NAME

irccd - irccd configuration file

DESCRIPTION

Both **irccd** and **irccdctl** use configuration file in a extended INI format. This section will briefly explain the additional extensions to the INI format.

The file syntax has following rules:

1. Each option is stored in a section,

2. Some sections may be redefined multiple times,

3. Empty option must have quotes (e.g. option = "").

The @include and @tryinclude statements

Irccd adds an extension to this format by adding an *@include keyword* which let you splitting your configuration file.

Note: this @include statement must be at the beginning of the file and must be surrounded by quotes if the file name has spaces.

You can use both relative or absolute paths. If relative paths are used, they are relative to the current file being parsed.

The alternative *@tryinclude* keyword is similar but does not fails if the requested file is not found.

Example of includes:

```
@include "rules.conf"
@include "servers.conf"

[mysection]
myoption = "1"
```

The list construct

When requested, an option can have multiples values in a list. The syntax uses parentheses and values are separated by commas.

If the list have only one value, you can just use a simple string.

Example of lists:

```
[rule]
servers = ( "server1", "server2" )

[rule]
servers = "only-one-server"
```

Note: spaces are completely optional.

Identifiers

Some sections require an identifier (specified as id) as parameter. They must be unique, not empty and can only contain characters, numbers, ’-’ and ’_’.

Example: both *abc* and *server-tz2* are valid.

CONFIGURATION SECTIONS

Configuration are always stored in dedicated section in the form *[name]* where name is one of the section defined below and are case-sensitive.

logs

This section can let you configure how irccd should log the messages.

Available options:

verbose (bool) Be verbose (Optional, default: false).

type (string) Which kind of logging, valid values are "console", "file" or "syslog" (Optional, default: console).

Note: syslog is not available on all platforms.

The following options are available for the *file* type:

path-logs (string) Path to the normal messages.

path-errors (string) Path to the error messages.

templates

The templates section let you change the irccd’s output. It uses the templates system (see irccd-templates(7) for more information about templates)

Only one keyword is defined, message which contains the message that irccd wants to output.

Note: colors and attributes are not supported on Windows.

Available options:

debug (string) Template to use to format debug messages (Optional, default: none).

info (string) Template to use to format information messages (Optional, default: none).

warning (string) Template to use to format warnings (Optional, default: none).

server

This section is used to connect to one or more server. Create a new server section for each IRC server you want to connect to.

Available options:

name (identifier) The unique id.

hostname (string) The server address or IP.

port (int) The server port (Optional, default: 6667).

password (string) An optional server password (Optional, default: none).

join-invite (bool) Automatically join channels upon invitation (Optional, default: false).

channels (list) List of channels to auto join (Optional, default: empty).

Note: if a channel requires a password, add it after a colon (e.g. "#channel:password").

command-char (string) The prefix for invoking special commands (Optional, default: !).

ssl (bool) Enable or disable SSL (Optional, default: false).

auto-reconnect (bool) Enable reconnection after failure (Optional, default: true).

auto-reconnect-delay (int) Number of seconds to wait before retrying (Optional, default: 30).

ping-timeout (int) Number of seconds before ping timeout (Optional, default: 300).

nickname (string) The nickname (Optional, default: irccd).

realname (string) The realname (Optional, default: IRC Client Daemon).

username (string) The username name (Optional, default: irccd).

ctcp-version (string) What version to respond to CTCP VERSION (Optional, default: IRC Client Daemon).

paths

The paths section defines common paths used as defaults for all plugins.

Any option in this section can be defined altough the following are used as common convention used in all plugins:

Available options:

cache (string) Path for data files written by the plugin.

data (string) Path for data files provided by the user.

config (string) Path for additional configuration from the user.

For each of these paths, "plugin/name" is appended with the appropriate plugin name when used.

The section is redefinable per plugin basis using the *[paths.<plugin>]* syntax.

plugins

This section is used to load plugins.

Just add any key you like to load a plugin. If the value is not specified, the plugin is searched through the standard directories, otherwise, provide the full path (including the .js extension).

Warning: remember to add an empty string for searching plugins.

transport

This section defines transports that are used to communicate through clients connected to irccd and to perform requests to irccd.

With transports, you may ask irccd to send a message, a notice or even to kick someone from a channel. Irccd will also notify all clients connected to this transport on IRC events.

There are two types of transport availables:

- Internet sockets, IPv4 and IPv6,

- Unix sockets, based on files (not available on Windows).

If SSL support was built in, both internet and unix sockets can be set to use encrypted connections.

Available options:

type (string) Type of transport "ip" or "unix".

password (string) An authentication password (Optional, default: none).

ssl (bool) Enable SSL (Optional, default: false),

key (string) Path to private key file (Required if ssl is true)

certificate (string) Path to certificate (Required if ssl is true)

The following options are available for the *ip* type:

port (int) Port number.

address (string) Address to bind or "*" for any (Optional, default: "*").

ipv4 (bool) Bind on IPv4 (Optional, default true).

ipv6 (bool) Bind on IPv6 (Optional, default true).

The following options are available for the *unix* type:

path (string) The file path to the socket.

rule

The rule section is one of the most powerful within irccd configuration. It lets you enable or disable plugins and IRC events for specific criterias. For instance, you may want to disable a plugin only for a specific channel on a specific server. And because rules are evaluated in the order they are defined, you can override rules.

Available options:

servers (list) A list of servers that will match the rule (Optional, default: empty).

channels (list) A list of channel (Optional, default: empty).

origins (list) A list of nicknames to check (Optional, default: empty).

plugins (list) Which plugins (Optional, default: empty),

events (list) Which events like onCommand, onMessage (Optional, default: empty).

action (string) Set to "accept" or "drop".

Warning: don’t make sensitive rules on origins option, irccd does not have any kind of nickname authentication. Thus, it may be very easy for someone to use a temporary nickname.

EXAMPLES

Full example of configuration file

```
# Add a transport that bind only to IPv6.
[transport]
type = ip
ipv4 = false
ipv6 = true
port = 12000

# A transport that binds to both IPv4 and IPv6.
[transport]
type = ip
port = 15000

# A server.
[server]
```

```
name = "foo"
host = "irc.foo.org"
port = "6667"
nickname = "superbot"
realname = "SuperBot v1.0"
username = "sp"

# An other server.
[server]
name = "wanadoo"
host = "chat.wanadoo.fr"
port = "6667"

# Load some plugins.
[plugins]
ask = ""                          # Search ask
myplugin = /path/to/myplugin.js   # Use absolute path

# This first rule disable the plugin reboot on all servers and channels.
[rule]
plugins = "reboot"
action = drop

# This rule enable the reboot plugin again on the server localhost,
# channel #staff.
[rule]
servers = "localhost"
channels = "#staff"
plugins = "reboot"
action = accept
```

SEE ALSO

irccd(1)

AUTHORS

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