Contents

The Internet Relay Chat

Mortuza Ahmmed

Essential IRC commands for everyday chat use. This quick reference covers basic connection, messaging, and channel management with practical examples for immediate application.

1 What is Internet Relay Chat?

IRC(Internet Relay Chat) is a text based chat system for instant messaging. It is a multi-user, real-time communication system hundreds of thousands of people all over the world using. It is mainly designed for group(many-to-many) communication in discussion forums called channels, but also allows one-to-one communication and data transfers via private message. IRC is more than entertainment. It's active communication.

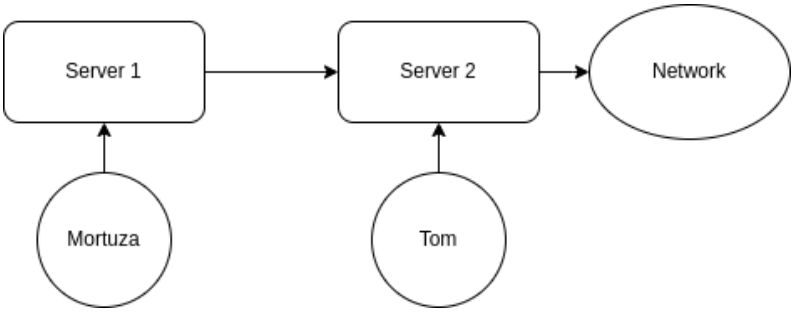
It was created by Jarkko "WiZ" Oikarinen[1] in late August 1988 to replace a program called MUT(MultiUser Talk) on a BBS called OuluBox in Finland. He found inspiration in a chat system known Bitnet Relay, which operated on the BITNET[2]. IRC gained prominence when it was used to report on the Soviet coup attempt of 1991 throughout a media blackout.

2 Technical Concept

IRC, in its simplest form, is made up of two programs—a *server* program that accepts connections and a *client* program that connects to the server. A client program handles some necessary procedures automatically and provides a better and simple user interface than the more technical messages the client and server exchange. IRC servers connect to each other via an IRC *network* of servers. Let's use a very simple model of an IRC network for our example:

Two servers and two clients. The servers are connected to each other, and each has client connected to it. Let's say Mortuza wishes to send a message to Tom. However,their machines don't connect directly. But each connects to a server, which is in turn connected to the server to which the other user is connected. Therefore, Mortuza can make use of the indirect route that exists between him and Tom. What Mortuza does is send server 1

[1] https://en.wikipedia.org/wiki/Jarkko_Oikarinen
[2] https://en.wikipedia.org/wiki/BITNET



client-server model

a message. In this message, he will tell server 1 the message's final destination(Tom) and its contents. Server 1 is aware of the existence of Tom, although he is not connected to it directly, and it knows that he is connected to server 2. It therefore forwards—*relays*—the message to server 2, which in turn sees that the recipient is one of its own clients and sends the message to Tom, who can then read it. Server 1 also adds the identity of the client sending it (Mortuza) before relaying it, so the recipient knows who it's from. This transfer of information between the servers and its users typically happens within milliseconds, thus making the exchange of messages swift enought to match that of real conversation. This is why Mortuza doesn't need to connect directly to Tom to send his message—the IRC environment permits an almost unlimited number of recipients for the same message and can relay this message to all those users at the same time. IRC permits one-on-one communication, but its real advantage is the ability to communicate with large numbers of people by sharing a common *channel* of conversation. Let's say we add a third user Rupa. Rupa is connected to server 2, just like Tom is. All three of them join a channel which they decide to call **#addakhana**[3]. They arrange the channel name by sending messages to each other. Establishing this channel gives them means of three-way communication. So, if Rupa wants to tell Mortuza, "I am a July warrior," and sends the message to the channel instead of sending it only to Mortuza, all users on the channel receive the message.

To truly understand IRC, you just have to try it. Once you connect, it won't take long to discover whether you love it, hate it, or find it nice for an occasional visit. Not everyone is thrilled by it—I've observed that it often does not appeal to people who find it necessary to be in visual contact with the people they are conversing with.

3 What do I need?

A computer, a network connection and an IRC client.

[3]This is the channel I registered on the Libera Chat network

3.1 The IRC client

The client is the piece of software you will use to connect to an IRC server over your network connection, much like you use an E-mail client to send and receive internet mail. There are quite a few different IRC clients around. If you are unsure of which one to choose, start with mIRC(if you run Windows) or Textual(if you run Mac OS). The following tables illustrate some IRC clients.

3.1.1 Mobile clients

Client	Platform(s)	Key Features
AndroIRC	Android	Feature-rich client, quick start wizard, multiple server support.
IRCCloud	iOS, Android	Native applications that sync with its web service.
Palaver	iOS	Intuitive interface, full IRCv3 support, push notifications.
LimeChat	iOS	Single window for multiple servers, fast and stable.

3.1.2 Web-based clients

Client	Platform(s)	Key Features
Kiwi IRC	Web	Modern web client, theming options, easy to embed on a website.
IRCCloud	Web, iOS, Android	Premium service with a web and mobile interface, offers a cloud bouncer for persistent connections.
TheLounge	Web	Self-hosted client that keeps you connected 24/7.
Mibbit	Web	Well-known web-based client that runs in any browser.

3.1.3 Desktop clients

Client	Platform(s)	Key Features
mIRC	Windows	Scripting language, customizable, buddy lists, multi-server connections.
HexChat	Windows, macOS, Linux	Easy to use, active development, multi-server support, customizable.
WeeChat	Windows, macOS, Linux	Lightweight terminal-based client, highly extensible with scripts and plugins.
Pidgin	Windows, macOS, Linux	Multi-protocol support, multi-server connections, open-source development.
Quassel IRC	Windows, macOS, LInux	Distributed client with a core component, allowing persistent connections.
Irssi	macOS, Linux	Terminal-based client, known for being robust and scriptable.
Textual	macOS	Modern user interface, integrates with iCloud and ZNC.
Konversation	Linux(KDE)	User-friendly graphical client for KDE desktop environments.

4 Where do I connect to?

Having installed a client program and familiarized yourself with IRC, you are now ready to connect to an IRC server. Depending on your client, there are different ways of setting the server or list of servers to which you intend to connect. Most clients let you set a default server to which your client automatically connects at start-up. Some also specify one or more alternative servers in case you fail to connect to the first. Let's not rush things, though. First of all, you will need to choose a network, and from its available servers select the one most likely to suit you.

4.1 Selecting a network and server, and connecting

In order to pick the most suitable network, you ask yourself what you are expecting to find on IRC. If you want to hang around, explore a few channels, and meet many different people, one of the major networks is probably the best choice. Once you find a

network, check its server list for the nearest server. This step is not absolutely necessary, but it is probably a good idea. Many networks have a generic server address that takes you a random server–for example, DALnet[4] has the irc.dal.net address for that purpose.

You can check the website `https://netsplit.de` and pick a network that seems likely to suit your needs. If your first choice doesn't work out, there are always many more. Remember, quantity doesn't necessarily mean quality—even if you are on a network with 30,000 or 40,000 users, you'll rarely contact more than a few dozen people at a time.

4.2 What's the difference between an IRC server and an IRC network?

The IRC server is the software that runs on a server machine, accepting connections from the users' IRC clients such as yours, and then handles traffic so that the clients can chat together.

An IRC network is formed when two or more servers are connected together. When you can chat on a server that is part of an IRC network, you will be able to talk not only other users on the same server, but also to any user on any other server on the same network. However, users on different networks, e.g., A on IRCnet and B on EFnet, will not be able to chat with each other using normal IRC functionality.

5 Things to Note

5.1 Your IRC nickname

Before logging on, you'll need to decide on a nick name, such as DrYunus or Hasina05. Your nick name is what other IRC users primarily will know you as; it will show up whenever you join or part a channel, send a private message, and so on. It can be up to a maximum nick name characters long which is specified by server and given to clients upon connecting. Unless your real first name is reasonably exotic it will probably already be taken, so try to make up something unique instead. Nick names on IRC are not owned, but trying to use the same nick as someone else will cause grief such as server kills and misplaced private messages.

5.2 Internet addressing

Two or three parts compose the name of any machine on the internet, using the following structure:

```
[arbitary.hostname.]domain.tld
```

TLD is the top level domain. It can be a generic one like COM, NET or EDU, which implies the nature of the organization using it. Some of these can be registered and used by anyone, anywhere in the world. In other cases it denotes a country—for example, BD for Bangladesh or IN for India. The country codes follow the ISO 3166 standard. Other essential is the second level domain which comes before the tld or indentifier.tld

[4] https://dal.net

combination. This is a name the site chooses which is registered with the competent authority. Everything before the second level domain is a machine name—an arbitrary choice of the domain's maintainers. Some indicate the function of the machine to which they point. For example, "www" indicates a Web server and "ftp" indicates an FTP server.

Many sites maintaining an IRC server follow the convention of using "irc" as the machine's name. Some typical examples are:

```
irc.dal.net (IRC server operated by DALnet)
irc.libera.chat (IRC server operated by LiberaChat)
```

5.3 The usermask

In addition to simple nicknames, a more versatile way of referring to user identities on IRC is the usermask, sometimes referred to as user@host or banmask. Usermasks are used when placing channel bans on the IRC server, and chances are that your IRC client uses them locally as well for specifying who should be ignored, automatically given channel operator status etc. The mask is constructed as follows:

1. the nickname
2. ! (separator character)
3. the user name
4. @ (separator character)
5. the host name or IP address

Wildcards are allowed, e.g., *!*@* would cover anyone on IRC. While you're connected, other users know you in two different ways. The first is simply your nickname, which people need in order to reach you and see you on IRC. It's used as a destination address for messages and an identifier for server queries regarding you. The final result looks like this:

```
SomeNick!mortuza@mtz.provider.com
Nickname!username@host.name
```

This is your identity for sending to the server. Every item the server receives from your connection automatically gets this added to it as the sender's identity. If you send a message destined for a user or another server, this full version will be forwarded to the recipient along with the message.

6 Basic IRC Usage

6.1 Connect to an IRC network

The specific command you use to connect to an IRC network depends on the client you are using(like mIRC, HexChat, Irssi, etc.), but the universal underlying command is:

`/SERVER [serveraddress] [port]`
You typically enter this command into the input line of your IRC client:
`/SERVER irc.dal.net 6697 or /CONNECT irc.dal.net 6697`
If you are unsure of your client, try /HELP or consult the program's documentation. If all goes well, you make a successful connection to an IRC server. The first four lines you see look something like this:

```
*** Welcome to the Internet Relay Network
SomeNick!mortuza@mtz.provider.com
*** Your host is irc.webber.net, running version 2.9.5/sc8a2/Mr2
*** This server was created Fri Oct 24 2025 at 21:03:58 EST
*** umods available oirw, channel modes abiklmnopqstv

    A report on the server's and network's current status follows:

*** There are 7444 users plus 28298 invisible and
    8 services on 66 servers
*** 159 operators online
*** 19 unknow connections
*** I have 2234 clients, 1 services and 1 server
*** Current local users: 2234 Max: 3068
*** Current global users: 35742 Max: 39293
```

6.2 Disconnecting from a server

At any time, you may close your connection to the IRC server. This is done via the **QUIT** command. Depending on the client you are using, there may be one or more synonyms for that command—for example,**BYE**, **EXIT** or **SIGNOFF**. If fact, some clients use one of these options rather than **QUIT**.
`/QUIT`
Technically, the **QUIT** command initiates the closure of your **TCP** connection to the server and exits the client program. **DISCONNECT** command closes a server connection without also exiting the client.
Note: The commonly accepted command character is forward slash (/). The client in turn sends commands to the server without the slash. Clients consider everything not preceded by command character to be a message.

6.3 Basic commands

The basic structure of a command is as follows:
`<command character>COMMAND <parameters to the command>`
For example, the **NICK** command looks like this:
`/NICK SomeNick`
There is a command character (/), a command (NICK), and a single parameter to the

command(a user's nickname). There must be no space between the command character and the command, or unexpected things may happen. There has to be a space between the command and its parameters, however. Commands are not case sensitive—you may use capital letters, lowercase, or even mixed case, if such is your desire. **NICK**, **nick**, and **Nick** all have the same meaning. Some commands accept more than one parameter, while others take none. The command may require parameters or they maybe optional. There's lots of IRC commands. Let's do the math:

$$-\frac{\hbar^2}{2m}\frac{d^2\psi(x)}{dx^2} + V(x)\psi(x) = E\psi(x)$$

I'm just kidding. Here is a summary of commands you can't live without, and a few more you could live without but wouldn't want to:

Command	**Required**	**Optional**	**Function**
nick	Nickname	Server name	Changes your nickname.
whois	Nickname	Server name	Requests information on a user.
join	Channel name	Password	Joins a channel.
server	Server name	Port	Connects you to the server specified as a parameter, disconnecting you from another you may be using.
part or leave	Channel name		Leaves a channel.
quit		Message	Disconnects and exits.
msg	Nicknames(s) and message		Sends a private message to that nickname.
away	Message		Sets you "away". Use it without the message to cancel "away" status.

7 User Modes

All servers permit a user to use certain settings that influence the status of the session. The number and function of these settings varies greatly among different server types, so we'll limit ourselves to looking only at those that are present on most servers or are most important. Setting or removing a user mode for yourself is simple:

```
/mode <your-nickname> +/-<mode>
```

Using the plus (+) before the letter corresponding to the user mode activates it, while the dash (-) unsets it. Omitting a prefix is the same as using a plus sign. User mode letters are case sensitive—**W** is not the same as **w** and may have a quite different effect or may not have a meaning at all.

User mode i: This is the most widely used user mode. User mode i (invisible), when set, makes a client invisible to certain types of user listings and scans. This is also the solution to the problem of being targeted by spam bots. Many servers automatically set it for a user upon connecting, in which case you'll see a notice that looks somewhat like this:

*** Mode change ''+i'' for user SomeNick by SomeNick

You can set it manually:

```
/mode SomeNick +i
```

User mode w: This lets you receive wallops, which are a special type of message servers or IRC operators send out, usually announcing some network event. Only IRC operators need to see them, so you might as well leave it off. On some networks, only IRC operators can receive them, and user mode **+w** does nothing for the regular user.

User mode s: User mode **s** is the perfect way of getting your screen flooded with countless useless server notices. User mode **s** sends you all sorts of server notices, and there are often additional user mode that allow someone to monitor only a certain type of notice.

User mode o: This user mode indicates that IRC operator status is active—therefore it's available only to IRC operators. Trying to set user mode **+o** is useless.

User mode r and d: Service robots often use user mode **d**(dumb), which is not available on all server type. On those networks that do allow users to use it, it prevents channel text from reaching the client and so is very useful. The **r** stands for **restricted**. This restriction lies in the fact that you may not change your nickname without disconnecting from the server and may not use channel operator commands, even if you are give channel operator status.

Other user modes: A variety of ther modes exist that IRC operators use for monitoring the server. None of these are of any interest to the average user, so you'll do fine without them. Some user modes might be in use on different types of servers, but function differently—**i**,**o**,**s**, and **w** are the only ones you can expect will do the same thing almost all servers. The most interesting mode available on several small network is known as **x**, **a**, or **z** depending on the network.

8 IRC Channels

One way of chatting is joining one or more channels. A channel is a group of users chatting (or idling) together. Channels are usually formed by people with something in common– teenagers living in the same city, fans of the same rock group, Linux enthusiasts…Any

text you send to a channel will be visible to all the users on the channel (unless they have set their clients to ignore you).

Your IRC client may come with a pre-loaded list of channels, but there's no guarantee that those channels happen to be in use on your particular server or network at any given time.

Channels are identified by a name, which usually reflects the topic of discussion in that channel or kinds of people that frequent it. The subject matter is entirely arbitrary, as is the name. Anyone can create a channel, name it whatever they like, and talk about anything they care to talk about. The only exception to this is channels on a network that enforces a policy about the kind of channels its users may create—for example, one that forbids sex channels.

Channels often also have a topic—a description of the channel and its purpose or a comment related to some event on the channel itself. Global channels, accessible from all servers of the given network, are invariably characterized by a leading hash mark (#) in their name. Local channels which only users of a specific server may join, begin with an ampersand (&).

To summarize this, a channel named **#addakhana** is global, whereas you can only reach a channel named **&santahar** from one server. Global channels are the majority, and have many more potential uses and problems.

8.1 List of available channels

All servers keep the current channel list in memory, and most allow a user to retrieve a large part of it. This is done using the **LIST** command. The basic syntax of the **LIST** command:

```
/list
```

Most clients allow you to limit the number of channels displayed. They filter the list for channels with characteristics you define by adding some parameters.

8.2 Joining a channel

Becoming part of a channel is simple. With the **JOIN** command, your client sends a join to your server, which in turn checks to see whether you have permission to join the channel you specified. Let's say you want to join a channel named **#santahar** (remember the hash mark is an integral part of the channel name, and you must use it). The command is simple:

```
/join #santahar
```

The server runs a brief check on the channel's setting and decides whether you may join it—usually you can for a public channel. If the server accepts your **JOIN** command, you'll see code resembling this:

```
*** mortuza (mortuza@mtz.santahar.net) has joined channel
#santahar
*** Users on #santahar: @mortuza rupa @tom +majedin diamond
+inclusive @alo kuddus2 hasina +sadia
```

```
*** Topic for #santahar: Hamra santahar bashi
```

This shows you the nickname of all the channel's user, which now include you, and the nickname and the host mask of user who just joined the channel—you again. Note that the at (@) sign before three of the nicknames is not part of the nick, but indicates channel operator status. And the plus (+) sign before three of the nicknames indicates voice status.

8.3 Channel operators

Channel operators are users with certain privileges on a channel. They control the channel by using a special set of commands that can change the channel's settings or modes and topic, invite people to the channel, or remove unwanted users. They are usually called **ops** for short—other terms are **chanops** or **chops**.

There are three ways of obtaining channel op status. The first is being the first user to join a channel, in which case the server automatically gives you ops. The second is getting assigned op status by a user who already is an op or by identifying yourself to a channel server that will check whether you are an authorized user and, if so, assign you op status. The third method, possible only on networks with a channel service, involves indentifying yourself to the service with a special password.

9 How Do I Communicate?

Now that you've successfully made a connection to an IRC server and joined a channel, you're likely to see all sorts of different messages, strange and plain, scroll down your screen. It's quite all right to remain silent for a while and simply follow the conversation. Types of messages you may receive:

Public messages: These are all probably the first messages you'll see, and they're what IRC is all about. Any user can send a message to the channel if the channel's setting permit it. A public message appears in your main screen or channel window like this:

```
<+rupa> Hello, mortuza :)
```

Private messages: It's possible to send a message to a single user or selected group of users without having to join a separate channel. The recipient of the message will see something like this:

```
*rupa* Hi,this is a private message that only you can see
```

Notices: You can send a **notice** to a channel, single user, or selected group, just as with private messages, but they appear different. Generally not used in personal communication, they appear in either your status window or the main window, with the following format:

```
-majedin- Thank you mortuza, for joining us
```

Actions: You'll often see someone on a channel "doing" something. A leading star characterizes this action, followed by a third-person description of the user's action or an emotional state he or she would like to describe in this manner. You can also use actions

in private messages, where they'll appear in a query window or with an asterisk-angle braket (*>) prefix to indicate the action is sent privately and not the channel.

```
* mortuza is eating rice
```

9.1 Channel modes

This is the most complex part of channel management and something a good channel op knows by heart. Let's take a look at the possible modes a channel may have. Depending on the type of server, there may be a few additional ones particular to that type and of varying usefulness. We'll concentrate on the modes available on all kinds of servers—those mentioned in "RFC[5] 1459," the document describing the IRC protocol.

You set or unset channel modes by prefixing their characteristic letter with a plus (+) or minus (-) sign, respectively. If you provide no prefix, the server assumes you mean plus. The characteristic letters of the modes are case sensitive—they must be lowercase. The general syntax of the **MODE** command is the following:

```
/mode #channel <[+|-]letter> [parameter]
```

If your channel supports the asterisk (*) as a substitute for the channel's name, you can also use this with **MODE**.

Mode b (ban): Mode **b** sets a ban forbidding any user to join the channel whose mask matches the string you added as a parameter if added and lifts the ban on the mask if removed.

Mode i (invite only) and I (invitation): Mode **i** is invite only mode. If a channel is **i**, only users invited by a channel operator may join.

```
/mode #santahar +i
```

Mode **I** takes a user mask as its parameter, in the same way mode **b** does.

Mode k (key): To set a password you add the **k** mode with password as its only parameter. If you wish to remove this mode, you must use the exact password in the same way with the **-k** mode change.

```
/mode #santahar +k iamgreat
```

Mode l (limit): This limits the possible number of users on the channel to the number given as a parameter to the **l** mode change. The number is not necessary to remove the **l** mode.

```
/mode #santahar +l 45
```

Mode m (moderated): You can set a channel to allow only its ops and certain other users (those a channel operator gives a voice) to send to it, while the rest can only listen. Mode **+m** adds this setting and **-m** removes it, allowing everyone to send.

```
/mode #santahar +m
```

Mode n (nonexternal): Mode **n** is one of the two modes practically no channel lacks. It disallows messages to the channel from any user who hasn't actually "joined" it.

[5] https://www.rfc-editor.org/rfc/rfc1459.html

Mode o (operator): Probably the most widely used mode is **o**. This assigns or removes ops from a user. The parameter to this mode, whether set or unset, is always user's nickname.

```
/mode #santahar +o majedin
```

Mode p (private): Mode **p** is rarely used but still possible on all server versions. It doesn't permit a channel's name to show on the channel list, but returns other information a use may request.

```
/mode #santahar +p
```

Mode s (secret): Mode **s** is a common setting that makes a channel return no information about its users and not appear on the channel list. A secret channel doesn't appear in its users' **WHOIS** info either.

```
/mode #santahar +s
```

Mode t (topic set by ops): Set mode **t** to prevent users who aren't channel ops from changing the channels topic. This along, with mode **n**, is the most common mode.

```
/mode #santahar +t
```

Mode v (voice): Mode **v** is unnecessary without mode **+m**. It takes the nickname of the user being given or taken away the voice as a parameter.

```
/mode #santahar +v rupa
```

9.2 File transfer

In addition to talking, IRC has also become a popular and convenient way to exchange a wide variety of files. Be forewarned, however, that many people are getting into serious trouble by downloading files that seem interesting or enticing, only to find out they are trojan horse attacks. These hacks allow strangers to take over your channels force you to disconnect, erase your hard disk or worse. The moral is clear: **Never accept candy from strangers**.

9.2.1 DCC send and get

DCC stands for Direct Client Communication, where you and your friend's client programs connect directly to each other, bypassing IRC servers and their occasional "lag" or "split" problems. Like **/msg**, the DCC chat is completely private.

```
/dcc chat username
```

DCC file transfer requires an exchange of commands between the sender and getter of each file. For example, if you as "YourNick" want to send the file "hello.jpg" to your friend "buddy", you would type:

```
/dcc send buddy hello.jpg
```

To get the file:

```
/dcc get YourNick
```

10 Conclusion

This article has provided a foundation in IRC basics, but there's much more to discover in the world of Internet Relay Chat. To continue your IRC journey, consider exploring:

- CTCP and DCC commands
- IRC bot programming and automation
- Network-specific services and features like NickServ and ChanServ
- IRC protocol specification (RFC 1459)
- Secure IRC connections using SSL/TLS
- IRC client customization and scripting

See you online! :-)