

Network Message Design Document

General Format

All messages sent through the network will be sent through port 6000. All messages will follow a general format, with a few exceptions. This format is as follows:

Name, Designation#, Role#, Action#, param1, param2, param3

- Name: Username of the user sending the message
- Designation#: Who is the message for (0 - Host, 1 - Client)
- Role#: Role of the user sending the message (0 - Spectator, 1 - Red, 2 - Black)
- Action#: The action being performed
 - Host Actions
 - 0: Client Connected
 - 1: Server Lobby Text
 - 2: Client Role Change
 - 3: Client Moved
 - 4: Client Chat
 - 5: Game Over
 - Client Actions
 - 0: New Client Connected
 - 1: Server Lobby Text
 - 2: Client Role Changed
 - 3: Host Started Game
 - 4: Host Selected a Theme
 - 5: Host Moved
 - 6: Spectators updated with the board
 - 7: Client Chat Updated
 - 8: Game Over
- param1, param2, param3: Customizable parameters that change depending on the data needing to be sent. Some examples of parameters:
 - Host action 0 (Client Connected): param1 is strPlayerList, param2 is roleData, param3 is the number of clients connected
 - Host action 4 (Client Chat Sent): param1 is the message, param2 and param3 are both null because no additional data needs to be sent
 - Host action 5 (Game Over): param1 is the winner, param2 and param3 are both null because no additional data needs to be sent

Exceptions

There are a couple exceptions to this general format. These exceptions are as follows:

- strMessage[0], 1, null, 1, Message, null, null
 - The username is the clients username
 - Differs from the format as this is sent to all spectators to update chat
- strMessage[0], 1, strMessage[4], 2, intRoleData[], null, null
 - The username is the clients username, strRole is the new role
 - Differs from the format as this is sent to all spectators to notify of a role switch
- strMessage[0], 1, strRole, 6, strBoard[], intRedPieces, intBlackPieces
 - The username is the clients username
 - Differs from the format as this is sent to all spectators to update the board
- strMessage[0], 1, strRole, 7, strMessage[4], null, null
 - The username is the clients username
 - Differs from the format as this is sent to the spectators to update their chat when a client sends a message in the in-game chat

Examples of Messages and Actions

Here are some examples of what the program performs upon a specific message, as well as a description of what that message means

- strUsername, 0, 0, 0, null, null, null
 - New client joined the lobby
 - Updates players connected, player list, and adds 1 to the amount of spectators. This is because a player is defaulted to a spectator when joining the lobby.
 - Updates with the number of spectators and updates the chat with the player joined
 - Send a message to the client with all the game data
 - strUsername, 1, strRole, 0, strPlayerList[], intRoleData[], intPlayersConnected
- strUsername, 1, strRole, 0, strPlayerList[], intRoleData[], intPlayersConnected
 - Message is sent back to all clients with the updated data as a new client has joined
 - Updates the player list, role data, and the players connected
 - Updates the panel for the red players, black players, spectators, and sets the title of the screen to "[host name's]'s Lobby"
 - Updates the chat with the new player connected
- strUsername, 1, strRole, 8, strWinner, null, null
 - Game is over, a winner has been declared
 - If the winner is red, sets the text of the game over screen to say red has won
 - If the winner is black, sets the text of the game over screen to say black has won
 - Switches to the game over screen