



# Qmailrocks.org

A Qmail installation documentary

[home](#) | [about](#) | [the installation](#) | [scripts & utilities](#) | [faq](#) | [contact](#) | [links](#) | [donations](#) | [journal](#) | [mailing list](#) | [list archive](#) | [qmr forum](#)



welcome to

## The qmailrocks.org qmail Installation guide for Solaris 9

**Warning:** This is the **beta release** of the Solaris QMR installation guide. It is **NOT a release version** and there may be mistakes and/or bugs in the guide. The purpose of the beta release is to allow visitors to try it out and then report back with and successes or failures they may encounter. All feedback received will be acted upon in order to publish the release version of this install guide.

### [Solaris 9 qmail installation guide \(beta release\)](#)

[qmailrocks@earthlink.net](mailto:qmailrocks@earthlink.net)

Color Coded Qmail Installation Key	
	Regular Black Text Qmail installation notes and summaries by the author. Me talking.
<b>Black</b>	<b>Bold Black Text</b> Commands to be run by you, the installer.
<b>Maroon</b>	<b>Bold Maroon Text</b> Special notes for Redhat 9 users.
<b>Red</b>	<b>Bold Red Text</b> Vital and/or critical information.
<b>Purple</b>	<b>Regular/Bold Purple text</b> Denotes helpful tips and hints or hyperlinks.
<b>Orange</b>	<b>Regular Orange Text</b> Command line output.
<b>Green</b>	<b>Regular green text</b> Denotes the contents of a file or script.

[home](#) | [about](#) | [the installation](#) | [scripts & utilities](#) | [faq](#) | [contact](#) | [links](#) | [donations](#) | [journal](#) | [mailing list](#) | [list archive](#) | [qmr forum](#)

Site last modified: **June 22, 2004 13:28:29**





welcome to

## The qmailrocks.org (beta) qmail Installation guide for Solaris 9

As it is now, I've split the Solaris qmail installation up into 20 sections which are listed below. Simply start at step 1 and follow it all the way to completion. By the time you reach step 20, you should have a smokin' qmail installation of qmail blazing away on your server.

**Warning:** This is the **beta release** of the Solaris QMR installation guide. It is **NOT a release version** and there may be mistakes and/or bugs in the guide. The purpose of the beta release is to allow visitors to try it out and then report back with and successes or failures they may encounter. All feedback received will be acted upon in order to publish the release version of this install guide.

For a quick look at what this installation will provide you with, [click here](#).

**Before You Start!** - When installing qmail, I would **STRONGLY** recommend that you first conduct a test installation on a test server if you have one available. I cannot stress this enough. If you're new to qmail, chances are you will probably screw the install up the first time through. Do yourself a favor and screw up on a test server that has no importance to you!

**A special note for Solaris 9 users** - When installing the QMR installation on Solaris 9, the pre-preparation of your system is EXTREMELY important. Please go over every step in **Part 0** (below) before proceeding to the rest of the install. E-mails concerning problems that can be solved by reading **Part 0** will be ignored.

Part 0 - [Pre-installation checklist / Getting your Solaris system prepared](#)

Part 1 - [Download all the needed items for the qmail installation](#)

Part 2 - [Installing qmail itself](#)

Part 3 - [Installing EZmlm](#)

Part 4 - [Installing autoresponder](#)

Part 5 - [Installing vpopmail](#)

Part 6 - [Installing vqadmin](#)

Part 7 - [Installing maildrop](#)

Part 8 - [Installing qmailadmin](#)

Part 9 - [Finalizing the qmail installation](#)

Part 10 - [Uninstalling Sendmail](#)

Part 11 - [Starting up qmail](#)

Part 12 - [Installing Courier-imap](#)

Part 13 - [Installing Squirrelmail](#)

Part 14 - [Installing Clam Antivirus & SpamAssassin](#)

Part 15 - [Installing qmail-Scanner w/qms-analog](#)

Part 16 - [Installing qmailanalog & qlogtools](#)

Part 17 - [Installing qtrap](#)

Part 18 - [Maintaining your qmail server](#)

Part 19 - [Mail client configuration](#)

Part 20 - [Feedback](#)

qmailrocks@earthlink.net

Color Coded Qmail Installation Key		
	Regular Black Text	Qmail installation notes and summaries by the author. Me talking.
■	<b>Black Text</b>	Commands to be run by you, the installer.
■	<b>Maroon Text</b>	Special notes for Redhat 9 users.
■	<b>Red Text</b>	Vital and/or critical information.
■	Regular/Bold Purple text	Denotes helpful tips and hints or hyperlinks.
■	Regular Orange Text	Command line output.
■	Regular green text	Denotes the contents of a file or script.

[home](#) | [about](#) | [the installation](#) | [scripts & utilities](#) | [faq](#) | [contact](#) | [links](#) | [donations](#) | [journal](#) | [mailing list](#) | [list archive](#) | [qmr forum](#)

Site last modified: **June 22, 2004 13:28:29**





## Qmailrocks.org Pre-Installation Checklist Solaris 9

A successful QMR qmail installation requires certain packages be installed and certain configurations be present on your server. I've put together this page to provide a **general** checklist for visitors to use before they begin the installation. Keep in mind that, since setups will vary from server to server, you may find some requirements that are not listed here. This list is by no means written in stone, so if you find a requirement that you feel is vital and is not present on this page, please feel free to let me know.

### How much disk space should I have available on my server?

Below I have provided some general estimates on how much disk space you will need in relation to how your disk is partitioned. The numbers you see below are not "bare minimum" numbers, but are rather my opinion as to the minimum amount of space needed for basic and safe operation. Keep in mind that it's really hard for me to tell you how much space you're going to need overall. The amount of space your mail server will need really depends on how much mail you intend on running through it. Obviously, if you intend on hosting the mail for 500 domains on your server, you'd better have a hefty drive to accommodate the amount of spooled mail and delivered mail that could be on your server at any given time. By the same logic, if you are only going to host 1 domain's mail on the server, you can probably get by with a lot less. The 2 areas of your mail server that are going to fluctuate the most are "/var/qmail" (where the spool is housed) and "/home/vpopmail" (where the mailboxes reside). When in doubt, get a bigger drive. It's always better to have too much space than too little. Once you have your qmail server up and running, enabling quotas for all domains can help you to better calculate the amount of disk space you will need.

If your server only has 1 giant "/" partition:

A safe amount of disk space would be about 80-90MB. This includes the download of the qmailrocks.tar.gz software bundle and it's extraction. After a successful installation, the total amount of needed space for a safely operating mail server could be brought down to about 15MB after the qmailrocks.tar.gz content is removed.

If your server is multiple partitions:

**/ partition:** About 70MB.

**/var partition:** About 10MB

**/home partition:** About 3MB for starters, although this **will** change as "/home/vpopmail" will be the place in which all e-mail is stored for all domains.

**Some free advise:** If the 80-90MB of estimated needed space is asking alot from your server, you might want to reconsider whether or not to use that server as a mail server. A mail server that is tight on disk space is a recipe for trouble. Just my opinion.

### What type of base installation does the QMR Solaris 9 start off with?

For my installation, I started off with a basic installation of Solaris 9 using the initial Solaris 9 install disk and then the 2 Solaris 9 software disks that go with it. I selected the default software setup. Basically, I stuck in Disk 1 and installed Solaris, then stuck in Software Disk 1 and then Software Disk 2 when told. I made no customizations except for when I setup the network connection, of course. This was all done on a Sun Sparc Ultra 5 with 128MB of RAM and 4.6Gb hard drive.

### What additional software packages should I already have installed on my server?

I added the following packages to my base system. Except where mentioned otherwise, I acquired all packages from ftp.sunfreeware.com. To access ftp.sunfreeware.com, simply use the following info:

**Host:** ftp.sunfreeware.com  
**Username:** anonymous  
**Password:** your\_email@somewhere.com

Anyway, I downloaded the latest versions of the following software and the installed them using the "pkgadd" utility. All of the following packages can be found at ftp.sunfreeware.com:

**gzip**

**gcc**

**diffutils**

**automake**

**autoconf**

**coreutils**

**wget** - for downloading software via the web.

**Apache 2.0.49** - The Apache webserver, duh.

**Perl** - I would strongly recommend that you install the sunfreeware version of Perl. It's a gcc compiled version and will therefore give you much less trouble when installing the needed modules later on. On my system, I uninstalled the default Sun Perl install that came with Solaris 9 and then installed the latest version from [ftp.sunfreeware.com](http://ftp.sunfreeware.com)

### What software packages should NOT be installed?

**1. Postfix** - If it's installed on your server, you will need to either uninstall it or disable it.

**2. Any POP service** - This includes Qpopper or any POP service that may be running out of xinetd or inetd. If your server has a POP service running, you will need to disable it.

### What Perl modules should be installed?

This list may vary depending on your setup, but here goes:

**Digest::SHA1**

**Digest::HMAC**

**Net::DNS**

**Time::HiRes**

**HTML::Tagset**

**HTML::Parser**

Again, these modules will be much easier to install if you have a gcc compiled version of Perl on your system rather than the default Sun Perl installation.

I'm almost positive that someone out there will need more, so if you come across any other needed modules please drop me a line.

### I'm running a firewall on my server. What ports should I open?

**IMPORTANT NOTE:** Keep in mind the the following ports are what are required to be open for only the QMR install. More than likely, a fully functioning webserver is going to have more ports open for various other services. So, do not use the port listing below to construct a NEW firewall without first determining what other ports you will need to open. If you are are interested in constructing a complete firewall for your server, check out the iptables tutorial at <http://www.iptablesrocks.org>.

#### Outbound ports (tcp)

25 - SMTP

110 - POP services

143 - IMAP

783 - Spamassassin

993 - IMAPS

#### Inbound Ports (tcp)

25 - SMTP  
 80 - HTTP  
 110 - POP services  
 143 - IMAP  
 443 - HTTPS  
 783 - Spamassassin  
 993 - IMAPS

### Environment Settings:

**vi /etc/profile**

Add the following line:

```
PATH=/usr/sbin:/usr/bin://bin:/usr/bin:/usr/sbin:/usr/local/sbin:/usr/local/bin:/sbin:/usr/ccs/bin:/usr/local/mysql/bin:/usr/local/ssl/bin
```

```
CPPFLAGS="-I/usr/include -I/usr/local/include -I/usr/local/ssl/include -I/usr/local/BerkeleyDB.4.2/include -I/usr/java/include -I/usr/sfw/include"
```

**vi /etc/default/login & /etc/default/su**

Add the following lines:

```
PATH=/usr/sbin:/usr/bin://bin:/usr/bin:/usr/sbin:/usr/local/sbin:/usr/local/bin:/sbin:/usr/ccs/bin:/usr/local/mysql/bin:/usr/local/ssl/bin
```

```
CPPFLAGS="-I/usr/include -I/usr/local/include -I/usr/local/ssl/include -I/usr/local/BerkeleyDB.4.2/include -I/usr/java/include -I/usr/sfw/include"
```

Run the following commands to set up the correct library paths:

```
crle -u -v -l /usr/lib:/lib:/etc/lib:/usr/local/lib:/usr/local/ssl/lib:/usr/local/BerkeleyDB.4.2/lib:/usr/share/lib: \  
/usr/xpg4/lib:/usr/sfw/lib:/usr/java/lib:/usr/ccs/lib:/usr/sadm/lib:/etc/security/lib:/opt/csw/lib
```







```
export LD_LIBRARY_PATH=/usr/local/lib
```

If /usr/ccs/bin/cc does not exist after installing all need packages above, add the following symlink to your system:

```
ln -s /usr/local/bin/gcc /usr/ccs/bin/cc
```

[Proceed to the Qmailrocks.org Qmail Installation](#)

[qmailrocks@earthlink.net](mailto:qmailrocks@earthlink.net)

Color Coded Qmail Installation Key	
	Regular Black Text   Qmail installation notes and summaries by the author. Me talking.
	<b>Regular Black Text</b>   Commands to be run by you, the installer.
	<b>Bold Maroon Text</b>   Special notes for Redhat 9 users.
	<b>Bold Red Text</b>   Vital and/or critical information.
	<b>Regular/Bold Purple text</b>   Denotes helpful tips and hints or hyperlinks.
	<b>Regular Orange Text</b>   Command line output.
	<b>Regular green text</b>   Denotes the contents of a file or script.

[home](#) | [about](#) | [the installation](#) | [scripts & utilities](#) | [faq](#) | [contact](#) | [links](#) | [donations](#) | [journal](#) | [mailing list](#) | [list archive](#) | [qmr forum](#)

Site last modified: **June 22, 2004 13:28:29**





## Part 1 - Download the Software

The first thing you'll need to do to get started is to download all the needed software packages for the entire qmail installation process. To make things easier for everyone, I've combined all the needed packages into 1 giant tarball (.tar.gz) bundle that you can download in one easy step. All of the included packages are the latest versions as of **June, 2004** and I will make efforts keep the software bundle up to date as new versions are released. If your going to be using the Qmailrocks installation guide to install qmail, I would strongly recommend that you download my software bundle. I've included several ready-made scripts and so forth, so everything will make more sense if you're on the same page as I am when going through the install.

So let's start the installation by getting the needed software. You will notice that below I create a new directory called /downloads and I place the Qmailrocks tarball in that directory before unpacking it. I would strongly recommend you do this as the rest of the instructions on this site are geared toward this source directory structure. Anyway, lets get down to business.

**mkdir /downloads**

**cd /downloads**

Now download the Qmailrocks.org software bundle:

**wget http://www.qmailrocks.org/downloads/qmailrocks.tar.gz**

(Alternatively, if you'd like to download individual packages or view a listing of all the packages, you may do so [right here](#).)

Once you've downloaded qmailrocks.tar.gz, were going to place in a directory called "downloads" and then unpack it...

**gunzip qmailrocks.tar.gz**

**tar xvf qmailrocks.tar**

[Proceed to Part 2](#)

[qmailrocks@earthlink.net](mailto:qmailrocks@earthlink.net)

Color Coded Qmail Installation Key		
	Regular Black Text	Qmail installation notes and summaries by the author. Me talking.
■	<b>Bold Black Text</b>	Commands to be run by you, the installer.
■	<b>Bold Maroon Text</b>	Special notes for Redhat 9 users.
■	<b>Bold Red Text</b>	Vital and/or critical information.
■	Regular/Bold Purple text	Denotes helpful tips and hints or hyperlinks.
■	Regular Orange Text	Command line output.
■	Regular green text	Denotes the contents of a file or script.





## Part 2 - Installing Qmail Itself

Now that you've downloaded all the needed packages, we can start the install. At this point you should have a qmailrocks source directory located at /downloads/qmailrocks. If you don't, go back to step 1. This step involves the setup of the very heart of your new qmail server. In this step, we'll install qmail itself, ucspi-tcp and daemontools. These 3 packages are the core of the qmail server and will be the foundation on which we build everything else. So don't screw it up!

I've created a handy little shell script that takes care of the first portion of getting qmail, ucspi-tcp and daemontools installed. Simply run this script from the command prompt of your Solaris box and you should be golden. The script will tell you what it's doing along the way.

### **/downloads/qmailrocks/scripts/install/qmr\_install\_solaris-s1.script**

If all goes well, you should have all the needed user and groups created as well as all the needed directories, permissions and ownership settings needed for the installation of qmail, ucspi-tcp and daemontools

Next, we apply 3 patches to qmail before we compile it. I've included a handy little script to apply all 3 patches at once. The 3 patches we are going to apply are: 1) The standard qmail-1.03 patch 2) The qmailqueue patch to enable the use of qmail--scanner later in this installation and 3) The auth-jms1.4a patch to enable SMTP authentication.

However, when using Solaris it is necessary to first install the GNU "diffutils" package so that you can use the "gpatch" utility instead of the typical "patch" utility. Lucky for you, I've included the "diffutils" package in the qmailrocks tarball...

### **pkgadd -d /downloads/qmailrocks/tools/diffutils-2.8.1-sol9-sparc-local**

Once you've got "diffutils" installed, let's patch qmail...

### **/downloads/qmailrocks/scripts/util/solaris\_qmail\_patches.script** ([click here to view this script](#))

Now we build Qmail...

### **cd /usr/src/qmail/qmail-1.03**

**make man && make setup check**

**./config-fast your\_fqdn\_hostname** (ex: ./config-fast mail.mydomain.com)

If there are no errors, Qmail has been built successfully!

Now we build ucspi-tcp...

### **cd /usr/src/qmail/ucspi-tcp-0.88/**

**make && make setup check**

If you don't get any errors, that's it for ucspi-tcp!

Now we build the daemontools....

### **cd /package/admin/daemontools-0.76**

**package/install**

If no errors are reported, you've successfully compiled the daemontools package!

## vi /etc/inittab

replace

```
SV:123456:respawn:/command/svscanboot
```

with

```
SV:123456:respawn:/command/svscanboot </dev/null >/var/log/svscan 2>&1
```

Now reboot your server:

## reboot

If you run take a look at the running processes on your server once it comes back up, you should see the daemon "svscanboot" running. You can usually do this with a "ps -ef" command. Here's a [screenshot](#) of it. If you see "svscanboot" running, you're in good shape.

OK, Qmail is almost totally installed but we're going to pause right here and install a bunch of handy tools and features that will make Qmail pretty and fun! After that, we'll make some final changes to qmail and then crank it up!

[Proceed to Part 3](#)

[qmailrocks@earthlink.net](mailto:qmailrocks@earthlink.net)

Color Coded Qmail Installation Key		
	Regular Black Text	Qmail installation notes and summaries by the author. Me talking.
■	<b>Bold Black Text</b>	Commands to be run by you, the installer.
■	<b>Bold Maroon Text</b>	Special notes for Redhat 9 users.
■	<b>Bold Red Text</b>	Vital and/or critical information.
■	Regular/Bold Purple text	Denotes helpful tips and hints or hyperlinks.
■	Regular Orange Text	Command line output.
■	Regular green text	Denotes the contents of a file or script.

[home](#) | [about](#) | [the installation](#) | [scripts & utilities](#) | [faq](#) | [contact](#) | [links](#) | [donations](#) | [journal](#) | [mailing list](#) | [list archive](#) | [qmr forum](#)

Site last modified: **June 22, 2004 13:28:29**





## Part 3 - EZmlm

EZmlm is a nice mailing list add-on to qmail. I've used it several times myself and its actually one of the better mailing list programs out there. When we install Qmailadmin later on, you'll see that EZmlm integrates seamlessly into Qmailadmin to provide a very user friendly mailing list management interfact. As an added bonus, Vpopmail (which we will install as well) will let you control what users can and cannot use mailing lists! Can't beat that!

So let's install it...

```
cd /downloads/qmailrocks/
```

```
gunzip ezmlm-0.53-idx-0.41.tar.gz
```

```
tar xvf ezmlm-0.53-idx-0.41.tar
```

```
cd ezmlm-0.53-idx-0.41
```

```
make && make setup
```

If you don't get any errors, then ezmlm is all set up and ready to go!

[Proceed to Part 4](#)

[qmailrocks@earthlink.net](mailto:qmailrocks@earthlink.net)

Color Coded Qmail Installation Key	
	Regular Black Text   Qmail installation notes and summaries by the author. Me talking.
<b>Black</b>	<b>Bold Black Text</b>   Commands to be run by you, the installer.
<b>Maroon</b>	<b>Bold Maroon Text</b>   Special notes for Redhat 9 users.
<b>Red</b>	<b>Bold Red Text</b>   Vital and/or critical information.
<b>Purple</b>	Regular/Bold Purple text   Denotes helpful tips and hints or hyperlinks.
<b>Orange</b>	Regular Orange Text   Command line output.
<b>Green</b>	Regular green text   Denotes the contents of a file or script.





# Qmailrocks.org

A Qmail installation documentary

[home](#) | [about](#) | [the installation](#) | [scripts & utilities](#) | [faq](#) | [contact](#) | [links](#) | [donations](#) | [journal](#) | [mailing list](#) | [list archive](#) | [qmr forum](#)

## Part 4 - Autoresponder

Autoresponder does exactly what you think it does. It allows us to set up autoresponders for mailboxes and so forth.

So let's install it...

```
cd /downloads/qmailrocks
```

```
gunzip autorespond-2.0.2.tar.gz
```

```
tar xvf autorespond-2.0.2.tar
```

```
cd autorespond-2.0.2
```

```
make && make install
```

If you don't get any errors, then autoresponder is all set up and ready to go!

[Proceed to Part 5](#)

[qmailrocks@earthlink.net](mailto:qmailrocks@earthlink.net)

Color Coded Qmail Installation Key		
	Regular Black Text	Qmail installation notes and summaries by the author. Me talking.
■	<b>Bold Black Text</b>	Commands to be run by you, the installer.
■	<b>Bold Maroon Text</b>	Special notes for Redhat 9 users.
■	<b>Bold Red Text</b>	Vital and/or critical information.
■	Regular/Bold Purple text	Denotes helpful tips and hints or hyperlinks.
■	Regular Orange Text	Command line output.
■	Regular green text	Denotes the contents of a file or script.

[home](#) | [about](#) | [the installation](#) | [scripts & utilities](#) | [faq](#) | [contact](#) | [links](#) | [donations](#) | [journal](#) | [mailing list](#) | [list archive](#) | [qmr forum](#)

Site last modified: **June 22, 2004 13:28:29**





## Part 5 - Vpopmail

Vpopmail is one of the major components of this installation. Vpopmail allows us to do virtual domain mail hosting. It's got a lot of built in tools and features that make it a dream to work with. Even if you don't want to host mail for multiple domains, I would still recommend installing Vpopmail. It just makes the whole mail game easier. Plus, my installation centers around it, so if you don't install it you're going to have a headache.

So let's install it...

```
cd /downloads/qmailrocks
```

```
gunzip vpopmail-5.4.4.tar.gz
```

```
tar xvf vpopmail-5.4.4.tar
```

```
cd vpopmail-5.4.4
```

### - Helpful Hint: Vpopmail configuration options -

The vpopmail "configure" command can have loads of options. Use "./configure --help" to see them all. In the syntax used in this installation, I specify to the type of logging that I want vpopmail to use. Vpopmail logs its activities to the server's syslog and there are several options you can use. I've used the "p" option, but feel free to adjust it to your needs. Here's are the details:

```
--enable-logging=n - logs nothing
```

```
--enable-logging=e - logs only errors (default)
```

```
--enable-logging=y - logs all attempts
```

```
--enable-logging=p - logs errors with passwords
```

```
--enable-logging=v - verbose. Logs all attempts with passwords
```

Now let's configure vpopmail...

```
./configure --enable-logging=p --prefix=/usr/local
```

```
make && make install-strip
```

If you don't get any errors, then Vpopmail is good to go!

### Wait! I'm getting an error! What do I do?

If you are using GCC version 3.x, you may get a compilation error similar to this:

```
In file included from vconvert.c:35:
vmysql.h:53:22: missing terminating " character
vmysql.h:60:35: missing terminating " character
make[2]: *** [vconvert.o] Error 1
```

If you get this error, you will need to apply a patch to Vpopmail:

```
patch < /downloads/qmailrocks/patches/vmysql.patch
```

Once you've applied the patch, try running "make" and "make install" again and you should

be ok.

So now let's go on to part 7...

[Proceed to Part 6](#)

[qmailrocks@earthlink.net](mailto:qmailrocks@earthlink.net)

Color Coded Qmail Installation Key		
	Regular Black Text	Qmail installation notes and summaries by the author. Me talking.
■	<b>Black Text</b>	Commands to be run by you, the installer.
■	<b>Maroon Text</b>	Special notes for Redhat 9 users.
■	<b>Red Text</b>	Vital and/or critical information.
■	Regular/Bold Purple text	Denotes helpful tips and hints or hyperlinks.
■	Regular Orange Text	Command line output.
■	Regular green text	Denotes the contents of a file or script.

[home](#) | [about](#) | [the installation](#) | [scripts & utilities](#) | [faq](#) | [contact](#) | [links](#) | [donations](#) | [journal](#) | [mailing list](#) | [list archive](#) | [qmr forum](#)

Site last modified: **June 22, 2004 13:28:29**





## Part 6 - Vqadmin

Vqadmin is simply a nice web based interface that will let us manage Vpopmail. Through the interface we can create new domains, new users, net quotas, enable services and much more.

So let's install it...

```
cd /downloads/qmailrocks
```

```
gunzip vqadmin-2.3.6.tar.gz
```

```
tar xvf vqadmin-2.3.6.tar
```

```
cd vqadmin-2.3.6
```

```
./configure --enable-cgibindir=/path/to/your/cgi-bin --enable-htmldir=/path/to/your/html/directory
```

(Example: ./configure --enable-cgibindir=/usr/local/apache2/cgi-bin --enable-htmldir=/usr/local/apache2/htdocs)

```
vi +535 Makefile.in
```

change the following line:

```
chgrp `id -g root` @cgibindir@/vqadmin/vqadmin.cgi
```

to this:

```
chgrp /usr/xpg4/bin/id -g root` @cgibindir@/vqadmin/vqadmin.cgi
```

```
make && make install-strip
```

If the installation is successful, Vqadmin should install itself in the cgi-bin directory of your default website. Unless you tell it otherwise, that usually defaults to /var/www/cgi-bin. You can specify another location in the ./configure command above.

Now you will need to add the following to your server's Apache configuration file (usually httpd.conf)

```
<Directory "/path/to/your/cgi-bin/vqadmin">
deny from all
Options ExecCGI
AllowOverride AuthConfig
Order deny,allow
</Directory>
```

In addition, within the Apache master config file you will want to set the "AllowOverride" option to "All". Example: **AllowOverride All**

```
cd /path/to/your/cgi-bin/vqadmin
```

Now you will want to create a .htaccess file to password protect the Vqadmin interface. There should already be a .htaccess file in the vqadmin directory, so all you need to do is configure it.

```
vi .htaccess
```

```
AuthType Basic
AuthUserFile /path/to/where/you/want/to/store/the/password/file/.htpasswd
AuthName vQadmin
```

require valid-user  
satisfy any

**chown nobody .htaccess**

**chmod 644 .htaccess**

Now you need to create a corresponding .htpasswd file that's going to contain the username and encrypted password for the Vqadmin administrator...

**htpasswd -bc /path/to/where/you/want/to/store/the/password/file/.htpasswd admin admin\_password**

**chmod 644 /path/to/where/you/want/to/store/the/password/file/.htpasswd**

**Question:** But what if I want to user another username other than "admin"?

**Answer::** You will notice that in the above line, I'm adding an admin user called "admin". The name of the user needs to be "admin" because that is the username which Vqadmin sets up by default to have full admin rights. If you want to use a username other than "admin", you will need to edit the /cgi-bin/vqadmin/vqadmin.acl file and add your custom user to that file along with whatever rights you want it to have. Within that file, you will see where the user called "admin" is already set up to have all rights. That line looks like this:

```
admin VIMUDCA admin1user
```

In this line, the "admin" part specifies the username, the "VIMUDCA" part specifies that user's rights (a chart of all the possible rights is right above this line), and the "admin1user" part specifies the common name for the user which will be displayed when you log into Vqadmin. The common name is purely for aesthetic purposes.

Now restart Apache...

**apachectl stop**

**apachectl start**

If all has gone well, you should now be able to browse (in your web browser) to:  
**http://www.yourdomain.com/cgi-bin/vqadmin/vqadmin.cgi**







**Note:** make sure you include "vqadmin.cgi" in the path or you will get a 403 forbidden error!

Enter the user "admin" and whatever password your assigned it.

You should now see the Vqadmin interface. Go ahead and add a new domain to your server! Pretty cool, huh? The "postmaster" user serves as the admin user for any new domain and we will use it to log into Qmailadmin, when we install that. As soon as we complete the install, that domain will be instantly able to get mail (assuming the MX is correctly pointing to your server). The nice thing about Vpopmail and Vqadmin is that you do not need to restart anything after you add a domain. Once you add it, it simply works! We're almost done!

[Proceed to Part 7](#)

[qmailrocks@earthlink.net](mailto:qmailrocks@earthlink.net)

Color Coded Qmail Installation Key		
	Regular Black Text	Qmail installation notes and summaries by the author. Me talking.
	<b>Bold Black Text</b>	Commands to be run by you, the installer.
	<b>Bold Maroon Text</b>	Special notes for Redhat 9 users.
	<b>Bold Red Text</b>	Vital and/or critical information.
	Regular/Bold Purple text	Denotes helpful tips and hints or hyperlinks.
	Regular Orange Text	Command line output.
	Regular green text	Denotes the contents of a file or script.

[home](#) | [about](#) | [the installation](#) | [scripts & utilities](#) | [faq](#) | [contact](#) | [links](#) | [donations](#) | [journal](#) | [mailing list](#) | [list archive](#) | [qmr forum](#)

Site last modified: **June 22, 2004 13:28:29**





## Part 7- Maildrop

Maildrop is a mail filtering agent which can be used to filter messages as they arrive on the server. You will probably notice, once this installation is complete, that you don't really use maildrop. However, it's a cool tool and it's worth having around if you ever decide to get crazy with filtering your incoming mail. You can find documentation on maildrop [right here](#).

Let's install it...

```
cd /downloads/qmailrocks
```

```
gunzip maildrop-1.6.3.tar.gz
```

```
tar xvf maildrop-1.6.3.tar
```

```
cd maildrop-1.6.3
```

```
./configure --prefix=/usr/local --exec-prefix=/usr/local --enable-maildrop-uid=root --enable-maildrop-gid=vchkw --enable-maildirquota
```

```
make && make install-strip && make install-man
```

If you didn't get any errors, maildrop should be all set!

[Proceed to Part 8](#)

[qmailrocks@earthlink.net](mailto:qmailrocks@earthlink.net)

Color Coded Qmail Installation Key		
	Regular Black Text	Qmail installation notes and summaries by the author. Me talking.
■	<b>Bold Black Text</b>	Commands to be run by you, the installer.
■	<b>Bold Maroon Text</b>	Special notes for Redhat 9 users.
■	<b>Bold Red Text</b>	Vital and/or critical information.
■	Regular/Bold Purple text	Denotes helpful tips and hints or hyperlinks.
■	Regular Orange Text	Command line output.
■	Regular green text	Denotes the contents of a file or script.





## Part 8 - Qmailadmin

Qmailadmin is going to provide us with a nice web based interface for administering mail accounts once they are setup through Vpopmail (or Vqadmin). From Qmailadmin we can create mailboxes, aliases, forwards, mail robots, mailing lists. You'll also find a few other handy functions as well. Qmailadmin is sort of the icing on the Qmail cake.



[click for full size preview](#)

Let's install it...

```
cd /downloads/qmailrocks
```

```
gunzip qmailadmin-1.2.0.tar.gz
```

```
tar xvf qmailadmin-1.2.0.tar
```

```
cd qmailadmin-1.2.0
```

```
./configure --enable-cgibindir=/path/to/your/cgi-bin --enable-htmldir=/path/to/your/html/directory
```

(Example: ./configure --enable-cgibindir=/usr/local/apache2/cgi-bin --enable-htmldir=/usr/local/apache2/htdocs)

```
make && make install-strip
```

That's it! Now browse to <http://www.yourdomain.com/cgi-bin/qmailadmin> and you should see the login screen. Login with the postmaster account and password for the domain that you created a while back using Vqadmin. Pretty cool, isn't it? Go ahead and create some additional mailboxes for your domain(s).

If you didn't get any errors, Qmailadmin should be all set!

**note:** if it's late and you're looking for a place to stop so you can sleep, this is a good place. Sendmail should still be handling mail on the server. After this page it's Qmail or bust!

[Proceed to Part 9](#)

[qmailrocks@earthlink.net](mailto:qmailrocks@earthlink.net)

Color Coded Qmail Installation Key		
	Regular Black Text	Qmail installation notes and summaries by the author. Me talking.
■	<b>Bold Black Text</b>	Commands to be run by you, the installer.
■	<b>Bold Maroon Text</b>	Special notes for Redhat 9 users.
■	<b>Bold Red Text</b>	Vital and/or critical information.
■	Regular/Bold Purple text	Denotes helpful tips and hints or hyperlinks.
■	Regular Orange Text	Command line output.
■	Regular green text	Denotes the contents of a file or script.





## Part 9 - Finalizing Qmail

Ok, we've installed a bunch of bells of whistles onto our qmail installation. Now it's time to wrap up the configuration for qmail itself. After that, we will stop and remove Sendmail from the server and then it's time to crank qmail up!

The first thing we're going to do is create the qmail supervise scripts, create the the qmail rc and qmailctl scripts and then set the needed permissions on all these scripts. Lucky for you, I've created a script to do all this for you. The script will give you a breakdown of what it is doing while it's running. If any errors occur, you'll see them. However, if you've configured everything right up until now, you shouldn't have any problems. You can check out the contents of this scripts [right here](#).

So let's run the script...

```
/downloads/qmailrocks/scripts/finalize/solaris/finalize_solaris.script
```

Hey, that was easy. Now there are just a couple tweaks to make to these new scripts we just created...

```
vi /var/qmail/supervise/qmail-pop3d/run
```

Find "**mail.example.com**" and change it to your server's hostname. [For example: mail.mydomain.com.](#)

```
vi /var/qmail/supervise/qmail-smtpd/run
```

Find "**mail.example.com**" and change it to your server's hostname. [For example: mail.mydomain.com](#)

OK, all done there. By now, you may notice that some Qmail functions are already up and running, so to finish the install, we will stop Qmail....

```
qmailctl stop
```

And setup selective relaying...

```
echo '127.:allow,RELAYCLIENT=""' >> /etc/tcp.smtp
```

```
qmailctl cdb
```

Now we create the common system aliases. These aliases are going to tell Qmail what to do with common server-generated mails. Stuff like bouncebacks, cron daily output and various other systemic sources. It's a good idea to redirect these aliases to a mailbox that you are going to check on a regular basis. You don't want to have your systemic mails piling up in some deep dark corner of your server doing no good and slowly filling your disk up.

```
echo some_address > /var/qmail/alias/qmail-root
```

where "some\_address" is the system user or email address you want these addresses aliased to.

```
echo some_address > /var/qmail/alias/qmail-postmaster
```

where "some\_address" is the system user or email address you want these addresses aliased to.

```
echo some_address > /var/qmail/alias/qmail-mailer-daemon
```

where "some\_address" is the system user or email address you want these addresses aliased to.







```
ln -s /var/qmail/alias/qmail-root /var/qmail/alias/qmail-anonymous
```

```
chmod 644 /var/qmail/alias/qmail*
```

Alright. We've got qmail ready to go. One of the last things we need to do is to disable/uninstall Sendmail on the server and replace the Sendmail binary with a symlink to qmail, so that our server won't freak out with Sendmail being gone.

[Proceed to Part 10](#)

[qmailrocks@earthlink.net](mailto:qmailrocks@earthlink.net)

Color Coded Qmail Installation Key		
	Regular Black Text	Qmail installation notes and summaries by the author. Me talking.
	<b>Bold Black Text</b>	Commands to be run by you, the installer.
	<b>Bold Maroon Text</b>	Special notes for Redhat 9 users.
	<b>Bold Red Text</b>	Vital and/or critical information.
	Regular/Bold Purple text	Denotes helpful tips and hints or hyperlinks.
	Regular Orange Text	Command line output.
	Regular green text	Denotes the contents of a file or script.

[home](#) | [about](#) | [the installation](#) | [scripts & utilities](#) | [faq](#) | [contact](#) | [links](#) | [donations](#) | [journal](#) | [mailing list](#) | [list archive](#) | [qmr forum](#)

Site last modified: **June 22, 2004 13:28:29**





## Part 10 - Uninstalling Sendmail

Well, the moment you've been waiting for is finally here. We're going to uninstall Sendmail from the server. However, since Sendmail is such a commonly used item among tons of server operations and cronjobs, you will see that, after we uninstall Sendmail, we will actually make an artificial Sendmail that is nothing more than a direct injection into qmail.

Anyway, let's uninstall it...

First, let's stop Sendmail

```
/etc/init.d/sendmail stop
```

Next, you'll want to check and see if your Sendmail installation is installed as a Sun package. You can do this by running a simple query...

```
pkginfo | grep sendmail
```

You should get an output similar to this:

```
system SUNWsndmr Sendmail root  
system SUNWsndmu Sendmail user
```

If so, you can remove the sendmail packages like so...

```
pkgrm SUNWsndmu
```

```
pkgrm SUNWsndmr
```

That's it! Sendmail is gone!

Now we will need to set up an "artificial" Sendmail, which is just a symbolic link to Qmail's Sendmail. This is needed to ensure that the myriad of systemic mail scripts are still able to send mail! qmail's "Sendmail" is nothing more than a direct injection into qmail itself...

```
In -s /var/qmail/bin/sendmail /usr/lib/sendmail
```

```
In -s /var/qmail/bin/sendmail /usr/sbin/sendmail
```

That's it! If all has gone well, Sendmail should be uninstalled and the Qmail Sendmail should be in its place.

Now it's time to give qmail a final test and then crank it up!

[Proceed to Part 11](#)

qmailrocks@earthlink.net

Color Coded Qmail Installation Key		
	Regular Black Text	Qmail installation notes and summaries by the author. Me talking.
■	<b>Black Text</b>	Commands to be run by you, the installer.
■	<b>Maroon Text</b>	Special notes for Redhat 9 users.
■	<b>Red Text</b>	Vital and/or critical information.
■	Regular/Bold Purple text	Denotes helpful tips and hints or hyperlinks.
■	Regular Orange Text	Command line output.
■	Regular green text	Denotes the contents of a file or script.

[home](#) | [about](#) | [the installation](#) | [scripts & utilities](#) | [faq](#) | [contact](#) | [links](#) | [donations](#) | [journal](#) | [mailing list](#) | [list archive](#) | [qmr forum](#)

Site last modified: **June 22, 2004 13:28:29**





## Part 11 - Starting qmail

Alright, qmail should be ready to go! But before we crank it up, let's run a script that will check the key components of the installation and make sure everything is alright.

To do this test, I have borrowed Dave Sill's "inst\_check" script, but I've made a few custom modifications to accommodate for the subtle differences between the Qmailrocks installation and the Life With Qmail installation. Basically, the Qmailrocks installation has a slightly different logging setup and some slight variations in permissions settings. If you've installed according to this site, use my version of the script, as using Dave's version will result in a lot of "error" detections that are false positives due to the differences in the 2 installs. When you run the script, it will check for some key required files and folders and will also check permissions and ownership settings on many key items. If a needed file does not exist or if the ownership/permissions settings are wrong on a key file, it will tell you and then make a suggestion as to how to correct the error. **This script does NOT check the CONTENT or SYNTAX of your scripts, but only for the scripts' existence and their ownership/permissions settings. If you've screwed up the syntax of on the run scripts, this tool will not detect it.** So you ready? Let's do it...

**/downloads/qmailrocks/scripts/util/qmr\_inst\_check**

If you get a "congratulations" type of message, you're all set. If you get some errors, just follow the directions to fix the errors and then re-run the script until you get all errors corrected and you get a "congratulations" message.

Assuming, you've passed the installation check script, let's crank Qmail up!

**qmailctl stop**

**qmailctl start**

You can find out how things are running by:

**qmailctl stat**

You should see an output like this:

```
/service/qmail-send: up (pid 18035) 34 seconds
/service/qmail-send/log: up (pid 18036) 34 seconds
/service/qmail-smtpd: up (pid 18270) 0 seconds
/service/qmail-smtpd/log: up (pid 18268) 1 seconds
messages in queue: 0
messages in queue but not yet preprocessed: 0
```

If you, don't see anything like that or if you see error messages, [click here](#) for troubleshooting tips.

Congratulations, Qmail is now officially up and running and you should be able to send and receive mail on the server. You can test it with the domain you created in Vqadmin a while back.

**telnet localhost 110**

you should see something like this:

```
Trying 192.168.1.10...
Connected to 192.168.1.10.
Escape character is '^]'.
+OK <16658.1054485137@yourserver.com>
user postmaster@mydomain.com (enter your username here. remember to use the full e-mail address)
+OK
pass your_password
+OK
quit
```

+OK  
Connection closed by foreign host.

This is the sign of a successful POP connection to the server!

Now try sending mail to that same user from another location. Telnet to 110 again and run the "list" command and you should see the message that your send...

**telnet localhost 110**

```
Trying 192.168.1.10...
Connected to 192.168.1.10.
Escape character is '^]'.
+OK <16658.1054485137@yourserver.comt>
user postmaster@mydomain.com (again, remember to log in with the full email address of the user)
+OK
pass your_password
+OK
list
+OK
1 323 (there's your message!)
.
quit
+OK
Connection closed by foreign host.
```

If you have reached this point, then Qmail is now successfully up and running. Technically speaking, you could quit right here and have a functioning mail server. However, we still have a few options that we're going to plug into Qmail before we're done. In the next pages we will:

Install Courier IMAP & IMAP SSL.

Install Squirrelmail web based mail interface (requires that IMAP be installed).

Install Qmail-Scanner, an alternative queueing device.

Install Spamassassin, to tag all incoming spam.

Install Clam Anti Virus - To quarantine e-mails containing known viruses.

Install Checkall, to filter out messages containing undesirable words that may make it past Spamassassin.

[Proceed to Part 12](#)

qmailrocks@earthlink.net

Color Coded Qmail Installation Key		
	Regular Black Text	Qmail installation notes and summaries by the author. Me talking.
■	<b>Bold Black Text</b>	Commands to be run by you, the installer.
■	<b>Bold Maroon Text</b>	Special notes for Redhat 9 users.
■	<b>Bold Red Text</b>	Vital and/or critical information.
■	Regular/Bold Purple text	Denotes helpful tips and hints or hyperlinks.
■	Regular Orange Text	Command line output.
■	Regular green text	Denotes the contents of a file or script.

[home](#) | [about](#) | [the installation](#) | [scripts & utilities](#) | [faq](#) | [contact](#) | [links](#) | [donations](#) | [journal](#) | [mailing list](#) | [list archive](#) | [qmr forum](#)

Site last modified: **June 22, 2004 13:28:29**





## Part 12 - Installing Courier-imap/imaps with Courierpassd

Now that you have qmail up and running, we're going to add a few extras onto it. For starters, we're going to install Courier-imap/imaps along with Courierpassd. Installing IMAP will, obviously, enable IMAP connections to the mail server and it is a necessary ingredient for most popular web based mail clients such as Horde, SQwebmail and Squirrelmail. Courier-imap is the preferred IMAP server to install because it has built in support the vchkw mail user setup that Vpopmail utilizes. In short, Courier IMAP works with Vpopmail and virtual domains. In addition to installing Courier-imap, we're going to install Courierpassd. Courierpassd is a utility that allows users to change their mailbox passwords remotely. This will come in handy when we install Squirrelmail in the next step of the installation. Courierpassd will allow your mail users to change their passwords using the Squirrelmail interface. This will give your users more power over their account settings and, more importantly, keep them from pestering you whenever they want to change their passwords. ;)

So let's start by installing Courier-imap/imaps

```
bunzip courier-imap-3.0.5.tar.bz2
```

```
tar xvf courier-imap-3.0.5.tar
```

```
cd courier-imap-3.0.5
```

```
./configure --prefix=/usr/local --exec-prefix=/usr/local --with-authvchkw --without-authldap --disable-root-check --with-ssl --with-authchangepwdir=/usr/local/libexec/authlib
```

**Hint: Since the above config line runs over 1 line, it'll be easier if you simply cut and past the entire config statement.**

**Note: the configure process will take a few minutes. Go grab a snack...**

```
make && make install-strip && make install-configure
```

If you get the following error upon make:

```
tlspasswordcache.c:11:26: openssl/rand.h: No such file or directory
```

Run the following commands:

```
In -s /usr/local/ssl/include/openssl /usr/local/include
```

```
In -s /usr/local/ssl/include/openssl /usr/include
```

Then run the "make" again and it should work correctly:

```
make && make install-strip && make install-configure
```

Now let's continue configuring imap...

```
cd /usr/local/etc
```

Make sure that the files "imapd" and "imapd-ssl" exist. If they do not exist, do the following:

```
cp imapd.dist imapd
```

```
cp imapd-ssl.dist imapd-ssl
```

Now let's create an SSL certificate for the IMAP-SSL server...

## **/usr/local/sbin/mkimapdcert**

This will start and automated process that creates a self-signed imap-ssl X.509 certificate called imapd.pem. It should create this new certificate at /usr/local/share/imapd.pem. If the certificate already exists, the "mkimapdcert" tool will not let you overwrite it.

**A Note on IMAP-SSL certificates:** Keep in mind that since this SSL certificate is self-signed and is not from a "trusted" authority such as Verisign or Thawte, mail clients such as Outlook will give a warning when they attempt to connect to your IMAP-SSL server on port 993. The warning will state that the certificate is not from a "trusted" authority. While the warning is a bit ugly, it does NOT mean your IMAP-SSL connection is any less secure than it would be with a real certificate from Verisign or Thawte. All it means is that the SSL certificate was not generated by a company which Microsoft recognizes as a "trusted" authority. From a security standpoint, however, your IMAP-SSL server is every bit as secure as it would be if you bought the certificate from Verisign or Thawte. If the warning is too inconvenient for your purposes, you will need to purchase a "real" certificate from a "trusted" authority such as Verisign or Thawte. Be prepared to shell out a good chunk of change if you do so.

## **vi +68 imapd-ssl**

Make sure that the following configuration exists: **IMAPDSSLSTART=YES**

Make sure that the following configuration exists around line 149: **TLS\_CERTFILE=/usr/local/share/imapd.pem**

Save and exit the file.

### **Special note for people running a small home or office network:**

If you are planning on having multiple users connect to your IMAP server from a single IP address, such as in a small home or office network, you may want to increase the "**MAXPERIP**" setting with the **/usr/local/etc/imapd** config file. This setting establishes the maximum number of IMAP connections that can be made from a single IP address. An example of this might be if you have a small office network running on a single DSL or Cable IP address and your mail server is outside of that network. While each computer in your internal network may have it's own private IP address, to the outside world anyone coming from your network has the single routeable IP address assigned to your DSL or Cable connection. The default setting for "**MAXPERIP**" is **4** so if you have a similar network setup and more than 4 people trying to access your IMAP server, you may want to increase this setting accordingly to avoid connection errors. Within the **/usr/local/etc/imapd** file, the line you are looking for looks like this:

```
MAXPERIP=4
```

Now we create the startup scripts...

```
cp /usr/local/libexec/imapd.rc /etc/init.d/imap
```

```
cp /usr/local/libexec/imapd-ssl.rc /etc/init.d/imaps
```

**Now let's start up IMAP and IMAP SSL...**

```
/etc/init.d/imap start
```

```
/etc/init.d/imaps start
```

If you run "nmap localhost", you should see both 143 and 993 now open and listening.

Now let's test it...

```
telnet localhost 143
```

```
Trying 192.168.1.10...
```

```
Connected to 192.168.1.10.
```

```
Escape character is '^['.
```

```
* OK [CAPABILITY IMAP4rev1 UIDPLUS CHILDREN NAMESPACE THREAD=ORDEREDSUBJECT THREAD=REFERENCES SORT QUOTA IDLE STARTTLS] Courier-IMAP ready. Copyright 1998-2003 Double Precision, Inc. See COPYING for distribution information.
```

```
a login postmaster@mydomain.com my_password
```

a OK LOGIN Ok. (successful login!)  
a logout (logs you out)  
\* BYE Courier-IMAP server shutting down  
a OK LOGOUT completed  
Connection closed by foreign host.

If you were able to log in , as in the example above, you're all set. IMAP is installed! For further testing, you can configure a mail client such as Outlook to test both the IMAP and IMAP-SSL connetion to your server. IMAPS runs on port 993.

Now that Couroier-imap is installed let's install our webmail client, Squirrelmail.

[Proceed to Part 13](#)

qmailrocks@earthlink.net

Color Coded Qmail Installation Key		
	Regular Black Text	Qmail installation notes and summaries by the author. Me talking.
■	<b>Bold Black Text</b>	Commands to be run by you, the installer.
■	<b>Bold Maroon Text</b>	Special notes for Redhat 9 users.
■	<b>Bold Red Text</b>	Vital and/or critical information.
■	Regular/Bold Purple text	Denotes helpful tips and hints or hyperlinks.
■	Regular Orange Text	Command line output.
■	Regular green text	Denotes the contents of a file or script.

[home](#) | [about](#) | [the installation](#) | [scripts & utilities](#) | [faq](#) | [contact](#) | [links](#) | [donations](#) | [journal](#) | [mailing list](#) | [list archive](#) | [qmr forum](#)

Site last modified: **June 22, 2004 13:28:29**





## Part 13 - Installing Squirrelmail

Now that we have Qmail running with IMAP, we can install a webmail client to make mail accessible via a web browser. My choice for this was [Squirrelmail](#). Squirrelmail is both easy to install and it has lots of nice plugins to broaden it's abilities. I know that a lot of poeple out there like to use [Horde](#). I like Horde myself and I've installed it on my Qmail server alongside Squirrelmail. However, Horde is a major pain in the ass to install. Anyone who's ever intalled it will tell you that. I just don't want to take the responsibility for it on this site. Some people also like to use SQwebmail. No offense, but I just don't care for SQwebmail.

Anyway, let's install Squirrelmail...

The first order of business to make sure PHP is installed and correctly configured. So let's get that out of the way...

In order for Squirrelmail to work correctly, you'll need to check a couple things about your PHP installation:

1. First of all, make sure some rendition of PHP 4 is installed. If it's not, kick yourself in the ass and then go install it. Sorry, I'm not going to give a PHP installation tutorial. To be safe, you will want the following config options to be active in your PHP installation.

```
--enable-track-vars
--enable-force-cgi-redirect
--with-gettext
--with-mysql
```

If you're running Redhat, however, PHP can be easily installed as an RPM either manually or with "up2date". A default RPM installation of PHP will usually cover you. The only extra RPMs you'll want to install is "php-mysql".

2. Make sure you have PHP uploads turned ON. This is done by editing a line in your php.ini file. The location of the php.ini file can vary, but it's usually located at /etc/php.ini. If it's not, don't panic. Just run the old "locate php.ini" command. ;) Here's the line you will want to check/edit:

```
file_uploads = On
```

That's it for the PHP setup. Now let's download Squirrelmail...

You can download the latest version of Squirrelmail from: <http://www.squirrelmail.org/download.php>

I recommend downloading the .tar.gz version of the latest release.

Now change directories to the web directory of the website you want to serve Squirrelmail off of. In my case, I used /var/www/html

```
tar zxvf /path/to/squirrelmail-x.x.x.tar.gz (enter whatever version you downloaded)
```

Now rename the untarred folder to something more friendly...

```
mv squirrelmail-x.x.x webmail
```

And now let's configure Squirrelmail...

```
mkdir /var/sqattachements
```

```
chown -R apache:apache /var/sqattachements (or whatever user apache runs as)
```

```
cd webmail
```

```
chown -R apache:apache data (or whatever user apache runs as)
```

**cd config****./conf.pl**

This will run the Squirrelmail setup script which will allow you to customize the installation as well as set your server settings. Most of the important settings are in area #2, which is dubbed "Server Settings". Here are the specs I recommend:

## General

- 
1. Domain : 1.2.3.4 (**Enter the IP of your server here. Don't be an idiot and actually use 1.2.3.4**)
  2. Invert Time : false
  3. Sendmail or SMTP : SMTP

## IMAP Settings

- 
4. IMAP Server : localhost
  5. IMAP Port : 143
  6. Authentication type : login
  7. Secure IMAP (TLS) : false
  8. Server software : other
  9. Delimiter : detect

## SMTP Settings

- 
4. SMTP Server : localhost
  5. SMTP Port : 25
  6. POP before SMTP : false
  7. SMTP Authentication : login
  8. Secure SMTP (TLS) : false

Depending on what version of Squirrelmail you are installing, the setup menu may differ slightly. But you get the idea. If you like, there are several other features of Squirrelmail you can customize that, while not critical, are sometimes fun. Also, check out [Squirrelmail's site](#) for tons of cool plugins. Make sure you save all settings before exiting the configuration menu.

Once you've configured Squirrelmail to your liking, it's time to configure Apache to serve our new webmail interface...

**Notice:** The following Apache configuration entry below will show you what I MYSELF have for my server's Apache configuration. Apache configurations will vary, so this may or may not work for you. Also, this is not meant to be a lesson in how to configure Apache. If you are confused about configuring Apache, I would recommend that you STOP here and go find a tutorial on Apache. Please do not email me asking me to explain Apache configuration methods to you. I am currently working on a comprehensive Apache tutorial site ([apacherocks.org](#)), but until it's complete, you will need to seek Apache help and advise elsewhere.

There are probably about a million ways to do this, but here's what I do. I edit the httpd.conf Apache configuration file and add the following block:

```
<VirtualHost 1.2.3.4:80>
ServerName mail.mydomain.com
ServerAlias mail.*
ServerAdmin postmaster@mydomain.com
DocumentRoot /var/www/webmail
</VirtualHost>
```

Here's a breakdown of what's above:

**<VirtualHost 1.2.3.4:80>** - This indicates I'm setting up my mail interface as a virtual host, rather than IP based. Obviously, you're going to want to replace 1.2.3.4 with the IP address of your web server. Additionally, what you have here may vary from server to server and is dependent on how you have your Apache configured. Be cautious!

**ServerName mail.mydomain.com** - The official name of the webmail server virtual host.

**ServerAlias mail.\*** - This line establishes a wildcard serveralias called mail.\*. With this setup, any domain that is pointed to your server and that has an A record called "mail", will be able to get to the webmail interface by simply going to <http://mail.whateverdomain.com>. This is a pretty cool little feature and makes accessing the webmail interface easy for all of your users.

**ServerAdmin** [postmaster@mydomain.com](mailto:postmaster@mydomain.com) - The server administrative contact. This is not required, but I like to include it.

**DocumentRoot** [/var/www/webmail](#) - The document root of your webmail interface. This will vary, depending on where you chose to install Squirrelmail. In this example, you can see that I installed it at [/var/www/webmail](#)

**</VirtualHost>** - The closing tag to the virtualhost.

Make sure you restart Apache after making the above changes.

Ok, now that Apache is all configured, let's test the new webmail interface...

**<http://www.yourdomain.com/webmail>**

We'll sign in with the postmaster account under the domain you should have created earlier using Vqadmin...

**Username:** [postmaster@yourdomain.com](#)

**Password:** [your\\_password](#)

If all has gone well, Squirrelmail should log you right into your account! From here you will be able to both send and receive mail as well as a host of other additional functions. Again, Squirrelmail has tons of really cool plugins, and you can check them out at [Squirrelmail's plugins page](#). Installing the plugins is pretty easy and their site can help you out. Now that was nice and painless, wasn't it? If I had tried to explain installing Horde instead, you would probably be holding a gun to your head right now, wishing for quick end to the misery. OK, I'm only kidding. :)

Now, I'm going to cover the addition of 1 Squirrelmail plugin. Keep in mind, there are tons of other plugins available. We're going to install the "change\_pass" plugin which will allow our mail users to change their passwords from the Squirrelmail interface. This is made possible by the installaiton of Courierpassd that we did when we installed Courier-imap in the previous step.

So here goes...

**cd /path/to/squirrelmail\_directory/plugins** (example: `cd /var/www/webmail/plugins`)

Download the module...

**wget [http://squirrelmail.org/countdl.php?fileurl=http%3A%2F%2Fwww.squirrelmail.org%2Fplugins%2Fchange\\_pass-2.6-1.4.x.tar.gz](http://squirrelmail.org/countdl.php?fileurl=http%3A%2F%2Fwww.squirrelmail.org%2Fplugins%2Fchange_pass-2.6-1.4.x.tar.gz)**

**Unpack the module...**

**tar zxvf change\_pass-2.6-1.4.x.tar.gz**

Remove the tarball of the module...

**rm -rf change\_pass-2.6-1.4.x.tar.gz**

Now let's go and add the module into Squirrelmail...

**cd /path/to/squirrelmail\_directory/config**

Run the Squirrelmail configuration tool...

**./conf.pl**







Choose the option for "plugins". On my version of Squirrelmail, this was option 8. Once you are in the modules menu you should see the "change\_pass" module on the list of available, but inactive, modules. You can add the "change\_pass" module by simply typing the number associated with the module and then hitting enter. Once the module appears on the active module list, go ahead and save the configuration changes and then exit out of the configuration tool.

Alright! You should be all set now. All that's left to do log into Squirrelmail and try out the password change tool!

That's it for Squirrelmail. Now let's move on to the next step.

[Proceed to Part 14](#)

[qmailrocks@earthlink.net](mailto:qmailrocks@earthlink.net)

Color Coded Qmail Installation Key		
	Regular Black Text	Qmail installation notes and summaries by the author. Me talking.
	<b>Black Text</b>	Commands to be run by you, the installer.
	<b>Maroon Text</b>	Special notes for Redhat 9 users.
	<b>Red Text</b>	Vital and/or critical information.
	Regular/Bold Purple text	Denotes helpful tips and hints or hyperlinks.
	Regular Orange Text	Command line output.
	Regular green text	Denotes the contents of a file or script.

[home](#) | [about](#) | [the installation](#) | [scripts & utilities](#) | [faq](#) | [contact](#) | [links](#) | [donations](#) | [journal](#) | [mailing list](#) | [list archive](#) | [qmr forum](#)

Site last modified: **June 22, 2004 13:28:29**





## Part 14 - Clam Antivirus & SpamAssassin

In this next step we will install Clam Antivirus and SpamAssassin. ClamAV is a global email virus scanner that scans all incoming email and quarantines infected files. SpamAssassin is a spam filter that grades each incoming e-mail based on a ruleset and then labels e-mail as spam should the e-mail message exceed a certain score. Once we have both of these jewels installed, we will tie them into qmail via qmail-scanner, which gets installed on the next step.

Now let's install Clam Antivirus and SpamAssassin...

First, let's make sure you have all the required perl modules and required packages. If you read Part 0 of this install guide, you should have made sure that you installed a version of Perl from ftp.sunfreeware.com. If you're running with a default Sun Perl installation, you're going to really have a hard time getting these Perl modules installed. All of the perl modules below are required for Spamassassin and qmail-Scanner to work. I'll talk more about this further down in the install.

You will need these Perl Modules:

**Digest::SHA1**  
**Digest::HMAC**  
**Net::DNS**  
**Time::HiRes**  
**HTML::Tagset**  
**HTML::Parser**

You will also need these other packages:

**unzip**

If you don't have perl-suidperl or unzip installed, you will find RPM's of these 2 packages included in the Qmailrocks package.

### - Helpful Hints -

#### Checking/Installing Perl Modules

##### How do I know if my server has these perl modules?

The following script will check your system for the above modules. You must have "perldoc" installed to run this script. If you've installed Perl from ftp.sunfreeware.com, perldoc should be ready to use.

**You MUST run this script as a NON-ROOT user!!**

**/downloads/qmailrocks/scripts/util/check\_perlmods.script**

If you see an output similar to this: `/usr/lib/perl5/5.6.1/i386-linux/Time/HiRes.pm` You are good to go! However, if you get a "not found" type response for any of the above modules, you will need to install them.

##### Ok, so if I don't have the module(s), how do I down and install it(them)?

There are 3 main ways you can do this:

- 1. Qmailrocks included source packages:** I've included source versions of all needed perl modules. These can be found in the "perlmods" directory of the Qmailrocks package.
- 2. Directly from CPAN:** Go to <http://www.cpan.org>, get the module and install it. Alternatively, you can use the command line CPAN utility to connect to CPAN and install the module.

Ok, so at this point you should have all 4 of the above Perl modules installed on your system. So let's install Clam Anti Virus and then Spamassassin.

Installing Clam Anti Virus...

```
cd /downloads/qmailrocks/
```

```
gunzip clamav-0.73.tar.gz
```

```
tar xvf clamav-0.73.tar
```

```
cd clamav-0.73.tar.gz
```

```
groupadd clamav
```

```
useradd -d /tmp -s /bin/false clamav
```

```
./configure
```

```
make && make check && make install && make clean
```

At this point, you'll need to edit the `/usr/local/etc/clamav.conf` file, modifying the following lines.

"Example" - should be commented out.

"LogFile" - should be set to `/var/log/clamav.log`

"LogTime" - should be uncommmented.

"LogSyslog" - should be uncommented.

"ScanMail" - should be uncommented.

Now I'm going to throw in a small customization to Clam AV...

```
mv /usr/local/bin/clamscan /usr/local/bin/clamscan.orig
```

```
ln -s /usr/local/bin/clamscan /usr/local/bin/clamscan
```

Ok, Clam AV is now installed, but let's go ahead and set it up so that it will auto-update every night with the latest virus definitions! First we will want to set up the proper logging for the updater....

```
touch /var/log/clam-update.log
```

```
chmod 600 /var/log/clam-update.log
```

```
chown clamav:clamav /var/log/clam-update.log
```

Now let's make sure the server is able to get updates for Clam Antivirus...

```
/usr/local/bin/freshclam -l /var/log/clam-update.log
```

If the server is able to get updates, you should see an output similar to this:

```
[root@somewhere /var/log]/usr/local/bin/freshclam -l /var/log/clam-update.log  
ClamAV update process started at Fri Jun 11 10:45:01 2004  
Reading CVD header (main.cvd): OK  
main.cvd is up to date (version: 23, sigs: 21096, f-level: 2, builder: ddm)  
Reading CVD header (daily.cvd): OK  
Downloading daily.cvd [*]  
daily.cvd updated (version: 349, sigs: 787, f-level: 2, builder: ccordes)  
Database updated (21883 signatures) from database.clamav.net (80.69.67.3).
```

Woohoo! You're updated with the latest virus definiations from the Clam database!

Now we just set a crontab to run every night, which will run the auto-update procedure! In the example below, I've set mine to run every day at 2:00 AM.

**crontab -e** (make sure you run this command as root)

```
0 2 * * * /usr/local/bin/freshclam --quiet -l /var/log/clam-update.log
```

Now save your new crontab and exit. That's it! We're all done with Clam Anti Virus! You will now have a server-wide e-mail virus scanner that updates itself every night with the latest and greatest virus definitions!

**- Helpful Hints -**

**Knowing Clam Anti Virus**

1) Clam AV works pretty well right out of the box. However, there is a configuration file created at /etc/clamav.conf in case you want to customize it to your liking.

3) Clam logs to /var/log/clamav.log (in above scenario)

4) When Clam detects an e-mail that contains a potential virus, the following will happen:

- a) Clam AV quarantines the e-mail in /var/spool/qmailscan/quarantine

- b) Clam AV will send a notification of the detection and quarantine to whoever you configure it to send notifications to. When we install Qmail-Scanner further down this page, we will tell it what people to notify when a virus is detected. If you configure Qmail-Scanner by my rules, it will send 1 notification to the system administrator. However, it can also send a notification to the sender and the recipient as well, if you configure it to do so.

Now we install SpamAssassin...

If you don't have the Mail::Spamassassin Perl module installed on your system, let's install it now...

```
cd /path/to/qmailrocks/perlmods/source
```

```
gunzip Mail-SpamAssassin-2.63.tar.gz
```

```
tar xvf Mail-SpamAssassin-2.63.tar
```

```
cd Mail-SpamAssassin-2.63
```

```
perl Makefile.PL (This will ask some configuration questions. Use your best judgment)
```

```
make && make install
```

Alright. Now we're going to add a user/group called "spamd" under which Spamassassin will be run...

```
groupadd spamd
```

```
useradd -g spamd -s /bin/false spamd
```

We'll need to create a startup script for spamassassin - normally stored as /etc/rc.d/rc.spamd

```
vi /etc/rc.d/rc.spamd
```

```
#!/bin/sh
```

```
# Spamd init script for Solaris 9  
# August, 2th 2003  
# Martin Ostlund, nomicon
```

**# Modified slightly by Troy Belding for Qmailrocks - February 23, 2004**  
**# Modified slightly again by Eric Siegel for Qmailrocks.org - June 21st, 2004**

```
PATH=/sbin:/bin:/usr/sbin:/usr/bin:/usr/local/bin
DAEMON=/usr/local/bin/spamd
NAME=spamd
SNAME=spamd
DESC="SpamAssassin Mail Filter Daemon"
PIDFILE="/var/run/$NAME.pid"
PNAME="spamd"

DOPTIONS="-x -u spamd -H /export/home/spamd -d --pidfile=$PIDFILE --syslog-socket=inet"

KILL="/bin/kill"
KILLALL="/bin/killall"
# Defaults - don't touch, edit /etc/mail/spamassassin/local.cf
ENABLED=0
OPTIONS=""

set -e

case "$1" in
start)
echo -n "Starting $DESC: "
$PNAME $OPTIONS $DOPTIONS

echo "$NAME."
;;
stop)
echo -n "Stopping $DESC: "
$KILL -9 `cat $PIDFILE`
/bin/rm $PIDFILE
echo "$NAME."
;;
restart|force-reload)
echo -n "Restarting $DESC: "
$0 stop
$0 start
;;
*)
ME=/etc/init.d/$SNAME
echo "Usage: $ME {start|stop|restart|force-reload}" >&2
exit 1
;;
esac

exit 0
```

Save and exit from this new init script and then set permissions on it...

```
chmod 755 /etc/init.d/spamd
```

SpamAssassin is now installed, so let's configure it...

```
cd /etc/mail/spamassassin (where x.x.x is the version of perl installed)
```

```
vi local.cf
```

Make sure the following lines exist and are not commented out.

```
rewrite_subject 1
```

## required\_hits 5

Now save and exit out of the file.

And now we will configure the server to start SpamAssassin on boot:

**In /etc/init.d/spamd /etc/rc3.d/S88spamd**

Now save and exit out of the file.

Start up SpamAssassin:

**/etc/init.d/spamd start**

If all has gone well, both Spamassassin and Clam Anti Virus should now be installed and SpamAssassin should be running. With both of these programs installed, we can now install Qmail-Scanner.

[Proceed to Part 14](#)

[qmailrocks@earthlink.net](mailto:qmailrocks@earthlink.net)

Color Coded Qmail Installation Key		
	Regular Black Text	Qmail installation notes and summaries by the author. Me talking.
■	<b>Bold Black Text</b>	Commands to be run by you, the installer.
■	<b>Bold Maroon Text</b>	Special notes for Redhat 9 users.
■	<b>Bold Red Text</b>	Vital and/or critical information.
■	Regular/Bold Purple text	Denotes helpful tips and hints or hyperlinks.
■	Regular Orange Text	Command line output.
■	Regular green text	Denotes the contents of a file or script.

[home](#) | [about](#) | [the installation](#) | [scripts & utilities](#) | [faq](#) | [contact](#) | [links](#) | [donations](#) | [journal](#) | [mailing list](#) | [list archive](#) | [qmr forum](#)

Site last modified: **June 22, 2004 13:28:29**





## Part 15 - qmail-scanner w/qms-analog

If you will recall, when we compiled qmail earlier in this installation, we applied a patch to qmail called "qmailqueue.patch". This patch allows qmail to be configured to run with a substitute queuing mechanism. That's exactly what we're about to do here. We're going to tell qmail to use Qmail-Scanner as the queuing mechanism. Qmail-scanner is going to allow us to integrate Clam Antivirus and SpamAssassin into our qmail server's mail queue. Once qmail-scanner is installed, there will be a master script that is filled with configuration options that help you to tailor the functionality of Clam Antivirus and SpamAssassin to your needs. To expand the number of configuration options, we are also going to apply a patch to qmail-scanner. For this patch, we will be using [Mark Teel's qms-analog patch](#). Qms-analog incorporated the widely used qmail-scanner-st patch but it also adds some cool reporting functionality as well which we will utilize later in this installation guide. So let's get on it!

**cd /downloads/qmailrocks**

Unpack qmail-scanner...

**gunzip qmail-scanner-1.22.tgz**

**tar xvf qmail-scanner-1.22.tar**

Now unpack qms-analog...

**gunzip qms-analog-0.3.3.tar.gz**

**tar xvf qms-analog-0.3.3.tar**

Install qms-analog itself. This will come in handy in the next step when we install Qmailanalog.

**cd qms-analog-0.3.3**

**make all**

Next, we copy needed qms-analog files to the qmail-scanner source directory...

**cp qmail-scanner-1.22-st-qms-*YYYYMMDD*.patch /downloads/qmailrocks/qmail-scanner-1.22/**

**cp qms-config-script /downloads/qmailrocks/qmail-scanner-1.22/**

Now, let's apply the qms-analog patch...

**cd /downloads/qmailrocks/qmail-scanner-1.22**

**chmod 755 qms-config-script**

**gpatch -p1 < qmail-scanner-1.22-st-qms-*YYYYMMDD*.patch**

(That's not a typo, it's "gpatch". The gnu patch utility is your friend!)

Now continue with the qmail-scanner installation...

**groupadd qscand**

**useradd -g qscand -s /bin/false -d /tmp qscand**

Now we will configure qmail-scanner and install it. Ordinarily, you would run the ./configure script to configure and install qmail-scanner. However, Mark Teel has donated a handy little config script that does most of the work for you. This script is called "qms-config-script" and, if you look above, you should have already copied this config script into the qmail-scanner source directory.

So let's do it...

First, you need to configure the script for your needs...

```
cd /downloads/qmailrocks/qmail-scanner-1.22
```

```
vi qms-config-script
```

You will notice several fields that need to be customized to fit your needs. Let's have a look. I've highlighted the fields you should customize in **RED**

```
#!/bin/sh
```

```
if [ "$1" != "install" ]; then
INSTALL=
else
INSTALL="--install"
fi
```

```
./configure --domain yourdomain.com \
--admin postmaster \
--local-domains "yourdomain.com,yourotherdomain.com" \
--add-dscr-hdrs yes \
--dscr-hdrs-text "X-Antivirus-MYDOMAIN" \
--ignore-eol-check yes \
--sa-quarantine 0 \
--sa-delete 0 \
--sa-reject no \
--sa-subject ":SPAM:" \
--sa-delta 0 \
--sa-alt yes \
--sa-debug no \
--notify admin \
"$INSTALL"
```

Now save and exit out of the config file. That was easy, wasn't it.

And now we will run a test config for qmail-scanner...

```
./qms-config-script
```

Answer YES to all questions. If you get no errors, you can then run the script in "install" mode and this will install qmail-scanner on your server.

If you get any errors that are not related to setuid, check out these [troubleshooting tips](#).

If you didn't get any errors on the test run above, then you should be ok to run the "real" installation script below. So let's do it...

```
./qms-config-script install
```

Again, answer YES to all questions. If you get no errors, you can then run the script in "install" mode and this will install qmail-scanner on your server. If you do get errors, check out these [troubleshooting tips](#).

And now all that's left for qmail-scanner is to initiate the version file and the perlscanner database...

First, we'll initialize the version file. This command also helps to keep your server's /var/spool/qmailscan folder clear of rogue files that can develop when SMTP sessions are dropped. You may want to stick this command into your server's crontab and run it once a day. You'll see more on this in the "maintaining your qmail server" step near the end of this tutorial.. So let's run it...

```
setuidgid qscand /var/qmail/bin/qmail-scanner-queue.pl -z
```

And now we will generate a new perlscanner database for qmailp-scanner. For future reference, it's a good idea to run this next command whenever you upgrade qmail-scanner. You'll see more on this in the "maintaining your qmail server" step near the end of this tutorial. So let's do it...

```
setuidgid qscand /var/qmail/bin/qmail-scanner-queue.pl -g
```

A successful database build should produce the following output:

```
perlscanner: generate new DB file from /var/spool/qmailscan/quarantine-attachments.txt
perlscanner: total of 9 entries.
```

And now one final ownership check...

```
chown -R qscand:qscand /var/spool/qmailscan
```

Woohoo, qmail-scanner is installed! Now it's time to tie qmail-scanner into qmail itself.

```
vi /var/qmail/supervise/qmail-smtpd/run
```

To instruct Qmail to use Qmail-Scanner as the alternative queuing mechanism, we add the following line to the SMTP "run" script right under the first line (#!/bin/sh):

```
QMAILQUEUE="/var/qmail/bin/qmail-scanner-queue.pl" export QMAILQUEUE
```

..and we change the "softlimit" in that same script...

```
change softlimit to 40000000
```

**Note: It is absolutely vital that you change the "Softlimit" setting in this script. If you don't, qmail may fail to deliver mail!!!**

So now the qmail-smtp/run file should look like this:

```
#!/bin/sh
QMAILQUEUE="/var/qmail/bin/qmail-scanner-queue.pl" export QMAILQUEUE
QMAILDUID=`id -u vpopmail`
NOFILESGID=`id -g vpopmail`
MAXSMTPD=`cat /var/qmail/control/concurrencyincoming`
LOCAL=`head -1 /var/qmail/control/me`
if [ -z "$QMAILDUID" -o -z "$NOFILESGID" -o -z "$MAXSMTPD" -o -z "$LOCAL" ]; then
echo QMAILDUID, NOFILESGID, MAXSMTPD, or LOCAL is unset in
echo /var/qmail/supervise/qmail-smtpd/run
exit 1
fi
if [ ! -f /var/qmail/control/rcpthosts ]; then
echo "No /var/qmail/control/rcpthosts!"
echo "Refusing to start SMTP listener because it'll create an open relay"
exit 1
fi
exec /usr/local/bin/softlimit -m 40000000 \
/usr/local/bin/tcpserver -v -R -l "$LOCAL" -x /etc/tcp.smtp.cdb -c "$MAXSMTPD" \
-u "$QMAILDUID" -g "$NOFILESGID" 0 smtp \
/var/qmail/bin/qmail-smtpd your_domain.com \
/home/vpopmail/bin/vchkpw /usr/bin/true 2>&1
```

To activate all the changes we just made, we're going to have to completely stop and restart qmail.

Stop it...

```
qmailctl stop
```

and start it...

## qmailctl start

And a quick check of the qmail processes, just to be safe..

## qmailctl stat

Now it's time to test the whole damn thing to see if Qmail-Scanner, Spamassassin and Clam AV are all working correctly. Fortunately, Qmail-Scanner comes with it's own testing script that does a fantastic job. So let's test it!

```
cd /downloads/qmailrocks/qmail-scanner-1.22/contrib
```

```
chmod 755 test_installation.sh
```

```
./test_installation.sh -doit
```

A successful test should produce the following output. 2 messages should be quarantined by Clam Antivirus in /var/spool/quarantine/new and 2 messages should be set to whatever mailbox you specified in the Qmail-scanner configuration script. Don't worry if you don't get virus notification emails. The normal notification emails that get sent out upon virus detection usually don't work during the test.

setting QMAILQUEUE to /var/qmail/bin/qmail-scanner-queue.pl for this test...

Sending standard test message - no viruses...  
done!

Sending eicar test virus - should be caught by perlscanner module...  
done!

Sending eicar test virus with altered filename - should only be caught by commercial anti-virus modules (if you have any)...

Sending bad spam message for anti-spam testing - In case you are using SpamAssassin...  
Done!

Finished test. Now go and check Email for postmaster@mydomain.com

When you check the mailbox to which the tests were sent, you should see 1 Qmail-scanner test message, 1 Spamassassin test message, 2 Clam AntiVirus virus detection messages. If the test script above does not produce any errors and if you get all 4 test messages in your administrative mailbox, you can pop open a beer and have a little celebration! You've successfully installed all 3 packages! Woohoo!

### - Helpful Hints -

#### Post Install configuration tips for Qmail-Scanner

Although Qmail-Scanner should work pretty much "out of the box" so to speak, you can make some customizations to it's configuration by editing the qmail-scanner-queue.pl script located at /var/qmail/bin/qmail-scanner-queue.pl. The qmail-scanner-queue.pl script controls a lot of the functionality of both Clam AV and Spamassassin. Check it out for yourself and you will see that there are quite a few items you have control over. I wouldn't recommend touching most of them. In fact, the only setting that I changed in mine is in the Spamassassin section:

#### Can I have Spamassassin tag suspected spam with a custom subject line?

Yes. Edit the /var/qmail/bin/qmail-scanner-queue.pl file and find the following line:

```
my $spamc_subject=:SPAM:;
```

Now type a custom spam subject. This subject line will be added to any mails that Spamassassin tags as suspected spam. Here's an example:

```
my $spamc_subject=`Hi, I'm Spam`;
```

The "spamc\_subject" setting determines what message Spamassassin will append to the "subject" of e-mails which it deems as SPAM.

#### Can I delete e-mails that Spamassassin labels as spam?

Yes. Edit the `/var/qmail/bin/qmail-scanner.pl` file and find the following line:

```
my $sa_delete='0';
```

Now replace the '0' with a number that represents how far above your SpamAssassin "required\_hits" variable that Qmail-scanner should start deleting messages at. For example, if you SpamAssassin required\_hits variable is set to "5" and you set the "sa\_delete" variable to "1.0", then any message that has a spam score of 1.0 over the "5" mark would be deleted. In other words, any mail with a score of 6 or more would be trashed automatically. So for this example, you would change the "sa\_delete" variable as follows:

```
my $sa_delete='1.0';
```

### Is it safe to tell qmail-scanner to delete e-mails that SpamAssassin marks as spam?

Spamassassin has been tested to have up to a 99% accuracy rating in terms of detecting real spam and leaving legitimate e-mail alone. I've been using it for over a year now and have never gotten a false positive. Therefore, I feel safe in telling it to just delete the stuff.

There are a host of other Spam and Virus handling directives that can be customized with the qmail-scanner.pl file. You can check out the qmail-scanner patch website at <http://xoomer.virgilio.it/j.toribio/qmail-scanner/> for all the details.







Other than that, I left my qmail-scanner-queue.pl script as is.

### Summary of functionality:

If you've gotten to this point, you should have Clam Anti-Virus, Spamassassin and Qmail-Scanner all working together. When a message comes into the server, Qmail-Scanner takes the message and pipes it out to both Clam Anti-Virus and Spamassassin. If the message contains a virus, Clam AV quarantines it in `/var/spool/qmailscan/quarantine` and then send a notification e-mail to whoever you specified in the Qmail-Scanner installation. If the message does not contain a virus, it is then scanned by Spamassassin. Depending on the score that Spamassassin assigns to the message and whether or not that score breaks the SPAM threshold set by you in the `/var/qmail/.spamassassin/user_prefs` file, Spamassassin will either let the message go unaltered to its destination or it will tag the message as SPAM. If the message is tagged as SPAM, it will still arrive at its destination, but with an altered "subject" that will signal to the recipient that this was tagged as SPAM. The text that gets appended to the "subject" of the e-mail is set in the `/var/qmail/bin/qmail-scanner-queue.pl` file. (For example: If you set qmail-scanner-queue.pl to tag all SPAM with "HI, I'M SPAM!", mail tagged as such will be delivered to the recipient with "HI, I'M SPAM" added to the subject. Once the message is tagged, the recipient can then configure his/her mail client to deal with those tagged message in whatever manner he/she sees fit. Alternatively, you can tell Spamassassin to delete all suspected spam messages (like I do). You can find directions for this in the "Hints" box above.

[Proceed to Part 16](#)

qmlrocks@earthlink.net

Color Coded Qmail Installation Key	
	Regular Black Text   Qmail installation notes and summaries by the author. Me talking.
	<b>Black Text</b>   Commands to be run by you, the installer.
	<b>Maroon Text</b>   Special notes for Redhat 9 users.
	<b>Red Text</b>   Vital and/or critical information.
	<b>Purple Text</b>   Denotes helpful tips and hints or hyperlinks.
	<b>Orange Text</b>   Command line output.
	<b>Green Text</b>   Denotes the contents of a file or script.

[home](#) | [about](#) | [the installation](#) | [scripts & utilities](#) | [faq](#) | [contact](#) | [links](#) | [donations](#) | [journal](#) | [mailing list](#) | [list archive](#) | [qmr forum](#)

Site last modified: **June 22, 2004 13:28:29**





## Part 16 - QmailAnalog w/qlogtools & qms-analog

Qmailanalog performs some basic log analysis on those qmail log files and then outputs them to a desired location. In my case, I run qmailanalog every night and output the results to e-mail. Along with qmailanalog, we're going to install the "qlogtools" package. Qlogtools, as its name implies, provides an array of tools which can be used to analyze the qmail logs. We're going to use one of the qlogtool packages, tai64n2tai, to convert the timestamps on the log files from a machine readable format to a human readable format which will come to us when we get the finished report. After we've installed both Qmailstats and Qlogtools, we will create a script which you can run on a nightly basis to generate e-mail stats. The script will also incorporate qms-analog, which we installed with qmail-scanner previously. The qms-analog output will give add qmail-scanner stats to our nightly report.

First, let's install qmailanalog...

```
cd /downloads/qmailrocks/
```

```
gunzip qmailanalog-0.70.tar.gz
```

```
tar xvf qmailanalog-0.70.tar
```

```
cd qmailanalog-0.70
```

```
make && make setup check
```

That's it. Qmailanalog is installed!

Now let's install qlogtools...

```
cd /downloads/qmailrocks/
```

```
gunzip qlogtools-solaris-3.1.tar.gz
```

```
tar xvf qlogtools-solaris-3.1.tar
```

```
cd qlogtools-solaris-3.1
```

```
mkdir /usr/local/man (if directory already exists, you're good to go)
```

```
make
```

```
./installer
```

OK. The qlogtools library of tools should now be installed.

Now we will implement a script to run Qmailanalog and then you can hook that script into the server's crontab to get stats generated every night.

The script below is a solid script that sends an email to the server administrator with both the qmailanalog output as well as qms-analog's readout of qmail-scanner's activities. Pretty sweet, huh?

```
vi /var/qmail/bin/qmailstats
```

```
#!/bin/sh
# Qmailanalog invocation script
PATH=/usr/local/qmailanalog/bin:/var/qmail/bin:/bin:/usr/bin:/usr/local/bin
QMAILSTATS="/tmp/q.$$"
EMAILMSG="/tmp/qms.$$"
umask 077
```

```
cat /var/log/qmail/qmail-send/* /var/log/qmail/qmail-pop3d/* /var/log/qmail/qmail-smtpd/* | tai64n2tai | awk  
'{$1=substr($1,1,index($1,"")+6);print}' | matchup > $QMAILSTATS 5>/dev/null
```

```
DATE=`date +%D`
```

```
echo "To: postmaster@yourdomain.com" > $EMAILMSG  
echo "From: postmaster@yourdomain.com" >> $EMAILMSG  
echo "Subject: Nightly Qmail Stats Report for $DATE" >> $EMAILMSG  
echo "" >> $EMAILMSG  
echo "" >> $EMAILMSG  
echo "-----" >> $EMAILMSG  
zoverall < $QMAILSTATS >> $EMAILMSG  
echo "" >> $EMAILMSG  
echo "" >> $EMAILMSG  
echo "-----" >> $EMAILMSG  
zfailures < $QMAILSTATS >> $EMAILMSG  
echo "" >> $EMAILMSG  
echo "" >> $EMAILMSG  
echo "-----" >> $EMAILMSG  
zdeferrals < $QMAILSTATS >> $EMAILMSG  
echo "" >> $EMAILMSG  
echo "" >> $EMAILMSG  
echo "-----" >> $EMAILMSG  
echo "----- L a s t 2 4 H o u r s -----" >> $EMAILMSG  
cat /var/spool/qmailscan/qms-events.log | qms-analog 24 >> $EMAILMSG  
echo "-----" >> $EMAILMSG  
echo "----- A l l R e c o r d s -----" >> $EMAILMSG  
cat /var/spool/qmailscan/qms-events.log | qms-analog 0 >> $EMAILMSG  
cat $EMAILMSG | qmail-inject
```

```
rm -f $QMAILSTATS  
rm -f $EMAILMSG
```

Now set the script executable...

```
chmod 750 /var/qmail/bin/qmailstats
```

Now run the script...

```
/var/qmail/bin/qmailstats
```

Check your email and you should get a report with some pretty cool stuff in it! Your report should look [something like this](#).

OK, if the qmailstats script is working, you will now want to create a crontab entry to run this script every night.

So, as the "root" user let's set up a cron entry...

```
crontab -e
```

```
0 3 * * * /var/qmail/bin/qmailstats 1>/dev/null 2>/dev/null
```

Save and exit from the crontab editor and you should be all set. The above entry will run the qmailstats script every night at 3:00AM.

[Proceed to Part 17](#)

Color Coded Qmail Installation Key		
	Regular Black Text	Qmail installation notes and summaries by the author. Me talking.
■	<b>Black Text</b>	Commands to be run by you, the installer.
■	<b>Maroon Text</b>	Special notes for Redhat 9 users.
■	<b>Red Text</b>	Vital and/or critical information.
■	Regular/Bold Purple text	Denotes helpful tips and hints or hyperlinks.
■	Regular Orange Text	Command line output.
■	Regular green text	Denotes the contents of a file or script.

[home](#) | [about](#) | [the installation](#) | [scripts & utilities](#) | [faq](#) | [contact](#) | [links](#) | [donations](#) | [journal](#) | [mailing list](#) | [list archive](#) | [qmr forum](#)

Site last modified: **June 22, 2004 13:28:29**





## Part 17 - Installing Qtrap

Our final ingredient in this installation is going to be a domain level word filter, which I've named "Qtrap". This script is applied on a per domain basis and serves as a "bad word" scanner to catch any spam that Spamassassin may have missed. This filter serves as the last defense against SPAM before it arrived in your inbox. I like this filter because it helps to get rid of any SPAM that happens to make it by Spamassassin. Without any protection at all, my mailbox gets a shit ton of SPAM every day. Within the first 3 months I enacted the Qtrap filter, Qtrap logged over 9,000 deleted SPAM messages, none of which were legitimate e-mails. My keyboard's delete key was very appreciated the extra rest.

This script is actually a hacked up version of another script I found somewhere that simply checked the SUBJECT line of incoming mail against a list of banned words. If a word in the subject matched, the mail would be bounced back to the sender with an explanation of why it was dropped.

My hacked up version of this script does it a little different: The entire body of all incoming messages to the designated domain is scanned and matched against a list of banned words. If a match is found, the message is simply dropped (deleted) instead of being bounced. Additionally, the drop action is then logged to a log file for later review and troubleshooting by the system administrator. I also expanded the logging features of the script to be a bit more useful and informative.

So let's install it...

```
cd /var/qmail/vpopmail
```

```
mkdir spamcheck
```

```
cd spamcheck
```

```
cp /downloads/qmailrocks/scripts/qtrap/qtrap-1.2 ./
```

```
vi qtrap-1.2
```

within the Qtrap-1.2 script, you should see a section of code that looks like this:

```
checkall () {
case $BANNED_WORDS in
porn|PORN|Sex|SEX)
printout $BANNED_WORDS
echo MESSAGE DROPPED from $SENDER because of $BANNED_WORDS on `date "+%D %H:%M:%S"` >>
/var/qmail/vpopmail/spamcheck/logs/qtrap.log
exit 99;;
*)
;;
esac
}
```

The portion of the above section that I've highlighted in **RED** is the array of "banned" words. Edit this array to your satisfaction. Make sure that each word is separated by a pipe "|" and keep in mind that the array is case sensitive. So the words "SEX" and "Sex" are 2 different words. Also, exercise caution here. You don't want to ban words that are used in everyday e-mails. For example, you wouldn't want to ban the word "hello" or something like that. You should only ban words that you are 100% sure you would never see in a legitimate e-mail.

Now let's set up the logging directory...

```
mkdir logs
```

```
cd logs
```

```
touch qtrap.log
```

```
chown -R vpopmail:vchkpw /var/qmail/vpopmail/spamcheck
```

```
chmod -R 755 /var/qmail/vpopmail/spamcheck
```

Now we will add this script into the mail path for a domain on our server.

```
cd /var/qmail/vpopmail/domains/yourdomain.com
```

```
vi .qmail-default
```

add the following line above the line that is already there

```
| /var/qmail/vpopmail/spamcheck/qtrap-1.2
```

Here's an example:

.qmail-default before:

```
| /var/qmail/vpopmail/bin/vdelivermail " delete
```

.qmail-default after:

```
| /var/qmail/vpopmail/spamcheck/qtrap-1.2
| /var/qmail/vpopmail/bin/vdelivermail " delete
```

Save these changes and that should be it. You don't have to restart anything. To test this last rule, try sending an e-mail to your mailbox and make sure that the test e-mail contains one of the words that you entered into the "bad word" list in the Qtrap script. If the filter is working right, the message should NOT arrive in your inbox. You should then be able to view the log file at /home/vpopmail/spamcheck/logs/qtrap.log and see a log of the dropeed message corresponding to the time at which you sent the test message. The drop log should look something like this:

**MESSAGE DROPPED from someone@somewhere.com because of some\_banned\_word on on 06/13/03 02:37:51**

If the test was successful, then that's it! Congratulations, you've completed the Qmailrocks.org Qmail installation. Have fun. The next couple steps discuss cleanup as well as some closing notes and suggestions.

[Proceed to Part 18](#)

[qmailrocks@earthlink.net](mailto:qmailrocks@earthlink.net)

Color Coded Qmail Installation Key	
	Regular Black Text   Qmail installation notes and summaries by the author. Me talking.
<b>Black</b>	<b>Bold Black Text</b>   Commands to be run by you, the installer.
<b>Maroon</b>	<b>Bold Maroon Text</b>   Special notes for Redhat 9 users.
<b>Red</b>	<b>Bold Red Text</b>   Vital and/or critical information.
<b>Purple</b>	<b>Regular/Bold Purple text</b>   Denotes helpful tips and hints or hyperlinks.
<b>Orange</b>	<b>Regular Orange Text</b>   Command line output.
<b>Green</b>	<b>Regular green text</b>   Denotes the contents of a file or script.

[home](#) | [about](#) | [the installation](#) | [scripts & utilities](#) | [faq](#) | [contact](#) | [links](#) | [donations](#) | [journal](#) | [mailing list](#) | [list archive](#) | [qmr forum](#)

Site last modified: **June 22, 2004 13:28:29**





## Part 18 - Maintaining your qmail server

Once you've got your qmail server up and running, how do you take care of it? This page will cover the many answers to that question. So here goes.

Table of contents:

- [Making sure that all services start on boot](#)
- [Maintaining the qmail queue](#)
- [Maintaining qmail-scanner](#)
- [Maintaining SpamAssassin](#)
- [Maintaining Clam Antivirus](#)
- [Maintaining current software versions](#)
- [Maintaining the qmail logs](#)
- [Maintaining administrative mailboxes](#)
- [Maintaining other mailboxes](#)
- [Backing up your qmail server](#)

### Making sure that all services start on boot

If you've installed qmail correctly, it should already automatically start when you boot your server. However, you will want to make sure that all of the other needed services start as well.

#### --For Redhat users--

Starting Courier-imap on boot - make sure the following 2 lines exist in your server's /etc/rc.local file:

```
/etc/rc.d/init.d/imap start  
/etc/rc.d/init.d/imapd start
```

Starting SpamAssassin on boot

Try running the "setup" command and check the system services area to see if SpamAssassin is listed and marked with a "\*" to start on boot. If SpamAssassin is not present in the system's "setup" tool, you can start it on boot by added the following line to the server's /etc/rc.local file

```
/etc/rc.d/init.d/spamd start
```

Starting Apache on boot

Try running the "setup" command and check the system services area to see if the "httpd" service is listed and marked with a "\*" to start on boot. If it is not present in the system's "setup" tool, you can start it on boot by added the following line to the server's /etc/rc.local file

### Maintaining the qmail queue

It's usually not a bad idea to keep your qmail server's queue in check. Your qmail server's queue is located at /var/qmail/queue. However, it's just about impossible to look directly at the queue folders and be able to tell what's going on.

**qmailctl stat** - This, of course, shows you your qmail server's current status. Included in the stats is the current condition of the queue. You can use this to see how many messages are sitting in the queue.

**qmHandle** - This add-on tool allows more in depth viewing of the queue and it also allows you to perform administrative functions on the queue. You can find instructions about installing qmHandle [here](#). Once qmHandle is properly installed, running the command "qmHandle" will provide a listing of all the possible commands possible. I'll go over a couple of them right now. The "qmHandle -l" command will give

you a complete listing all every message in the queue and a summary containing the date the message was sent, the sender and the intended recipient. The output for a single message might look like this:

**6406395 (195, R)**  
**Return-path: bob@somewhere.com[]**  
**From: Bob Smith <bob@somewhere.com>**  
**To: Frank Smith <frank@wherever.com>**  
**Subject: Re: This weekend**  
**Date: Mon, 16 Feb 2004 12:14:31 -0700**  
**Size: 1482 bytes**

The message number, 6406395, can be seen at the top of the message. If we wanted to delete this single message from the queue, we could do so with a command of "qmHandle -d6406395". The entire queue can be cleared out with the command "qmHandle -D".

**queuelifetime** - The "queuelifetime" setting for qmail determines how long messages can stay in the queue. By default, your qmail server will keep messages in the queue for 604,800 seconds, or 7 days. However, you can set a custom queuelifetime by creating a file called /var/qmail/control/queuelifetime. The content of that file is a single line containing a number which represents the number of seconds the queue will hold any given message. If you want to set a custom queuelifetime, you might want to whip out the old calculator.

### A little advice on handling your queue

You need to keep in mind that a queued message is NOT necessarily a BAD thing. The only time a message is queued for any length of time is when it is undeliverable at the time it is originally sent. A message is usually undeliverable for 1 of 2 reasons:

**1)** The receiving mail server is offline. If this is the case, when that mail server comes back online at a later time your qmail server will then be able to deliver that message. A perfect example of this is the Qmailrocks.org mailing list. At any given time there lots of messages in the mailing list server's queue. However, these messages usually get delivered eventually. As you can imagine, some people on the mailing list are probably using a new qmail server to house the mailbox with which they have signed up on the list with. Well, of course, since that person is new the qmail there is a chance that their server may go down for periods of time while they are working on it and perfecting their qmail skills. It is at these times when their server is unreachable and when my list server then queues the message for later delivery. No big deal.

**2)** The recipient address is bogus or incorrect. If you get hit with a lot of spam on your server, or if you have a spammer on your server, this will probably happen to you at some point. Your queue gets filled with message bound for bogus addresses OR it gets filled with bounce messages that your server is trying to deliver back to spammers who sent spam to bogus addresses on your server. A good way to cut down on this is to set the domains on your server to "delete" catchall mail instead of bouncing it. This can be done from within the Qmailadmin interface. Setting a domain's catchall setting to "bounce" is a bad idea in my experience as it only results in a queue full of bounced bounce messages. If you don't need a catchall for your domain, do yourself a favor and set it to "delete". This is also the case if you are running Qmail-scanner. Qmail-scanner has an option to notify the sender when a virus is found in an e-mail. Bad idea. Most of the time, the address from which that virus laden e-mail came is bogus. So trying to "notify" the sender usually results in nothing but a bunch of double bounces flying all over your server.

Fortunately, since your qmail server has a built in queue lifetime, messages will eventually drop out of the queue if they are undeliverable. The last thing I want to mention is that it is a common misconception that if your queue is full, mail being sent presently will not be delivered quickly. This is a misconception. As I said, the queue is a repository for messages that are not immediately deliverable. If a message is being sent to a valid address it will get sent immediately, regardless of the size of the queue. So, in summary, you don't need to panic every time you have messages in your queue. Most of the time, the best thing to do is to just let your queue take care of itself. However, there are time when you may deem it necessary to clean out the queue or take other administrative action, and that's what the "qmHandle" tool above is good for.

### Maintaining qmail-scanner

Qmail-scanner is fairly easy to maintain. Once you get it configured how you want it to be, the main items your going to want to worry about are 1) the log files and 2) the virus quarantine area. Here's some info on both.

**1) qmail-scanner logs** - Qmail-scanner, when Clam Anti Virus and SpamAssassin are hooked into it, logs the virus scanning activities to /var/spool/qmailscan/qmail-queue.log. This log file can get REALLY big, so you will want to keep it in check. You may want to set up a rotation schedule for this log file or some other sort of scheduled task that deals with this log file on a routine basis. If this log file exceeds the linux file size limit of 2GB, your mail server will start freaking out and all hell will break lose. So do yourself a favor and keep an eye on this log file.

**2) The virus quarantine area** - When qmail-scanner pipes the mail out to Clam Anti Virus and virus is found, the virus laden message is quarantined at /var/spool/qmailscan/quarantine/new. Those e-mails will usually not pose any threat to your server, since there are very few Linux/Unix viruses and since they are not being executed. However, on a busy mail server, that folder can get filled up pretty quickly, so you may want to keep an eye on it and have it emptied on a routine basis. I empty mine out with a crontab that runs once a week.

Anytime you upgrade qmail-scanner, it's a good idea to refresh the qmail-scanner perlscanner database:

**setuidgid qscand /var/qmail/bin/qmail-scanner-queue.pl -g**

### Maintaining SpamAssassin

SpamAssassin is relatively maintenance free one you get it up and running. It logs it's activities to /var/log/maillog, so you can always reference the logs for any investigations. New versions of SpamAssassin are released fairly often, so you may want to occasionally check <http://www.spamassassin.org> to see if there have been any new releases. In my experience, I've always been able to install the new version over the older version with no problems. If you upgrade, just make sure the /etc/mail/spamassassin/local.cf file still contains the setting you want and you should be in good shape.

Anytime you upgrade SpamAssassin, it's a good idea to refresh the qmail-scanner perlscanner database:

**setuidgid qscand /var/qmail/bin/qmail-scanner-queue.pl -g**

### Maintaining Clam Antivirus

The only things you really need to do to maintain Clam AV are:

1) Make sure your server is running the automatic virus definition updates on a regular basis. I run the following command out of crontab on a routine basis:

```
/usr/bin/freshclam --quiet -l /var/log/clamav/clam-update.log
```

2) Keep your version relatively current. New releases are put out fairly often, so keep an eye on it. Again, in my experience I've been able to install the newer version over the older one with no problems.

3) Anytime you do decide to upgrade Clam Antivirus, you will need to update qmail-scanner's version file. This is easily done by running the following command:

**setuidgid qscand /var/qmail/bin/qmail-scanner-queue.pl -z**

### Maintaining current software versions in general

Naturally, as time passes new and improved versions of software will be released from their respective vendors. It's always nice to have to latest versions of everything, but unless the newer version fixes a security hole or a major bug you don't need to lose sleep over it. You can check the vendors' websites every now and then to see if a new version is out. I try to keep the QMR software package as current as possible and I'm pretty good at it. Most of the time, the qmailrocks.tar.gz package will contain the latest versions of everything.

I've never had any big problems with upgrading any of the software. For source packages, you can usually simply compile and install the latest version right over the older version with no problems. I mention this in the above paragraphs on SpamAssassin and Clam AV, but it generally applies to all of the software found on this site.

Qmail itself has not had a new version release in a very long time, but you can bet if there is a new release I will have it here. As you probably know by this point, the current qmail version is version 1.03. Many people have noticed that qmail.org offers a version of qmail called "netqmail-1.05". At first glance this may appear to be a newer version of qmail. IT IS NOT. Netqmail is simply qmail-1.03 with some of the patches (which I use on this site anyway) already built in. If you follow the QMR installation guide using 1.03, the resulting installation of qmail is not different that if you used netqmail. Understand? If you don't trust me, check out the full description of netqmail [here](#).

### Maintaining the qmail logs

Fortunately, qmail's logs take care of themselves. They automatically rotate all on their own, so you never have to worry about them. The only thing worth noting is that you CAN customize the rotation schedule for the logs. This is done the "logs" supervise script for each supervised function. Confused? Ok, I'll explain.

In the QMR qmail installation, there are 3 supervise scripts and, subsequently, 3 logs for those scripts.

**/var/qmail/supervise/qmail-pop3d/run** operated the pop server and is logged via **/var/qmail/supervise/qmail-pop3d/log/run**

**/var/qmail/supervise/qmail-smtpd/run** operates the smtp server and is logged via **/var/qmail/supervise/qmail-smtpd/log/run**

**/var/qmail/supervise/qmail-send/run** operates the mail processor and is logged via **/var/qmail/supervise/qmail-send/log/run**

Each of the above "log/run" scripts tells the server how it wants those activities logged. Let's take the /var/qmail/supervise/qmail-pop3d/log/run logging script as an example:

```
#!/bin/sh
PATH=/var/qmail/bin:/usr/local/bin:/usr/bin:/bin

export PATH

exec setuidgid qmail multilog t s100000 n20 /var/log/qmail/qmail-pop3d 2>&1
```

The last line of this script sets some of the logging options. We can break that last line into several parts:

```
exec setuidgid qmail multilog t s100000 n20 /var/log/qmail/qmail-pop3d 2>&1
```

So let's break it down:

**exec setuidgid qmail multilog** - run the multilog program as the "qmail" user.

**t** - inserts an @, a precise timestamp, and a space in front of each line.

The above "t" is why the rotated logs take on names like this: @40000000402d1c562cbf3534.s

**s100000** - logs will rotate when they reach 100000 bytes.

**n20** - number of rotations to keep on hand before they fall off.

**/var/log/qmail/qmail-pop3d 2>&1** - the directory to where the logs will be written, silently.

## Maintaining administrative mailboxes

This may seem a bit silly, but you'd be surprised how many people neglect the administrative e-mail account on their server. But what is the administrative e-mail account? Well, that depends on you. Administrative mail would be mail destined for the server's root user as well as any notification settings you may have. The destination for most of the server's administrative addresses is usually determined by the aliases you have created at /var/qmail/aliases. If you've followed the QMR install guide, you should have a "postmaster" alias as well as "root", "mailer-daemon" and "anonymous" aliases. If you're like me, you direct all of these aliases to a single mailbox on your qmail server. For example, on the qmailrocks.org server, I direct all of these aliases to the qmailrocks.org postmaster account. This means that my postmaster account gets all of the administrative mail on the server. I get bounce failure messages, crontab reports, log watch reports and many other admin type emails. In addition, since I also have qmail-scanner set up to send virus reports to this same address. So, as you can see, my postmaster account is the central locus for all the server's administrative mail. You probably want to do the same with your server, as these administrative e-mails can often help you to find and correct problems that might otherwise go unnoticed. A mistake that people make a lot is to have the administrative mail directed to some mail account that they never check. This inevitably leads to that person being surprised when they find out that they have a mailbox on their server that's a couple Gigs in size. These people will also be surprised when they find that that mailbox is full of error messages that have been coming in for months indicating that something on the server is misconfigured. They never checked the mailbox, so they never knew. Well, I guess ignorance is bliss. So my point of this whole paragraph is that you should keep your administrative mail configuration organized and well cared for. It will save you a lot of heartache down the road.

## Maintaining other mailboxes

Managing all other mailboxes on your server is made easy by simply setting quotas on all domains. It is inevitable that if you host mail on your server, there will be some idiot who either never checks his mailbox or decides that he has to store a lifetime worth of mail on the server. Setting quotas for your domains is a way of keeping these idiots in check and preventing you from having a disk space crisis because of these idiots. If you have a 120GB drive in your server and no quotas, there WILL be some fool who fills up all 120GB with his mail. So take my advice and set rigid quotas for your domains.

## Backing up your qmail server

Backing up a qmail server is relatively easy. While different people may give you slightly different recommendations, you can ensure a safe backup of your qmail server if you backup the following directory:

**/var/qmail** - backs up all of your qmail settings as well as all vpopmail data. Wasn't that easy?

[Proceed to Part 19](#)

[qmailrocks@earthlink.net](mailto:qmailrocks@earthlink.net)

Color Coded Qmail Installation Key		
	Regular Black Text	Qmail installation notes and summaries by the author. Me talking.
■	<b>Black Text</b>	Commands to be run by you, the installer.
■	<b>Maroon Text</b>	Special notes for Redhat 9 users.
■	<b>Red Text</b>	Vital and/or critical information.
■	<b>Purple text</b>	Denotes helpful tips and hints or hyperlinks.
■	<b>Orange Text</b>	Command line output.
■	<b>Green text</b>	Denotes the contents of a file or script.

[home](#) | [about](#) | [the installation](#) | [scripts & utilities](#) | [faq](#) | [contact](#) | [links](#) | [donations](#) | [journal](#) | [mailing list](#) | [list archive](#) | [qmr forum](#)

Site last modified: **June 22, 2004 13:28:29**





## Part 19 - Mail Client Configuration

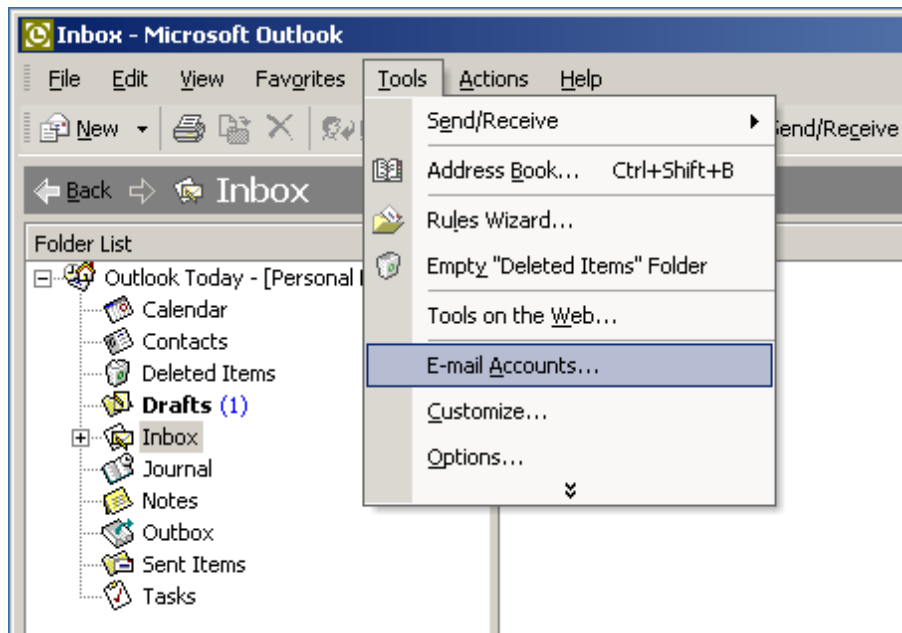
So now you've got a fully functioning mail server. Well, now it's time to make use of that mail server and configure a remote mail client to retrieve mail from the server and send mail through the server. For my example, I've chosen Microsoft's Outlook XP. Outlook Express varies slightly, but you should be able to get the general idea. Likewise, if you using some other mail client you should be able to get the idea. Please do not e-mail me asking for instructions on another mail client. This is the only one I'm going to provide for now.

Anyway, let's configure Outlook...

For this example, let's set the following pre-conditions:

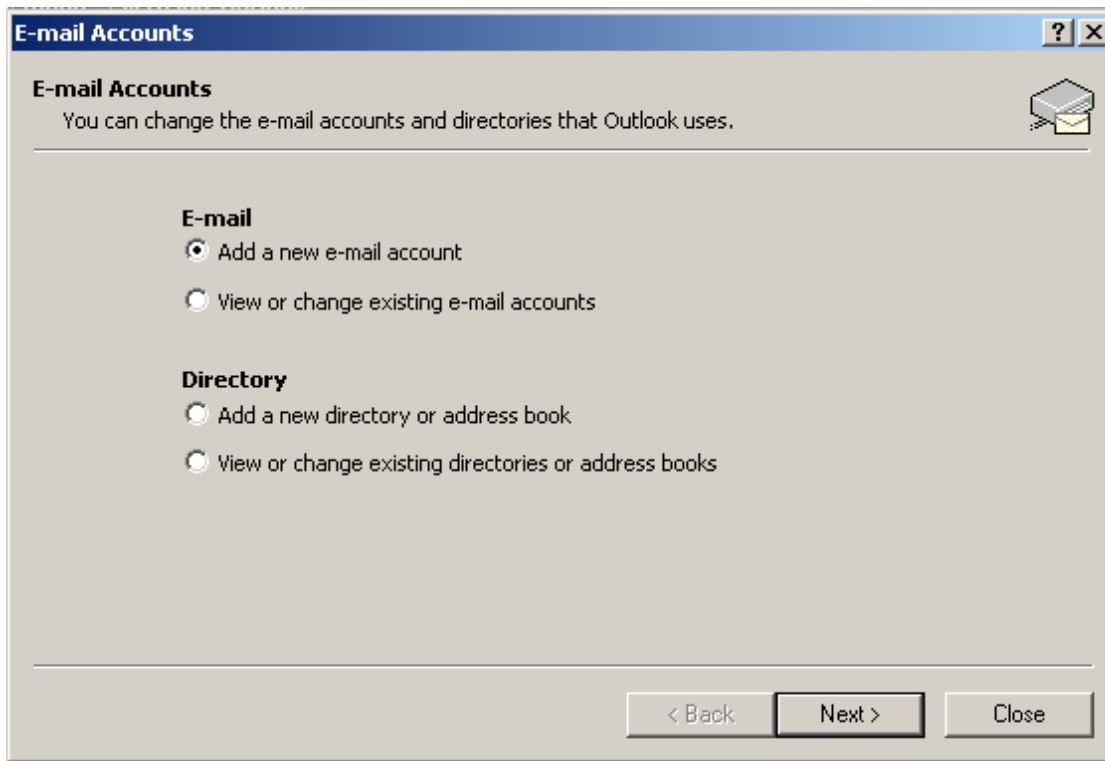
1. I have used Vqadmin to set up a new domain on my server called "domain.com"
2. Since I have set the domain up using Vqadmin, there should already be a main user for the domain called "postmaster@domain.com".
3. I'm going to set up Outlook to check mail for and "postmaster@domain.com" and will also use that user for SMTP authentication, so that I can send mail.

### Step 1: Go to E-mail account configuration



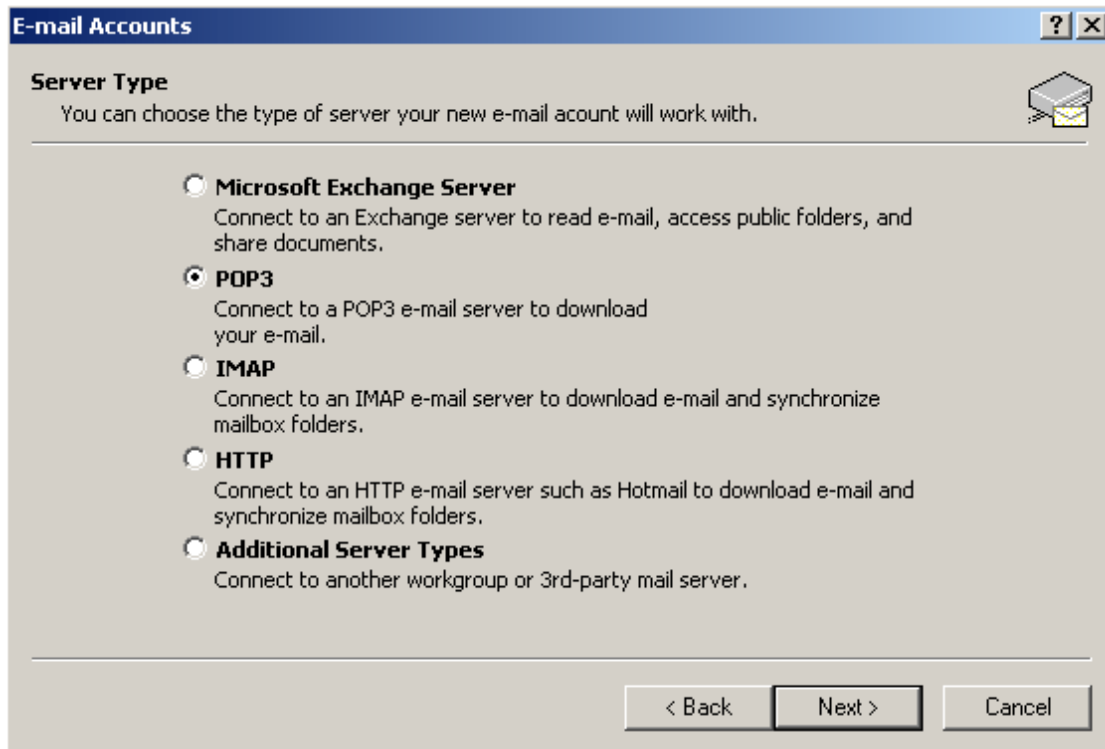
Select "E-mail Accounts" under Tools.

### Step 2: Add a new E-mail account



When finished, click "Next".

### Step 3: It's a POP3 type of account



When finished, click "Next".

### Step 4: Configure the account

**E-mail Accounts** ? X

**Internet E-mail Settings (POP3)**  
Each of these settings are required to get your e-mail account working.

**User Information**

Your Name:

E-mail Address:

**Server Information**

Incoming mail server (POP3):

Outgoing mail server (SMTP):

**Logon Information**

User Name:

Password:

Remember password

Log on using Secure Password Authentication (SPA)

**Test Settings**

After filling out the information on this screen, we recommend you test your account by clicking the button below. (Requires network connection)

**Click on "More Settings" once you've filled out the above information.**

Step 5: Enable SMTP Authentication (under the "more settings" area):

**Internet E-mail Settings** ? X

General | **Outgoing Server** | Connection | Advanced

My outgoing server (SMTP) requires authentication

Use same settings as my incoming mail server

Log on using

User Name:

Password:

Remember password

Log on using Secure Password Authentication

Log on to incoming mail server before sending mail

**Save all changes and give it a go! You should be all set. Now that wasn't so hard, was it?**

**SMTP Settings Troubleshooting Tip**







You may find that when you configure Outlook to use your Qmail server as the outgoing SMTP server, those connection attempts get blocked. You may get an error message like "Could not connect to server". Often, this is due to your ISP blocking foreign connections to port 25 (smtp). Most large ISP's these days will block attempts to connect to a foreign SMTP server. This is an anti-spam measure. The solution is to use your ISP's SMTP server instead of your own.

Let's take a real-life example:

Earthlink, for example, blocks connection attempts to foreign SMTP servers. I use Earthlink DSL, therefore I have to configure Outlook to use Earthlink's SMTP servers (mail.earthlink.net) instead of my Qmail server's. So, in the SMTP server setting in Outlook, I stick in "mail.earthlink.net" instead of my server's address. Of course, whether or not you configure Outlook to use SMTP authentication at this point is strictly determined by what your ISP's SMTP connection rules are. If you need help, contact your ISP.

[Proceed to Closing Notes](#)

[qmailrocks@earthlink.net](mailto:qmailrocks@earthlink.net)

Color Coded Qmail Installation Key		
	Regular Black Text	Qmail installation notes and summaries by the author. Me talking.
	<b>Black Text</b>	Commands to be run by you, the installer.
	<b>Maroon Text</b>	Special notes for Redhat 9 users.
	<b>Red Text</b>	Vital and/or critical information.
	Regular/Bold Purple text	Denotes helpful tips and hints or hyperlinks.
	Regular Orange Text	Command line output.
	Regular green text	Denotes the contents of a file or script.

[home](#) | [about](#) | [the installation](#) | [scripts & utilities](#) | [faq](#) | [contact](#) | [links](#) | [donations](#) | [journal](#) | [mailing list](#) | [list archive](#) | [qmr forum](#)

Site last modified: **June 22, 2004 13:28:29**

