

User commands

ADMIN

Syntax:

```
ADMIN [<target>]
```

Instructs the server to return information about the administrators of the server specified by <target>, where <target> is either a server or a user. If <target> is omitted, the server should return information about the administrators of the current server.^[1]

AWAY

Syntax:

```
AWAY [<message>]
```

Provides the server with a message to automatically send in reply to a PRIVMSG directed at the user, but not to a channel they are on.^[2] If <message> is omitted, the away status is removed. Defined in RFC 1459.

CNOTICE

Syntax:

```
CNOTICE <nickname> <channel> :<message>
```

Sends a channel NOTICE message to <nickname> on <channel> that bypasses flood protection limits. The target nickname must be in the same channel as the client issuing the command, and the client must be a channel operator.

Normally an IRC server will limit the number of different targets a client can send messages to within a certain time frame to prevent spammers or bots from mass-messaging users on the network, however this command can be used by channel operators to bypass that limit in their channel. For example, it is often used by help operators that may be communicating with a large number of users in a help channel at one time.

This command is not formally defined in an RFC, but is in use by some IRC networks. Support is indicated in a RPL_ISUPPORT reply (numeric 005) with the CNOTICE keyword

CPRIVMSG

Syntax:

```
CPRIVMSG <nickname> <channel> :<message>
```

Sends a private message to <nickname> on <channel> that bypasses flood protection limits. The target nickname must be in the same channel as the client issuing the command, and the client must be a channel operator.

Normally an IRC server will limit the number of different targets a client can send messages to within a certain time frame to prevent spammers or bots from mass-messaging users on the network, however this command can be used by channel operators to bypass that limit in their channel. For example, it is often used by help operators that may be communicating with a large number of users in a help channel at one time.

This command is not formally defined in an RFC, but is in use by some IRC networks. Support is indicated in a RPL_ISUPPORT reply (numeric 005) with the CPRIVMSG keyword

CONNECT

Syntax:

```
CONNECT <target server> [<port> [<remote server>]] (RFC 1459)
CONNECT <target server> <port> [<remote server>] (RFC 2812)
```

Instructs the server <remote server> (or the current server, if <remote server> is omitted) to connect to <target server> on port <port>.^{[3][4]} This command should only be available to IRC operators. Defined in RFC 1459; the <port> parameter became mandatory in RFC 2812.

DIE

Syntax:

```
DIE
```

Instructs the server to shut down.^[5] This command may only be issued by IRC server operators. Defined in RFC 2812.

ENCAP

Syntax:

```
:<source> ENCAP <destination> <subcommand> <parameters>
```

This command is for use by servers to encapsulate commands so that they will propagate across hub servers not yet updated to support them, and indicates the subcommand and its parameters should be passed unaltered to the destination, where it will be unencapsulated and parsed. This facilitates implementation of new features without a need to restart all servers before they are usable across the network.^[6]

ERROR

Syntax:

ERROR <error message>

This command is for use by servers to report errors to other servers. It is also used before terminating client connections.^[7] Defined in RFC 1459.

HELP

Syntax:

HELP

Requests the server to display the help file. This command is not formally defined in an RFC, but is in use by most major IRC daemons.

INFO

Syntax:

INFO [<target>]

Returns information about the <target> server, or the current server if <target> is omitted.^[8] Information returned includes the server's version, when it was compiled, the patch level, when it was started, and any other information which may be considered to be relevant. Defined in RFC 1459.

INVITE

Syntax:

INVITE <nickname> <channel>

Invites <nickname> to the channel <channel>.^[9] <channel> does not have to exist, but if it does, only members of the channel are allowed to invite other clients. If the channel mode **i** is set, only channel operators may invite other clients. Defined in RFC 1459.

ISON

Syntax:

ISON <nicknames>

Queries the server to see if the clients in the space-separated list <nicknames> are currently on the network.^[10] The server returns only the nicknames that are on the network in a space-separated list. If none of the clients are on the network the server returns an empty list. Defined in RFC 1459.

JOIN

Syntax:

```
JOIN <channels> [<keys>]
```

Makes the client join the channels in the comma-separated list <channels>, specifying the passwords, if needed, in the comma-separated list <keys>.^[11] If the channel(s) do not exist then they will be created. Defined in [RFC 1459](#).

KICK

Syntax:

```
KICK <channel> <client> [<message>]
```

Forcibly removes <client> from <channel>.^[12] This command may only be issued by channel operators. Defined in [RFC 1459](#).

KILL

Syntax:

```
KILL <client> <comment>
```

Forcibly removes <client> from the network.^[13] This command may only be issued by IRC operators. Defined in [RFC 1459](#).

KNOCK

Syntax:

```
KNOCK <channel> [<message>]
```

Sends a NOTICE to an invitation-only <channel> with an optional <message>, requesting an invite. This command is not formally defined by an RFC, but is supported by most major IRC daemons. Support is indicated in a RPL_ISUPPORT reply (numeric 005) with the KNOCK keyword.

LINKS

Syntax:

```
LINKS [<remote server> [<server mask>]]
```

Lists all server links matching <server mask>, if given, on <remote server>, or the current server if omitted.^[14] Defined in [RFC 1459](#).

LIST

Syntax:

```
LIST [<channels> [<server>]]
```

Lists all channels on the server.^[15] If the comma-separated list <channels> is given, it will return the channel topics. If <server> is given, the command will be forwarded to <server> for evaluation. Defined in [RFC 1459](#).

LUSERS

Syntax:

```
LUSERS [<mask> [<server>]]
```

Returns statistics about the size of the network.^[16] If called with no arguments, the statistics will reflect the entire network. If <mask> is given, it will return only statistics reflecting the masked subset of the network. If <target> is given, the command will be forwarded to <server> for evaluation. Defined in [RFC 2812](#).

MODE

Syntax:

```
MODE <nickname> <flags> (user)  
MODE <channel> <flags> [<args>]
```

The MODE command is dual-purpose. It can be used to set both user and channel modes.^[17] Defined in [RFC 1459](#).

MOTD

Syntax:

```
MOTD [<server>]
```

Returns the message of the day on <server> or the current server if it is omitted.^[18] Defined in [RFC 2812](#).

NAMES

Syntax:

```
NAMES [<channels>] (RFC 1459)  
NAMES [<channels> [<server>]] (RFC 2812)
```

Returns a list of who is on the comma-separated list of <channels>, by channel name.^[19] If <channels> is omitted, all users are shown, grouped by channel name with all users who are not on a channel being shown as part of channel "**".

If <server> is specified, the command is sent to <server> for evaluation.^[20] Defined in RFC 1459; the optional <server> parameter was added in RFC 2812.

The response contains all nicknames in the channel prefixed with the highest channel status prefix of that user, for example like this (with @ being the highest status prefix)

```
:irc.server.net 353 Phyre = #SomeChannel :@WiZ
```

If a client wants to receive all the channel status prefixes of a user and not only their current highest one, the IRCv3 multi-prefix extension can be enabled (@ is the channel operator prefix, and + the lower voice status prefix).^[21]

```
:irc.server.net 353 Phyre = #SomeChannel :@+WiZ
```

See also NAMESX below for an alternate, older approach to achieve the same effect. However, by today most clients and servers support the new IRCv3 standard.^[22]

NAMESX

Syntax:

```
PROTOCTL NAMESX
```

Instructs the server to send names in an RPL_NAMES reply prefixed with all their respective channel statuses instead of just the highest one (similar to IRCv3's multi-prefix).

For example:

With NAMESX

```
:irc.server.net 353 Phyre = #SomeChannel :@+WiZ
```

Without NAMESX

```
:irc.server.net 353 Phyre = #SomeChannel :@WiZ
```

This command can ONLY be used if the NAMESX keyword is returned in an RPL_ISUPPORT (numeric 005) reply. It may also be combined with the UHNAMES command.

This command is not formally defined in an RFC, but is recognized by most major IRC daemons. The newer modern approach is to use IRCv3 protocol extensions to activate the multi-prefix extension for the regular NAMES command.^[21]

NICK

Syntax:

NICK <nickname> [<hopcount>] (RFC 1459)

NICK <nickname> (RFC 2812)

Allows a client to change their IRC nickname. Hopcount is for use between servers to specify how far away a nickname is from its home server.^{[23][24]} Defined in RFC 1459; the optional <hopcount> parameter was removed in RFC 2812.

NOTICE

Syntax:

NOTICE <msgtarget> <message>

This command works similarly to PRIVMSG, except automatic replies must never be sent in reply to NOTICE messages.^[25] Defined in RFC 1459.

OPER

Syntax:

OPER <username> <password>

Authenticates a user as an IRC operator on that server/network.^[26] Defined in RFC 1459.

PART

Syntax:

PART <channels> [<message>]

Causes a user to leave the channels in the comma-separated list <channels>.^[27] Defined in RFC 1459.

PASS

Syntax:

PASS <password>

Sets a connection password.^[28] This command must be sent before the NICK/USER registration combination. Defined in RFC 1459.

PING

Syntax:

PING <server1> [<server2>]

Tests the presence of a connection.^[29] A PING message results in a PONG reply. If <server2> is specified, the message gets passed on to it. Defined in RFC 1459.

PONG

Syntax:

```
PONG <server1> [<server2>]
```

This command is a reply to the PING command and works in much the same way.^[30] Defined in RFC 1459.

PRIVMSG

Syntax:

```
PRIVMSG <msgtarget> <message>
```

Sends <message> to <msgtarget>, which is usually a user or channel.^[31] Defined in RFC 1459.

QUIT

Syntax:

```
QUIT [<message>]
```

Disconnects the user from the server.^[32] Defined in RFC 1459.

REHASH

Syntax:

```
REHASH
```

Causes the server to re-read and re-process its configuration file(s).^[33] This command can only be sent by IRC operators. Defined in RFC 1459.

RESTART

Syntax:

```
RESTART
```

Restarts a server.^[34] It may only be sent by IRC operator. Defined in RFC 1459.

RULES

Syntax:

```
RULES
```

Requests the server rules. This command is not formally defined in an RFC, but is used by most major IRC daemons.

SERVER

Syntax:

```
SERVER <servername> <hopcount> <info>
```

The server message is used to tell a server that the other end of a new connection is a server.^[35] This message is also used to pass server data over the whole network. <hopcount> details how many hops (server connections) away <servername> is. <info> contains addition human-readable information about the server.

Defined in [RFC 1459](#).

SERVICE

Syntax:

```
SERVICE <nickname> <reserved> <distribution> <type> <reserved> <info>
```

Registers a new service on the network.^[36] Defined in [RFC 2812](#).

SERVLIST

Syntax:

```
SERVLIST [<mask> [<type>]]
```

Lists the services currently on the network.^[37] Defined in [RFC 2812](#).

SQUERY

Syntax:

```
SQUERY <servicename> <text>
```

Identical to PRIVMSG except the recipient must be a service.^[38] Defined in [RFC 2812](#).

SQUIT

Syntax:

```
SQUIT <server> <comment>
```

Causes <server> to quit the network.^[39] Defined in RFC 1459.

SETNAME

Syntax:

```
SETNAME <new real name>
```

Allows a client to change the "real name" specified when registering a connection.

This command is not formally defined by an RFC, but is in use by some IRC daemons. Support is indicated in a RPL_ISUPPORT reply (numeric 005) with the SETNAME keyword

SILENCE

Syntax:

```
SILENCE [+/-<hostmask>]
```

Adds or removes a host mask to a server-side ignore list that prevents matching users from sending the client messages. More than one mask may be specified in a space-separated list, each item prefixed with a "+" or "-" to designate whether it is being added or removed. Sending the command with no parameters returns the entries in the client's ignore list.

This command is not formally defined in an RFC, but is supported by most major IRC daemons. Support is indicated in a RPL_ISUPPORT reply (numeric 005) with the SILENCE keyword and the maximum number of entries a client may have in its ignore list. For example:

```
:irc.server.net 005 WiZ WALLCHOPS WATCH=128 SILENCE=15 MODES=12 CHANTYPES=#
```

STATS

Syntax:

```
STATS <query> [<server>]
```

Returns statistics about the current server, or <server> if it's specified.^[40] Defined in RFC 1459.

SUMMON

Syntax:

SUMMON <user> [<server>] ([RFC 1459](#))

SUMMON <user> [<server> [<channel>]] ([RFC 2812](#))

Gives users who are on the same host as <server> a message asking them to join IRC.^{[41][42]} Defined in [RFC 1459](#); the optional <channel> parameter was added in [RFC 2812](#).

TIME

Syntax:

TIME [<server>]

Returns the local time on the current server, or <server> if specified.^[43] Defined in [RFC 1459](#).

TOPIC

Syntax:

TOPIC <channel> [<topic>]

Allows the client to query or set the channel topic on <channel>.^[44] If <topic> is given, it sets the channel topic to <topic>. If channel mode +t is set, only a channel operator may set the topic. Defined in [RFC 1459](#).

TRACE

Syntax:

TRACE [<target>]

Trace a path across the IRC network to a specific server or client, in a similar method to [traceroute](#).^[45] Defined in [RFC 1459](#).

UHNAMES

Syntax:

PROTOCTL UHNAMES

Instructs the server to send names in an RPL_NAMES reply in the long format:

With UHNAMES

```
:irc.server.net 353 Phyre = #SomeChannel :WiZ!user@somehost
```

Without UHNAMES

```
:irc.server.net 353 Phyre = #SomeChannel :WiZ
```

This command can ONLY be used if the UHNAMES keyword is returned in an RPL_ISUPPORT (numeric 005) reply. It may also be combined with the NAMESX command.

This command is not formally defined in an RFC, but is recognized by most major IRC daemons.

USER

Syntax:

```
USER <username> <hostname> <servername> <realname> (RFC 1459)  
USER <user> <mode> <unused> <realname> (RFC 2812)
```

This command is used at the beginning of a connection to specify the username, hostname, real name and initial user modes of the connecting client.^{[46][47]} <realname> may contain spaces, and thus must be prefixed with a colon. Defined in RFC 1459, modified in RFC 2812.

USERHOST

Syntax:

```
USERHOST <nickname> [<nickname> <nickname> ...]
```

Returns a list of information about the nicknames specified.^[48] Defined in RFC 1459.

USERIP

Syntax:

```
USERIP <nickname>
```

Requests the direct IP address of the user with the specified nickname. This command is often used to obtain the IP of an abusive user to more effectively perform a ban. It is unclear what, if any, privileges are required to execute this command on a server.

This command is not formally defined by an RFC, but is in use by some IRC daemons. Support is indicated in a RPL_ISUPPORT reply (numeric 005) with the USERIP keyword.

USERS

Syntax:

```
USERS [<server>]
```

Returns a list of users and information about those users in a format similar to the UNIX commands who, rusers and finger.^[49] Defined in RFC 1459.

VERSION

Syntax:

```
VERSION [<server>]
```

Returns the version of <server>, or the current server if omitted.^[50] Defined in RFC 1459.

WALLOPS

Syntax:

```
WALLOPS <message>
```

Sends <message> to all operators connected to the server (RFC 1459), or all users with user mode 'w' set (RFC 2812).^{[51][52]} Defined in RFC 1459.

WATCH

Syntax:

```
WATCH [+/-<nicknames>]
```

Adds or removes a user to a client's server-side friends list. More than one nickname may be specified in a space-separated list, each item prefixed with a "+" or "-" to designate whether it is being added or removed. Sending the command with no parameters returns the entries in the client's friends list.

This command is not formally defined in an RFC, but is supported by most major IRC daemons. Support is indicated in a RPL_ISUPPORT reply (numeric 005) with the WATCH keyword and the maximum number of entries a client may have in its friends list. For example:

```
:irc.server.net 005 WiZ WALLCHOPS WATCH=128 SILENCE=15 MODES=12 CHANTYPES=#
```

WHO

Syntax:

```
WHO [<name> ["o"]]
```

Returns a list of users who match <name>.^[53] If the flag "o" is given, the server will only return information about IRC operators. Defined in RFC 1459.

WHOIS

Syntax:

```
WHOIS [<server>] <nicknames>
```

Returns information about the comma-separated list of nicknames masks <nicknames>.^[54] If <server> is given, the command is forwarded to it for processing. Defined in [RFC 1459](#).

WHOWAS

Syntax:

```
WHOWAS <nickname> [<count> [<server>]]
```

Used to return information about a nickname that is no longer in use (due to client disconnection, or nickname changes).^[55] If given, the server will return information from the last <count> times the nickname has been used. If <server> is given, the command is forwarded to it for processing. In [RFC 2812](#), <nickname> can be a comma-separated list of nicknames.^[56]

Defined in [RFC 1459](#).

See also

- [IRCd](#)
- [IRCX](#)
- [Server](#)

References

1. "Admin command" (<https://tools.ietf.org/html/rfc1459#section-4.3.7>). p. 31. sec. 4.3.7. doi:10.17487/RFC1459 (<http://dx.doi.org/10.17487%2FRFC1459>). [RFC 1459](#). <https://tools.ietf.org/html/rfc1459#section-4.3.7>.
2. "Away" (<https://tools.ietf.org/html/rfc1459#section-5.1>). pp. 38 – 39. sec. 5.1. doi:10.17487/RFC1459 (<http://dx.doi.org/10.17487%2FRFC1459>). [RFC 1459](#). <https://tools.ietf.org/html/rfc1459#section-5.1>.
3. "Connect message" (<https://tools.ietf.org/html/rfc1459#section-4.3.5>). pp. 29 – 30. sec. 4.3.5. doi:10.17487/RFC1459 (<http://dx.doi.org/10.17487%2FRFC1459>). [RFC 1459](#). <https://tools.ietf.org/html/rfc1459#section-4.3.5>.
4. "Connect message" (<https://tools.ietf.org/html/rfc2812#section-3.4.7>). pp. 28 – 29. sec. 3.4.7. doi:10.17487/RFC2812 (<http://dx.doi.org/10.17487%2FRFC2812>). [RFC 2812](#). <https://tools.ietf.org/html/rfc2812#section-3.4.7>.