

Chat Network

Documentation for our hybrid IRC / XMPP / Matrix chat system.

- [Introduction to our IRC network](#)
- [IRC Clients](#)
- [Common IRC commands](#)
- [Moderation](#)
- [Channel bot](#)
- [Gateways for XMPP and Matrix](#)
- [Bridges to other chat systems](#)

Introduction to our IRC network



Our IRC network can be accessed with an IRC client on `irc.f-hub.org` via port `6697` (SSL only). It runs on the modern IRCv3 capable IRCd [Ergo.chat](#). It therefore comes with built in history and multi-client connections out of the box and no additional [bouncer](#) is required. We also fully integrate with XMPP and Matrix via our own [gateways](#) that allow federated access to any IRC channel.

Registering accounts

Accounts from F-hub.org are automatically registered on our IRC network and can be used to authenticate via SASL login. In addition it is possible to register independent accounts through NICKSERV like on other IRC networks. Please contact us in those rare cases where there is a conflict with an existing user account name.

Bouncer functionality

Once registered you can connect with multiple clients to the same account like common with IRC bouncers. Our Ergo IRCd tries to emulate ZNC to some extent in that regard. By default, persistent connection mode is enabled, but you can opt-out of it via: `/msg NickServ set always-on false`

Channel history

Channel history is automatically logged by the network itself (via IRCv3 history standard) and stored for 30 days. It can be easily accessed by history capable IRC clients or alternatively via

`/history #channel 6h` or similar, please see the [full guide here](#).

IRC tutorial

If you are new to IRC, please refer to our [IRC tutorial that explains basic features and commands](#).

Recommended clients

[IRCv3 capable clients](#) are strongly recommended for accessing our network and we provide easy to use web-clients. You can learn more about our [recommended IRC clients here](#).

Helper bot

We run a helpful little bot that can be invited to any channel via `/invite bot`. Type `@help` to ask the bot for instructions. You can learn more about it on our [dedicated wiki page here](#).

Gateways

Our IRC network also offers [gateway access from federated XMPP and Matrix accounts](#). Please note that due to the federated nature of these systems chat-logs may be stored for a longer period on connected third party servers and the gateways themselves.

Connecting from Tor

Due to repeated abuse we automatically filter all connection attempts against IP addresses of Tor exit nodes and other known abusive source IPs via automated real-time block lists. As a result many Tor exit node IPs are blocked and the remaining ones require SASL account authentication on connection.

IRC Clients

[IRCV3 capable clients](#) are strongly recommended for accessing our network, but any IRC client that supports direct SSL connection and SASL auth is supported.

Mobile clients

For Android, [Goguma](#) is well supported and recommended. It can be installed easily for free via [F-Droid](#).

Goguma can also be [found on the Apple store for iOS](#).

Please let us know if you have other good suggestions.

Web-clients

We offer unauthenticated access via an easy to use [KiwiIRC webclient](#) that also supports IRCv3 history. This is ideal if you want to share a link to a channel somewhere externally like this:

```
https://irc.f-hub.org/kiwiirc/#support
```

You can also use our [Gamja client](#) for the more authentic IRC experience.

Desktop clients

Sadly there are not a lot of desktop IRC clients that support the advanced IRCv3 features already. But if the client supports ZNC bouncers, that gives at least partial functionality.

For TUI clients [Senpai](#) is a good choice as it supports many modern IRCv3 features.

XMPP and Matrix clients

Since our IRC network offers [first-party gateways for XMPP and Matrix](#), you can also use any client for these federated networks to connect with your external account from another server. For XMPP we recommend using the cross-platform desktop client [Gajim](#).

Common IRC commands

Ergo IRCd

Our IRC network runs on the modern IRCv3 capable IRCd [Ergo.chat](https://ergo.chat). It therefore comes with additional built in features and network services that can be accessed via in client commands.

Quick cheat sheet

Accounts

Note: The preferred methode for account creation is to [apply for an account](#) on F-hub.org.

Register a legacy account (for multi-client use) by sending this on IRC as an guest:

```
/msg NickServ register mySecretPassword validEmailAddress@example.com
```

Afterwards, you will get an automated email with further instructions on how to verify this account.

Note: Only SASL auth is supported and you can not change your nickname after registering it.

Delete an account with (follow instructions): `/msg NickServ unregister *yournick*`

Turn off persistent connection mode (bouncer): `/msg NickServ set always-on false`

Play back channel history on legacy clients: `/history #channel 12h (time duration or message count)`

Or set it to autoreply the latest messages: `/msg NickServ set autoreplay-lines 25`

Channels

Register a channel: `/msg ChanServ register #myChannel`

Set channel mode: `/mode #channel +/-attribute`

`+p` - Sets the channel as invisible in `/list`.

`+m` - Mute users who are not opped/hopped/voiced.

`+R` - Only registered nicks are allowed in.

`+M` - Only registered nicks are allowed to talk.

Kick user from channel: `kick #channel nickname [reason]`

Set user mode in current channel: `/mode #channel +/-attributes nickname`

`+o` - User is an operator (prefix @).

`+h` - User is a half-op (prefix %).

`+v` - User has is unmuted (prefix +).

Help

Help messages for service:

`/msg NickServ HELP`

`/msg ChanServ HELP`

You can use `/NS` or `/CS` instead.

Other useful commands:

`/join #channel` - Joins the specified channel.

`/part #channel` - Leave the specified channel.

`/nick nickname` - Changes your nick (unless SASL authed).

`/names #channel` - Shows the nicks of all users on #channel.

`/msg nickname message` - Sends a private message to a user.

`/me action` - Prints "yourname action"

Moderation

This page provides guidance for channel owners and operators on how to deal with spam and abuse. If you are interested in general guidance on what moderation to expect and is expected from you, please refer to our Code of Conduct.

First you need to be channel owner or operator. When you first create a new channel make sure to immediately register it so that it recognizes you as the channel owner in the future. If you for some reason forgot this or lost operator rights please talk to us on the #support channel.

Banning users

Typically a `/kick usernick` will be sufficient to send a message to disruptive users, but they are free to join again.

For more permanent measures you can add a ban mask to the channel with the `/mode +b` command. `/mode +b nick!*@*` would for example ban the user with this nick, although that would of course not be very efficient as changing the nick is very easy.

Masks have the format `nick!user@host`, where `nick` is a nickname, `user` is a username/account, and `host` is a hostmask. The hostmask can be: an IP address, a domain name or a cloak. Masks take wildcards. You can use the `*` wildcard to represent strings of zero or more characters.

Users can be muted rather than banned by prefixing a ban mask with `m:`. For example: `/mode +b m:nick!*@*`.

IP addresses are not visible to non-admins, but are tracked internally based on an obfuscated user ID or so called cloak by default. Please note that sometimes IP addresses are shared or the spammers use tor exit nodes, so banning on IP might cause collateral damage to other legitimate users.

Channel modes

Please note that some of these modes differ from what is commonly in use on other networks or are Ergo.chat specific.

`/mode #channelname +m` will make a channel moderated, meaning everyone is muted by default and needs to request permission to speak. This can be temporarily done via `/mode #channelname +v usernick`. Registered users can be given permanent permission to speak ("voice") via `/amode #channelname +v usernick`. There is also a `+M` mode that does this for registered users automatically, but sadly `+M` isn't currently understood by the Matrix bridge, resulting in the unfortunate situation that spam messages remain possible and visible on the Matrix side (the Matrix bridge does understand regular `+m` though).

`/mode #channelname +R` will only allow registered users to enter the channel. Note that already joined non-registered users need to be kicked for this to have an effect on them. Non-authenticated bridged users will be prevented from sending messages.

`/mode #channelname +i` turns the channel into invite-only mode. Users can be invited with `/invite usernick`. Setting `+I` with a mask can exempt for example bridged users from this restriction.

If you temporarily restrict access to a channel, it can make sense to redirect everyone else to another channel by setting `/mode #channelname +f #support` to redirect users to the support channel or some other channel.

`/mode #channelname +U` messages from unprivileged clients are sent only to channel operators.

`/mode #channelname +u` will hide system messages from unvoiced clients. This can be for example useful when someone uses nick changes to spam a moderated channel.

All channel modes can be reversed by using the above commands, but with a `-` instead of a `+`.

Use for example `/mode #channelname -m` will remove voice restrictions for all channel participants.

Additional moderators

You can permanently add additional moderators to a channel you registered like this: `/msg ChanServ AMODE #example +o bob`.

It is also possible to transfer ownership of a registered channel `/msg ChanServ transfer bob`. It will ask for adding a transfer code and the receiving user must accept the transfer. Just follow the on-screen instructions.

If you lost ownership of an unregistered channel, please contact the administrators on the `#support` channel.

We are also looking for volunteer network wide moderators. Please let us know if you are interested in that.

Moderation bot

For this to work you need to add the `bot` account as a moderator to your channel (see above).

More details on a how to configure the channel bot to help with moderation (flood protection for example) will be added here soon.

Network wide options

Network administrators also have some options to deal with network-wide spam and abuse. Please get in contact with us if you think this is an required response.

`/SAMODE` can be used to grant or remove channel privileges. For example, to create an operator in a channel that has no operators: `/SAMODE #channel +o nickname`.

`/SAJOIN` lets operators join channels despite restrictions, or forcibly join another user to a channel. For example, `/SAJOIN #channel` or `/SAJOIN nickname #channel`.

`/uban add hostmask` will permanently ban an IP from the network. You can add a reason and an expiry time or restrict the IP to SASL authenticated connections.

`/whowas nickname` can help with identifying IPs for users that already left but are likely to return.

The `/defcon 1-5` command can be a quick fix for spam and abuse in multiple channels, but should be used only in extreme cases.

Channel bot

We provide an advanced IRC channel bot that provides many small usability improvements for IRC users and channel admins. The bot is based on [Limnoria](#) (formerly known as Supybot) and supports a wide range of modules.

Features for users

(this is a stub, check back later for more details)

The bot responds if its name is added to the front of a message like this `bot` or to certain commands starting with "@" like `@help`. You can also `/msg` the bot privately for registering your IRC nick with the bot (more on this further below).

The bot will join a channel by inviting it via `/invite bot`. Once it joined the channel, basic commands are available to all users. You can try out most of the commands on our `#botplayground` channel. Please do not spam other channels with this :)

Other features of the bot are for example to [send notes to unregistered users](#), create [polls](#) or [schedule reminders and tasks](#).

Features for channel operators

(this is work in progress, check back later)

Limnoria also offers a range of features to channel operators such as automatic link previews, automatic RSS/ATOM feeds etc.

Once you invited the the Limnoria channel bot via `/invite bot` to your channel it will automatically start previewing URLs. It is possible to disable the automatic link previews in a channel via `bot config channel #channel supybot.plugins.Web.titleSnarfer False`. For more details on this feature see the [official Limnoria documentation](#).

RSS or ATOM feeds can be added via the [RSS plugin](#). Add a feed using `bot rss add somefeedname https://example.com/feed.xml`. Where `somefeedname` is an internal arbitrary name for the RSS source you can chose freely. To have new news automatically announced on the channel, use `bot rss announce add somefeedname`.

Git forge integration

Limnoria can also be integrated with git-forge webhooks for automatically posting about new issues and PRs on your project's Github or Forgejo instance.

Please check back later for how to configure this.

Slack webhook integration

Via Limnoria's [Grapnel plugin](#) channels can be integrated with Slack like webhooks. You can add a channel specific webhook endpoint via `bot grapnel add` which will make the bot send you the secret webhook URL in a private message. You can replace `your-cool-app-name` at the end of the URL with the name of the sending app; it will be then reflected in the output notice by the bot.

Auto-moderation features

(this is a stub, check back later for more details)

Limnoria can help protect against channel flooding and nick spam with some built in features and the [AttackProtector plugin](#). We are also looking into how compatible the [Sigyn extension](#) is with our IRC network.

Other features

Fediverse integration

(this is a stub, check back later for more details)

Limnoria allows interacting with ActivityPub compatible social media websites.

Authentication

(this is a stub, check back later for more details)

For some of the more advanced features you need to first register your IRC nick with the bot.

Once you are registered with the bot you can ask one of our network operators to get elevated permissions for further bot configuration.

Gateways for XMPP and Matrix

Built-in gateways

The main IRC network on F-hub.org provides gateways (sometimes called portals) to allow connections from XMPP and Matrix accounts. For this we host our own instances of [Biboumi \(XMPP\)](#) and [Appservice-IRC \(Matrix\)](#). All IRC channels are automatically available via these gateways and should work almost like native XMPP or Matrix channels.

How to connect from XMPP

IRC Channels can be accessed like regular XMPP MUCs via `#channel@irc.f-hub.org`. Please be assured that despite the warning in your XMPP client, our IRC network does not expose your JID on public channels. For a permanent connection (IRC bouncer like) double check that the "persistence" channel setting is enabled or configure it via an ad-hoc command.

It is possible to authenticate via this bridge with your IRC SASL account through ad-hoc commands, learn more about the [gateway usage here](#).

If you are for example using the Cheogram mobile client you can authenticate yourself this way: First add the "irc.f-hub.org" gateway as a contact. If you open the chat there are two tabs at the top. Click on "Commands" and select "Configure a few settings for that IRC server" from the menu. Then you can specify your IRC username and SASL password and complete it by pressing Finish at the bottom.

You can create easy to share invite-links for XMPP [here](#).

How to connect from Matrix

Room addresses syntax is something like `#irc_#channel:matrix.f-hub.org`. This makes it possible to connect with external Matrix accounts. It is possible to authenticate via this bridge with your IRC

SASL account, learn more about the [portal usage here](#).

After setting up a user account on the IRC server, you can login to your registered irc account through the Matrix bridge. This is easiest to do if you haven't yet connected to the IRC server through the bridge bot. The commands for the Matrix bridge bot (`@appservice-irc:matrix.f-hub.org`) are:

`!storepass (password)` and `!username (username)` and then `!reconnect`

The simplest way to join an irc channel through the bridge service is to message the Matrix bridge bot: `!join [irc.example.net] #channel [key]` or you can use invite-links for Matrix created [here](#).

Privacy notice

Please be aware that due to the federated nature of these services chat-logs and user names will become available to 3rd party services and may be retained indefinitely by them. No sensitive internal user-data from F-hub.org is shared however.

Bridges to other chat systems

We provide chat bridges/gateways to other networks, but please be aware that they are often in breach of those services' terms and various privacy caveats apply.

Matterbridge bot bridge

We run a [Matterbridge](#) instance that allows linking channels through a single bot account. Due to the RELAYMSG support of our IRC network, user-names are displayed nicely on the IRC side, but overall the bridging quality is not comparable to other options below. The bridge needs to be set up manually, so please get in contact with us if you want a specific channel bridged to an external service this way. Due to the complexity involved, we can not offer all options theoretically provided by Matterbridge, but services such as Discord, Mumble, Telegram, Mattermost, Rocket-Chat and many others should work. Depending on the external network this can sometimes require admin permissions on the remote channel.

Advanced Discord bridge

While we do not recommend using Discord, we are aware that many FOSS communities have channels there. Luckily there is a very nice bridge available to link Discord channels to IRC called [Dibridge](#). We have been running several instances of this bridge for a while now and the usability is quite nice from both sides of the bridge. Please get in contact with us if you want a Discord channel bridged this way. You will need to have admin rights for the Discord server to configure the bridge with us. Please note that Discord bots and other Discord bridges are currently not relayed to IRC by this bridge.

External XMPP channels

We are still investigating how to best bridge external XMPP channels, although basic bot bridging via Matterbridge is already possible. Please get in contact with us if you have recommendation or would like such a channel bridged.

Room plumbing with Matrix bridge

If you have an existing Matrix room you would like to plumb to an IRC channel on our network, please get in contact with us. Usually it is sufficient to invite the bridge-bot (appservice-irc:matrix.f-hub.org) to the channel and then we can plumb it to our IRC network.

Matrix double-bridging

Due to certain technical considerations it is against the ToS of some other IRC networks like libera.chat to directly plumb Matrix rooms that are already bridged with our appservice-irc bridge/gateway. As a work-around we can bridge such rooms with Matterbridge, but the usability will be somewhat degraded. Please get in contact with us if you need such bridging and we can discuss the details.

Built-in gateways for XMPP and Matrix

The main IRC network on F-hub.org also provides gateways (sometimes called portals) to allow connections from XMPP and Matrix accounts. All our IRC channels are automatically available via these gateways and should work almost like native XMPP or Matrix channels. Learn more about this [here](#).