

Multiplayer Game Programming A2 Documentation

a. Feature List

Features:

Packet Management

- Both server and client can add data to packets, send and receive encoded packets from each other, decode packets, and find data from decoded packets.
- Server packet data: Message, Timestamp, RoomName(Optional), SessionID(Optional)
- Client packet data: Message, RoomName
- Packet Management Codes in packet_manager.h/.cpp
- Implementation example in IO_Multiplexing_Server.cpp (Lines 718 - 730)

Welcome Message

- User will be sent a welcome message with their Session ID when they join the server
- send_welcome_message function in IO_Multiplexing_Server.cpp (Lines 30 - 50)

```
[23:22:18] -Welcome to my I/O multiplexing server! Your Session ID is (316)-
```

Info Message

- User will be sent a info message with a list of all the currently connected users (besides themselves), after the welcome message
- session_info_message function in IO_Multiplexing_Server.cpp (Lines 52 - 94)

```
[23:22:18] -No other clients are connected-
```

```
[23:23:15] -Already connected to 1 other client(s) with session ID(s) (316)-
```

Notice Message

- Whenever a user joins the server, all other users will be sent a notice message which informs them about the user joining
- send_notice_message function in IO_Multiplexing_Server.cpp (Lines 96 - 123)

```
[23:24:12] -New client (248) has connected-  
(260)>  
[23:24:17] -New client (336) has connected-  
(260)>
```

Timestamps

- Each packet sent by the server includes a timestamp(HH:MM:SS)
- Function to get time: get_time in IO_Multiplexing_Server.cpp (Lines 18 - 28)

Rooms

- Each room has a room master (user that made the room), room name and password (set by room master), and a list of room users
- Room struct in room_stuff.h
- Functions using rooms are below

Commands

- /all

- Broadcast a message to all users anywhere
- Broadcasting user gets a confirmation message if sent successfully, or a error message if failed to send
- Other users can see which user broadcasted what
- send_to_all function in IO_Multiplexing_Server.cpp (Lines 126 - 208)

```
[23:24:17] New client (260) has connected  
(260)> /all test  
Sending Message...  
[23:24:58] -Successfully broadcasted message-  
(260)>
```

```
(260)> /all  
Sending Message...  
[23:25:08] -Failed to broadcast message, Message not found-  
(260)>
```

```
[23:24:58] Client (260) broadcasted : test
```

- /createroom

- Create a room with a unique name and a password (no spaces)
- User that created the room becomes the room master
- User also leaves the current room and joins the newly created room
- User gets a confirmation message if created successfully, or a error message if failed to create
- try_create_room function in IO_Multiplexing_Server.cpp (Lines 1128 - 1175)
- create_room function in IO_Multiplexing_Server.cpp (Lines 546 - 639)

```
(260)> /createroom room 123  
Sending Message...  
[23:25:51] -Room (room) created, Joined room (room)-  
(260)>
```

```
[23:25:51] Room (room) created, Joined room (room)  
(260)> /createroom m  
Sending Message...  
[23:26:27] -Failed to create room, Missing room name and/or password-  
(260)> _
```

- /deleteroom

- Delete the current room with a password
- User must be room master to delete room
- All users in the room will leave the deleted room and join the lobby room
- All users gets a message about room deleted if deleted successfully
- User gets a error message if failed to delete
- try_delete_room function in IO_Multiplexing_Server.cpp (Lines 1224 - 1266)
- delete_room function in IO_Multiplexing_Server.cpp (Lines 891 - 1022)

```
(260)> /deleteroom w
Sending Message...
[23:28:28] -Failed to delete room (Lobby), Wrong password-
(260)>
```

```
(260)> /deleteroom 123
Sending Message...
[23:28:41] -Room (room) has been deleted, Joining back room (Lobby)-
(260)>
```

- /inforoom
 - Look at information about the room like room master, password (If user is master), and users
 - Shows who is the user
 - show_room_info function in IO_Multiplexing_Server.cpp (Lines 1062 - 1125)

```
[23:28:58] -Client (248) has
(260)> /inforoom
Sending Message...
[23:27:59]
*** Room (room) info ***
> Room Password : 123
| Room Master |
> (260) <- You
| 2 Room Users |
> (260) <- You
> (248)
*** End of Room Info ***
```

- /joinroom
 - Join a room with a password
 - User leaves the current room before joining the room
 - User gets a confirmation message if joined successfully, or a error message if failed to join
 - Other users will see a notification message about the user joining
 - try_join_room function in IO_Multiplexing_Server.cpp (Lines 1176 - 1223)
 - join_room function in IO_Multiplexing_Server.cpp (Lines 546 - 639)

```
(248)> /joinroom r
Sending Message...
[23:27:25] -Failed to join room, Missing room name and/or password-
(248)>
```

```
(316)> /j 1 3
Sending Message...
[23:41:14] -Failed to join room (1), Wrong password given-
(316)>
```

```
[23:25:54] -Client (248) has
(248)> /joinroom room 123
Sending Message...
[23:26:58] -Joined room (room)-
(248)>
```

```
[23:41:35] -Client (316) has joined room (1)-
(320)>
```

- /leaveroom
 - o Leave the current room
 - o User will join the lobby room after leaving
 - o leave_room function in IO_Multiplexing_Server.cpp (Lines 788 - 890)

```
(248)> /leaveroom
Sending Message...
[23:29:50] -Left room (room)-
[23:29:50] -Client (248) has left room (room)-
```

- /help
 - o Shows the help menu
 - o send_help_message function in IO_Multiplexing_Server.cpp (Lines 209 - 241)

```
(260)> /help
Sending Message...
[21:30:25]
*** How to use this program ***
> /all [Message] - Broadcast message [Message] to all connected clients
> /createroom OR /c [RoomName] [RoomPassword] - Create a room named [RoomName] with password [RoomPassword] and become Room Master
> /deleteroom OR /d [RoomPassword] - Delete current room with password [RoomPassword] (Only for Room Master)
> /info OR /i - Show name, owner and users of current room
> /joinroom OR /j [RoomName] [RoomPassword] - Join a room named [RoomName] with password [RoomPassword]
> /leaveroom OR /l - Leave current room
> /help OR /h - Show this message
> /quit OR /q - Quit and close the connection
> /rooms OR /r - Show a list of all rooms
> /users OR /u - Show a list of all connected users
> /whisper OR /w [SessionID] [Message] - Send direct message [Message] to user [SessionID]
> Text not starting with / will be sent to all connected clients in the same room
*** End of Help ***
(260)> .
```

- /quit
 - o Quit and closes the connection
 - o Other users will see a notification message about the user leaving
 - o quit_user function in IO_Multiplexing_Server.cpp (Lines 280 - 325)

```
(344)> /q
Sending Message...
[23:49:41] -Successfully quit server-
```

```
[23:49:41] -Client (344) left the server-
```

- /rooms
 - o Shows a list of all the rooms
 - o Shows which room user is in
 - o show_rooms function in IO_Multiplexing_Server.cpp (Lines 1023 - 1061)

```
(248)> /rooms
Sending Message...
[23:31:21]
*** 3 rooms are available ***
> (Lobby) <- You are here
> (1)
> (room)
*** End of Room List ***
```

- /users
 - Shows a list of all the users
 - Shows who is the user
 - send_client_list function in IO_Multiplexing_Server.cpp (Lines 242 - 279)

```

(248)> /users
Sending Message...
[23:31:37]
*** 2 clients online ***
> (260)
> (248) <- You
*** End of Client List ***

```

- /whisper
 - Send a direct message to a specific user
 - Whispering user gets a confirmation message if sent successfully, or a error message if failed to send
 - Only target user can see the sent message
 - whisper_to_one function in IO_Multiplexing_Server.cpp (Lines 326 - 466)

```
(248)> /whisper
Sending Message...
[23:32:02] -Failed to send a message, Target Client/Message not found-
```

```
(248)> /whisper wfwe
Sending Message...
[23:32:16] -Failed to send a message, Invalid target client-
```

```
(248)> /whisper 123
Sending Message...
[23:32:25] -Failed to send a message, Target Client/Message not found-
```

```
(248)> /whisper 260 hello
Sending Message...
[23:32:43] -Successfully sent message to Client (260)-
```

```
[23:32:43] Client (248) whispered : hello
```

- Sending message to all users in the same room
 - Sending user gets a confirmation message if sent successfully, or a error message if failed to send
 - Only users in the same room can see the sent message

- send_to_room function in IO_Multiplexing_Server.cpp (Lines 467 - 524)

```
(248)> hello room  
Sending Message...  
[23:33:40] -Successfully sent message to room-
```

```
[23:34:21] Client (248) sent : hello room
```

b. User Guide

- /all [Message] - Broadcast message [Message] to all connected clients
- /createroom OR /c [RoomName] [RoomPassword] - Create a room named [RoomName] with password [RoomPassword] and become Room Master
- /deleteroom OR /d [RoomPassword] - Delete current room with password [RoomPassword] (Only for Room Master)
- /inforoom OR /i - Show name, owner and users of current room
- /joinroom OR /j [RoomName] [RoomPassword] - Join a room named [RoomName] with password [RoomPassword]
- /leaveroom OR /l - Leave current room
- /help OR /h - Show this message (Help menu)
- /quit OR /q - Quit and close the connection
- /rooms or /r - Show a list of all rooms
- /users OR /u - Show a list of all connected users
- /whisper OR /w [SessionID] [Message] - Send direct message [Message] to user [SessionID]
- Text not starting with / will be sent to all connected clients in the same room