from pyclient.api import \*  
client = ClientAPI()

# **Core**

client.echo(line=’’)

Echoes a line of text to the primary output tab. Color codes are allowed.

client.execute(line)

Executes a line as if it was typed into the input bar. If it does not contain a command, the contents are sent to the server.

# **Connection**

client.connect(host=None, port=None)

Connects to the server. If host or port are not specified, the host or port from the config is used. If already connected, or connecting, disconnect first.

TODO: Maybe don't have it default to connection settings -- Connection system is self-contained?

client.disconnect()

Disconnect from the current server if we are connected; otherwise do nothing.

client.isConnected

True if the client is connected, false otherwise.

TODO: More connection state info - preferably a list of states

client.host

Host name of the server we are connected to (or None if not connected.)

client.port

Port number of the server we are connected to (or None if not connected.)

client.send(line)

Send a single line of text to the server, if connected. Automatically inserts CRLF at the end.

# Configuration

hasConfigKey(key)  
getConfigBool(key, default=False)  
getConfigInt(key, default=0)  
getConfigStr(key, default=’’)  
setConfigBool(key, val)  
setConfigInt(key, val)  
setConfigStr(key, val)  
  
These functions read from and write to the configuration dictionary.

saveConfig()

Saves the current configuration dictionary to file. You shouldn’t have to do this, as the configuration is saved when the client exits, and when the configuration dialog is closed.