**Chat Protocol**

**Introduction**

This is a network protocol specification document for a simple real-time chat communication protocol. The document specifies the network messages understood by servers and client using this communication protocol.

**Connect/Disconnect**

**Connect**

**Command:** CONNECT

**Parameters:** [host] [port]

**Example payload:** CONNECT chat.phixyn.com 17666

**Description:** Send a connect request to a server.

**Responses:** OK, REQ\_AUTH, ERR\_UNAVAILABLE, ERR\_SERVERFULL, USER\_PERMABANNED, USER\_TEMPBANNED, ALREADY\_CONNECTED (?)

TODO: Not sure if host and port are needed. Clients can handle sending this. We need some kind of handshake system.

**Disconnect**

**Command:** DISCONNECT

**Parameters:** None

**Example payload:** DISCONNECT

**Description:** Disconnect from a/the server but keep a user's authentication token?

**Responses:** OK

**Ping**

**Command:** PING

**Parameters:** None

**Example payload:** PING

**Description:** Polls clients for connection status/to see if they are connected. (?)

Can also ping servers?

**Responses:** OK, PONG (?)

**Users**

**Username rules**

Servers should only accept usernames that are a single alphanumeric string ...

Usernames should be unique! If a user tries to join using an existing username, an error is sent by the server (which one?).

Usernames must be a single alphanumeric string with no spaces. The maximum length of the username is 14 characters. Underscores are allowed to separate words (e.g. Ashleigh\_Smith).

The following characters are valid:

\* [0-9]

\* [Aa-Zz]

\* \_

\* [

\* ]

The following characters are invalid:

**Register new user**

**Command:** REGISTER

**Parameters:** [user] [password]

**Payload:** REGISTER Ashleigh a\_secure\_password

**Description:** When a client first connects, the server will send an REQ\_AUTH message?

**Responses:** OK, USER\_EXISTS, USER\_INVALIDNAME

**Login**

**Command:** LOGIN

**Parameters:** [user] (password)

**Payload:** LOGIN Ashleigh a\_secure\_password

**Description:** Send a login request to a server.

**Responses:** OK, ERR\_NOSUCHUSER, ERR\_AUTH

**Auth/Login (todo)**

**Command:** AUTH

**Parameters:** [user] [password]

**Payload:** AUTH Ashleigh a\_secure\_password

**Description:** Send an authentication request to a server. If the credentials are incorrect, the response will be ERR\_AUTH (?). If the user is not found, USER\_NOEXIST is returned (ERR\_NOSUCHUSER?).

**Responses:** OK, ERR\_NOSUCHUSER, ERR\_AUTH

**Logout**

**Command:** LOGOUT

**Parameters:** [user] [auth\_token]

**Payload:** LOGOUT Ashleigh some\_stored\_auth\_token

**Description:** Logout from a user, but don't disconnect from the server? o.o

**Responses:** OK

**Change password**

**Command:** USER\_PASS

**Parameters:** [password]

**Payload:** USER\_PASS a\_secure\_password

**Description:** Change \_*your\_* password. If a user has set a password, which they are forced to do, they can change it in the future using a command in the client. (implementation)

**Responses:**

**Admin system**

This should be described by adding "ERR\_PERMS" to the list of responses in commands that would require special privilege.

Examples: Kick, ban, mute, server restart, server shutdown, changing channel topic, etc

**Channels**

**Joining/leaving channels**

**Command:** CH\_JOIN

**Command:** CH\_LEAVE

**Parameters:** [channel]

**Payload:** CH\_JOIN #foo

**Description:** Clients use these network packets to join and leave channels. The channel name must be specified in both commands. Multiple channel names cannot be passed. However, clients can implement join and leave commands that accept multiple channels, in which case they would have to send a CH\_JOIN or CH\_LEAVE for each channel.

**Responses:** OK

**Listing channels**

**Command:** CH\_LIST

**Parameters:** (public/private)

**Payload:** CH\_LIST

**Description:** When the server receives this message, it will send a list existing channels to the client.

The optional parameter specifies whether to list \_only\_ public or \_only\_ private channels. The default behaviour, if no parameters are supplied, is the following: if this message (CH\_LIST) came from a non-admin user or a user who cannot see certain private channels, the server will only send the channels that the user has access to. If the user is an admin, all channels will be sent back to the client. The same behaviour happens when public or private is supplied as a non-admin user.

**TODO:** Format of the response (i.e. list/array, dictionary, JSON?)

**Response examples:**

#foo

#bar

#super-secret-chat

or

{"#foo","#bar","#super-secret-chat"}

**Channel names**

Channel names can only be alphanumeric and must not contain any spaces. The size limit for channel names is 20 characters. Dashes or underscores may be used for channel names with multiple words (e.g. super-secret-chat).

**Listing users of channels**

**Command:** CH\_USERLIST

**Parameters:** [channel]

**Payload:** CH\_USERLIST #foo

**Description:** Returns a list of users currently in a channel to the client.

**Responses:**

**Creating a channel**

**Command:** CH\_CREATE

**Parameters:** [name]

**Payload:** CH\_CREATE #foo

**Description:** Create a new channel with specified name.

**Responses:**

**Deleting a channel**

**Command:** CH\_DESTROY

**Parameters:** [name]

**Payload:** CH\_DESTROY #foo

**Description:** Delete an existing channel with the specified name

**Responses:**

**Channel topic**

**Command:** CH\_TOPIC

**Parameters:** (message)

**Payload:** CH\_TOPIC A channel for general discussion

**Description:** Read or set current channel topic

\*\*TODO:\*\* Consider adding a second parameter for channel name?

**Responses:**

**Messages**

**Command:** MSG

**Parameters:** [channel/user]

**Payload:** MSG Ashleigh Don't forget the milk!

MSG #general Does anyone else want to go home and play FFXIV?

**Description:** Send a message to an existing channel or a connected user

Public messages from a client are broadcast to all connected clients (except its origin) in the channel.

Private messages are only broadcast to the clients they are intended for, as opposed to all connected clients in a channel. Private messages are only sent to one client at a time.

**Responses:** OK

**Message length restrictions?**

The maximum length of a message in the MSG payload is x. This is because the maximum size of a packet is x ... (where is this defined though? :-{ )

**Other**

**MOTD**

**Command:** MOTD

**Parameters:** (message)

**Payload:** MOTD Welcome to Zion's chat system.

**Description:** Return or set current Message Of The Day on the server. (server wide)

**Responses:** (Current MOTD message) (+ OK?), NO\_MOTD???

**Uptime**

**Command:** UPTIME

**Parameters:** None

**Payload:** UPTIME

**Description:** Return the server's current uptime.

**Responses:** UPTIME string (e.g. 7d, 11:03:02), OK (?), ERR\_PERMS (?)

**Errors codes/messages**

**Common error cases**

Trying to send messages before authenticating

No permissions for certain actions (admin system)

ERR\_AUTH

ERR\_PERMS

**Registration/Authentication errors**

ERR\_NOSUCHUSER

USER\_EXISTS

USER\_PERMABANNED

USER\_TEMPBANNED

USER\_INVALIDNAME

???

ERR\_UNAVAILABLE

ERR\_SERVERFULL

Not sure about these

ERR\_USER [message]

ERR\_CHANNEL [message]

**Appendix**

Maybe put all error messages here?

**References**