verlihub Manual

This is an **unofficial** guide for verlihub, created with some spare time and patience. This manual specifically focuses on linux installations and verlihub 0.9.7. **Specific examples using the gentoo distribution will be provided.** Other distributions have package management systems available similar to gentoo, so research and find the equivalent commands.

verlihub's official website is located here. The official forums are located here (no, the forums on sourceforge are not frequented regularly.)

This document was last updated on **August 26, 2004**.

What is verlihub?

VerliHub is a Direct Connect protocol server (Hub), runs on linux, written in c++, relatively very low cpu-ram-bandwidth usage, and many useful features. Accepts plugins, and uses a MySQL database.

Table of contents

1. Getting Started

First of all, you will need the source for verlihub from sourceforge. Download the release and save it in /root (root home folder). Throughout the course of this manual, it will be assumed you are logged in as root during the compiling and installation phase (root access is not generally needed during compiling, if you know what you are doing and want to use a different username, make sure you correct all the path references from here on) and the source is located in the root home folder (/root).

Use the following command to extract the files:

```
tar zxvf verlihub-0.9.7.tar.gz
```

This will extract the source into it's own folder. (In the example above, it will be in /root/verlihub-0.9.7).

2. Installation

There are some requirements that need to be fulfilled before you attempt to compile Verlihub. **Each linux distribution has it's own type of package management system, it is up to you to research it. From here on, gentoo distribution specific instructions will be used.** For example, debian has apt-get, fedora/suse/mandrake use RPMs. Go to rpmfind.net to find RPMs for your distribution. These are the requirements:

- gcc version >= 3.2 (gcc --version will tell you the version number)
- mysql (3.23 or higher) **Slackware users only: use 4.0.20 or higher**
- Perl-compatible regular expression library (pcre)
- GeoIP
- **Optional:** A remote way to administer mysql. (This guide is not dependent on it, and it **IS NOT** a requirement. For those that use mysql, this will make life a lot easier.)

The package distribution needs to be updated prior to installing any packages. For gentoo, use:

```
emerge sync
```

Then proceed to the next section.

2.1 Installing geoIP

Here is the source for GeoIP. Follow the instructions provided with the package to compile it.

Alternatively, install it using the package management system using:

```
    emerge geoip
```

This will download, compile and install it for you.

2.2 Installing Perl-compatible regular expression library (pcre)

This is what is required: libpcre0-devel or libpcre1-devel. Google around to locate the source, or use rpmfind.net if your distribution supports it.

Again, you can install it using the package management system:

```
    emerge libpcre
```

This will download, compile and install it for you.

2.3 Installing mysql

Once again, Google around to locate the source, or use rpmfind.net if your distribution supports it.

Yet again, you can install it using the package management system:

```
    emerge mysql
```

This will download, compile and install it for you. After mysql installs, there still are two things you need to do. Firstly, mysql should start at boot up. Use the following gentoo distribution command (other distributions will have a command similar to it):

```
    rc-update add mysql default
```

and this will automagically add mysql to your startup scripts.

Now you need to start mysql. You can do this by rebooting or use:

```
    /etc/init.d/mysql start
```

to start it.

At this point in this guide, the commands are not gentoo-distribution specific; they should work across different platforms unless specifically noted.

You should remove the two anonymous user accounts from mysql. Use this to accomplish it:

```
    mysql -u root
    mysql> DELETE FROM mysql.user WHERE User = '';
    mysql> FLUSH PRIVILEGES;
```

Lastly, you **have** to set a root password for mysql, and remove the anonymous accounts installed by default. Reset the root password by using:

```
    mysql -u root
    mysql> SET PASSWORD FOR 'root'@'localhost' = PASSWORD('newpwd');
    mysql> SET PASSWORD FOR 'root'@'host_name' = PASSWORD('newpwd');
```

Replace the password between the two single quotes with a password of your choice. **You will need this password later on!** For the second password statement, replace 'host_name' with the IP address of the computer mysql is running on.

Now mysql is ready to go. Keep in mind that mysql does not necessarily need to run on the same machine as the hub software. It can be located (preferably) on the same LAN for latency reasons. Instructions to set this up however are out of the scope of this document.

2.4 Preparing and compiling the source

Now that the prerequisites are installed, there should be no problems building the source. To build the source use:

```
cd /root/verlihub-0.9.7
./configure
make
```

This could take several minutes to compile. After it's done, see Installing the fresh build.

2.5 Installing the fresh build

If the compile has completed successfully, then install it using:

```
make install
```

DO NOT START THE HUB YET! Proceed to the next section for some initial configuration of the hub before the initial start-up.

3. Starting up for the first time

First of all we need to prepare the database. If you look where you extracted the tar.gz file from Getting Started, you will notice a /scripts folder. We are going to run the installer script located in this directory. The output is shown below.

```
cd /root/verlihub-0.9.7/scripts
./install
mysql database for verlihub will ba called?(verlihub) press enter
mysql database is verlihub
mysql user to access verlihub gonna be?(verlihub) press enter
mysql user is verlihub
mysql password to access verlihub be?(verlihub) press enter
mysql pass is verlihub
mysql server will run where?(localhost) press enter
mysql server will be on localhost
Do you want to prepare database? (Y/N)y
Ok preparing mysql..
You probably need administrator access to mysql database
mysql administrator username?(root) press enter
mysql administrator is root
You'll be now twice promted by mysql client for password of root@localhost
Enter password: enter the mysql root password you assigned above
Enter password: enter the mysql root password you assigned above
the directory where you wish to install configuration and from which you want
to run hub?(/root) press enter
running directory is /root
configuration directory is /root/.verlihub
```

```
Written: /root/.verlihub/dbconfig

Db script is a tiny but handy script that helps you easily connect to mysql,
without all typing.. It will contain the password for verlihub's db
Do you want to prepare the db script? (Y/N)y
Created /root/.verlihub/db
use this command cd /root
ok, run verlihub executable and then kill it, it will create it's tables
then you can use regnick script to create admin user (e.g: regnick -n root -p
blabla_password -c 10)
after it you can run verlihub again already using the runhub script, and enjoy
it..
done
```

After this script is complete, then all we have to do is move the configuration files into /etc/verlihub. The installer will tell you where it is placing these files and in the example above, the output folder is in green.

So to set up the configuration files:

```
mkdir /etc/verlihub
mv /root/.verlihub/* /etc/verlihub
```

After this is done, verlihub needs to be started then stopped immediately (this will allow it to create all of the tables it needs.) So just start it manually by typing:

```
/usr/local/bin/verlihub
```

Alternatively, you can specify the port on the command line. This will ignore any port settings specified in the database. Here is how to start verlihub on port 4000:

```
/usr/local/bin/verlihub 4000
```

Wait until it stops outputting text to the screen, then press CTRL+C to exit the program.

3.1 Creating the Master user

Now that the database has been prepared, you need to create the Master user. This user has all of the commands available to them for configuring the way verlihub works. All of this is done through the master user. To create the master user, we need to use mysql to manually add a record to the database (once this is done, the Master user will be able to create users while actually in the Hub. You need to think of a nickname you wish to use and replace *Your_Admin_Nick* below with it.

To create the master user:

```
mysql --user=verlihub --password=verlihub -D verlihub
mysql> INSERT INTO reglist (nick, class) VALUES ('Your_Admin_Nick',10);
mysql> quit
```

Substitute the correct username and password in the first line if you specified different ones when preparing the verlihub database. After this is done, start the hub once again by:

```
/usr/local/bin/verlihub
```

Remember: if you specify a port on the command line it overrides the port that is stored in the database! Here's how to start verlihub and force it to use port 4000:

```
/usr/local/bin/verlihub 4000
```

Now use a Direct Connect client and connect to your hub! (**REMEMBER:** The hub listens on **port 4111** by default! Change this accordingly if you decide to run on a different port. You need to specify this port if it is NOT port 411 in the hub address in a format like "localhost:4111" or you won't connect!) Login using the admin nick you provided above, and when you connect, type:

```
+passwd <new_pass>
```

in the chat window to register your admin password. Make sure you choose a password that you can remember. After you set the new password, reconnect to the hub.

Congratulations! The hub is now up and running, and you are connected as the master user.

4. Running Verlihub

Now that the Hub has been run for the first time, now you need to decide how it starts. In this guide so far, verlihub has been invoked manually. It can also be run as a daemon.

4.1 Running normally

This is how it has been invoked so far. You can log on as a normal user and start the hub manually each time by using:

```
/usr/local/bin/verlihub
```

When running the hub this way, all you need to do is press CTRL+C to stop the hub software, or if you are logged in as the Master user type !quit in the chat window. **Please note:** When you run the hub as a non-root user you CANNOT set the hub to listen on ports below 1024!

4.2 Running as a daemon

When verlihub is configured to run as a daemon, it is generally loaded in the startup scripts. You don't need to be there manually to login and start up the hub. The problem is that different distributions have different ways of writing scripts and adding them to the startup sequence. You should really check out the forums (link is at the top of this page) for possible help with certain distributions.

Verlihub comes with a script to launch as a daemon, but it may not work in all instances (see Starting and Stopping verlihub as a daemon for how try try to use the provided script.) The following is **gentoo-distribution specific**. This is a script that can be used with gentoo to start and stop verlihub. Save this into /etc/init.d/verlihub:

```
#!/sbin/runscript
# Verlihub boot script for gentoo distributions

# feel free to change these
confdir=/etc/verlihub
logdir=/var/log/verlihub
verlihubexe=/usr/local/bin/verlihub

# don't change these ones, though!
loghistorydir=$logdir/history
logfile=$logdir/log
errfile=$logdir/err
loghistfile=$loghistorydir/log
```

```
errhistfile=$loghistorydir/err
pidfile=/var/run/verlihub.pid
requiredfiles="dbconfig db"
optionalfiles="motd faq rules help_admin help_master help_op help_usr help_reg"

depend() {
 need net
 need mysql
}

start() {
ebegin "Starting verlihub"
 # make sure the configuration directory exists
 if [ ! -d $confdir ]; then
  eerror FATAL ERROR: missing configuration directory $confdir
  return 1
 fi;

 # ensure log directory exists
 if [ ! -d $logdir ]; then
  mkdir $logdir
 fi;

 # ensure log history directory exists
 if [ ! -d $loghistorydir ]; then
  mkdir $loghistorydir
 fi;

 # check for the required files
 for f in $requiredfiles; do
  file=$confdir/$f
  if [ ! -e $file ]; then
   eerror "FATAL ERROR: missing required file $file"
   return 1
  fi;
 done;

 # check for the optional files
 for f in $optionalfiles; do
  file=$confdir/$f
  if [ ! -e $file ]; then
   eerror "WARNING: missing file $file, some functionality will be disabled"
  fi;
 done;

 # backup logs and date/timestamp them
 if [ -e $errfile ]; then
  mv -f $errfile $errhistfile.`date +%Y_%m_%d_%k:%M:%S`.old
 fi
 if [ -e $logfile ]; then
  mv -f $logfile $loghistfile.`date +%Y_%m_%d_%k:%M:%S`.old
 fi

 # start the hub
 $verlihubexe &gt;$logfile 2&gt;$errfile &
 pid=$!
 disown $pid
 echo $pid &gt; $pidfile
eend $?
}

stop() {
ebegin "Stopping verlihub"
 pid=`cat $pidfile`
 kill $pid
 rm $pidfile
eend $?
}

restart()
 svc_stop
 sleep 5
```

```
     svc_start
    }
```

After you save that script, use:

```
rc-update add verlihub default
```

to add it to your startup scripts. See the next section for how to use the gentoo script.

4.2.1 Starting and Stopping verlihub as a daemon

For other scripts see Helper scripts for more information. These scripts are in CVS and versions dated > August 26, 2004.

In the /scripts folder there is a runhub command. Be warned that this script **may not work** in all instances! To start it use:

```
/root/verlihub-0.9.7/scripts/runhub
```

To try to stop it with the runhub script try:

```
/root/verlihub-0.9.7/scripts/runhub -s
```

The following is **gentoo-distribution specific**, and you need to set up the script in the previous step for this to work.

To start verlihub:

```
/etc/init.d/verlihub start
```

To stop verlihub:

```
/etc/init.d/verlihub stop
```

To restart verlihub:

```
/etc/init.d/verlihub restart
```

4.3 Helper scripts

There are helper scripts in the CVS version of verlihub (and/or versions later than August 26, 2004.) They have been listed here in a table with a brief description of what they are capable of:

Script	Description
vh_getcfg	Run this script and see which configuration folder verlihub is most likely to use
vh_runhub	start, stop, restart verlihub as a daemon; creates log and err files; displays missing files
vh_getdb	Finds the dbconfig files and parses it, then allows you to do multiple things with that information such as: connect; query; gethost; getdata; getuser; getpass
vh_regnick	Creates a registered user
vh_trigger	Creates a file_trigger

5. Configuring your Hub

All of the settings are stored in the database. All administration is done from within the hub, by a person logged on as the Master user.

To get the current configuration settings type:

```
!getconfig
```

To change a setting to a new value type:

```
!set <variable> <value>
```

Once changes are made, you will need to refresh the hub. (Some changes will require you to stop and start the hub again, however.) To do this type:

```
!reload
```

If your changes don't apply, restart the hub manually.

5.1 Basic Configuration

There are a few components to the basic configuration of your hub. There is some settings that need to be changed in the database, as well as some files need to be added (such as a Message of the Day, FAQ, amongst other things...) to the configuration directory.

5.1.1 Hub Variables

In the following table is a list of the "basic" things that should be set in your hub.

Variable	Explanation
listen_port	The port the hub listens on. Other hubs use port 411, but you need to run verlihub as root to do that. Default=4111.
listen_ip	If you have more than one network card in your PC, enter the IP address to listen on. If you don't have more than one network card, no change is necessary.
hub_name	The name of your hub.
hub_desc	A description of your hub.
hub_host	What people will use to connect to your hub. (An example: myhub.no-ip.com)
hub_owner	Owner that runs the hub.
min_nick	Minimum nickname length. Default of 3.
max_nick	Maximum nickname length.
max_users	Maximum number of users allowed in your hub.
min_share	Minimum amount that people must share in your hub in megabytes. People that share less will not be allowed to join.
tban_kick	Length of time people must wait after being kicked to rejoin. In seconds. Default=300 seconds.
send_user_info	When people join, the hub will send them information about themselves. Set to 0 (zero) to disable, or 1 to enable. Default=On.

hub_security	This is the name of the Hub Security bot.
opchat_name	This is the name of the OpChat bot.

For example, to set the hub name to 'My Hub' use this command:

```
!set hub_name My Hub
```

5.1.2 Message of the Day

The message of the day will be shown to every member joining the hub. All that needs to be done is a text file needs to be saved as 'motd' in the configuration directory (in this guide /etc/verlihub/motd) and verlihub will automatically use it.

Note: If you don't use CRLF encoding in your text file, the Windows clients will display the whole file **on one line**.

5.1.3 FAQ

A FAQ (Frequently Asked Questions) is available to users using +faq in the hub chat window. Again, all that needs to be done is a text file needs to be saved as 'faq' in the configuration directory (in this guide /etc/verlihub/faq) and verlihub will automatically use it.

Note: If you don't use CRLF encoding in your text file, the Windows clients will display the whole file **on one line**.

5.1.4 Hub Rules

A rules trigger is also automatically provided. When people type +rules in the hub chat window, the contents of 'rules' in the configuration directory (in this guide /etc/verlihub/rules) are sent to the user.

Note: If you don't use CRLF encoding in your text file, the Windows clients will display the whole file **on one line**.

5.1.5 Hub Help

Interestingly enough, there is no help files that come with verlihub (at least I couldn't find any.) So you will have to create 7 different help files. Before you do this, it may help to understand how verlihub deals with help files. There are several "classes" of users that can exist in verlihub. Each one has a help file associated with it. The help will be sent by class, for example, a regular user will only get class 0 help when they use +help. A regular operator will get class 3, 2, 1, and 0 help sent to them when they use +help. Because of this, it is important to create all the help files, as certain commands are only available to certain users. The files need to be placed in the configuration directory (in this guide it's /etc/verlihub). Here is a table outlining the files needed for the help system to work:

Class	Tag	Help file
0	Regular users	/etc/verlihub/help_usr
1	Registered users	/etc/verlihub/help_reg
2	VIP users	/etc/verlihub/help_vip
3	Operator user	/etc/verlihub/help_op

4	Cheef user	/etc/verlihub/help_cheef
5	Admin user	/etc/verlihub/help_admin
10	Master user	/etc/verlihub/help_master

As for a guide for what to put in these files, refer to the Hub Command list for examples of what to put in these files.

Note: If you don't use CRLF encoding in your text file, the Windows clients will display the whole file **on one line**.

5.2 Advanced Configuration

The items here are not necessary to configure for the hub to function. It is a good idea to be aware of some of these configurable options, however. All of these are configured as the Master user logged into the hub.

5.2.1 Listening on more than one port

Verlihub has the ability to listen on more than one port for incoming clients, if you need to do this. For example, to listen on 411 **(the hub needs to run as root to listen on this port)** use this command:

```
!set extra_listen_ports 411
```

5.2.2 File Triggers

Verlihub has a trigger feature, which makes possible to show any text file on any specified command beginning with "+" sign. There is a mysql table called 'file_triggers' that contains definitions of these triggers.

There is a provided script called 'trigger' to create triggers in the database. It is located in the scripts folder (sorry, there is no way to add one in the Hub itself.) The trigger script will show you the mysql command you need to enter the trigger data into the database. In this guide, that script would be located in /root/verlihub-0.9.7/scripts/trigger.

We will use mysql directly to add a trigger called '+testtrigger', and have it display the contents of file /etc/verlihub/test. Use this command to do this:

```
mysql --user=verlihub --password=verlihub -D verlihub -e "INSERT INTO
file_trigger (command,def) VALUES ('+testtrigger','/etc/verlihub/test')"
```

The changes will NOT take effect until you refresh the hub either by restarting it or invoking the command !reload.

Once you add the trigger into the database, and refresh the Hub using !reload, then you can use the triggers. Of course, if you make changes to the contents of a file that is already added as a trigger, the changes will take effect immediately.

Verlihub also provides you with a quick way to change the trigger's contents quickly. For example, if you wanted to replace the text of +testtrigger, use:

```
!ftedit +testtrigger This is the new contents for this trigger.
```

This will overwrite the trigger file contents!

5.2.3 Flood protection

Verlihub has diverse parameters to protect chat from flooding. There must be some compromise between heavy restrictions and the possibility to chat; there is not a full set of protection against this.

There are two hub variables that can help prevent mass flooding: *max_chat_msg* and *max_chat_lines*. *max_chat_msg* is the maximum length of a message allowed to be sent to your hub (default is 256 characters long). *max_chat_lines* is the maximum number of lines that can be sent to your hub (default is 5). As an example, to set a maximum length of 512 characters and 10 lines:

```
!set max_chat_msg 512
!set max_chat_lines 10
```

If you have active bots that are frequently kicking naughty users, you may want to get rid of the kicking messages. You can use the variable *hide_all_kicks* to do this; 1 hides all kick messages, 0 shows messages. Use this to hide all messages:

```
!set hide_all_kicks 1
```

Another possibility for muting kick messages is using commands **!hidekick <nick>** and **!unhidekick <nick>** which will kick the one user and not display anything.

There is nothing in verlihub that prevents user from repeating the same message. You may want to get a client that can monitor this and remove users automatically. (A dcgui-qt client can do this for you.)

Verlihub does not prevent users saying rude words. There is a possibility to configure VerliBot to kick/ban such users.

None of these restrictions apply to OPs.

5.2.4 Hub Redirections

You can set up to 10 hubs where verlihub will try to redirect every disconnected user. The choice is made randomly among all ten of them. If you want only one hub to redirect to you would set all ten of them to same hub address.

If you would like to disable redirection, set redir_host0 to your hub's IP address or hostname, and set redir_host_max to 0. For example:

```
!set redir_host0 my-hub.no-ip.com
!set redir_host_max 0
```

The variables are redir_host0 to redir_host9. To set it to redirect to one hub only:

```
!set redir_host0 somehub.org
!set redir_host1 somehub.org
!set redir_host2 somehub.org
!set redir_host3 somehub.org
!set redir_host4 somehub.org
!set redir_host5 somehub.org
!set redir_host6 somehub.org
!set redir_host7 somehub.org
!set redir_host8 somehub.org
!set redir_host9 somehub.org
```

5.2.5 Customizing Error Messages

Verlihub allows you to customize the following error messages:

- User is banned message (msg_banned)
- Hub full message (msg_hub_full)
- Invalid nickname prefix (msg_nick_prefix)
- Client version too new (msg_downgrade)
- Client version too old (msg_upgrade)

Here is an example of how to set the messages:

```
!set msg_banned You are banned from this hub.
!set msg_hub_full This hub is full, try again later.
!set msg_nick_prefix You do not have the desired nickname prefix in order to join this hub.
!set msg_downgrade You are using too new of a client. This hub does not support untested clients at this time.
!set msg_upgrade You are using an outdated client. Upgrade your client and rejoin.
```

5.2.6 Customizing Welcome Messages

Verlihub allows you to greet members as they join. Newly registered users will be greeting with a message to change their password, and if the main chat window is disabled in the hub, they will see the command for how to turn the chat window on and off.

To change the two default messages (password and chat) use this:

```
!set msg_chat_onoff "Use !chaton to display messages in the chat window."
!set msg_change_pwd "You have 5 minutes to change your password."
```

As far as welcoming users, you can welcome Class 2 and up users specially by using:

```
!set msg_welcome_vip VIP member %[nick] has joined!
!set msg_welcome_op OP %[nick] has joined.
!set msg_welcome_cheef OP %[nick] has joined.
!set msg_welcome_admin Admin OP %[nick] has joined.
!set msg_welcome_master Hub owner %[nick] has joined. Everyone hide!
```

Tip: If you want to use the user's nickname in the message, use the variable %[nick] as shown above.

5.2.7 Advanced Share Limit control

Verlihub allows you to set different levels for registered users and hub operators. You can also specify a overall maximum allowed share, and a maximum allowed share for registered users (all are in megabytes). For example, to set a minimum share of 2 gigabytes for registered users and no share for OPs, use:

```
!set min_share_reg 2000
!set min_share_ops 0
```

To set an overall maximum share of 1000 gigabytes and a maximum share of 500 gigabytes for registered users use:

```
!set max_share 1000000
!set max_share_reg 500000
```

5.2.8 Restricting Clients

Verlihub also recognizes DC tags. Using these tags, you can restrict users access to your hub.

5.2.8.1 Hub/slot ratios & Maximum hubs allowed in

Most hubs allow you to restrict how many hubs your users are in, and also a hub/slot ratio. Verlihub is no different, for example, if you want to restrict your users to be in 4 hubs maximum, and have 1 slot per hub, use:

```
!set tag_max_hs_ratio 1
!set tag_max_hubs 4
```

5.2.8.2 Minimum/Maximum slots

Verlihub allows you to restrict the number of slots the client is using. You can specify default minimum and maximum values, as well as values that are specific to the type of connection. Below is a list of the variables needed. To set them, use the !set command as illustrated elsewhere in this manual.

- tag_min_slots_default
- tag_min_slots_28kbps
- tag_min_slots_33kbps
- tag_min_slots_56kbps
- tag_min_slots_modem
- tag_min_slots_isdn
- tag_min_slots_cable
- tag_min_slots_dsl
- tag_min_slots_satellite
- tag_min_slots_microwave
- tag_min_slots_wireless
- tag_min_slots_lant1
- tag_min_slots_lant3

- tag_max_slots_default
- tag_max_slots_28kbps
- tag_max_slots_33kbps
- tag_max_slots_56kbps
- tag_max_slots_modem
- tag_max_slots_isdn
- tag_max_slots_cable
- tag_max_slots_dsl
- tag_max_slots_satellite
- tag_max_slots_microwave
- tag_max_slots_wireless
- tag_max_slots_lant1
- tag_max_slots_lant3

5.2.8.3 Speed per available slot

Verlihub also allows you to set a minimum speed **per slot** for the client. Users that are below it are not allowed a connection to the hub. You can specify an overall default, as well as connection-specific limits.

Below is a list of tags to do this. To set them, use the !set command as illustrated elsewhere in this manual.

- tag_min_ls_ratio_default
- tag_min_ls_ratio_28kbps
- tag_min_ls_ratio_33kbps
- tag_min_ls_ratio_56kbps
- tag_min_ls_ratio_modem
- tag_min_ls_ratio_isdn
- tag_min_ls_ratio_cable
- tag_min_ls_ratio_dsl
- tag_min_ls_ratio_satellite
- tag_min_ls_ratio_microwave
- tag_min_ls_ratio_wireless
- tag_min_ls_ratio_lant1
- tag_min_ls_ratio_lant3

5.2.8.4 Upload capping

If the client connected is capable of upload capping, you can specify a minimum cap allowed before they are allowed to join in the hub. All that needs to be set is the default cap, however, you can also specify caps depending on the type of connection the user is reporting. These values are in KB/sec.

In the example below, a 25KB/sec default cap is set, and various caps are set depending on the connection:

```
!set tag_min_limit_default 25
!set tag_min_limit_28kbps 2
!set tag_min_limit_33kbps 3
!set tag_min_limit_56kbps 5
!set tag_min_limit_modem 5
!set tag_min_limit_isdn 13
!set tag_min_limit_cable 13
!set tag_min_limit_dsl 13
!set tag_min_limit_satellite 20
!set tag_min_limit_microwave 20
!set tag_min_limit_wireless 20
!set tag_min_limit_lant1 50
!set tag_min_limit_lant3 100
```

5.2.9 Hublist registering

Verlihub has also a very simple and a bit incomplete hublist registering support. Well there are several variables and one command that make it work. Those that are used are:

- hublist_host - the ip address of the hublist registering host
- hublist_port - usually 2501 which is default
- hub_host
- listen_port
- hub_name
- hub_desc

The command to send the registration off is:

```
!hublist
```

You can use this command once the above variables are set correctly. Do not use it more than once every 20 minutes, as the hublists tend to ban hubs that update too frequently.

5.2.10 Setting up a Registered Users only Hub

Verlihub can restrict users to member-only visitors. For example, use this to not allow unregistered users and 1000 registered users:

```
!set max_users 0
!set max_extra_regs 1000
```

5.2.11 Setting up a nationalistic Hub

Verlihub can also restrict users to be within certain country zones. To restrict to US and Canada use:

```
!set cc_zone1 :CA:US:
!set max_users 7000
!set max_users0 100
!set max_users1 = 6900
```

5.2.12 Multiple Language support

Some text of Verlihub can be translated, specifically those that hub sends to users. By default, these special fields are not in the database. To enable this, you need to:

```
!set save_lang 1
```

Restart the hub, then:

```
!set save_lang 0
```

You need to reset *save_lang* to 0, it is only needed once to create special settings in the database.

Now in the database you have (in the SetupList table) some variables with their english values (they have the format lang_en in the table). Whatever you change in these variables will be loaded next time. If you like you can keep the lang_en variables and make a copy to lang_whatever. If you do this, you need to tell Verlihub to use the new ones on next startup, by adding a line in dbconfig in the configuration folder (in this guide, the file will be /etc/verlihub/dbconfig). All you need to do is add:

```
lang_name = lang_whatever
```

Verlihub will then use the language specified when the hub is restarted.

5.2.13 Bending user limit rules

Verlihub allows you to "bend" the rules for certain people joining the hub. When the limit in max_users is reached, and either a Registered user, an OP, or an Admin OP joins, Verlihub checks these three values:

- max_extra_regs - Amount of registered users allowed to join hub after the hub is full
- max_extra_ops - Amount of OPs allowed to join hub after the hub is full
- max_extra_admins - Amount of Admin OPs allowed to join hub after the hub is full

So if you would like to have no registered members join after the hub is full, but you still want 5 OPs and 5 Admin OPs allowed to join, use this set of commands:

```
!set max_extra_regs 0
!set max_extra_ops 5
```

```
!set max_extra_admins 5
```

5.2.14 Increasing file descriptor limits for connections >= 10mbit

If for example you have a 10mbit line, you need to increase the default value for the file descriptor limits to 4096. If however, you have a 100mbit line, you need to increase your file descriptor limits by a larger amount or you will be limited to how many users are in your hub (for example, increase it to 10240.)

To do this, you need to change the following files:

In **/etc/security/limits.conf**, add the lines:

```
*     soft    nofile  4096
*     hard    nofile  4096
```

In **/etc/pam.d/login**, add:

```
session required /lib/security/pam_limits.so
```

Then increase the system-wide file descriptor limit by adding the following lines to the **/etc/rc.d/rc.local** startup script:

```
echo 16256 > /proc/sys/fs/file-max
echo 24576 > /proc/sys/fs/inode-max
```

Also, you can run these commands at the command line and/or you can add them to your .bash_profile so you don't need to re-enter them.

You will then need to tell the system to use the new limits (bash):

```
ulimit -n unlimited
```

For csh/tcsh:

```
unlimit descriptors
```

Verify this has raised the limit by checking the output of `ulimit -a` (bash) or `limit` (csh, tcsh).

Then you can start your hub.

5.3 Configurable variables

The table below lists all of variables that are configurable in verlihub (this list is more than likely incomplete). To change their values, use the !set command when logged into the hub as a Master user.

General Hub settings	
Variable	**Explanation**
hub_name	The name of your hub.
hub_desc	A description of your hub.
hub_host	What people will use to connect to your hub. (An example: myhub.no-ip.com)
hub_owner	Owner that runs the hub.

hub_security	This is the name of the Hub Security bot.
opchat_name	This is the name of the OpChat bot.
show_tags	Show tags to users? 0=hide from all, 1=show tags, 2=show tags to OPs only. Default=2.
Hub Connection settings	
Variable	**Explanation**
listen_port	The port the hub listens on. Other hubs use port 411, but you need to run verlihub as root to do that. Default=4111.
extra_listen_ports	Extra ports the hub listens on.
listen_ip	If you have more than one network card in your PC, enter the IP address to listen on. If you don't have more than one network card, no change is necessary.
dns_lookup	Reverse DNS lookup when user joins. Needed to see hostmask of user. If you have a slow DNS server this can slow down the hub.
max_users	Maximum number of users allowed in your hub.
max_upload_kbps	Maximum upload speed hub is allowed to use.
min_frequency	Somewhat automatic antilag system, higher values will allow lower lag; 0 means disabled. Largest usable value is 3. Normal values would be between 0.1 and 1.5. If you want to try to break records, set to 0 (though it's not recommended.)
Nickname Rules	
Variable	**Explanation**
min_nick	Minimum nickname length. Default of 3.
max_nick	Maximum nickname length.
nick_chars	Allowed characters in username.
nick_prefix	
nick_prefix_cc	
Share Rules	
Variable	**Explanation**
min_share	Minimum amount that people must share in your hub in megabytes. People that share less will not be allowed to join.
min_share_reg	If you want your registered users to have a different minimum share level, use this.
min_share_ops	If you want your OPs to have a different minimum share level, use this.
min_share_factor_passive	Multiplication factor for the minimum share of passive users (a setting of 2.0 means passive have to share twice the min_share setting)
min_share_use_hub	If this is higher then min_share then users that have less then this will not be able to search and download, but will be allowed to join the hub
max_share	If you want to prevent users from sharing too much, enter the max allowed to share in MB.

max_share_reg	If you want to prevent registered users from sharing too much, enter the max allowed to share in MB.
Ban settings	
Variable	**Explanation**
tban_kick	Length of time people must wait after being kicked to rejoin. In seconds. Default=300 seconds.
tban_max	Maximum length of time an OP can ban someone for. In seconds.
hide_all_kicks	Hides all kick messages from users.
Login settings	
Variable	**Explanation**
int_login	Number of seconds user has to wait after disconnection before being allowed to reconnect.
send_user_info	When people join, the hub will send them information about themselves. Set to 0 (zero) to disable, or 1 to enable. Default=On.
send_user_ip	When people join, the hub will send them their IP address. Set to 0 (zero) to disable, or 1 to enable. Default=On.
nicklist_on_login	Show nicklist when users log in.
allways_ask_password	Always ask users for a password, whether they are registered or not.
chat_default_on	Enable the chat window by default. 1=yes, 0=no. If no, users need to use !chaton to get messages.
Custom Hub Messages	
Variable	**Explanation**
msg_chat_onoff	
msg_change_pwd	You can customize your user's change password message by putting a message in this variable.
Custom Error Messages	
Variable	**Explanation**
msg_banned	You can customize the error message that indicates the user is banned by putting a message in this variable.
msg_hub_full	You can customize the error message that indicates the hub is full by putting a message in this variable.
msg_nick_prefix	
msg_downgrade	You can customize the error message that indicates the client is too new by putting a message in this variable.
msg_upgrade	You can customize the error message that indicates the client is outdated by putting a message in this variable.
msg_replace_ban	
Custom Welcome Messages	
Variable	**Explanation**
msg_welcome_vip	If you want to welcome your VIP members, fill in a welcome message here. Use %[nick] to use the person's nickname in the message.

msg_welcome_op	If you want to welcome your OPs, fill in a welcome message here. Use %[nick] to use the person's nickname in the message.
msg_welcome_cheef	If you want to welcome your Cheef OPs, fill in a welcome message here. Use %[nick] to use the person's nickname in the message.
msg_welcome_admin	If you want to welcome your Admin OPs, fill in a welcome message here. Use %[nick] to use the person's nickname in the message.
msg_welcome_master	If you want to welcome your Master user, fill in a welcome message here. Use %[nick] to use the person's nickname in the message.
Maximum User Extensions	
Variable	**Explanation**
max_extra_regs	If the hub is full, allow this many more extra registered users to join.
max_extra_ops	If the hub is full, allow this many more OPs to join.
max_extra_admins	If the hub is full, allow this many more Admin OPs to join.
Public Hub List settings	
Variable	**Explanation**
hublist_host	
hublist_port	
hublist_send_minshare	When set to 1, a minimum share field is added automatically to the hublist registration description.
timer_hublist_period	Number of seconds between two hublists registrations.
Hub redirection	
Variable	**Explanation**
redir_host_max	Set this to 0 (zero) and redir_host0 to your hub's DNS name to disable redirection.
redir_host0	Enter hosts to redirect to in case users aren't allowed in your hub, or leave them blank.
redir_host1	
redir_host2	
redir_host3	
redir_host4	
redir_host5	
redir_host6	
redir_host7	
redir_host8	
redir_host9	
Search settings	
Variable	**Explanation**
int_search	Minimum number of seconds user has to wait before being allowed to search again.
min_search_chars	Minimum number of characters allowed in a search.

DC Tag settings	
Variable	**Explanation**
tag_allow_none	Allow clients to NOT report a tag? 1=yes, 0=no. Yes by default.
tag_sum_hubs	The count of numbers to sum and use as the HUBS count from the H:X/Y/Z part of tag
tag_min_class_ignore	Which lowest class users are not to be checked for tags
tag_max_slots_default	If you want to restrict users from having too many slots open, enter maximum here.
tag_max_slots_28kbps	
tag_max_slots_33_6kbps	
tag_max_slots_56kbps	
tag_max_slots_modem	
tag_max_slots_isdn	
tag_max_slots_cable	
tag_max_slots_dsl	
tag_max_slots_satellite	
tag_max_slots_microwave	
tag_max_slots_wireless	
tag_max_slots_lant1	
tag_max_slots_lant3	
tag_min_slots_default	Enter the minimum number of slots open to be allowed to connect to the hub.
tag_min_slots_28kbps	
tag_min_slots_33_6kbps	
tag_min_slots_56kbps	
tag_min_slots_modem	
tag_min_slots_isdn	
tag_min_slots_cable	
tag_min_slots_dsl	
tag_min_slots_satellite	
tag_min_slots_microwave	
tag_min_slots_wireless	
tag_min_slots_lant1	
tag_min_slots_lant3	
tag_min_ls_ratio_default	If you want to limit users by speed per available slot, enter the minimum speed per slot here, in KB/sec. Quick example: 2 slots capped at 10KB/sec is 5KB/sec per slot.
tag_min_ls_ratio_28kbps	
tag_min_ls_ratio_33_6kbps	
tag_min_ls_ratio_56kbps	

tag_min_ls_ratio_modem	
tag_min_ls_ratio_isdn	
tag_min_ls_ratio_cable	
tag_min_ls_ratio_dsl	
tag_min_ls_ratio_satellite	
tag_min_ls_ratio_microwave	
tag_min_ls_ratio_wireless	
tag_min_ls_ratio_lant1	
tag_min_ls_ratio_lant3	
tag_min_limit_default	If your user is limiting his/her upload speed, specify here the minimum speed allowed in order to join the hub, in KB/sec.
tag_min_limit_28kbps	
tag_min_limit_33_6kbps	
tag_min_limit_56kbps	
tag_min_limit_modem	
tag_min_limit_isdn	
tag_min_limit_cable	
tag_min_limit_dsl	
tag_min_limit_satellite	
tag_min_limit_microwave	
tag_min_limit_wireless	
tag_min_limit_lant1	
tag_min_limit_lant3	
tag_max_hs_ratio	Enter the minimum hub/slot ratio here. Users that are below this are refused connections to the hub. For example, setting to 1 means user needs to have 1 slot open per hub.
tag_max_hubs	Set this to restrict how many hubs your user can be in concurrently.
tag_min_version_plusplus	Use the _min_version_ tags to specify a minimum version for the client allowed in the hub.
tag_min_version_dcgui	
tag_min_version_odc	
tag_min_version_dc	
tag_min_version_dcpro	
tag_min_version_strongdc	
tag_max_version_plusplus	Use the _max_version_ tags to specify a maximum version for the client allowed in the hub.
tag_max_version_dcgui	
tag_max_version_odc	
tag_max_version_dc	

tag_max_version_dcpro	
tag_max_version_strongdc	

Country Code User Definitions

Variable	Explanation
cc_zone1	Enter the countries for the first country code zone. The format is `:XX:YY:ZZ:`. It has to start and end with a colon, and all country codes are separated by colons. There can be any number of codes (example codes: CZ; UK; US; SE. **NOTE:** These need to be capitalized!)
cc_zone2	Enter the countries for the second country code zone. The format is `:XX:YY:ZZ:`. It has to start and end with a colon, and all country codes are separated by colons. There can be any number of codes (example codes: CZ; UK; US; SE. **NOTE:** These need to be capitalized!)
cc_zone3	Enter the countries for the third country code zone. The format is `:XX:YY:ZZ:`. It has to start and end with a colon, and all country codes are separated by colons. There can be any number of codes (example codes: CZ; UK; US; SE. **NOTE:** These need to be capitalized!)
max_users0	If a user joins that doesn't belong in the Country Code list specified below, this user limit applies.
max_users1	User limit that applies to cc_zone1
max_users2	User limit that applies to cc_zone2
max_users3	User limit that applies to cc_zone3

IP Address User Definitions

Variable	Explanation
ip_zone4_min	If you want to restrict users by IP address, enter the minimum IP address for the first IP zone here. (Verlihub allows you to define 3 IP zones.)
ip_zone4_max	If you want to restrict users by IP address, enter the maximum IP address for the first IP zone here. (Verlihub allows you to define 3 IP zones.)
ip_zone5_min	If you want to restrict users by IP address, enter the minimum IP address for the second IP zone here. (Verlihub allows you to define 3 IP zones.)
ip_zone5_max	If you want to restrict users by IP address, enter the maximum IP address for the second IP zone here. (Verlihub allows you to define 3 IP zones.)
ip_zone6_min	If you want to restrict users by IP address, enter the minimum IP address for the third IP zone here. (Verlihub allows you to define 3 IP zones.)
ip_zone6_max	If you want to restrict users by IP address, enter the maximum IP address for the third IP zone here. (Verlihub allows you to define 3 IP zones.)
max_users0	If a user joins that doesn't belong in the IP range specified below, this user limit applies.
max_users4	User limit that applies to ip_zone4_*
max_users5	User limit that applies to ip_zone5_*
max_users6	User limit that applies to ip_zone6_*

Miscellaneous settings

Variable	Explanation

max_message_size	
desc_insert_mode	Insert 'A', 'P', or '5' into user's description ? (1 = yes/0 = no)
use_reglist_cache	
use_penlist_cache	
save_lang	Set to 1 to enable multi-language. See this guide for more details.
timer_conn_period	
timer_serv_period	
timer_reloadcfg_period	
delayed_search	
delayed_myinfo	
reg_class_difference	The minimum difference between a class of user that is registering and the new user's class (for OPs adding new users)
step_delay	
bc_reply	When you set to 0 (zero), then the !bc commands don't send you replies from all back to your PM
log_level	
check_ctm	
check_rctm	
optimize_userlist	
ul_portion	

6. Managing your hub

Verlihub has various commands that range from adding registered users to banning them from the hub. This section has been broken into 5 parts: User Management, Kicking users, Banning users, Sending messages to users, and Muting users.

6.1 User management

This section of the manual deals specifically with commands and instructions for user management. Before you do any user management, make sure you are familiar with the next section, User classes.

6.1.1 User classes

There are several "classes" of users that can exist in verlihub. Below is a summary of the class types:

Class	Tag	Description
0	Regular users	Any user that connects to the hub
1	Registered users	Password protected nickname
2	VIP users	Special share limit
3	Operator user	Special share limit, kick/ban users, can register level 1 users

4	Cheef user	Mainly is allowed to register users and VIP members, but not allowed to create OPs
5	Admin user	Can create OPs, broadcast messages and change hub settings via !set
10	Master user	This user can do everything, including create more admin users, can't create another master user

6.1.2 Adding a registered user

First some clarification on who can add users. Normal users, VIP users, and Normal OPs (Level 3) can not add a registered user. Cheef OPs (Level 4) can add Registered users, and Admin OPs (Level 5) can create VIP users and Registered users. Master users (Level 10) can NOT create another Master user, however they can create Admin OPs, Cheef OPs, OPs, VIP users, and Registered users. All of this is summarized in this table below:

User Level	Can create users of level
0 (Normal users)	can't add users
1 (Registered user)	can't add users
2 (VIP user)	can't add users
3 (Regular OP)	can't add users
4 (Cheef user)	1 (Registered users)
5 (Admin user)	2 (VIP user) 1 (Registered user)
10 (Master user)	5 (Admin user) 4 (Cheef user) 3 (Regular OP) 2 (VIP user) 1 (Registered user)

With that out of the way, to register a user use the command **!regnewuser <nick> <class>**. As an example, here's how to create the user 'ARegUser' as a normal registered user:

```
!regnewuser ARegUser 1
```

NOTE: The user will need to use the +passwd command to register a password! They will be kicked from the hub if they don't change the password in 5 minutes!

TIP: If you have some users that can't figure out how to change their password you can do it for them by using the command **!regpass <nick> <password>**.

6.1.3 Removing a registered user

Only Admin OPs (Class 5) are allowed to remove registered users. Removing a registered user in done using the **!regdelete <nick>** command. For example, to remove user 'BadUser' use:

```
!regdelete BadUser
```

6.1.4 Temporarily disabling an account

Verlihub allows you to temporarily disable a registered nickname without permanently removing it from the database. This is done using the **!regdisable <nick>** command. **!regenable <nick>**. This is example of disabling user 'DisableMe' and then enabling it right away:

```
!regdisable DisableMe
!regenable DisableMe
```

6.1.5 Changing a user class

Ah yes, promotions (or demotions?) always seem to happen depending on user's behaviour. Should this ever happen, you can simply change the user's class with the command **!regclass <nick> <newclass>** command. The rules for adding new users apply here (see here). For example to change the user 'NewOP' to class 3 (Normal OP) use:

```
!regclass NewOP 3
```

To temporarily OP someone, use the command **!class <nick> <newclass>**. So to temporarily OP the user 'TempOP' use:

```
!class TempOP 3
```

6.1.6 Getting information on a user

There are two ways to get information about a user in Verlihub, both with slightly different information. If you want to see the user's registration information, use **!reginfo <nick>** and it will show you the information related to the user's registration. For example, to see information on user 'Someuser':

```
!reginfo Someuser
```

The other information will show simply the IP address, and the Hostname (if dns_lookup = 1 in the configuration). This command is simply **!getinfo <nick>**. So, to get information on 'Someuser' again, use:

```
!getinfo Someuser
```

If you want to get a list of nicknames by providing the IP address, use the command **!whoip <IP address>**. There are also two other ways to get information by IP:

- **!whorange <lower IP address>..<upper IP address>** (shows nicknames for a range of addresses)
- **!whorange <ip address>/<left mask bits>** (shows nicknames by subnet)

6.1.7 Registered user kick/ban protection

If you want to protect a registered user from accidental kicks, you can do this with the command **!regprotect <nick> <protect_class>**. What this does is allow you to grant protection from being kicked from certain classes of users. For example, to protect 'VIPUser' from normal OPs (Class 3) use this command:

```
!regprotect VIPUser 3
```

and this will prevent user from being kicked by any Class 3 user.

6.1.8 Recording a note about a user

To add a note in the registration database, use the **!regset <nick> note_op <msg>** command. An example:

```
!regset NewRegUser note_op <This is my friend, that's why I regged him>
```

6.1.9 Changing passwords

Once a user has registered a password once, he/she will need to inform the OP if for whatever reason they want to change it. The user will not be allowed to change the password without an OP knowing. To allow the user to set the password use the command **!regpass <nick>**. So, to allow user 'NeedPWChanged' to have their password changed:

```
!regpass NeedPWChanged
```

After this has been done, the user will be prompted to change their password. They will be kicked from the hub if they don't change the password in 5 minutes.

TIP: If you have some users that can't figure out how to change their password you can do it for them by using the command **!regpass <nick> <password>**.

6.1.10 Muting a user

You can also "mute" a given user temporarily from chatting. Using command **!gag <nick> [<time>]** will keep the user quiet for the time specified (the default is 2 days) or !ungag <nick> is used. For example to mute user 'Annoying':

```
!gag Annoying
```

And to unmute the user:

```
!ungag Annoying
```

6.1.11 Preventing a user from searching the hub

You can prevent users from searching the hub if they are abusing it. Using command **!nosearch <nick> [<time>]** will prevent the user from searching for the time specified (default is 2 days) or !unnosearch <nick> is used. For example to prevent user 'Annoying' from searching:

```
!nosearch Annoying
```

And to allow searches again:

```
!unnosearch Annoying
```

6.1.12 Preventing a user from private messaging others

You can prevent users from messaging others if they are abusing it. Using command **!nopm <nick> [<time>]** will prevent the user from sending PMs for the time specified (default is 2 days) or !unnopm <nick> is used. For example to prevent user 'Annoying' from sending messages:

```
!nopm Annoying
```

And to allow messages:

```
!unnopm Annoying
```

6.1.13 Preventing a user from connecting to others

You can prevent a user from connecting to others in your hub. Using command **!noctm <nick> [<time>]** will not allow the nickname specified to connect to anyone in the hub for the time specified (default is 2 days). For example to prevent user 'Annoying' from connecting to anyone:

```
!noctm Annoying
```

And to allow messages:

```
!unnoctm Annoying
```

6.2 Kicking users

Kicking users temporarily removes them from the channel. They will not be able to rejoin immediately, depending on the settings of *tban_kick* in the Hub settings. (The default is 5 minutes.) Kicking someone is done using the **!kick <nick> <reason>** command. You have to specify a reason to kick the user from the hub. For example to kick 'BadUser' because he isn't sharing enough use:

```
!kick BadUser You are not sharing enough.
```

If you don't want a public kick message to show in the chat window, use **!hidekick**. Alternatively, set the configuration variable *hide_all_kicks* to 1 to hide all kick message.

If you want to ban the IP of a user you just kicked and can only remember the nickname, use **!banip this nick <nick> <reason>**.

6.2.1 Banning during a kick

If you want to kick and ban a user with one command, simply add '_BAN_' somewhere in the kick reason. For example, to kick and ban 'BadUser' for not sharing enough:

```
!kick BadUser You are not sharing enough. _BAN_
```

Also, you can temporarily ban a user by appending the time unit after _BAN_ (for example, a 2 day ban you would use _BAN_2d). See Temporarily banning a user for more information.
This method bans both IP address and nickname.

6.3 Banning users

There are various ways to ban users fromt the hub. They are outlined below. **The best way to ban is during a kick. See Banning during a kick for information.**

6.3.1 Temporarily banning a user

If you need to, you can temporarily ban a user. First of all, you will need to know the time units that are available to you. They are illustrated in this table:

Time Unit	Abbreviation
Second	s
Minute	m

Hour	h
Day	d
Week	w
Month	M
Year	y

With that out of the way, just append the time unit after any of the ban commands, with an underscore seperating them. For example, to temporarily ban someone for 2 weeks, use:

```
!ban_2w BadPerson You have been temporarily banned.
```

6.3.2 Adding a ban (IP and nickname)

See Banning during a kick for information on how to do this.

6.3.3 Adding an IP ban

There are two commands to ban an IP address. The first is **!ban <ip> <reason>**, and the second is **!banip <ip> <reason>**. These commands are the same, it does not matter which one you use.

6.3.4 Adding a nickname ban

To ban a nickname, use the command **!bannick <nick> <reason>**.

If you want to ban the IP of a user you just kicked and can only remember the nickname, use **!banip this nick <nick> <reason>**.

6.3.5 Adding a hostname ban

Note: You need to configure variable *dns_lookup* for this to work. If you have a slow DNS server this can slow down the hub.

All of the examples will use hostname someuglynumbers.provider.whatever.com as an example.

Ban Command	Explanation
!banhost2 <hostname> <reason>	This will ban the second level of the given (in the example, whatever.com would be banned.)
!banhost3 <hostname> <reason>	This will ban the third level of the given (in the example, provider.whatever.com would be banned.)
!banhostr1 <hostname> <reason>	Bans the leftmost part of the hostname (in the example, "someuglynumbers" would be banned.)

6.3.6 Adding an email ban

To ban an email address, use the command **!banemail <email> <reason>**.

6.3.7 Adding a nick prefix ban

To ban a nickname prefix, use the command **!banprefix <nick_prefix> <reason>**.

NOTE: This one will affect the performance of users joining the hub, the more prefixes you ban the slower it will be!

6.3.8 Adding a share size ban

To ban a share size, use the command **!banshare <exact_share_size_in_bytes> <reason>**.

6.3.9 Banning a range of IP addresses

There are two ways to ban a range of addresses. The first way is simply to specify a range of IP addresses to ban using **!banrange <ipmin>..<ipmax> <reason>**.

The second way is to ban by subnet. To do this, specify an IP and the subnet mask to ban, using the command **!banrange <ip>/<left_mask_bits> <reason>**

NOTE: This may not work on big-endian machines!

6.3.10 Getting information on bans

If you want to get information about a ban, there is the **!infoban <nick_or_ip_or_whatever>** command. This method will try to list all possible matches for what is entered.

If you want to specify a specific type of ban use these three commands:

- **!infobannnick <nick>**
- **!infobanip <ip>**
- **!infobanrange <any_ip_from_the_range>**

6.3.11 Removing bans

You have to provide a reason for the unban. Even a dot (".") is allowed for the reason. The unbans are stored in database, and the Master user can view them.

This is a list of the various unban commands:

Unban Command	**Explanation**
!unban <ip_or_nick> <unban_reason>	Standard method for removing bans.
!unbanip <ip> <unban_reason>	Will remove only existing IP bans.
!unbannick <nick> <unban_reason>	Removes nickname bans.
!unbanhost2 <hostname> <unban_reason>	Removes a second-level hostname ban.
!unbanhost3 <hostname> <unban_reason>	Removes a third-level hostname ban.
!unbanhostr1 <hostname> <unban_reason>	Removes the leftmost part of a hostname ban.
!unbanemail <email> <unban_reason>	Removes an email ban.
!unbanprefix <prefix> <unban_reason>	Removes a nickname prefix ban.

!unbanshare <exact_share_in_bytes> <unban_reason>	Removes a share size ban.
!unbanrange <low IP value>..<high IP value> <unban_reason>	Removes a range of IP bans.

6.4 Sending messages to multiple users

Verlihub provides three methods for sending messages to multiple users: a simple broadcast to everyone, a message to OPs only, and a message to all registered users.

6.4.1 Sending a message to everyone

If you ever need to send a message to all users connected to your hub, use the **!broadcast <message>** command. For example, to send a message to everyone that the hub is going down for maintenance:

```
!broadcast This hub is going down for maintenance and will be back in 2
minutes.
```

6.4.2 Sending a message to OPs only

There are two ways to send a message to OPs only. First of all, if there is an OPChat bot in your hub, simply send it a message and it will relay that message to all of the other OPs.

Secondly, use the **!ops <message>** command. For example:

```
!ops We don't have an OPChat bot!
```

6.4.2 Sending a message to registered users only

If you want to send a message to your registered users, use the **!regs <message>** command. For example, to tell the registered users they need to change their password:

```
!regs You need to reset your password. PM an OP for more details.
```

7. Plugins

Verlihub supports the use of plugins. The actual plugin development is beyond the scope of this guide. If you are wondering how to compile a plugin, refer to the documentation that came with the plugin. The commands are listed here in case you decide to install one. **If you upgrade Verlihub, IT IS VITAL you recompile ALL of the plugins you have loaded!**

7.1 Listing loaded plugins

To list all loaded plugins use:

```
!pluglist all
```

7.2 Loading a plugin

After you have downloaded and compiled a plugin, use this command to load it:

```
!plugin <plugin_filename.so>
```

7.3 Unloading a plugin

To unload a plugin use:

```
!plugout <plugin_name>
```

7.4 Reloading a plugin

To reload a plugin, use:

```
!plugreload <plugin_name>
```

8. Hub command list index

This command list is more than likely incomplete, but should give you an idea of what verlihub can do.

Normal User Commands				
Level	**Command**	**Aliases**	**Parameters**	**Description**

9. Troubleshooting

At this point, this section will remain empty unless something with a real need deserves to go here. =P

9.1 Reporting Bugs

Please use this forum to report bugs.