**Chat Protocol Specification**

1. General  
   1. Transport Layer

The server and its clients use the Transmission Control Protocol to connect and communicate with one another.

* 1. Server Responses

All commands sent from client to the server can receive the following responses

(SUCCESS) Command executed successfully  
(FAIL) Command execution failed  
(ERR\_INVALID) Invalid command or missing parameters

* 1. Command Prefix

Commands sent from a client must be prefixed with a period and forward slash to prevent accidental occurrences (e.g. “./MKROOM johns\_room 12 secretpassword”)

* 1. Character Limitations

The limitations on the length of messages and room names are in place to prevent exploitation/misuse.

1. Chat Commands  
   1. Send Message (MSG)

**Command:** MSG  
**Parameters:** <message>  
**Example Usage:** MSG “Hello”  
  
Send a message to users within the same room. Messages must be between 1 and 128 characters long.  
  
**Server Responses**  
  
(ERR\_MUTED) User is muted  
(ERR\_SHORT) Message too short  
(ERR\_LONG) Message too long

* 1. Private Message (PM)

**Command**: PM

**Parameters:** <username> <message>

**Example Usage:** PM johnsmith “hello john”

Sends a private message to the user. Recipient must be in the same room as the sender. Private messages must be between 1 and 128 characters long.

**Server Responses**

(ERR\_NOUSR) User does not exist  
(ERR\_BLOCKED) Communication with the user is blocked  
(ERR\_SHORT) Message too short  
(ERR\_LONG) Message too long

* 1. Block User (BLOCK)

**Command:** BLOCK

**Parameters:** <username>

**Example Usage:** BLOCK johnsmith

Block the user, disabling all communication to and from them.  
  
**Server Responses**

(ERR\_NOUSR) User does not exist

* 1. Unblock User (UNBLOCK)

**Command:** UNBLOCK

**Parameters:** <username>

**Example Usage:** UNBLOCK johnsmith

Unblock the user, enabling communication to and from them.

**Server Responses**

(ERR\_NOUSR) User does not exist

1. Room Commands  
   1. Create Room (MKROOM)  
        
      **Command:** MKROOM  
      **Parameters:** <room name> [<capacity>] [<password>]

**Example Usage:** MKROOM johns-room  
  
Creates a room. The creator of the room is given ownership privileges (see 3.4 - 3.10). If a password is specified, the room is locked and requires entry of the password to be accessed. Rooms remain active until deleted by the room’s owner or automatically after a period of inactivity. A room can host between 2 and 128 users; an alternative maximum capacity can be specified but must be within this range. Room names are comprised of alphanumeric characters and hyphens and may contain a minimum of 3 and a maximum of 24 characters. There may only be one instance of any given room name present on the server at any one time. Room names may not contain any combination of words that are restricted by the server.   
  
**Server Responses**  
  
(ERR\_EXISTS) Room name already in use  
(ERR\_BADCAP) Invalid capacity  
(ERR\_SHORT) Room name too short  
(ERR\_LONG) Room name too long  
(ERR\_ILLEGAL) Room name contains illegal characters  
(ERR\_RESTRICTED) Room name contains restricted words

* 1. Join Room (ENTER)

**Command**: ENTER

**Parameters:** <room name> [<password>]

**Example Usage:** ENTER johns-room secretpassword

Enter an existing room on the server. If the destination room is locked, the correct password must be provided to enter.

**Server Responses**

(ERR\_BADPASS) Incorrect password

(ERR\_FULL) Room currently full

(ERR\_EXISTS) Room does not exist

* 1. Exit Room (EXIT)

**Command**: EXIT

Exits the current room and return to the lobby.

**Server Responses**

(ERR\_EXIT) User is not within a room

* 1. Kick User (KICK)

**Command:** BAN

**Parameters:** <username> [<message>]

**Example Usage:** KICK johnsmith

Removes a user from the room. An optional message can be supplied to give reasoning for the action. Requires room ownership privileges.

**Server Responses**

(ERR\_RIGHTS) Insufficient privileges

(ERR\_NOUSR) User does not exist

* 1. Ban User (BAN)

**Command**: BAN

**Parameters**: <username> <minutes> [<message>]

**Example Usage**: BAN johnsmith 60 “offensive language”

Removes a user from the room and prevents them from re-joining for a specified period. An optional message can be specified to give reasoning for the action. Requires room ownership privileges.

**Server Responses**

(ERR\_RIGHTS) Insufficient privileges

(ERR\_NOUSR) User does not exist

* 1. Mute User (MUTE)

**Command:** Mute

**Parameters:** <username> <minutes> [<message>]

**Example Usage:** MUTE johnsmith 5 “spamming”

Prevents a user from chatting within the room for the given period. Requires room ownership privileges.

**Server Responses**

(ERR\_RIGHTS) Insufficient privileges

(ERR\_NOUSR) User does not exist

* 1. Unmute User (UNMUTE)

**Command:** Mute

**Parameters:** <username>

**Example Usage:** UNMUTE johnsmith

Remove a user’s mute, allowing them to chat. Requires room ownership privileges.

**Server Responses**

(ERR\_RIGHTS) Insufficient privileges

(ERR\_NOUSR) User does not exist

* 1. Promote User (PROMOTE)

**Command:** PROMOTE

**Parameters:** <username>

**Example Usage:** PROMOTE johnsmith

Transfers ownership privileges to the user. Requires room ownership privileges.

**Server Responses**

(ERR\_RIGHTS) Insufficient privileges

(ERR\_NOUSR) User does not exist

* 1. Lock Room (LOCK)

**Command:** LOCK

**Parameters:** <password>

**Example Usage:** LOCK secretpassword  
  
Locks the room with the password, overwriting any previously set password. Room passwords may consist of alphanumeric characters and be between 8 and 24 characters long. Requires room ownership privileges.

**Server Responses**

(ERR\_RIGHTS) Insufficient privileges

(ERR\_BADPASS) Invalid password

**3.10.** Unlock Room (UNLOCK)

**Command:** UNLOCK

Unlocks the room, removing any previously set password. Requires room ownership privileges.

**Server Responses**

(ERR\_RIGHTS) Insufficient privileges

**3.11.** List Rooms (LSROOMS)

**Command: LSROOMS**

Outputs a list of all existing room names. Password protected rooms are suffixed with “[\*]”.  
  
**Server Responses**  
  
(ERR\_NOROOMS) No user created rooms exist

1. User Specific Commands
   1. Grant Room Ownership (GIVEROOM)

**Command:** GIVEROOM  
**Parameters:** <username> <room>  
**Example Usage** GIVEROOM johnsmith johns-room  
  
Gives a user ownership privileges of a room.  
  
**Server Responses**  
  
(ERR\_NOUSR) User does not exist  
(ERR\_EXISTS) Room does not exist

* 1. Revoke Room Ownership (TAKEROOM)  
       
     **Command:** TAKEROOM  
     **Parameters:** <username> <room>  
     **Example Usage** TAKEROOM johnsmith johns-room  
       
     Removes a user’s ownership privileges of a room.  
       
     **Server Responses**  
       
     (ERR\_NOUSR) User does not exist  
     (ERR\_EXISTS) Room does not exist
  2. Discard User (DISCARD)  
       
     **Command:** DISCARD  
     **Parameters:** <username>  
     **Example Usage:** DISCARD johnsmith  
       
     Close the connection between the user’s client and the server.  
       
     **Server Responses**  
       
     (ERR\_NOUSR) User does not exist
  3. Set/Change Username (UNAME)

**Command:** UNAME  
**Parameters:** <username>  
**Example Usage:** UNAME johnsmith  
  
Sets or change a user’s username. Usernames must be alphanumeric and comprised of a minimum of 3 and a maximum of 12 characters. There may only be one instance of any given username connected to the server at any one time. Usernames must not contain any combination of words that are restricted by the server.  
  
**Server Responses**  
  
(ERR\_EXISTS) Username already in us  
(ERR\_SHORT) Username too short  
(ERR\_LONG) Username too long  
(ERR\_ILLEGAL) Username contains illegal characters

(ERR\_RESTRICTED) Username contains restricted words