

Gentoo Guide:

- [Installation](#)
 - [Finalizing The Installation](#)
 - [Tools](#)
 - [MySQL Database Server](#)
 - [Apache Web Server](#)
 - [PHP](#)
 - [Mail \(Sendmail/SSMTP\)](#)
 - [MySQL Backup](#)
 - [Protecting Your Web Directories With .htaccess](#)
 - [PHPMyAdmin](#)
 - [Webalizer](#)
 - [TeamSpeak2 Server](#)
 - [GenSplash Framebuffer](#)
 - [Getting a GUI, Gentoo and X](#)
 - [Sound, Gentoo and ALSA](#)
 - [Window Managers](#)
 - [IRC Server](#)
-

Installation

Gentoo Linux is my OS of choice. It is highly customizable, has no extra bloat, and can be tailored and fine tuned to the system it is running on. If you really want to learn how to use Linux as well as what makes it tick then install Gentoo from scratch! You will be amazed at how much you will learn, not only about Gentoo and Linux, but also about the hardware inside your PC.

The best way to install gentoo is to follow the handbook for your particular arch found [here](#). Then download the Gentoo Minimal/Install CD found [here](#).

Follow the handbook and it will get you up and running with the latest updated version of Gentoo. I use the handbook for every installation I do, it is an excellent resource. Once you are done you should have a basic Gentoo installation with a user created. When you get to page 12 "*Where to go from here?*" check out the links it offers then come back and check out the next section of this guide: [Finalizing the Installation](#).

Finalizing the Installation

Ok, so you followed the handbook and completed your installation. Now what? Well one of the last things the guide had you do was create a user. Here is some info about the groups that you added your user to and some others that are available. Note that some of these groups may not be available until you install a certain package.

- disk - Direct access to disks
- wheel - su (change to super user / root)
- floppy - Direct access to floppies
- audio - Access to audio devices
- cdrom - Direct access to CD-ROM's (includes CD-RW and DVD-ROM/RAM/RW)
- video - Access to video capture hardware, hardware acceleration
- games - Access to all games
- usb - Use of USB devices
- users - Standard group of users
- apache - Allows write access to Apache files
- mysql - Allows read/write access to MySQL files
- plugdev - Allows user to mount and use pluggable devices such as cameras and USB sticks

In order to add a user to any of these groups do the following:

- Log in as root.
- Type `gpasswd -a username group`
- Then log the user out then back in again to have the change take effect.

Ok now that thats done lets update the portage database:

- Log in as root
- Type `emerge --sync`
- You will probably be told that you need to update portage so type `emerge -av portage` and select yes.

Now its time to get your base install up to date. So type `emerge -avtDNu world`,

this will spew up a long list of packages, review them and make sure that the use flags are correct. If you installed by the manual you will know what I mean. If not go back and read the manual! Then select yes to run the updates. This will probably take a few hours as it is going to download all of the files and compile them. I usually like to run this one right before I go to bed at night and it is usually done by morning.

When the updates are complete you will need to update the config files that need changes. So type `etc-update`, this will probably spew up a list of files that need changing and you need to go through each one individually. This is one of those the more you do it the easier it gets kind of things. A general rule to follow however is:

Always overwrite files in `/etc/init.d/` with the new versions. Sometimes overwrite files in `/etc/udev/` or `/etc/pam.d/` , and Never overwrite files you changed yourself.

Usually when dealing with these I tend to overwrite anything I did not change myself, then I really have to look at the files I did make changes to, to make sure that the updates they are making are necessary or not. Sometimes they are just comments. If they are necessary then run the interactive merge option in `etc-update`. Otherwise just use `overwrite` or `replace`.

Now we need to remove any of the old packages and dependences that are no longer needed by the system. Type `emerge -av --depclean`, make sure you review the packages before you type yes to make sure its not trying to remove a package that you need.

Next thing to do is to install the `gentoolkit`. This package includes `revdep-rebuild` which is a necessary tool to help fix broken dependencies and packages. Type `emerge -av gentoolkit` and select yes. Once the install is done type `revdep-rebuild` and press enter. This will take a few mins and if all is good it will just end, if not then it will recompile any necessary packages which could take some time.

Once all this is completed reboot your PC by typing `reboot` at the command prompt. Make note of those commands since they are the commands you will use every time you want to update your Gentoo install.

To review:

- `emerge --sync`
- `emerge -avtDNu world`

- `etc-update`
- `emerge -av --depclean`
- `revdep-rebuild`

You can also use the following three commands before you use *revdep-rebuild* to further check and clean your system:

- `eclean distfiles` - Cleans out obsolete distfiles to free up space
- `glsa-check -f affected` - Checks installed packages to see if they have any identified security risks and installs the appropriate updates.

[Back to Top](#)

Tools

In the last part we finalized the installation and updated our system. Here are a couple of scripts that will automate portage to automatically sync itself once a week. If you followed the Gentoo Handbook when you did your initial base install you should have installed `vixie-cron`, which we will use to perform this task.

This script will update the portage tree once a week:

- As root type `nano -w /etc/cron.weekly/portage.cron` this will open up a new file called `portage.cron`.
- In the new file type the following line: `exec /usr/bin/nice /usr/bin/emerge --sync --quiet`
- Save the file and exit then type `chmod +x /etc/cron.weekly/portage.cron` to make it executable.

Now let's pull down some handy tools to make working on our new Gentoo Linux box a bit easier. First things first we need to enable SSH. By default it is already installed but not running. So as root do the following:

- Type `/etc/init.d/sshd start` to start the ssh daemon.
- Type `rc-update add sshd default` to add it to the default run level so the service starts when you boot your PC.

Great now we have SSH enabled so you can now work on that PC from anywhere as long as you have access to your LAN via VPN, or sitting on

your couch with a laptop watching Family Guy. You can also forward port 22 (be careful!) to your PC's IP address. Once this is done you can access it via putty on a windows pc or from a terminal on another Linux box with the command: `ssh XXX.XXX.XXX.XXX` - obviously insert the PC's ip address instead of the X's.

SSH is great and all but what happens when you remotely access your box and you start a compile or update and your connection drops, or you have to close your laptop and go? Well then your compile or update or any other process you were running in the remote shell is lost. But there is an easy fix to that. A wonderful program called screen! Screen lets you run a process in a terminal, close the window and the process will still be running for you to call back up later. To install screen simply use emerge:

```
emerge -av screen
```

Once installed just type `screen` in the terminal to open up a new screen. You can run multiple screen sessions and you can list your screens by typing `screen -list`. You call up existing screen sessions by typing `screen -dr xxxxx`, where xxxxx is the number that is listed when you run `screen -list`. It is a very simple tool, and once you use it you will find you can't live with out it. Type `screen -help` OR `man screen` for more info.

Next we are going to need to install some packages to handle .zip and .rar files. By default our install already takes care of .tar .gz and .bz2 files. To get zip and rar do the following: As root type `emerge -av zip unzip rar unrar`.

For more info on how to use these packages or any package for that matter type `xxx -help` OR `man xxx`.

Lets grab a few more tools that will make hardware detection and setup a bit easier. As root type the following: `emerge -av netselect eselect pciutils usbutils`

Netselect and eselect are utilities that help to select certain devices on your system such as your opengl provider and kernel source version which will be used in later guides. Pciutils and usbutils are both useful tools that are used to tell you what hardware is connected to your system. Once these two packages are installed you can run the following commands to get a listing of hardware on your machine. These are very helpful when trying to figure out what kind of chipset your devices have for when you manually build your kernel.

- For a listing of PCI devices on your system, as root type: `lspci`
- For a listing of USB devices connected to your system, as root type: `lsusb`

Now that we got all that done it's time to set up our system so it automatically prunes log files when they get to large and cleans out the temporary files.

To keep log files from getting to large install logrotate as root. This is a one shot no setup deal: `emerge -av logrotate`

For temp files we will use two programs that require some setup. The first is tmpwatch so as root do the following.

- Type `nano -w /etc/conf.d/bootmisc` and change the line `WIPE_TMP="no"` to read `WIPE_TMP="yes"`
- Save the file and exit. Be sure to watch for this file the next time you update your PC and do an `etc-update`, since you manually edited it!
- Type `emerge -av tmpwatch`
- When thats done type `nano -w /etc/cron.daily/tmpwatch` and uncomment all of the example code by removing the `#`'s. Make sure you leave the actual comments commented though! Then save the file and exit.

Next we will install tmpreaper which is another tmp cleaner. To install as root do the following:

- Type `emerge -av tmpreaper`

The last thing we are going to do in this part of the guide is make sure our new Gentoo box stays synced with a world atomic clock via NTP. So fire up that root terminal and lets get it done!

- First we need to add the caps use flag to the ntp package so type `echo "net-misc/ntp caps" >> /etc/portage/package.use`
- Then type `emerge -av ntp` and press enter. Note that the caps use flag is now listed in red. Type yes to complete the emerge.
- Now we need to set up our time server pool. Remember installing netselect above? I bet you thought it was useless, well type `netselect -s 3 pool.ntp.org` this will automatically select the 3 best (fastest) servers for your PC to use to sync to.
- Now type `/etc/init.d/ntp-client start` to start the client update service, then type `rc-update add ntp-client default` to add it to the default runlevel.
- Now we need to configure the ntpd daemon to help our PC's clock keep the right time continually and not just during updates. So type `nano -w /etc/conf.d/ntpd` and comment out the line `NTPD_OPTS="-u ntp:ntp"` by putting a `#` in front of it so it reads `#NTPD_OPTS="-u ntp:ntp"`
- Then type `nano -w /etc/conf.d/clock` and change the line `CLOCK_SYSTOHC="no"` to `CLOCK_SYSTOHC="yes"` which will automatically adjust the hardware clock

during shutdown so that it is accurate.

- If you are using dhcp to provide your PC with an IP address then type `nano -w /etc/conf.d/net` and add the line `dhcpcd_eth0="-N"` to prevent some conflicts. Replace `eth0` with the name of your network interface if it is not `eth0`, ie `eth1` or `wlan1`.
- Last but not least start the `ntpd` service by running `/etc/init.d/ntpd start` and set it to the default runlevel by running `rc-update add ntpd default`

[Back to Top](#)

MySQL Database Server

In this part of the guide I will discuss how to setup a basic MySQL database server. If you have followed the rest of the guide up to this point you should have a fully updated base Gentoo system, with some useful tools installed. Note if you are planning on setting up a desktop machine and do not wish to have MySQL installed skip this section.

To install MySQL on your Gentoo system log in as root and type `emerge -av mysql` type yes and hit enter. This will take some time to compile so you may wanna go grab a snack and watch some tv or something.

15 to 30 mins later ...

Once the compile finishes type the following line and press enter: (note: where the * is don't type the * but use the tab key so it auto fills in the right version.)

- `ebuild /var/db/pkg/dev-db/mysql-*/mysql-*.ebuild config`

This will prompt you to set the root password for the MySQL server to use. This does not have to be the same password you used for your root Gentoo user. It can be any thing you want.

Now start the server by typing `/etc/init.d/mysql start` and then type `rc-update add mysql default` to set it to the default runlevel.

Now let's test the new MySQL install. Type `mysql -u root -p` at the command prompt. (Note: you can issue this command as your regular user.) It will then prompt you for your root password. Enter the password it and press

enter. You should now see the following in your terminal: `mysql>`

- Type `use mysql;` to select the mysql database
- Type `show tables;` to view a listing of the tables that are in the mysql database
- Type `select * from user;`

After issuing the last command it will spew up a bunch of characters and stuff on the screen, but there should be no errors. If there are no errors your install was a success! Type `exit` to exit the mysql command line interface.

Now we need to set the mysql use flag in our `make.conf` file. I know you read the handbook so you know exactly what I am talking about right? Well to review, you need to add mysql to your use flag list so that when you compile packages that need to use it, it automatically sets that flag and compiles mysql support into the package. Here is how we add it:

- As root type `nano -w /etc/make.conf`
- You should now be looking at your `make.conf` file which you created during your initial install when you followed the handbook. In this file you have a line that reads: `USE= ""`
- Maybe you have a bunch of stuff in between the "" and maybe you only have a few things. A desktop system will probably have quite a few use flags listed here for multimedia codecs and such where a server may only have a very few. So we want to add `mysql` in between the "" so the line looks like this: `USE="mysql"`

Now we need to recompile anything on our system that may use that flag. This is a very simple command. As root type `emerge -av --newuse world`, this will search your system and see if any packages that you have installed will make use of the new mysql use flag that was added to your `make.conf` file. If it finds any packages it will list the packages, and show `mysql` in red as a use flag indicating that it will be compiled into the package. Review the list and type `yes` to update the packages.

Your MySQL installation is now complete.

[Back to Top](#)

Apache Web Server

In the last part of the guide I discussed how to set up MySQL on your Gentoo Linux box. One great use for MySQL is to combine it with an Apache web server and PHP to create content rich, dynamic websites. In this part of the guide I will discuss how to set up an Apache web server. Note, if you are setting up a desktop system and do not wish to have a web server installed skip this part.

Installing Apache is pretty straight forward, there are a few configuration files that need to be edited but nothing too difficult. First lets `su -` to log in as root then type `emerge -av apache`, once apache finishes compiling and installs type `/etc/init.d/apache2 start` to start the service and `rc-update add apache2 default` to set it to the default runlevel. We also need to add the `apache2` use flag to our `/etc/make.conf` file. As root type `nano -w /etc/make.conf`, and add `apache2` to your list of use flags. Then type `emerge -av --newuse world` to recompile any packages on your system that will use the new use flag.

Now we need to make some changes to the config files. Still as root type `nano -w /etc/apache2/httpd.conf`, at the very end of this file add the line `ServerName localhost` to stop apache from spamming about the server name every time the service is started. Type `/etc/init.d/apache2 restart` to restart the apache server so the change takes effect.

By default apache looks in `/var/www/localhost/htdocs` for your web files. You should now be able to open a web browser on another pc on your network and type `http://enter_your_server's_ip_here`, if you see the apache test page you are good to go. You can now delete all of the files in the `htdocs` folder and replace them with some `.html` or `.htm` files.

This is fine if you only want to host one website on your server, but apache is much more powerful than that. I prefer to use the default location for some of the nice web based server admin tools available such as `webalizer` and `phpmyadmin` (which I will discuss in later guides), and set up virtual hosts for all of my websites. Apache has a config file called `00_default_vhosts.conf` which is where you can specify the locations of other websites you wish to host on your server.

Type `nano -w /etc/apache2/vhosts/00_default_vhost.conf`

At the end of this file you can add as many virtual hosts as you would like. Each one would correspond to a domain name that you own that points to your IP address. This way you can keep the `localhost` folder access limited

to the internal network only (cause you don't want everyone to have access to your web server stats and other tools), and specify virtual hosts that will be accessed from the web via port 80.

Here is how you do it:

- First you will need to create the directory for your new virtual host in the `/var/www` folder. You will also want to include the same folders that are in the `localhost` folder such as `cgi-bin`, `icons`, `htdocs`, and `error`.
- Log in as root
- Type `cd /var/www`
- Type `mkdir somedomain` to create the new directory
- Type `cd localhost`
- Type `cp -a * ../somedomain` to copy all of the files in the `localhost` directory to the `somedomain` directory. If you had anything in `htdocs` that copied over just delete the files in the new directory.

Now you need to edit the `/etc/apache2/vhosts/00_default_vhosts.conf` and add your new virtual host entry to the end of the file. So as root type `nano -w /etc/apach2/vhosts/00_default_vhosts.conf` and add your vhost.

Here is an example of what a virtual host entry would look like:

```
#####  
#Your_domain_name_here  
#####  
<VirtualHost *:80>  
ServerName somedomain.com  
ServerAlias somedomain.com  
DocumentRoot /var/www/somedomain/htdocs/  
DirectoryIndex index.php index.html index.htm  
<Directory /var/www/somedomain/htdocs>  
Options -Indexes FollowSymlinks  
AllowOverride All  
Order allow,deny  
Allow from all  
</Directory>  
</VirtualHost>
```

Once you finish adding your new vhost entry you need to restart apache so as root type `/etc/init.d/apache2 restart`

Now, if you have your domain name registered and pointing to your public IP address, and port 80 is forwarded to your server in your router, you

should be able to open a web browser and type `http://somedomain.com` and see the website that you put in that folder on your server.

Rinse and repeat to add more domains.

[Back to Top](#)

PHP

In the last part I discussed how to set up an Apache web server on your Gentoo box. In this part I will show you how to set up PHP so it works with Apache and MySQL so you can host dynamic content rich websites. Plus by having MySQL, Apache, and PHP installed and working together you have access to a whole slew of web based tools that make managing your server much simpler

In order to instal PHP we must do the following:

- As root type `echo "dev-lang/php ctype sockets unicode pdo" >> /etc/portage/package.use`
- Then type `emerge -av php`

PHP will take a while to compile even on a very fast PC so take a break and go do something for a half hour to an hour ...

It's done? Ok, now we need to make sure that apache is set up to use PHP. So as root type `nano -w /etc/conf.d/apache2` and press enter. In this file you will find a line that reads: `APACHE2_OPTS="-D DEFAULT_VHOST -D INFO -D LANGUAGE -D SSL -D SSL_DEFAULT_VHOST"`

At the end of this line, still in the quotes add: `-D PHP5` if it is not there already. The newer PHP ebuidls have been adding this automatically as of lately, but check it just to be sure. Once you have added or verified that it is there then save the file and exit. Then as root type `/etc/init.d/apache2 restart` to restart the apache2 service so that the change takes effect.

Now lets test to make sure that PHP is installed correctly. We will put a test file in our localhost folder that will give us a ton of information about our web server setup. So as root again do the following:

- Type `cd /var/www/localhost/htdocs` to enter your default web directory.
- Type `nano -w phpinfo.php` and press enter
- In the editor type the following `<? phpinfo() ?>`
- Save the file and exit.
- In a web browser on another PC on the local network type the following: `http://server_ip_address/phpinfo.php`
- If PHP is installed correctly you should see a fancy PHP info page with a lot of information.

Thats it! Simple aye? Try installing PHP and get it working that fast on a Windows/IIS server, hehe good luck!

[Back to Top](#)

Sendmail/SSMTP

Now that I finished showing you how to set up your LAMP server you are probably going to want to set up sendmail so that some of your new fancy websites can take advantage of it for their email services. Lets get started!

We are going to be using ssmtp which will replace sendmail as the MTA for the server. So as root type `emerge -av ssmtp`

Once that completes we need to edit two config files.

Type `nano -w /etc/ssmtp/ssmtp.conf` and edit the following lines to reflect your email settings.

(Settings for a regular email account)

```
root=youremailaccount@somedomain.com
mailhub=smtplib.domain.com
rewriteDomain=
hostname=smtplib.domain.com
AuthUser=your_user_name
AuthPass=your_password
FromLineOverride=Yes
```

(Settings for a gmail account)

```
root=youraccount@gmail.com
mailhub=smtp.gmail.com:587
rewriteDomain=
hostname=smtp.gmail.com:587
UseSTARTTLS=YES
AuthUser=your_account
AuthPass=your_password
FromLineOverride=Yes
```

Once you are done editing `/etc/ssmtp/ssmtp.conf` save the file and exit. Then type `nano -w /etc/ssmtp/revaliases` and press enter to edit the following lines.

(Settings for a regular email account)

```
root:youraccount@domain.com:smtp.domain.com
yournormaluser:youraccount@domain.com:smtp.domain.com
```

(Settings for a gmail account)

```
root:youraccount@gmail.com:smtp.gmail.com:587
yournormaluser:youraccount@gmail.com:smtp.gmail.com:587
```

When you are done save the file and exit. What this did was set up accounts for root and your main user to use to send emails from. Now we need to install mailx which will be used when sending emails from the server. So as root type `emerge -av mailx`

Now we need to create a symlink from ssmtp to sendmail since sendmail has been replaced by ssmtp but some packages still like to use sendmail to send well mail! So as root do the following:

- Type `cd /usr/sbin`
- Type `ln -s ssmtp sendmail` to create the symlink, that's it!

Now lets test the new mailer. As root do the following:

- Type `mailx test@domain.com` (where `test@domain.com` is the email address you want to send to)
- You should now see `Subject: ,` type your subject and press enter.
- You will now have a blank area, this is your email body, just start typing your email, when done press `ctrl+d` to send the email.
- You will now see `Cc: displayed,` if you wish to copy the email to someone enter that address now otherwise just hit enter and the email

will be sent.

Now you need to add you regular user to the ssmtp group by using the command: `gpasswd -a username ssmtp`. Also note that if you want the ability to send emails through a website using php you will need to add the user apache to the ssmtp group by doing: `gpasswd -a apache ssmtp`

Congratulations! Your server can now send mail, and you can also send emails directly from your terminal.

[Back to Top](#)

MySQL Backup Script

Now that we got our LAMP server pretty well setup and ready to roll, we need to set up a way to back up our MySQL databases. I stumbled accross this nifty script a while back and have been using it ever since. You can find the script as well as detailed instructions on how to set it up [here](#).

This script requires that PHP is installed in order to work, which if you have been following along should not be an issue. It also requires vixie-cron to be installed, which should already be taken care of if you followed the handbook.

One note: when his directions tell you to type `/etc/init.d/crond restart` disregard it and substitute that line with `/etc/init.d/vixie-cron restart`, since our systems are using vixie-cron as the crontab daemon.

[Back to Top](#)

Protecting Your Web Directories With .htaccess

In this section I will show you how to protect your localhost web root with `.htaccess` and `.htpasswd` files. By doing this a username and password will be required to enter the site. Let's get started!

- As root type `cd /var/www/localhost/htdocs`
- then type `touch .htaccess`
- then type `touch .htpasswd`

Ok now we need to generate an encrypted user/passwd combo for our `.htpasswd` file so head on over to [this link](#) to use the handy `.htpasswd` generator!

Now type in the username you want to use and the password in the generator and it will output some encrypted text for you to put in the `.htpasswd` file. Here is an example with the username `root` and the password `12345`:

```
root:MTEWszYnI5ilw
```

Now as root and still in the `localhost/htdocs` folder type `nano -w .htpasswd` and add `root:MTEWszYnI5ilw` to the first line of the file. Save the file and close it.

Now type `nano -w .htaccess` and insert the following lines into the file:

```
AuthUserFile /var/www/localhost/htdocs/.htpasswd
AuthType Basic
AuthName restricted
Require valid-user
```

Thats it! Now when you go to `http://your_server_ip` you will be prompted for a username and password. There is so much more that you can do with `.htaccess` files, this is just a start. Try doing a google search on `.htaccess` and do some experimenting!

[Back to Top](#)

PHPMyAdmin

This part of the guide is going to cover how to set up PHPMyAdmin on your new Gentoo LAMP server. PHPMyAdmin is a very useful tool that allows to you manage your MYSQL databases from a convenient and easy to use web based interface. Lets get to it!

First `su -` to root and type `emerge -av phpmyadmin` and type `yes`. This will

automatically install PHPMysqlAdmin to your `/var/www/localhost/htdocs/` folder in a folder called `phpmyadmin`.

Note: This is why I discussed setting up vhosts in the apache guide. If you were to host your main website from the localhost folder then anyone could go to `www.yourdomain.com/phpmyadmin` and pull up your web interface. Granted they will have to guess your password, but you still don't want that exposed. With your websites set up as vhosts each domain name you have pointed to your server is routed to the correct folder on the server. If set up properly the only way you should be able to get to the localhost folder is by typing `http://localhost` at the server in a browser, or by typing the IP address of the server in a web browser from another pc. You can then tighten up the security of that folder by using `.htaccess` and `.htpasswd` files.

Ok enough of the review and back to the guide...

After the install is complete type: `mysql -u root -p < /var/www/localhost/htdocs /phpmyadmin/scripts/create_tables.sql`

Then restart apache by typing `/etc/init.d/apache2 restart`

Now we need to edit the PHPMysqlAdmin config file. As root do the following:

- Type: `cd /var/www/localhost/htdocs/phpmyadmin`
- Then type: `cp config.sample.inc.php config.inc.php`
- Then: `nano -w config.inc.php`

In this file change the following:

- Comment out `"$cfg['blowfish_secret'] = ;"` by putting a `#` in front of it.
- Change `"$cfg['Servers'][$i]['auth_type'] = 'config';"` to `"$cfg['Servers'][$i]['auth_type'] = 'http';"`
- Change the line `"$cfg['Servers'][$i]['controluser'] = ;"` to `"$cfg['Servers'][$i]['controluser'] = 'root';"`
- Change the line `"$cfg['Servers'][$i]['controlpass'] = ;"` to `"$cfg['Servers'][$i]['controlpass'] = 'root_mysql_password';"`

Save the file and exit. Then go to a computer with a web browser on the LAN and type `http://server's_ip_here/phpmyadmin`, you will be prompted for a username and password. Use the username and password that you supplied in the config file to log in. You should now see a nice web interface that will allow you to manage your MYSQL server.

[Back to Top](#)

Webalizer

So now you got a nice shiny new web server, and your websites are all setup in their own virtual hosts. How bout a handy web based application that can display traffic statistics for you with nice graphs and charts? That is where webalizer comes into play.

First we need to add a line to make.conf so as root type:

- `nano -w /etc/make.conf`
- Add the line: `LINGUAS="en"`

Save the file and exit. Then type `emerge -av webalizer` to install the package. The install should default to the localhost folder. When it is done type `/etc/init.d/apache2 restart` to restart the apache2 service.

Now we need to setup a cron job so webalizer will refresh its data daily. So as root type: `nano -w /etc/cron.daily/webalizer.cron`

in this file add the following lines:

```
#!/bin/sh
/usr/bin/webalizer
```

Save the file and exit. Then type `chmod +x /etc/cron.daily/webalizer.cron` to make the file executable. Type `/etc/cron.daily/webalizer.cron` to test the script, which will also allow webalizer to display the current statistics. Then open up a web browser and point to `http://server_ip_here/webalizer/index.html` to view the page.

Note: if you want webalizer to display hostname and visits by country information you will need to add `HostnameLookups On` to your `/etc/apache2/http.conf` file. This can cause some performance decrease however because the server needs to resolve each ip address that accesses it.

Thats it! Your LAMP server is now complete!

[Back to Top](#)

TeamSpeak Server

In this section I will discuss how to set up a Teamspeak 2 Server on Gentoo Linux. Teamspeak is a very popular tool used by gamers and online collaborators to provide an online group oriented voice chat environment. First thing you need to do is su to root, then type:

```
emerge -av teamspeak2-server-bin
```

Once that is done make sure to check the `/opt/teamspeak2-server/server.log` file for the auto generated superuser name and password then type:

- `cd /opt/teamspeak2-server/`
- then: `./server_linux`

The server should now be running in the background, you can verify it by issuing the `ps x` command and look to see if it is listed in the process list.

To stop the server just type `kill xxxxx` where xxxxx is the process number corresponding to `server_linux` in the list provided by the `ps x` command.

The teamspeak server comes with a web based interface that you can use to configure the server and add users. To get to the web interface open a browser on any pc that is connected to the same LAN as the server and type `http://insert-server-ip-here:14534`

If you want to allow external access to your teamspeak server make sure you forward TCP port 8767 to your server's ip on your router.

Note: You can install the Teamspeak2 client on a Gentoo Linux workstation that has X installed by using `emerge -av teamspeak2-client-bin` then as a regular user just type `TeamSpeak` in the terminal to launch the program. (The capitalization in the command `TeamSpeak` is important)

[Back to Top](#)

GenSplash Framebuffer

If you are tired of staring at a black console screen on your Gentoo box you can trick it out a bit using Gensplash. Gensplash or "fbcondecor" allows you to use different background images, fonts and colors to decorate you console so you don't have the standard black background and white text. It also allows you to have nice boot and shutdown screens. (Think of the boot process and console on the install CD). The install process can vary depending on your hardware and kernel config.

The basics are:

- Configure your kernel for framebuffer console image support and vesafb support
- `echo "media-gfx/splashutils fbcondecor" >> /etc/portage/package.use`
- `emerge -av splashutils splash-themes-livecd splash-themes-gentoo`
- `genkernel --splash=themename initrd`
- `rc-update add fbcondecor default`
- `reboot`

Note: You can now use the new uvesafb instead of vesafb for framebuffer support check [this link](#) for the information on how to set it up.

[Back to Top](#)

Getting a GUI, Gentoo and X

If you would like to install a GUI on your Gentoo PC, such as Xfce4, Gnome or KDE then you will first need to get X up and running. Installing X can differ depending on what kind of video card you have installed. I suggest reading and following the definitive Gentoo X guide found [here](#).

[Back to Top](#)

Sound, Gentoo and Alsa

In order to get sound working in Gentoo you need to set up ALSA. If you have not set up your kernel with ALSA support you will need to do so, see [here](#) for more information. Recompile your kernel with ALSA support and your sound card built in. Then do the following:

- `emerge -av alsa-utils alsa-tools alsa-oss mpg123 vorbis-tools`
- `rc-update add alsasound boot`
- `/etc/init.d/alsasound start`
- `alsamixer`
- Set all of the levels and unmute the necessary channels.
- `/etc/init.d/alsasound save`
- `gpasswd -a username audio`

Reboot your pc and you should now be able to play a .mp3 file by using the following command: `mpg123 filename.mp3` if you hear the song playing alsa is working properly.

You may want to consult the [Gentoo ALSA how-to](#) for further information and kernel specifics.

[Back to Top](#)

Window Managers

Now that you have X and ALSA installed it's time to set up a window manager. The 3 major players are Gnome, Xfce4, and KDE. You can find official Gentoo guides for installing these window managers at the following locations:

- [Xfce4 on Gentoo](#)
- [Gnome on Gentoo](#)
- [KDE on Gentoo](#)

[Back to Top](#)

IRC Server

Want to host your own IRC server on your gentoo box? Well here is how you do it:

- `su` to root
- `type emerge -av ngircd`
- `type cd /etc/ngircd/`
- `type cp ngircd.conf ngircd.conf.bak` to make a backup copy of the original config file

Now we need to edit the config file so `type nano -w ngircd.conf`

Under the [Global] section uncomment and edit the following lines:

```
Name = somename.com
Info = IRC Server Info
Password = Password
AdminInfo1 = Your name
AdminInfo2 = Your Location
AdminEmail = Your Email
Ports = 6667, 6668
Listen =
MaxConnections = 1024
MotdPhrase = Your MOTD here
ServerUID = your user here
ServerGUID = wheel
```

Then uncomment `PingTimeout`, `PongTimeout`, `ConnectRetry`, `OperCanUseMode`, `MaxConnections`, `MaxConnectionsIP`, `MaxJoins` and leave them at their default settings.

Next under the [Operator] section you can add your server op info:

```
Name = SomeName
Password = SomePass
```

Now lets add a channel under the [Channel] section:

```
Name = ChannelName
Topic = ChannelTopic
```

Save the file and exit. You can now test your install by typing `#ngircd --configtest` which should give you no errors and display your config. If there are errors correct them and try the `configtest` again. Now make sure your forward ports 6667 and 6668 to your server on your router then start your server and add it to the default runlevel by doing the following:

- /etc/init.d/ngircd start
- rc-update add ngircd default

You should now be able to connect to your new IRC server with an IRC client such as XChat or Pidjin!

[Back to Top](#)

Updated by [Ronny L. Bull](#) on 05-26-2009