NAME

libirccd-rule - create and match rules

SYNOPSIS

#include <irccd/rule.h>

*struct irc_rule **
irc_rule_new(*enum irc_rule_action action*);

int
irc_rule_add(*char *list, const char *value*);

void
irc_rule_remove(*char *list, const char *value*);

int
irc_rule_match(*const struct irc_rule *rule, const char *server, const char *channel, const char *origin, const char *plugin, const char *event*);

int
irc_rule_matchlist(*const struct irc_rule_list *list, const char *server, const char *channel, const char *origin, const char *plugin, const char *event*);

void
irc_rule_finish(*struct irc_rule *rule*);

DESCRIPTION

The function in this header provides rule matching for filtering plugins depending on IRC events.

Rules are defined in the *struct irc_rule* declared as following:

```
struct irc_rule {
        enum irc_rule_action action;
        char servers[IRC_RULE_LEN];
        char channels[IRC_RULE_LEN];
        char origins[IRC_RULE_LEN];
        char plugins[IRC_RULE_LEN];
        char events[IRC_RULE_LEN];
        struct irc_rule *next;
        struct irc_rule *prev;
```

};

The fields of *struct irc_rule* are:

action One of the *enum irc_rule_action* enumeration.

servers A colon separated list of servers identifiers to match.

channels A colon separated list of channels to match.

origins A colon separated list of origins to match.

plugins A colon separated list of plugins to match.

events A colon separated list of events to match.

next Pointer to the next rule.

perv Pointer to the previous rule.

The *enum irc_rule_action* is declared as:

```
enum irc_rule_action {
        IRC_RULE_ACCEPT,
        IRC_RULE_DROP
};
```

The following enumerators are available:

IRC_RULE_ACCEPT Allows the current event.

IRC_RULE_DROP Drop the current event.

The **irc_rule_new**() allocates a new rule with the given *action* and return it.

The **irc_rule_add**() function adds the new *value* to the char array *list* which should be one of the member field from the *struct irc_rule*.

The **irc_rule_remove**() removes the existing *value* from the char array *list*.

The **irc_rule_match**() function tests if the criteria given as arguments is allowed for this *rule*. All of *server*, *channel*, *origin*, *plugin*, and *event* can be NULL, in that case the rule is considered as not matching only if the rule does not contain a criterion for one of each. For example, if the rule must match a server "example" and argument *server* is NULL, then the rule will not match. Otherwise, if the rule does not have a server criterion then argument *server* is ignored entirely and this specific server criterion matches.

The **irc_rule_matchlist**() function is similar to **irc_rule_match**() except that it analyze the whole linked *list* instead.

The **irc_rule_finish**() clears resources allocated for the *rule*. Make sure to remove it from the linked list where it is attached to before calling this function.

RETURN VALUES

The function **irc_rule_add**() returns 0 on success and -1 on errors. In that case *errno* is set to indicate the error.

The functions **irc_rule_match**() and **irc_rule_matchlist**() returns non-zero if the rule is allowed.

EXAMPLES

Create a rule that matches servers "example" or "wanadoo" on channel "#staff" for the plugin "hangman" and drop it.

```
struct irc_rule *r;

r = irc_rule_new(IRC_RULE_DROP);
irc_rule_add(r->servers, "example");
irc_rule_add(r->servers, "wanadoo");
irc_rule_add(r->channels, "#staff");
irc_rule_add(r->plugins, "hangman");
```

ERRORS

The function **irc_rule_add**() may set one of the following error:

[ENOMEM] When the limit of a rule criterion has been reached, which is IRC_RULE_LEN.

SEE ALSO

libirccd(3)

AUTHORS

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