

NAME

irccd-ipc - irccd network protocol

SYNOPSIS

HOOK-ADD *name path*
HOOK-LIST
HOOK-REMOVE *name*
PLUGIN-CONFIG *name [variable [value]]*
PLUGIN-INFO *name*
PLUGIN-LOAD *name*
PLUGIN-PATH *name [variable [value]]*
PLUGIN-LIST
PLUGIN-RELOAD *name*
PLUGIN-TEMPLATE *name [variable [value]]*
PLUGIN-UNLOAD *name*
RULE-ADD *accept/drop [ceiops=value]*
RULE-EDIT *index [ceops(+|-)value] [a=accept|drop]*
RULE-LIST
RULE-MOVE *from to*
RULE-REMOVE *index*
SERVER-DISCONNECT [*name*]
SERVER-MESSAGE *name channel message*
SERVER-ME *name channel message*
SERVER-MODE *name channel mode [args]*
SERVER-NOTICE *name channel notice*
SERVER-INFO *name*
SERVER-INVITE *name channel target*
SERVER-JOIN *name channel [password]*
SERVER-KICK *name channel target [reason]*
SERVER-LIST
SERVER-PART *name channel [reason]*
SERVER-TOPIC *name channel topic*
WATCH

DESCRIPTION

This guide will help you controlling irccd via sockets.

For this to work, you need to define a transport in your *irccd.conf* file, for more information see *irccd.conf(5)*.

Most of the commands are similar to **ircdctl** and mapped to the same network command. For more detailed information about each command, take a look at `ircdctl(1)` before.

SYNTAX

The `ircd` program uses plain text for networking messages that consist of a command name, its arguments and ends with a CRLF delimiter.

Example of command:

```
SERVER-MESSAGE malikania #test Hello world!
```

RESPONSES

All command emit a response even though if they don't produce output. If a command succeeds, a plain "OK" is returned, otherwise "ERROR" followed by a message is sent instead.

Example with a successful message:

```
OK
```

Example of error:

```
ERROR server not found
```

Some command may produce more output that are printed under several lines for an easier interpretation client side. In that case commands usually return "OK" followed by the number of lines provided.

Example with *PLUGIN-CONFIG*:

```
OK 2
collaborative=true
level=hard
```

COMMANDS

The following commands are available.

HOOK-ADD Add the hook *name* at the given *path*.

HOOK-LIST Returns the list of hooks by their names separated by a space immediately after "OK" status.

Example:

OK irc-notify mail-notify

HOOK-REMOVE

Removes the hook specified by *name*.

PLUGIN-CONFIG

Set or get *variable* and its *value* if specified for the plugin *name*. Returns the list of configuration variables if there were only the plugin *name* or return only the value if *variable* was specified.

Example if *variable* was specified:

OK 1
current value

Example if *variable* was not specified:

OK 2
collaborative=true
level=hard

PLUGIN-INFO

Return the information about the plugin *name* in the form:

OK 5
name
long description
version
license
author

PLUGIN-LOAD

Find and load the plugin specified by *name*.

PLUGIN-PATH

Exactly the same usage as **PLUGIN-CONFIG** but for the plugin paths.

PLUGIN-LIST

Returns the list of plugins by their names separated by a space immediately after "OK" status.

Example:

OK ask auth hangman

- PLUGIN-RELOAD** Reload the plugin specified by *name*.
- PLUGIN-TEMPLATE** Exactly the same usage as **PLUGIN-CONFIG** but for the plugin paths.
- PLUGIN-UNLOAD** Unload and remove the plugin *name*.
- RULE-ADD** Add a new rule with matching action to *accept* or *drop*. Then by a list separated by spaces, add any key=value pair where the key defines the criterion to set from "ceops" which adds a channel, event, origin, plugin or server respectively. The *i* character is the optional index where to insert the new rule.

Example of client request:

```
RULE-ADD accept c=#test s=example i=1
```

- RULE-EDIT** Edit the rule at the given *index*. Similarly to **RULE-ADD** command, you can specify a criterion modification using the "ceops" characters followed by a minus or plus string to either remove or add the criterion respectively. It is also possible to change the rule action with the key value pair *a=accept* Or *a=drop*

Example of client request:

```
RULE-EDIT c-#staff s+example a=drop
```

- RULE-LIST** Return the list of rules loaded in the form:

```
OK 1
accept
list of servers
list of channels
list of origins
list of plugins
list of events
```

For each rule, the block of 6 lines is repeated for every rule returned in the "OK" response. So if you have 4 rules, you'll have to read 24 lines total.

- RULE-MOVE** Move the rule at the position *from* to the position specified by *to*.

RULE-REMOVE	Remove the rule at the given <i>index</i> .
SERVER-DISCONNECT	Disconnect and remove the server specified by <i>name</i> if specified otherwise, remove all.
SERVER-MESSAGE	Send the private <i>message</i> to the <i>channel</i> into the server <i>name</i> .
SERVER-ME	Send an action emote <i>message</i> to the <i>channel</i> into the server <i>name</i> .
SERVER-MODE	Change <i>channel</i> to the new <i>mode</i> with optional list of mode arguments specified by <i>args</i> .
SERVER-NOTICE	Send the <i>notice</i> to the <i>channel</i> into the server <i>name</i> .
SERVER-INFO	Get the information about the server <i>name</i> in the form: OK name hostname port [ssl] nickname username real name #channels #channels...
SERVER-INVITE	Invite the <i>target</i> to the <i>channel</i> into the server <i>name</i> .
SERVER-JOIN	Join a <i>channel</i> into the server <i>name</i> using an optional <i>password</i> .
SERVER-KICK	Kick the <i>target</i> from the <i>channel</i> in the server <i>name</i> using an optional <i>reason</i> .
SERVER-LIST	Returns the list of servers by their names separated by a space immediately after "OK" status. Example: OK caramail wanadoo
SERVER-PART	Leaves the <i>channel</i> from the server <i>name</i> using an optional <i>reason</i> .
SERVER-TOPIC	Sets the new <i>topic</i> to the <i>channel</i> into the server <i>name</i> .
WATCH	Enable watch mode.

When set, irccd will notify the client about new IRC event incoming using the syntax:

EVENT-<NAME> arguments...

Example:

EVENT-CONNECT wanadoo

EVENT-MESSAGE wanadoo jean!jean@caramail.com #games hello guys!

SEE ALSO

irccd(1), irccdctl(1)

AUTHORS

The **irccd** daemon was written by David Demelier <*markand@malikania.fr*>.