QIRC

A Class Object for PyQt5

Version 0.012

Summary

The *QIRC* class provides a multi-threaded Internet relay chat (IRC) client for use with PyQt programs.

To use *QIRC*, understanding IRC and the IRC protocol is a necessity. *QIRC* is designed to be low level, meaning its interface is influenced by the protocol itself; *QIRC* can be hard to understand if you don't understand the underlying protocol.

The IRC protocol is defined in a series of RFC documents:

- RFC 1459
- RFC 2812

Methods

<table>
<tr><td></td><td>QIRC(**kwargs)
Any of the keywords available with the configure() method can be used here.</td></tr>
<tr><td>None</td><td>configure(**kwargs)
Configures the IRC client.

<table>
<tr><th>Keyword</th><th>Type</th><th>Default</th><th>Description</th></tr>
<tr><td>server</td><td>String</td><td>""</td><td>The server the IRC client will connect to.</td></tr>
<tr><td>port</td><td>Integer</td><td>6667</td><td>The port the IRC client will connect to.</td></tr>
<tr><td>password</td><td>String</td><td>None</td><td>The password needed to connect to the IRC server (if required).</td></tr>
<tr><td>encoding</td><td>String</td><td>"utf-8"</td><td>What string encoding to use with the server.</td></tr>
<tr><td>ssl</td><td>Boolean</td><td>False</td><td>Set to True to use SSL/TLS to connect to the IRC server.</td></tr>
<tr><td>verify_hostname</td><td>Boolean</td><td>False</td><td>Set to True to verify the IRC server's host name (requires ssl to be set to True).</td></tr>
<tr><td>verify_certificate</td><td>Boolean</td><td>False</td><td>Set to True to verify the IRC server's certificate (requires ssl to be set to True).</td></tr>
<tr><td>nickname</td><td>String</td><td>"qircclient"</td><td>The nickname the IRC client will use.</td></tr>
<tr><td>alternate</td><td>String</td><td>"qirc_client"</td><td>The nickname the IRC client will try to use if the first choice is already taken.</td></tr>
<tr><td>username</td><td>String</td><td>"qircclient"</td><td>The username the IRC client will use.</td></tr>
<tr><td>realname</td><td>String</td><td>"qircclient"</td><td>The "real name" the IRC client will use.</td></tr>
<tr><td>flood_protection</td><td>Boolean</td><td>True</td><td>Set to False to turn off flood protection.</td></tr>
<tr><td>flood_protection_send_rate</td><td>Float</td><td>1.5</td><td>If flood_protection is set to True, messages will be sent to the server at a rate of one message every X seconds, with X equal to the value set here.</td></tr>
</table>
</td></tr>
</table>

None	start() Connects the IRC client to the IRC server.
None	stop() Disconnects the IRC client from the IRC server and terminates the client's thread.
None	privmsg(String *target*, String *message*) Sends a PRIVMSG to the currently connected IRC server.
None	join(String *channel*, String *key*=None) Sends a JOIN command to the currently connected IRC server.
None	part(String *channel*, String *message*=None) Sends a PART command to the currently connected IRC server.
None	quit(String *message*=None) Sends a QUIT command to the currently connected IRC server, disconnects from the IRC server, and kills the thread that the QIRC instance is using.
None	send(String *message*) Sends a raw message to the currently connected IRC server. This method allows users of this class to send commands that are not otherwise supported. The data is sent to the server unchanged, other than being properly encoded and delimited (as documented in RFC 1459).

Signals

None	**connected(Dictionary *data*)** Emitted when the IRC client connects to the IRC server. The *data* dictionary contains three keys: • **client** – The QIRC instance that emitted the signal • **server** – String: The IP or host name of the server connected to • **port** – Integer: The port on the server connected to
None	**registered(Dictionary *data*)** Emitted when the IRC client successfully registers with the IRC server. The *data* dictionary contains three keys: • **client** – The QIRC instance that emitted the signal • **server** – String: The IP or host name of the server connected to • **port** – Integer: The port on the server connected to
None	**nick_collision(Dictionary *data*)** Emitted when the nickname the IRC client tries to register with is already taken. The IRC client will try nicknames until it finds one that is not taken (these can be set with the **configure()** function). The *data* dictionary contains three keys: • **client** – The QIRC instance that emitted the signal • **old** – The invalid nickname • **new** – The new nickname
None	**message(Dictionary *data*)** Emitted when the IRC client receives a PRIVMSG from the server. The *data* dictionary contains five keys: • **client** – The QIRC instance that emitted the signal • **nickname** – The nickname of the message sender • **host** – The username and host of the message sender • **target** – The channel or user the message was sent to • **message** – The message
None	**public(Dictionary *data*)** Emitted when the IRC client receives a PRIVMSG sent to a channel the client is in. The *data* dictionary contains five keys: • **client** – The QIRC instance that emitted the signal • **nickname** – The nickname of the message sender • **host** – The username and host of the message sender • **target** – The channel message was sent to • **message** – The message
None	**private(Dictionary *data*)** Emitted when the IRC client receives a PRIVMSG sent directly to the client. The *data* dictionary contains five keys: • **client** – The QIRC instance that emitted the signal • **nickname** – The nickname of the message sender • **host** – The username and host of the message sender • **target** – The IRC client's nickname • **message** – The message

None	**action(Dictionary *data*)** Emitted when the IRC client receives a CTCP action PRIVMSG. The *data* dictionary contains five keys: • **client** – The QIRC instance that emitted the signal • **nickname** – The nickname of the message sender • **host** – The username and host of the message sender • **target** – The channel or user the message was sent to • **message** – The message
None	**ping()** Emitted when the IRC client receives a PING from the server. The IRC client automatically responds to the server; this just is a notification that a PING has been received.
None	**tick(Integer *uptime*)** Emitted once a second while the IRC client is connected to the server. The *uptime* integer contains the length of time the IRC client has been connected, in seconds.
None	**user_part(Dictionary *data*)** Emitted when the IRC client receives a PART message from the server. The *data* dictionary contains five keys: • **client** – The QIRC instance that emitted the signal • **nickname** – The nickname of the user leaving the channel • **host** – The username and host of the user • **channel** – The channel the user is leaving • **reason** – The reason the user left (or an empty string)
None	**user_join(Dictionary *data*)** Emitted when the IRC client receives a JOIN message from the server. The *data* dictionary contains four keys: • **client** – The QIRC instance that emitted the signal • **nickname** – The nickname of the user joining the channel • **host** – The username and host of the user • **channel** – The channel the user is joining
None	**user_quit(Dictionary *data*)** Emitted when the IRC client receives a QUIT message from the server. The *data* dictionary contains four keys: • **client** – The QIRC instance that emitted the signal • **nickname** – The nickname of the user quitting IRC • **host** – The username and host of the user • **reason** – The reason the user quit (or an empty string)
None	**nick_change(Dictionary *data*)** Emitted when the IRC client receives a NICK message from the server. The *data* dictionary contains four keys: • **client** – The QIRC instance that emitted the signal • **nickname** – The old nickname of the user changing their nickname • **host** – The username and host of the user • **new** – The user's new nickname

None	**user_list (Dictionary *users*)** Emitted when the IRC client receives a NAMES message from the server. The *users* dictionary contains five keys: • **client** – The QIRC instance that emitted the signal • **channel** – The name of the channel the user list belongs to • **users** – A Python list containing all users in the channel. Each entry in the list can either contain a nickname with status prefixes, or a user's nickname (with status prefixes), username, and host in **nickname! username@host** format.

Attributes

String	**server**	The server the IRC client is connected to.
Integer	**port**	The port the IRC client is connected to.
String	**password**	The password used to connect to the IRC server.
String	**nickname**	The nickname the IRC client is currently using.
String	**alternate**	The alternate nickname the IRC client was set to use.
String	**username**	The username the IRC client is using.
String	**realname**	The "real name" the IRC client is using.
String	**encoding**	What string encoding the IRC client is using.
Boolean	**ssl**	*True* if the IRC client is connected to the server via SSL/TLS, and *False* if otherwise.
Integer	**uptime**	How long the IRC client has been connected to the server, in seconds.
socket object	**socket**	The socket object the IRC client is using for connectivity.