CS494 Erik Jastad
Internet Draft Portland State University
Intended status: IRC class Project Specification June 8, 2019
Expires: December 2019

Internet Relay Chat Class Project

Status of this Memo

This document may not be modified, and derivative works of it may not be created, except to publish it as an RFC and to translate it into languages other than English.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet- Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on December 9, 2019

Abstract

This memo describes the communication protocol for an IRC-style client/server system for the Internetworking Protocols class at Portland State University.

Table of Contents

1. Introduction

This specification describes a simple Internet Relay Chat (IRC) protocol for clients to communicate with each other. This system uses a central server which relays messages that are sent to it to other connected clients.

Clients can create,join,leave and delete rooms. Rooms are groups of users that are part of the same message stream. Any message sent to a room is forwarded to all clients currently in that room.

Clients can also broadcast a message to all clients currently connected to the server or they can send a private message directly to other clients.

2. Conventions used in this document

The key words "MUST" and "MAY" in this document are to be interpreted as described in RFC 2119 [RFC2119].

In this document, these words will appear with that interpretation only when in ALL CAPS. Lower case uses of these words are not to be interpreted as carrying significance described in RFC 2119.

3. Basic Information

All communication described in this protocol takes place over TCP/IP, with the server listening for connections on a default port of 65432. If this default port is in use the host application will prompt for a new port to be used. Clients connect to the port and maintain a persistent connection to the server. The client can send messages and requests to the server over this established channel, and the server replies via the same channel. The system is asynchronous in the fact a client can send messages to the server at any time and the server sends messages to the clients at any time.

Both the server and client MAY terminate the connection at any time.

The server MAY choose to allow only a finite number of users and rooms. This is resource and system dependent.

4. Message Infrastructure

A client MAY type in anything they want to be broadcasted to all rooms they are currently in.

Special commands are initiated with a ‘/’ character preceding the command. This command is received by the server and parsed and validated for a matching operation to perform.

4.1 Special Operations

/create
/join
/leave
/delete
/rooms
/broadcast
/whisper
/roomlist
/userlist
/myrooms
/help

4.1.1 Usage

The create command MUST be followed by a single room or a list of comma separated rooms. This creates each room specified unless the room name already exists.

The join command MUST be followed by a single room or a list of comma separated rooms. This puts the client in each room specified if it exists.

The leave command MUST be followed by a single room or a list of comma separated rooms. This removes the client from each room specified if they were currently in the room.

The delete command MUST be followed by a single room or a list of comma separated rooms. This removes each room specified only if the room exists and no clients are currently in the room.

The rooms command MUST be followed by a single room or a list of comma separated rooms which is followed by the intended message. Each room specified will have the message relayed to the clients in the room only if the room exists.

The broadcast command MUST be followed by a message. This will relay the message to every client in the IRC.

The whisper command MUST be followed by a client name and a message. This will relay the message to the specified client if they exist.

The roomslist command will print each room that is available on the server.

The userlist command can have a room follow it or it can be used standalone. The standalone command will print a list of every user in the IRC. If the command is followed by a room then every user in the specified room is listed only if the room exists.

The help command is executed client side and prints a list just like 4.1 noted above.

4.2 Error Messages
Default port is not available!
Username is already in use, please enter a different name!
Please enter a valid user name!
Unrecognized Command!
Room was not found!
User was not found!
Room has active users and could not be deleted!
The general room can not be deleted!
You are currently not in the room!
The room already exists!
Host connection lost!

5. Special Operation Semantics

- MUST start with a '/'
- MUST contain a valid operation listed in 4.1
- All messages sent/received MUST be 1024 bytes or less
- MUST follow the guidelines in 4.1.1 for formatting

6. Client Messages

The first message from the client will be their username that other clients identify them by. If it already exists they will be prompted for a different name.

To send a message a client can simply type their message followed by the enter key. This message will be sent to each room the client is a part of and those rooms will send the message to their connected clients.

7. Server Messages

Broadcasted messages are sent by the server to each connected client whenever a client joins or leaves the IRC.

Room messages are sent by the server whenever a client joins or leaves a room.

8. Error Handling

Both the server and client detect when the socket connection linking them is terminated. If the server detects a client is disconnected it MUST remove the client from the connected clients list, by implementation this removes the user from the rooms as well.

9. Extra Features

Private messaging is supported to a single client or multiple clients.

10. Conclusion

This specification provides a generic messaging framework for multiple clients via a central server.

This application could be improved to include a GUI. This would allow for separate windows to send and receive messages. This would also make it easier for clients to manage rooms.

This application could also be improved by providing better data structures to handle operations. Originally it was constructed this way, but do a problem not recognized by my inexperience with python classes I changed how this is being handled. After developing the application I realized the mistake but due to time constraints I could not reimplement in time.

11. Security Considerations

Messages sent using this system have no additional protection of any kind. Any protection that is in place is due to the network,system or other processes and not provided by this IRC system.

12. References

594-SampleRFC.

13. Acknowledgments

This document was prepared using google docs.