

Eggdrop 1.4

DCC-Commands

Core Eggdrop help

This is the help provided in the core part of eggdrop, for module specific help, see later

+bot <bot> <address:botport#[/userport#]> [hostmask]

Available to: {+t}

Creates a user record for a new bot with the nickname given. The hostmask table will have one entry, either that specified, a host from a current user with the given nick, or "none", and the bot (b) flag will be set for the user. The internet address field of the user will also be set to the address given.

You can use this command and '**chpass**' to completely set up a record for a future bot, or you can let the two bots negotiate a password for themselves the first time they link. If the bot has a separate port for bots and users they should be separated with a slash (/).

-bot <bot>

Available to: {+t}

Exactly the same as '**-user**', but is included for convenience. It erases a user record.

See also: +bot, +user, -user

+host <nickname> <hostmask>

Available to: {+m|m}

Adds a hostmask to a user's record on the bot. The hostmasks are where the bot will identify that user from. Usually you will not need to use this command since a user can add hostmasks to her own record via the **/MSG IDENT** command, but it's here if you need it.

See also: -host

-host <hostmask>

Available to: {-}

Allows you to remove a host for yourself.

-host <nickname> <hostmask>

Available to: {+m|m}

Removes a hostmask from another user's record on the bot.

See also: +host

+ignore <hostmask> [comment]

Available to: {+m|-}

Adds a hostmask to the ignore list, with your nickname and an optional comment. An ignore added manually like this becomes "permenent", ie, it will not expire automatically. To make the bot stop ignoring this hostmask, you must eventually use a '**ignore**' command. This command can be used to either ignore users on irc, or to ignore incoming telnet connections.

See also: -ignore, ignores

-ignore <hostmask OR number>

Available to: {+m|-}

Removes the ignore from the list of ignores stored on the bot. You may also reference the ignore by the number shown in '.ignores'.

See also: +ignore, ignores

+lang <language>

Available to: {+n|-}

Adds a new language to the top of the language list. All sections are rechecked to test if there are language files supporting the language.

See also: -lang, +lsec, -lsec, lstat, relang

-lang <language>

Available to: {+n|-}

Removes a language from the language list. All sections are checked if they are using the language and if so, try to load the next available language instead.

See also: +lang, +lsec, -lsec, lstat, relang

+lsec <section>

Available to: {+n|-}

Adds a new section. The bot tries to load the section with the preferred language if possible.

See also: +lang, -lang, -lsec, lstat, relang

-lsec <section>

Available to: {+n|-}

Removes a section. The language bindings are not removed though.

See also: +lang, -lang, +lsec, lstat, relang

+user <nickname> <hostmask>

Available to: {+m}

Creates a new user record for the nickname given, with one entry in the hostmask table. The new user record will have no flags (i.e. not be a master, op, friend, etc) and no password.

-user <nickname>

Available to: {+m|-}

Erases the user record for the nickname given.

See also: +user, +bot, -bot

addlog <text>

Available to: {+to|o}

Writes your comment into the bot's log file. Bot masters can go back later and review the log, and will see your comment (with your handle attached). This is useful for explaining confusing activity.

away [away-message]

Available to: {-|}

Marks you as "away" on the party line. Your away message will show up in the **'who'** list, and will be displayed to anyone who tries to send you a note. Your notes will be stored, and then displayed to you as soon as you are no longer away. Saying something on the party line will automatically remove your "away" status, or you can type **'back'** or **'away'** by itself.

back

Available to: {-|}

This marks you un-away on the party line.

backup

Available to: {+m|+m}

makes the bot write a backup of its entire user list to the disk. this is useful if you feel the need to backup the user list.

See also: reload, rehash

banner <text>

Available to: {+t}

Displays a message to everyone currently using the bot's party line or file area -- useful for announcing that the bot will go down, etc.

binds [type]

Available to: {+m|-}

Shows the Tcl bindings in effect, in a list similar to this:

Command bindings:

TYPE	FLGS	COMMAND	HITS	BINDING (TCL)
Msg	-	rose	0	msg_rose
msg	-	go	0	msg_go
dcc	m -	bind	0	md_bind
pub	-	gross	0	pub_gross

The fields should be self-explanatory, except for HITS, which records the number of times that binding has been called.

If not, go read the file '[tcl-commands.pdf](#)' for help on the Tcl bind command. Note that the builtin commands are not shown.

You may also specify a type of binding to show (ie, '**.binds msg**') if you want to narrow the field a bit.

binds [type] all

Available to: {+m|-}

Displays all the Tcl bindings of every type (or the specified type), including the bindings for built in commands such as:

TYPE	FLGS	COMMAND	HITS	BINDING (TCL)
Msg	-	op	0	*msg:op

boot <nickname[@bot]> [reason]

Available to: {+t|-}

Will kick a user off the party line, and display the reason if you give one. You can attempt to boot someone from another bot (in a botnet), though it may be rejected if that bot does not allow remote boots. You can not boot the bot's owner.

botattr <nickname> [attributes] [channel]

Available to: {+t|-}

Lets you view and change the attributes (flags) field for a bot. Example: Sets Fred1 +share and -hub.

.botattr Fred1 +s-h

Whether or not you change any flags, it will show you the bot's attributes afterwards. To get a list of the flags possible, do '**help whois**'.

NOTE: This command is NOT used to replace .chattr, it modifies botflags such as +s, +h, +a, +u -bot specific flags only.

NOTE: You can't use this command on bots which are directly linked to your bot at the current moment.

See also: whois, chattr

botinfo

Available to: {+t|-}

Requests information from every bot currently in the botnet. Each bot should eventually send you one line listing its version and other information.

bots

Available to: {-|-}

Shows the list of bots currently in the botnet. Example:

Bots: cEvin, ruthie, Killa1

There is no indication of which bots are directly connected to this one. Use '**who**' or '**bottree**' for that information.

[v]bottree

Available to: {+t|-}

Shows a tree-format diagram of the bots currently in the botnet. It's just a nice way to get a feel for how the bots are connected physically. if 2 bots are sharing, a + will be indicated, or a ? if nothing is known.

Use vbottree if you want to know bot versions as well.

See also: bots, botinfo

chaddr <bot> <address:botport#/userport#>

Available to: {+t|-}

Changes the internet address for a bot. This is the address your bot will try to telnet to in order to create a connection and link up. If the bot has a separate port for bots and users they should be separated with a slash (/).

See also: link, +bot

chat <channel # or name or off>

Available to: {-|-}

Changes your channel on the dcc chat connection. when you first connect to the bot, it places you on channel 0 (the party line). You can move to another channel where basically nobody can see you (except anyone else who decides to join that channel). Valid channel numbers are 1 thru 99999.

Some channels may have assigned names if the assoc module is loaded. For those, you can specify the channel by name instead of number if you wish.

'**chat off**' removes you from any channel at all. You can still use normal bot commands and see the console, but you can't talk to anyone except via '.note'

'**chat on**' returns you to the party line (channel 0) if you were elsewhere.

chat <*channel# or name>

Available to: {-|-}

Same as above, but for channels available only to the bot you are on.

chattr <nickname> [attributes] [channel]

Available to: {+m|+m}

Lets you view and change the attributes (flags) field for a user. For example, to give Lamer the p and f flags:

```
.chattr Lamer +pf
```

Or to remove Denali from the global op list:

```
.chattr Denali -o
```

You may also do any combination of the above:

```
.chattr Fred1 -m+xj-o
```

You can also change the flags for users on a specific channel by supplying the channel after the attributes:

```
.chattr Usagi -m+dk-o #blah
```

Changing global and channel specific flags within the same command line is now possible! Example:

```
.chattr Bill f|o #lamer (global +f, +o #lamer)
```

Whether or not you change any flags, it will show you the user's attributes afterwards.

To get a list of the flags possible, do **'help whois'**.

NOTE: Only the owner may add or remove the 'n' (owner), 'm' (master) and 't' (botnet master) flags.

NOTE: It is pointless to -n a permanent owner. You must remove the permanent owner in the config file.

NOTE: This command can no longer be used to change bot flags, they are a separated entity, changeable with the **'botattr'** command.

See also: whois

chnick <oldnick> <newnick>

Available to: {+t|-}

Changes the handle/nickname of a user record. For example, to change the nickname of user 'gavroche' to 'jamie', you would type: 'chnick gavroche jamie'

chpass <handle> [newpassword]

Available to: {+t|-}

Changes a user's password on the bot. If you leave off the new password, the user effectively no longer has a password set. A password is needed to get ops, join the party line, and other things (but only required if one is set).

NOTE: In previous versions, setting a password to "nopass" would clear a user's password -- with encrypted passwords, this no longer works!

comment <user> <comment...>

Available to: {+m|-}

Creates or changes the comment field for a user. The comment field can only be seen via 'whois' or 'match'. Non-masters cannot see the comment field.

common (The c flag)

Available to: {+o|o}

Sometimes a lot of people use irc from the same account, i.e. example, a public telnet site. In that case, many people can have the same user@host. For that reason, they can't use a traditional user record because they will appear to all be the same person.

In such cases, create a user record for the account, and mark it +c, or "common access". Marking it as "common access" means that it shouldn't be used as a definitive address. When someone introduces themselves to the bot (ie, by /msg'ing "hello") and matches as a +c user, they will be given a new user record which identifies them BY NICKNAME.

Obviously people with that kind of user record shouldn't be given much special access, as when they change nicknames the bot will identify them as someone else.

console [channel] [modes]

Available to: {+to|+o}

Changes your console level, so that you will see only those types of console messages that you want to. Your current console channel is the channel (that the bot is on) which you can view from the party line, and which channel-specific commands (like 'say' and 'op') take affect on.

The mode can also be a modifier like '+p' or '-jk' or '+mp-b'. If you omit the channel and modes, it will show your current console channel and setting.

Valid levels are:

- m display private msgs/ctcps to the bot
- p display public talk and ctcps on the channel
- k display kicks/bans/mode changes on the channel
- j display joins/parts/nick changes/signoffs/etc on the channel
- b display bot links/unlinks/userfile-sharing
- s display server messages and connect/disconnects
- w display msgs between IRCops (wallops)

Available to: {+m|+m}

- c display user commands (dcc and msg)
- o splay other bot notices [HIGHLY RECOMMENDED]

Available to: {+m|-}

- x display file transfers and file-area commands
- d display debug messages that only coders would care about

Available to: {+n|-}

- r display all raw text from the server (if enabled)
- v display raw text SENT to the server (if enabled)

-- There are also 8 user-defined console modes '1' thru '8' --

console <user> [channel] [modes]

Available to: {+m|+m}

Is used to set the console level of another user. This can even be used on users who normally would not be able to set their own console mode.

dccstat

Available to: {+m|-}

Displays a table-format list of all the "dcc" connections the bot is handling. "dcc" means "direct client-to-client communication" and eggdrop expands this to cover every open socket. so any type of network connection to the bot is considered a "dcc" connection.

the headings of the table are:

- SOCK the socket of this connection (always unique)
- ADDR the IP-number of the host the connection is to, if applicable
- PORT the port number being used for communication
- NICK the nickname of the user or bot, if it's a user or bot
- HOST sometimes, the hostname corresponding to the IP address
- TYPE the type of dcc connection (see below)

the types of connection currently possible are as follows (but more are being added literally all the time):

- chat user in dcc-chat command mode
- pass user entering dcc chat (being asked for her password)
- send raw data connection: user sending a file
- get raw data connection: sending a file to a user
- getp pending get (waiting for the user to acknowledge)
- lstn telnet listening port (in place of a hostname, it will show the proc to call, or mask of acceptable nicks)
- t-in incoming telnet user (being asked for his nickname)
- file user in dcc-chat file area
- bot bot linked in (aka botnet connection)
- bot* pending bot link (waiting for acknowledgement)
- rela user in relay connection to another bot
- >rly bot being relay'd to (one for each "rela")
- conn pending telnet connection (chat, relay, bot-link, etc)
- new new user via telnet, entering a handle
- newp new user via telnet, entering a password

in addition, 'chat' and 'bot' have flags listed for each connection. capital letters mean the flag is on, and lowercase letters mean the flag is off.

For 'chat', the flags are:

- C in file area, but allowed to return to party line
- P party line access only (no +o access)
- T telnet connection (instead of dcc chat)
- E echo is on
- P use is paging

For 'bot', the flags are:

- P ping sent, waiting for reply
- U user-file sharing is active
- C this bot initiated the connection
- O user-file offered, waiting for reply
- S in the process of sending the user-file
- G in the process of getting the user-file
- W warned this bot to stop hubbing
- L leafed bot (not allowed to hub)
- I bot is currently in the 'linking' stage
- A bot is being aggressively shared with for 'chat' users, the party-line channel is also listed.

debug

Available to: {+m|-}

will display a dump of memory allocation, assuming the bot was compiled with DEBUG defined. it's useless to anyone but program- mers hacking on the bot and trying to find memory leaks.

die [reason]

Available to: {+n|-}

kills the bot. the bot goes offline immediately, logging who gave the 'die' command. you shouldn't have to use this too often. if you specify a reason, it's logged, otherwise the reason is "authorized by <nickname>".

echo <on/off>

Available to: {-}

sets whether you want your messages echoed back to you. if it's on, then when you say something on the party line, it will be displayed to you just like everyone else will see it. if it's off, then that won't happen.

fixcodes

use this in those situations where the bot gets mixed up about your type of connect, eg: when you /ctcp chat the bot & it thinks you're telnetting, and you're actually using dcc, it effectively switches telnet on and off.

help [<command>/ <module> module]

Available to: {-}

All commands begin with '.' and all else goes to the party line. Text starting with ', ' is sent ONLY to bot-masters. Text starting with '' is NOT send over the botnet. You may receive a list of all standart commands by '.help'. You can get help on individual commands: '.help <command>'

Extra help relating to loaded modules may be obtained by typing '.help <module> module', possible modules include: **assoc channels console ctcp transfer irc notes seen server share**

These are distributed with the bot, and they may be loaded, others can be added also. You may receive a list of commands in all loaded modules by '.help all'.

If you only remember a part of the command's name you are searching for, just use wildcards (e.g. '.help *bot*') and all matching help texts will be displayed.

ignores [wildcard]

Available to: {+m|-}

lists the hostmasks that the bot is currently discarding msgs, etc from. the ignores are numbered, so they can be easily removed with '-ignore' by number. a permanent ignore looks like this:

```
[ 1] *!onaji@* (perm)
```

```
Xerxes: spamming me
```

```
Started 18:38
```

this kind of ignore can only be stopped by using '-ignore'.

Xerxes is the one who added it, and his comment is "spamming me". he added it at about 18:38 today. a temporary ignore looks like this:

```
[ 2] *!*@shellx.best.com (expires at 19:59)
```

```
MyBot: msg/notice flood
```

```
Started 18:59
```

that means the ignore was made automatically by the bot at 18:59, because someone flooded it. it can be removed with

'**ignore**', or it will expire an hour after it started (this is set in '**ignore-time**').

if you use '**ignores <wildcard>**' it will list all the ignores that match against your wildcard.

See also: +ignore, -ignore, set ignore-time

link [some-bot] <new-bot>

Available to: {+t|-}

tries to link in a new bot. if you use the form

.link newbot

then your bot must have a record for the new bot (see '**+bot**'). you can either set a password on each end or they will negotiate one the first time they link. if you use the form .link somebot newbot then the request will be sent to "somebot" for it to link to "newbot".

loadmod <module>

Available to: {+n|-}

Loads a module.

See also: unloadmod

lstat

Available to: {+n|-}

Shows the language statistics, all selected languages and a list of all language sections.

See also: +lang, -lang, +lsec, -lsec, relang

match <attr> [channel] [[start] limit]

Available to: {+o|+o}

shows you user records with the attributes requested:

<attr> is of the form: <+/-><global>[&/<channel>[&/<bot>]] specifying '&' as the speperator, will cause AND style matching, '|' will cause OR style matching, channel flags are match vs the specified channel (or your console channel), and bot is match vs the sepreate bot flags.

some examples:

.match +o will match any GLOBAL +o's

.match -o&+o will match anyone without a global +o AND with a channel +o on your console channel

`.match +f|+f` will match anyone with a global +f or a channel +f on your console channel
`.match |f #fred` will match anyone with +f on channel #fred

match <wildcard-string> [[start] limit]

Available to: {+o|+o}

shows you user records where the nickname or any of the hostmasks match the wildcard string shown.

the default limit is 20, meaning that if more than 20 user records match your requirements, only the first 20 will be shown. you can change this limit by specifying one on the command line, though. if you include a limit, you may also include a starting place (the default is 1). for example, to list the first 15 users who have hostmasks from clemson.edu:

```
.match *clemson.edu 15
```

to get the next 10 (if there are more than 15), you ask to list numbers 16 thru 25:

```
.match *clemson.edu 16 25
```

me <text>

Available to: {-|}

performs an action on the party line.

module [modulename]

Available to: {+n|-}

Reports the names of currently loaded modules, including dependencies & status report, you can get info only on a

specific module by specifying it.

modules <botname>

Available to: {+n|-}

requests a listing of the modules running on the remote bot

motd

Available to: {-|}

will redisplay the party-line message of the day, which was shown when you first joined.

newpass <password>

Available to: {-}

changes your password on the bot, just like the /msg command 'pass', except you don't need to type your old password.

nick <new-nick>

Available to: {-}

changes your nickname on the bot. that's the handle that the bot will know you as forever and ever, and what you have to use when telnetting in or IDENT'ing yourself, so remember it.

note <nickname[@bot]> <message>

sends a private note to a user on the party line. if that user is currently on the party line, and not marked as away, she will receive the message immediately. otherwise it may be stored and displayed the next time that user joins the party line. If you join the channel, and have notes stored for you on the bot, it will tell you. To send a note to someone on a different bot, use "nick@bot" for the nickname.

See also: whom, notes

page <# or off>

Available to: {-}

allows you to slow down the number of lines the bot sends to you at once while you're in chat mode. with this on, any commands that send greater than the specified number of lines will stop when the number is reached, and wait for you to type another command (or press enter) to continue. if you have too many lines waiting to be seen you may be booted off the bot.

quit [comment]

Available to: {-}

removes you from the party line and disconnects your dcc chat session with the bot. if you specify a comment, it will be

displayed to other users as you leave.

rehash

Available to: {+m|-}

reloads the config file for the bot, which resets all the 'set' variables and reloads any TCL scripts your config loads. It also saves and reloads the user file from disk, just to give your disk a little extra wear.

See also: save, reload, restart

rehelp

Available to: {+n|-}

This command gets the bot to recheck all it's help files for new commands that have been added.

relang

Available to: {+n|-}

Try to find language files with a more preferred language than right now.

See also: +lang, -lang, +lsec, -lsec, lstat

relay <botname>

Available to: {+t|-}

will relay you via telnet to another bot that your bot knows of, whether or not they are currently connected. your dcc-chat/telnet connection to this bot will be relayed to the other bot until the other bot drops your relay, or until you send "*bye*" on a line by itself.

See also: bots

reload

Available to: {+m|+m}

reloads the user file of the bot, discarding any changes made since the last '.save' or hourly user file save. sharebots

should probably not ever do this.

See also: rehash, save

restart

Available to: {+m|-}

restarts the Tcl interpreter, wipes all timers and utimers, and reloads the config file for the bot, which resets all the 'set' variables and reloads any TCL scripts your config loads. It also saves and reloads the user file from disk, just to give your disk a little extra wear. it's also the most risky command to run, because it's never ever really worked properly.

See also: rehash, reload, save

save

Available to: {+m|+m}

makes the bot write its entire user list to the disk. this is useful if you think the bot is about to crash or something, since the user file is only written to disk about once an hour.

See also: reload, rehash

set <variable> [value]

Available to: {+n|-}

changes the values of internal settings of the bot, or, if used without a value parameter, shows the current value of a setting. the bot starts out with settings as given in the configuration file. here are the variables eggdrop cares about:

mod-path	nat-ip	dcc-flood-thr	die-on-sighup
die-on-sigterm	Remote-boots	max-dcc	max-logs
enable-simul	Protect-telnet	ident-timeout	share-unlinks
allow-dk-cmds	botnet-nick	help-path	temp-path
userfile	Motd	notify-newusers	admin
owner	my-hostname	my-ip	network
save-users-at	switch-logfiles-at	connect-timeout	reserved-port
log-time	Require-p	keep-all-logs	open-telnets
firewall	Console	default-flags	whois-fields
hourly-updates	Username	ignore-time	uptime
Version	Numversion	telnet-flood	

use **'`.help set <variable>`'** for more info on each variable.

simul <nickname> <text>

Available to: {+n|-}

if the user is on the party line, the bot will simulate them typing the text you specify. example:

```
.simul dweeb .quit
```

makes it just as if "dweeb" typed ".quit". this command will not work unless eggdrop has simul enabled in the config file. Please use this command ethically.

status [all]

displays a condensed block of information about how the bot is running. here's an example from Snowbot:

I am Snowbot, running eggdrop v1.1.1: 400 users (mem: 42k)

Running on BSDI 2.5

Admin: Robey <robey@wc130.residence.gatech.edu>

#turtle : 2 members, enforcing "+tn-mlk" (greet, bitch)

#gloom : 3 members, enforcing "+tn-i" (greet)

Server minnie.cc.utexas.edu:6667

Online for 4 days, 06:18 (background) CPU 14:18 cache hit 55.3%%

the first line tells you the bot's name, what version of eggdrop it's running, the number of users the bot has records of, and the amount of memory being used.

the second line, tell what operating system the bot is running under.

the third line, if present, is the contents of your 'admin' line in the config file.

next is a line for each of the channels your bot is currently monitoring. for each channel, it lists the number of people on the channel ("members"), what channel modes it's trying to enforce, and which options are set for that channel.

if your bot is sharing user files with anyone, and is currently either transmitting or receiving the userfile from a bot, it will say so here, and tell how many users have been received so far, or how many more users need to be transmitted.

the next few lines show the current server, and if the helpbot is in use, its nickname and current server. to prevent server flooding, both the bot and its helpbot use queues to store replies to queries from users. if either queue has a significant amount of information in it, this status display will show how full (%) the queue is.

the next line tells you how long the bot has been online, and whether it is operating in the background or foreground. if it's in the foreground, it will tell you whether it's in terminal mode or channel display mode. (terminal mode makes the local console simulate a dcc chat connection, and channel display mode displays a '.status' and '.channel' output every few seconds.) it will also give you an estimate of the amount of CPU time (minutes and seconds) the bot has used so far, and how well the cache is working. (the cache is an attempt to keep the CPU time down, and anything over 50% or so is just great.)

if you do **'.status all'** you will see another pageful of info, showing every internal setting as specified in the config file. it should be self-explanatory to anyone who needs to use it.

See also: set, channel, debug

strip [modes]

Available to: {-}

enables you to remove embedded 'attribute' codes from within a section of text. Valid options are:

- b remove all boldface codes
- c remove all color codes
- r remove all reverse video codes

- u remove all underline codes
- a remove all ansi codes
- g remove all ctrl-g (bell) codes

the mode can also be a modifier like '+c' or '-bu' or '+ru-c'. if you omit modes, it will show your current setting.

strip <user> [modes]

Available to: {+m|-}

is used to set the strip level of another user, a master can't set their own strip flags without starting the modes with a +/-

su <user>

Available to: {-|-}

changes your username to that of another. If you are an owner this does not require a password otherwise, you must enter the password of <user> **.quit** returns you to your original user.

trace <bot>

sends out a trace signal to another bot. if that trace signal returns (and it should!) you will get an output something like this:

```
Trace result -> Valis:Stonewall:NoBoty:SomeBoty
```

it's a list of the bots between you and the destination bot. it should also return the time in seconds it took for the trace to occur.

See also: bots, bottree

unlink <botname>

Available to: {+t|-}

disconnects the named bot from the botnet, assuming it was connected in the first place. some bots (sharebots in particular) might not allow you to unlink them.

See also: link, bots, bottree

unlink *

Available to: {+t|-}

disconnects from all bots

See also: link, bots, bottree

unloadmod <module>

Available to: {+n-}

Unloads a module.

See also: loadmod

uptime

Available to: {+m|m}

displays the bot's current uptime like .status would show.

See also: status

who [bot]

Available to: {-}

lists people on this bot, or bots connected to this bot. The first section is people on your current channel (the party line, if you haven't changed channels) who are on the bot. a '*' next to the nickname means that user is an owner; '+' means they're a master; '@' means they're an op, '%' means they are a botnet master. the user's nickname, hostname, and possibly an idle time and/or away message will be displayed:

NOTE: Owners will also see the users dcc idx

*Robey telnet: nowhere.com

the next section is bots directly linked to this one:

-> MagField (14 Oct 01:16) eggdrop v1.0j

(btw, if you're still using 1.0j, you need to upgrade ;)

the arrow indicates who initiated the connection -- the right arrow means this bot connected to MagField. a left arrow means they connected to us. a '+' next to the arrow indicates that the bot is sharing userfiles with us. the connection time (14 Oct 01:16) and bot version are also shown.

Nowadays it may also show the network the bot resides on.

NOTE: Owners will also see the bots dcc idx

the final section is a list of users on this bot who are not on your channel. (this section may be omitted if everyone is on the same channel as you.)

if you specify a bot's name, as in **'.who valis'**, the who request will be sent to that bot, as long as that bot is on the botnet.

NOTE: masters may also see "(con)" after a user's entry, this shows that user's console modes (see the **'.console'** help page). And in the final section (users that aren't on this channel), masters will see the actual channel other users are on. also, people in the file system will be listed (as being in channel "files") -- a '+' next to the nickname here means the user has access to return to the party line.

See also: whom

whois <nickname>

Available to: {+o|+o}

will show you the bot information about a user record. there are five headings:

HANDLE the handle (nickname) of the user

PASS "yes" if she has a password set, "no" if not

NOTES number of notes stored waiting for the user

FLAGS the list of flags for this user (see below)

LAST the time or date that the user was last on the channel

the valid flags under FLAGS are:

- o global op (bot will op this user on any channel)
- m master (user is a bot master)
- n owner (user is the bot owner)
- t botnet master (user is a botnet master)
- x xfer (user has file-area access)
- j janitor (user is a file-area master)
- p party-line (user has party-line access)
- c common (user record is a common-access site) [see 'help common']
- u unshared (not sent to other share-bots)
- b bot (user is another bot)
- d global deop (user cannot get ops)
- k global auto-kick (user kicked & banned automatically)
- f global friend (user is not punished for doing bad things)
- v global voice (user get +v automatically on +autovoice channels)
- a global auto-op (always op this user on join)
- h high-light flag, user sees highlighting in text output
- w wasop-test (needs wasop test for +stopnethack procedure)
- g give auto-voice (works as +a but for voice instead)

(plus 26 user-defined flags, A-Z)

each channel that the user has joined will have a specific record for it, with the channel-specific flags and possibly an info line. the channel-specific flags are:

- m master (user is a master for the channel)
- n owner (user is an owner for the channel)
- o op (bot will give this user chanop)
- d deop (bot will not allow this user to become a chanop)
- f friend (user is not punished for doing bad things)
- k kick (user is auto-kicked)
- v voice (user gets +v automatically on +autovoice channels)
- q quiet (user never gets +v on +autovoice channels)
- a auto-op (always op this user on join)
- w wasop-test (needs wasop test for +stopnethack procedure)
- g give auto-voice (works as +a but for voice instead)

(plus 26 user-defined flags, A-Z)

hostmasks for the user are displayed on the following lines. if the user entry is for a bot, there will be a line below which says "ADDRESS:" and gives the bot's telnet address. some user entries may have "EMAIL:" and "INFO:" entries too.

There may be additional information displayed depending on the modules loaded.

for bots, there are additional flags seen under the BOTATTR: entry, these include:

- s share (bot is sharing user records, aggressively)
- p share (bot is sharing user records, passively)
- g global share (share all channels)
- h hub (bot is auto-linked at highest priority)
- a alternate (bot is auto-linked if no hub bots can be linked)
- l leaf (bot is not allowed to link in other bots)
- r reject (bot will not be allowed on the net)
- i isolate (isolate the party line across a botlink)

(plus 10 user-defined flags, 0-9)

bots may also have +s (share) settings for individual channels

NOTE: masters: if there is a comment, you will see it under "COMMENT:"

See also: match

whom [channel]

Available to: {-}

will display a table of the users currently on the joint partyline (formed when you have two or more bots linked together). it will show each user's nickname, preceded by a "*" if they are the owner of their bot, "+" if they are a master, "%" if they are a botnet master or "@" if they are an op. in other columns it will show which bot the user is on, and his or her host.

if the user is away, the away message will be shown. likewise, if they are idle, the idle time will be displayed.

if you are on a different channel than the party line (channel 0), then this will show the people on every bot who are on your current channel. you can also optionally give the "channel" option to list who is on a particular channel.

See also: who, bots, chat

Module help

IRC Module Help

the irc module provides the following dcc commands:

act [channel] <text>

Available to: {+o|+o}

Performs an action on the current console channel (or otherwise specified channel), as if the bot did it. Just like the /me command in IRC.

See also: console

adduser [!]<nickname> [handle]

Available to: {+m|+m}

Creates a new user record for a user on the channel, using their current hostname. It's similar to a user msg'ing the bot 'hello' except that no information is sent to that user. If the bot already knows someone by that nickname, and the user on the channel doesn't have a bot record, then it does the equivalent of an 'ident' for that user -- except that, again, no information is sent to the user telling them that anything was done.

If the user is using a different nickname than the bot normally knows her by, you can specify her "handle" (the nickname that the bot remembers).

If you want to add a user using a static hostmask, prefix their nick with a '!'. ie .adduser !Lamer

See also: +host, -host, +user, -user

channel [channel-name]

Available to: {+o|+o}

Shows you an extensive display of the users on a channel, and the current channel attributes. By default, it shows you the channel you are currently viewing on the console, but you can specify another channel if you wish.

The first line will look like:

Channel #hiya, 8 members, 45 users, mode +tn:

This means that the bot is sitting on channel #hiya, where 8 other irc'ers are. There are 45 people that the bot knows by hostmask, and the channel mode is +tn. If the bot isn't on the channel it is supposed to be on, it will say "Desiring channel #hiya" instead. Next is a list of the users on the channel, with each entry looking like this:

NICKNAME	HANDLE	JOIN	IDLE	USER@HOST
----------	--------	------	------	-----------

@kantSF kantSF 14:53 o 6m josh@random.edu

The "@kantSF" means that the user's nickname is kantSF and that he is a chanop. The second "kantSF" is the nickname that the bot knows him by. Sometimes this will differ from the nickname a person is using. The time displayed is the time the user joined the channel. The next field is the attributes:

- n bot owner
- o can get ops (+o)
- m bot master or owner
- f channel friend
- b another bot
- d cannot get ops (+d)

The last field is the user@host he is using irc from.

See also: status, whois

deluser <nickname>

Available to: {+m|+m}

deletes a user record for a user on the channel, using their current hostname. Channel masters can remove users so long as the user isn't a bot master.

See also: adduser, +user, -user

deop <nickname> [channel]

Available to: {+o|+o}

will remove chanop from the person you specify, so long as the bot is opped on that channel, and the person you specify isn't on the bot's list of authorized chanops.

See also: op, console

invite <nickname> [channel]

Available to: {+o|+o}

invites someone from irc into your current console channel (or specified other channel). this is most useful when the channel is +i. a user with the +o flag can also request an invite from the bot with /MSG INVITE

See also: console

kick [channel] <nickname> [reason]

Available to: {+o|+o}

will kick a user off your current console channel (or specified other channel) with the comment given. if you omit the reason, the default kick comment is "requested".

See also: kickban, console

kickban [channel] [-|@]<nickname> [comment]

Available to: {+o|+o}

kicks a user off the channel and bans her by a reasonable hostmask. your nickname will be attached to the ban in the bot's internal ban list, and the ban will last for whatever is set in ban-time -- only on this channel. use '+ban' for a more permanent ban which will be activated on every channel the bot monitors. if you use a comment, that will also be attached to the ban in the ban list, and used as the kick comment. appending a prefix of '-' or '@' to a nickname changes the ban mask used:

eg. with a host of nick!ident@host.name.domain

- | command | banmask |
|------------------|--------------------------|
| • .kickban nick | *!*dent@*.name.domain |
| • .kickban -nick | *!*dent@host.name.domain |
| • .kickban @nick | *!*@host.name.domain |

with a host of nick!~ident@host.name.domain (strict-host set to 1)

- | command | banmask |
|------------------|---------------------------|
| • .kickban nick | *!*ident@*.name.domain |
| • .kickban -nick | *!*ident@host.name.domain |
| • .kickban @nick | *!*@host.name.domain |

See also: +ban, bans, stick

msg <nickname> <text>

Available to: {+o|-}

sends a private message to someone from the bot, just as if the bot had typed /msg.

op <nickname> [channel]

Available to: {+o|+o}

will grant chanop to the person you specify, so long as the bot is opped on that channel, and the person you specify isn't being actively deopped by the bot.

See also: deop, console

reset [channel]

Available to: {+m}

clears out the bot's channel information and makes it gather the information from the server all over again, as if it had just joined that channel. it's not really useful much, but could be if an odd bug causes the channel information to get scrambled. Unfortunately this command used to get a lot of use. you can omit the channel name to make it reset ALL channels.

resetbans [channel]

Available to: {+o|+o}

resets the bot's ban list for the channel. any bans on the channel that aren't in the ban list (either the global list or the local channel ban list) will be removed, and if there are any bans in the global ban list or channel ban list that are not currently on the channel, they will be added.

See also: bans, reset, console

resetexempts [channel]

Available to: {+o|+o}

resets the bot's exemption list for the channel. this command behaves exactly like resetbans, except it is for exempts.

See also: resetbans, resetinvites

resetinvites [channel]

Available to: {+o|+o}

resets the bot's invitation list for the channel. this command behaves exactly like resetbans, except it is for invites.

See also: resetbans, resetinvites

say [channel] <text>

Available to: {+o|+o}

dumps the text to your current console channel (or other specified channel), as if the bot "said" it.

topic <text>

Available to: {+o|+o}

changes the channel's topic, assuming the bot is a chanop or the channel is not +t (uses your current console channel).

See also: console

Assoc Module Help

The Assoc module provides botnet channel naming and have following dcc commmands

assoc

Available to: {-|}

Displays a list of current botnet channel names.

assoc <channel #> <name>

Available to: {+t|}

Creates a name for a channel. The name will propagate across your botnet (if you're hooked into one) and can be used instead of the channel # when changing chat channels (see '.chat'). If you omit the channel # and name, it will just dump a list of the current named channels.

See Also: chat

assoc <*channel #> <name>

Available to: {+t|}

Creates a name for a local channel. These channels are local to the bot itself.

See Also: chat

assoc <[*]channel #>

Available to: {+t|}

Wipes out the name for a channel, if there was one.

Channels Module Help

The channels module provides the means for STORING channel information about users, it DOES NOT provide the means for sitting on a channel (this is in the irc module). The following commands are provided by the channels module:

+ban <hostmask> [channel] [%bantime <XdXhXm>] [comment]

Available to: {+o|o}

Adds a ban to the list of permanent bans stored on the bot, with optional comment and bantime. This ban will be in effect for every channel on the bot if no channel is supplied, and is stored with your nickname and comment. Prefixing a comment with @ will make it only visible within the bot. Bantime has to be expressed in days, hours and minutes.

See also: bans, -ban, stick, unstick

-ban <hostmask OR number>

Available to: {+o|+o}

Removes the ban from the list of permanent bans stored on the bot -- you may reference the ban by the number shown in **.bans**.

See also: bans, +ban, stick, unstick

+chan <channel>

Available to: {+n}

Lets you add another channel to the bot, this cause the bot to join the channel, and for you to make changes to it's settings. This channel is not permanent unless "**chanfile**" has been defined in the config file, and is saved everytime the userfile is saved, or by using **.chansave**

See also: +chan, -chan, chanload, chanset, chaninfo

-chan <channel>

Available to: {+n|-}

This removes ALL information about a channel from the bot. It's not made permanent unless a channel-file has been defined in the user file, in which case it's saved each time the usefile is, or by using **.chansave**.

***** IMPORTANT *****

This erases ALL information about the channel, including channel settings and channel records for users -- **EVERYTHING**.

DO NOT use it to have the bot temporarily leave a channel. This command is for abandoning a channel (e.g. the channel will have to be redefined and all user flags for that channel will have to be redone).

Even if you don't have chanfile defined, it will still erase all the channel records for users on the next userfile save.

See also: +chan, chansave, chanload, chanset, chaninfo

+chrec <handle> [channel]

Available to: {+m|+m}

Adds an empty channel record for the user so that channel lastons and info lines can be saved. No flags are associated with the channel.

See also: chattr

-chrec <handle> [channel]

Available to: {+m|m}

Deletes a channel record for the user, including channel lastons, info lines, and flags.

See also: chattr

+exempt <hostmask> [channel] [%exempttime <XdXhXm>] [comment]

Available to: {+o|o}

Adds a ban exempt to the list of permanent exempts stored on the bot, with optional comment and exempttime. This exempt will be in effect for every channel on the bot if no channel is supplied, and is stored with your nickname and comment. Prefixing a comment with @ will make it only visible within the bot. Exempttime has to be expressed in days, hours and minutes.

See also: exempts, -exempt, stick, unstick

-exempt <hostmask OR number>

Available to: {+o|+o}

Removes the exemption from the list of permanent exempts stored on the bot -- you may reference the exempt by the number shown in **'.exempts'**.

See also: exempts, +exempt, stick, unstick

+invite <hostmask> [channel] [%invitetime <XdXhXm>] [comment]

Available to: {+o|o}

Adds a invite to the list of permanent invites stored on the bot, with optional comment and invitetime. This invite will be in effect for every channel on the bot if no channel is supplied, and is stored with your nickname and comment. Prefixing a comment with @ will make it only visible within the bot. invitetime has to be expressed in days, hours and minutes.

See also: invites, -invite, stick, unstick

-invite <hostmask OR number>

Available to: {+o|+o}

Removes the invite from the list of permanent invites stored on the bot -- you may reference the ban by the number shown in **'.invites'**.

See also: invites, +invite, stick, unstick

bans [[channel] wildcard]

bans all

Available to: {+o|+o}

Shows you a list of the global bans active on the current channel, and the list of channel-specific bans, as well as any bans that are on the channel but weren't placed by the bot.

Here's a sample entry;

```
[ 5] !!*habib@*frys.com (perm)
```

paulie: revolving check policy

Created 15:10

The number (5) can be used to reference the ban if you wish to remove it (see `!-ban`). Next is the actual hostmask being banned. The "(perm)" means that the ban is "permanent": that is, it doesn't automatically expire. If there is an elapsed time showing instead, the time displayed is how long the ban has been active. Those kind of bans expire after one hour. The second line of the ban entry is the comment ("revolving check policy" – it would seem that paulie had to stand in line for a while) and who put the ban there (paulie). The last line shows when the ban was added, and possibly the last time the ban was activated on the channel (if it's different from the creation time).

Sometimes there will be a "!" or "*" right before the number. a "!" means the ban is in the bot's permaban list, but is not currently on the channel. a "*" marks a ban which is NOT in the permaban list but IS on the channel.

As you use 'bans' it will show you only the bans which are currently in action on the channel. If you use 'bans all' it will show you every ban in memory (with 'perm-bans' on, these are identical).

If you use 'bans <wildcard>' it will list all the bans (active or not) that match against your wildcard. Consider it a 'bans all' list matched against your wildcard.

The ban list may change according to which channel you're currently viewing in the console. Different bans may be active on different channels. If you specify a channel name, that channel will be used instead of your current console channel.

See also: `-ban`, `+ban`, `console`, `set ban-time`, `stick`, `unstick`

chaninfo <channel>

Available to: {+m|m}

This lists all the settings for the bot on the given channel. It shows any of the following:

- `chanmode` which modes are enforced on the channel, both + and - modes can be enforced
- `idle-kick` kick idle users (non +f's anyway) on the channel after how many minutes (use 0, or `dont-idle-kick` to turn this off)

ATTENTION: Due to security concerns, only permanent owners can set the `need-*` modes.

- `need-op` tcl command to execute when the bot doesn't have ops

- need-invite tcl command to execute when the bot needs to be invited to the channel (ie channel has mode +i)
- need-key tcl command to execute when the bot needs to find out the key to the channel
- need-unban tcl command to execute when the bot is banned and can't get on the channel
- need-limit tcl command to execute when the bot can't join the channel because the channel is full

The following can be set + or - (eg .chanset #channel -clearbans) clearbans clear all the channel bans when the bot joins the channel?

- need-limit tcl command to execute when the bot can't join the channel because the channel is full
- enforcebans kicks people who match channel bans
- dynamicbans only activate bans on channel when they are needed
- userbans allow other users to place bans on the channel
- dynamicexempts only activate exempts on channel when needed
- userexempts allow exempts to be set by users directly
- dynamicinvites only activate invites on channel when needed?
- userinvites allow invites to be made by users directly?
- autoop automatically op users when they join the channel
- bitch only allow users with the +o flag to be ops on the channel
- greet say a users info line when they join the channel
- protectops reop a +o user who gets deopped
- protectfriends reop a +f user who gets deopped?
- dontkickops never kick +o flag people
- wasoptest does a wasop test for +stopnethacks
- inactive never join this channel, or leave if already joined. Usefull as you can make the bot leave a channel without loosing it's settings or affecting any sharing.
- statuslog log the channel status every 5 minutes
- stopnethack deop anyone who enters the channel with serverops
- cycle cycle the channel when it becomes op-less
- revenge take revenge on anyone who deops or kicks friends or ops revengebot take revenge on anyone who deops or kicks the bot

- secret don't provide info about the channel over the botnet
- shared share user settings for this channel (**NOTE:** this can be set online now)
- autovoice this causes the bot to also monitor channel voice settings, using the +v/+q settings.

The following are flood settings, they are set by going

`.chanset <channel> flood-type number:seconds`

where number & seconds are integers indicating the number of times in how many seconds the flood will be triggered.

`.chanset <channel> flood-type 0:0`

(or 0:1 or 1:0) will deactivate the respective flood setting.

- flood-chan this defines the flood level for public chatter & actions
- flood-ctcp this defines the flood level for ctcp's to the channel (include PING & VERSION), except for ACTION's
- flood-join this defines the number of joins from a give user@host that constituts a flood
- flood-kick how many kicks in the given time are a flood
- flood-deop deops in the given time

See also: +chan, -chan, chansave, chanload, chanset

chanload

Available to: {+n|+n}

Reloads the channel settings from the file defined by the "chanfile" setting. This allows you to return to a previous setting if you've really messed things up.

See also: +chan, -chan, chanload, chanset, chaninfo

chanset <channel> <settings>

Available to: {+n|+n}

Allows you to change the channel settings for a channel. (See '**help chaninfo**' for the settings) These are used until the next rehash/chanload/restart, and are saved whenever the channel settings are saved.

See also: +chan, -chan chanload, chanset, chaninfo

chansave

Available to: {+n|+n}

Saves the channel settings to the file defined by the "**chanfile**" setting. This file is reloaded during rehash and restarting the bot. This allows you to change channel modes without having to edit the config file every time.

See also: +chan, -chan% chanload, chanset, chaninfo

chinfo <user> [channel] [info-line]

Available to: {+m|m}

Sets the information line for a user. This line is shown via /msg commands '**who**' and '**whois**', and if you have set greet on, it is shown when a user joins the channel. If the info line begins with a '@', then it is "locked", and that user may no longer change it. If the channel name is omitted, the default info line is changed.

See also: info

chinfo <user> [channel] none

Available to: {+m|m}

Erases a user's info line.

See also: info

info [channel] [info-line]

Available to: {-}

sets your info line. this is shown via /msg commands '**who**' and '**whois**', and possibly when you join the channel. if you omit the info line, it will just show you what your current info line is. if you omit the channel name, your default info line will be displayed/changed. (this is the info line shown for channels where you have no specific info line set.)

info [channel] none

Available to: {-}

removes your info line.

invites [[channel] wildcard]

invites all

Available to: {+o|o}

Shows you a list of the global invites active on the current channel, and the list of channel-specific invites, as well as any invites that are on the channel but weren't placed by the bot.

Here's a sample entry;

```
[ 5] *!*habib@*frys.com (perm)
```

paulie: revolving check policy

Created 15:10

The number (5) can be used to reference the invite if you wish to remove it (see '-invite'). Next is the actual hostmask being invited. The "(perm)" means that the invite is "permanent": that is, it doesn't automatically expire. If there is an elapsed time showing instead, the time displayed is how long the invite has been active. Those kind of invites expire after one hour. The second line of the invite entry is the comment ("revolving check policy" – it would seem that paulie had to stand in line for a while) and who put the invite there (paulie). The last line shows when the invite was added, and possibly the last time the invite was activated on the channel (if it's different from the creation time).

Sometimes there will be a "!" or "*" right before the number. a "!" means the invite is in the bot's perminvite list, but is not currently on the channel. a "*" marks a invite which is NOT in the perminvite list but IS on the channel.

As you use 'invites' it will show you only the invites which are currently in action on the channel. If you use 'invites all' it will show you every invite in memory (with 'perm-invites' on, these are identical).

If you use 'invites <wildcard>' it will list all the invites (active or not) that match against your wildcard. Consider it a 'invites all' list matched against your wildcard.

The invite list may change according to which channel you're currently viewing in the console. Different invites may be active on different channels. If you specify a channel name, that channel will be used instead of your current console channel.

See also: -invite, +invite, console, set invite-time, stick, unstick

exempts [[channel] wildcard]

exempts all

Available to: {+o|+o}

Shows you a list of the global ban exemptions active on the current channel, and the list of channel-specific ban exemptions, as well as any ban exemptions that are on the channel but weren't placed by the bot.

Here's a sample entry;

```
[ 5] !*habib@*frys.com (perm)
```

paulie: revolving check policy

Created 15:10

The number (5) can be used to reference the ban exemption if you wish to remove it (see '-exempt'). Next is the actual hostmask being exempted. The "(perm)" means that the ban exemption is "permanent": that is, it doesn't automatically expire. If there is an elapsed time showing instead, the time displayed is how long the ban exemption has been active. Those kind of ban exemptions expire after one hour. The second line of the ban exemption entry is the comment ("revolving check policy" -- it would seem that paulie had to stand in line for a while) and who put the ban exemption there (paulie). The last line shows when the ban exemption was added, and possibly the last time the ban exemption was activated on the channel (if it's different from the creation time).

Sometimes there will be a "!" or "*" right before the number. a "!" means the ban exemption is in the bot's permaban exemption list, but is not currently on the channel. a "*" marks a ban exemption which is NOT in the permaban exemption list but IS on the channel.

Af you use 'exempts' it will show you only the ban exemptions which are currently in action on the channel. If you use 'ban exemptions all' it will show you every ban exemption in memory (with 'perm-exempts' on,these are identical).

If you use 'exempts <wildcard>' it will list all the ban exemptions (active or not) that match against your wildcard. Consider it a 'exempts all' list matched against your wildcard.

The ban exemption list may change according to which channel you're currently viewing in the console. Different ban exemptions may be active on different channels. If you specify a channel name, that channel will be used instead of your current console channel.

See also: -exempt, +exempt, console, set exempt-time, stick, unstick

stick [ban/exempt/invite] <hostmask OR number>

Available to: {+o|o}

makes a ban/exempt/invite "sticky" -- meaning the bot will always try to keep it active on the channel, even if the channel is using dynamic bans. obviously if the channel isn't using dynamic bans, this has no effect.

See also: bans, exempts, invites, unstick, +ban, +exempt, +invite

unstick [ban/exempt/invite] <hostmask OR number>

Available to: {+o|o}

turns a "sticky" ban/exempt/invite normal again.

See also: bans, exempts, invites, stick, -ban, -exempt, -invite

Wire Module Help

This module provides a more secure, encrypted method of communicating over the botnet.

wire <encryption-key>

Available to: {-|}

This places you on a secure chat channel using the encryption key provided, only people who also know that key will be able to chat to you. All text starting with ';' will go over the wire. It works just like ''(local bot only) or ',' (local global masters only), except it goes over the botnet too.

wire off

Available to: {-|}

Removes you from the secure chat channel you were using .

wire info

Available to: {-|}

Tells you which secure chat channel you are currently on.

onwire

Available to: {-|}

Shows information about who is currently on the secure chat channel you are using from which bots. This command is similar to '**whom**' for normal channel channels.

wirelist

Available to: {+n|-}

Shows debugging information about who is using a wire at any moment, it doesn't however reveal what the keys are.

Seen Module Help

This module provides the classic '**seen**' command, which returns the last time a user was seen by the bot, both dcc & msg commands are provided.

seen <handle>

Available to: {-|}

Returns the last time <handle> was seen by the bot.

Console Module Help

The console module provides the ability to store your console settings so that next time you use the party line they are set. You can use the '**store**' command at any time to store it, or it may be configured to save automatically.

store

Available to: {-|}

Stores your console settings so that they are set automatically next time you join the party line.

See Also: set console-autosave

Filesys Module Help

files

Available to: {+x|-}

will move you into the file transfer sub-system, if it has been enabled on this bot. from there you can browse through the files online and use dcc file transfers to download and upload.

Commands in the filesystem

cancel <file(s)>

Available to: {+x|-}

tells the bot to stop sending a file that is pending (either queued, waiting, or in the process of being transferred).

See also: pending

cd <directory>

Available to: {+x|-}

changes your current directory if possible. this works exactly like the unix command.

See also: pwd

cp <source> <dest>

Available to: {+j|-}

copies a file or group of files from one place to another.

see also: mv

desc <file> <description>

Available to: {+x|-}

Changes the description for a file. If you are a master or file janitor, you can change the description for any file. otherwise you can only change the descriptions for files you have uploaded.

The description is restricted to 3 lines of 60 characters each, and is broken up between words. you can force a line break by putting a '|' in the comment.

filestats <user>

Available to: {+j|-}

Reports on the users upload & download statistics.

filestats <user> clear

Available to: {+j|-}

Clears a users upload & download statistics.

stats

Available to: {+j|-}

Reports your upload & download statistics.

get <filename(s)> [nickname]

Available to: {+x|-}

Sends you the file(s) requested, over IRC. you should get a DCC SEND notice on IRC, and have your client accept it. If your nickname on IRC is different than the one you use on the bot, you should specify a nickname too. you can also use that to send files to other people. if a file you want is actually on another bot, it may take a little bit longer so be patient. if that bot isn't on the botnet right now, it will say the file isn't available.

There is a limit to the number of transfers you can have going simultaneously, so if you go over this limit, the remainder of your file requests will be queued. as the first files finish transferring, the queued files will be sent.

See also: pending, cancel

hide <file(s)>

Available to: {+j-}

marks a file as hidden, so that normal users can't see it. only a master or file janitor using 'lsa' can see hidden files.

See also: unhide, lsa

In <bot:filepath> <localfile>

Available to: {+x|-}

Creates a link to a file on another bot. the filepath has to be complete, like '/gifs/uglyman.gif'. if the bot is not connected to the botnet, nobody will be able to download the file until that bot connects again. the local filename can be anything you want. example: In Snowbot:/gifs/uglyman.gif ugly.gif

See also: share, unshare, get

ls [filemask]

Available to: {+x|-}

Displays the files in the current directory. Subdirectories are shown with "<DIR>" next to them, and other files will display their size (typically in kilobytes), who uploaded them (and when), and how many times each file has been downloaded. if a description of the file exists, it is displayed below the filename. You can restrict the file listing by specifying a mask, just like in unix.

See also: cd, pwd, get

lsa [filemask]

Available to: {+x|-}

Works just like ls, but hidden files are displayed too. hidden files are displayed with a "(hid)" next to them, and shared files are displayed with a "(shr)" next to them.

See also: hide, share

mkdir <dir> [flags [channel]]

Available to: {+j|-}

Creates a subdirectory from this one, with the given name. If flags are specified, then those flags are required to enter or even see the directory. you can even specify a channel that the flags are matched against. you can use the '**mkdir**' command again to alter or remove those flags.

See also: rmdir

mv <source> <dest>

Available to: {+x|-}

Moves a file or group of files from one place to another. (it can also be used to rename files.)

See also: cp

pending

Available to: {+x|-}

Gives you a listing of every file you've requested which is still waiting, queued, or in the process of transferring. it shows you the nickname on IRC that the file is being sent to, and, if the transfer is in progress, tells you how far along the transfer is.

See also: cancel

pwd

Available to: {+x|-}

Tells you what your current directory is.

See also: cd

quit

Available to: {+x|-}

Exits the file system

rm <file(s)>

Available to: {+j|-}

Erase a file for good.

rmdir <dir>

Available to: {+j|-}

Removes an existing directory, if there are no files in it.

See also: mkdir

share <file(s)>

Available to: {+x|-}

Marks a file as shared. this means that other bots can get the file remotely for users on their file systems. by default, files are marked as unshared.

See also: unshare, lsa, ln

sort

Available to: {+j|-}

Sorts the files in the current directory into ascending alphabetical order.

unhide <file(s)>

Available to: {+j|-}

Makes a file be not hidden any more.

See also: hide, lsa

unshare <file(s)>

Available to: {+j|-}

Removes the shared tag from a file.

See also: share, lsa, ln

Share Module Help

This module provides userfile sharing between two bots, it transfers the userfile when they first connect, and then send updates whilst they are connected to keep the userfiles in sync.

how to setup sharing with the share module

First you need to decide 2 things:

- which bot will be 'active' and which 'passive', the passive bot's userfile WILL GET OVERWRITTEN so choose carefully. We shall call the ACTIVE bot A, the PASSIVE bot B
- which channels you want to share between the 2 bots. Only the channels you choose to share will have their Info shared, ALL global info will be shared.

Steps:

1. Do for each of the channels you want to share a chanset #channel +shared
 - on A type: '.botattr B +s' +s == actively share
 - on B type: '.botattr A +p' +p == passively share
2. for each channel you want shared between the two,
 - on A type: '.botattr B |+s #channel'
 - on B type: '.botattr A |+s #channel'
3. link them together, they should start sharing immediately

flush <bot>

Available to: {+n|-}

Clears the resync buffer for a share-bot. this is useful if you want to start over with a userfile transfer: you can unlink the sharebots, flush the resync buffer, and relink.

see also: link, chattr, unlink

Notes Module Help

This module provides the means for storing & retrieving notes at a later stage.

notes index

notes read <# or ALL>

notes erase <# or ALL>

Available to: {-}

Lets you manipulate notes that have been stored up for you while you were gone.

'**notes index**' gives a listing of all the notes stored up: who they are from, and when they were left. '**notes read**' lets you read some or all notes, according to a list of numbers and/or intervals separated by semicolon. And '**notes erase**' erases notes after you are done with them. ex: notes erase 2-4;8;16-

see also: note, whom, noteigns

fwd <handle> [desto]

Available to: {+m|-}

This allows you to set a note forwarding address for a user, this means if a note needs to be stored for the user, the bot will attempt to pass it on to the given user@bot, if the bot is not online then the note is still stored locally, if the other user doesn't exist, the note is lost. Boohoo.

+noteign <ignoremask>

Available to: {-}

Add a new ignore to your note ignore list.

See also: -noteign noteigns

+noteign <user> <ignoremask>

Available to: {+m|-}

Add a new ignore to the user's note ignore list.

See also: -noteign noteigns

-noteign <ignoremask>

Available to: {-}

Remove an existing ignore from your note ignore list.

See also: +noteign noteigns

-noteign <user> <ignoremask>

Available to: {+m|-}

Remove an existing ignore from the user's note ignore list.

See also: +noteign noteigns

noteigns

Available to: {-}

List all note ignores. All notes sent from users who match one of the ignore masks will be rejected.

See also: +noteign -noteign

noteigns <user>

Available to: {+m|-}

List all note ignores for user. All notes sent from users who match one of the ignore masks will be rejected.

See also: +noteign -noteign

note <nickname[@bot]> <message>

Available to: {-|}

sends a private note to a user on the party line. if that user is currently on the party line, and not marked as away, she will receive the message immediately. otherwise it may be stored and displayed the next time that user joins the party line. If you join the channel, and have notes stored for you on the bot, it will tell you. to send a note to someone on a different bot, use "nick@bot" for the nickname.

see also: whom, notes, noteigns

Scripts Help

Help for Scripts made by Nils Østbjerg <shorty@business.auc.dk>

Flagnote Script Help

Version: 1.4.0

This script lets you send notes to anyone with one or more specific flags using the new matching system (the same used with: .match)

flagnote <flags> [#channel/All] <message>

Available to: {+m|m}

Sends <message> to users with flag(s) requested <flags> is of the form: <+/-><global>[&/<channel>] specifying & as the speperator, will cause AND style matching, | will cause OR style matching, channel flags are match vs the specified channel, all channel or your console channel if non specified.

This is the same matching style as the '**match**' command uses.

A '%nick' in message to be replaced with destination handle.

IWeed Script Help

Version: 1.4.0

This script in a partyline version of the weed scripts that comes with the bot and it weeds out specified entries in the userlist.

weed [options]

Available to: {+n|-}

This weeds out users, bans and ignores from the userlist

Possible options:

-<nick>	exempt this user from weeding
^o ^m ^f ^v	exempt ops or masters or friends or voices
^co ^cm ^cf ^cv	exempt chanops or chanmasters or chanfriends or chanvoices
^p ^x	exempt party-line or file-area users
^i ^w	exempt users with info or password set
^t	exempt botnet masters

+<days>	weed: haven't been seen in N days
:n	weed: haven't EVER been seen
:o :m	weed: ops or masters with no password set
:co :cm	weed: chanops or chanmasters with no password set
:t	weed: botmasters with on password set
:a	weed: anyone with no pasword set
o m	unop/unmaster: ops or masters with no password
co cm	unop/unmaster: chanops or chanmasters with no password
t	unmaster: botmasters with no password
b<days>	weed: bans not used in N days
i<days>	weed: ignores created over N days ago
e<days>	weed: exempts not used in N days
v<days>	weed: invites not used in N days

Users with +n, +b or +W will not be affected by this commmand nor will sticky bans, exempts and invites.

BotRequest Script Help

Version: 1.3.0

This script will allow bots to request ops and other thing form each other. So far it only work fully on IRCnet type nets because of my lazyness and lack of knowledge of how the other nets works. On IRCnet types nets /invite overrides bans and limit, so there is not unban or lower limit routine

doneed <channel>

Available to: {+n|+n}

Set the <channel>'s setting so that the BotRequest script is used.

ChanBackup Script Help

Version: 1.3.0

This script will enable you to copy the setting of a channel into another channel this includes channel setting and userflags.

sync <from channel> <to channel> [options]

Available to: {+n|-}

Syncs setting, flags, infoline and bans from one channel to another.

Possible options:

- :s sync: channel settings
- :i sync: userinfo for channel
- :f sync: channel flags
- :b merge: channel bans

chanbackup <from channel> <to channel>

Available to: {+m|-}

This create a channel identical to <from channel> this include channel setting, flags, infolines and channel bans. <to channel> needs to be a channel not currently monitored by the bot.

Password Script Help

Version: 1.4.0

This script will stop users setting bad/easy to guess passwords. To do this it uses dictionary files which per default is placed in filearea/Password/ all files places here with the .dic extention will be used. The files should have one word per line.

checkpass [days]

Available to: {+n|-}

This check to see if users password is to a blacklisted password and if [days] are specified it will also check to see if users have updated their password with in the number of [days].If the password is either blacklisted or older then [days] the user will be set to +B, and you can use Flagnote to message all users with bad passwords.

The B flag will be removed next time the command is run if the password is valid at that time.

The lists of blacklisted passwords are located in the filearea under /Password/. Any files in that directorey with the extention .dic are part of the list. They should contain one word per line.

NetCommand Script Help

Version: 1.4.0

This script will allow masters to execute the same command on all sharebots or execute a command on a remote sharebot.

It will work on any command, SO BE CAREFULL IT IS A SECURITY RISK. You have been warned.

net <command>

Available to: {+m|-}

This will execute the same <command> on all sharebots where you got flags to execute it.

Remote <sharebot> <command>

Available to: {+m|-}

This will execute the <command> on <sharebot> if you got flags on that to execute the command.