NAME

irccd - IRC Client Daemon

SYNOPSIS

irccd [**-c** *file*] [**-v**]
irccd *info*
irccd *version*

DESCRIPTION

The **irccd** program is an IRC bot which connects to one or more severs and dispatches events to plugins and connected clients.

The following options are available:

-c *file*	specify the configuration file.
-v	be verbose.

When ran without arguments, **irccd** will read your configuration file and dispatch IRC events to the plugins and connected clients indefinitely.

Otherwise, the following commands are available:

info	Show in a scriptable manner the options that were selected during irccd’s build.
version	Get the irccd version. if Mercurial was available when building irccd then the current revision is bundled in the output.

PLUGINS

The **irccd** program can runs plugins once IRC events are received. For example, if someone sends you a private message plugins will be invoked with that event. Both native plugins written in C++ and Javascript are supported (if enabled at compile time).

The following IRC events are supported:

onCommand	This is a special event that does not exist in IRC context. It calls the plugin special invocation command using *!name arguments*... syntax. The exclamation mark is configured by default to be the command character prefix, then if the adjacent name is known to be a loaded plugin it is invoked with the additional arguments.

Example: to call the **hangman** plugin, one may use !hangman to start a game.

See also irccd.conf(5) on how to change the command character prefix under a *[server]* section.

onConnect	When a server successfully connects to an IRC server.
onDisconnect	When a server disconnected from an IRC server both in case of failures or explicit user disconnection.
onInvite	Event called when the bot itself has been invited to a channel.
onJoin	When someone joins a channel.
onKick	When someone has been kicked from a channel, irccd may be included.
onLoad	This is a special event that does not exist in IRC context. It is invoked when the plugin is initialized.
onMessage	Upon private message.
onMe	On action emote, also most known as */me* command.
onMode	When a user or channel mode change.
onNames	When a list of nicknames has been received.
onNick	On nick change, irccd may be included.
onNotice	On private notice.
onPart	When someone leaves a channel.
onReload	This is a special event that does not exist in IRC context. It is invoked when the user asks to reload a plugin.
onTopic	When a channel topic has been changed.
onUnload	This is a special event that does not exist in IRC context. It is invoked when the user asks to unload a plugin and before exiting.

onWhois	When a whois information has been received.

The following plugins are provided with irccd:

- ask
- auth
- hangman
- history
- joke
- links
- logger
- plugin
- roulette
- tictactoe

See additional documentation in their own manual page in the form irccd-plugin-name(7) where name is the actual plugin name.

TRANSPORTS

The daemon can be controlled at runtime using the dedicated **irccdctl** tool or using sockets.

Both TCP/IP and UNIX sockets are supported and SSL layer may be enabled over it for a secure connection. If authentication is desired, it can be enabled too.

See also the *[transport]* section in the irccd.conf(5) manual page.

RULES

irccd supports a feature called rules which allows you to define a fine-grained set of rules allowed for specific plugins. For instance, you may want to disable some IRC events for some plugins depending on your set of parameters. This is useful for plugins that generates huge traffic.

Rule events are matched using the same name as plugin events described in the section above. For example, to disable a private message event you must use the *onCommand* value.

See also the *[rule]* section in the irccd.conf(5) manual page.

DIRECTORIES

irccd uses different types of paths depending on the context.

Paths prefixed by (W) means they are only used on Windows, others prefixed by (U) means they are used on UNIX systems.

Configuration

The following directories are searched in the specified order for configuration files. For example, the files *irccd.conf* and *irccdctl.conf* will be searched there.

- (W) %APPDATA%/irccd/config
- (U) ${XDG_CONFIG_HOME}/irccd
- (U) ${HOME}/.config/irccd (if XDG_CONFIG_HOME is not set)

SEE ALSO

irccd-api(7), irccd-templates(7), irccd-ipc(7), irccd-plugin-ask(7), irccd-plugin-auth(7), irccd-plugin-hangman(7), irccd-plugin-history(7), irccd-plugin-joke(7), irccd-plugin-links(7), irccd-plugin-logger(7), irccd-plugin-plugin(7), irccd-plugin-roulette(7), irccd-plugin-tictactoe(7), irccd-test(1), irccd.conf(5), irccdctl(1), irccdctl.conf(5)