
RITlug TeleIRC Documentation

Release 1.2.2

Mark Repka, Seth Hendrick, Nate Levesque, Robby O'Connor, Jus

Aug 01, 2019

Contents

1	Quick install	3
1.1	Create a Telegram bot	3
1.2	Configure IRC channel	4
1.3	Configure and run TeleIRC	4
2	How to deploy TeleIRC	7
2.1	systemd	7
2.2	pm2	7
2.3	Arch Linux AUR	7
2.4	Docker	8
3	Config file glossary	9
3.1	IRC settings	9
3.2	Telegram settings	10
3.3	Imgur settings	10
3.4	Miscellaneous settings	10
4	Frequently asked questions	11
4.1	Telegram	11
5	Who uses TeleIRC?	13
5.1	How to get on this list	14
6	Live demo	15
7	Indices and tables	17

RITlug TeleIRC is a NodeJS implementation of a [Telegram](#) <=> [IRC](#) bridge. TeleIRC works with any IRC channel and Telegram group. It bridges messages between a Telegram group and an IRC channel.

This bot was originally written for [RITlug](#). Today, it is used by various communities.

This is a quick installation guide for an administrator to configure and deploy TeleIRC. TeleIRC configuration is divided into these steps:

1. Create a Telegram bot
2. Configure IRC channel
3. Configure and run TeleIRC

1.1 Create a Telegram bot

Note: TeleIRC **DOES NOT** support channels, only groups. Read more about channels vs groups [here](#).

Create a new Telegram bot to act as a bridge from the Telegram side. The bot API provides a token key for the bot. Use the bot to discover the unique chat ID of your Telegram group.

1.1.1 Create bot with BotFather

1. Start new message to `@BotFather` user on Telegram
2. Send `/start` to `@BotFather`¹
3. Follow instructions to create new bot (e.g. name, username, description, etc.)
4. Receive **token key** for new bot (used to access Telegram API)
5. **(REQUIRED)** Set `/setprivacy` to **DISABLED** (so bot can see messages)²

¹ `@BotFather` is the Telegram bot for creating Telegram bots

² Privacy setting must be disabled for TeleIRC bot to see messages in the Telegram group. By default, bots cannot see messages unless a person uses a command to interact with the bot. Since TeleIRC forwards all messages, it needs to see all messages. Messages are not stored or tracked by TeleIRC.

6. Add bot to Telegram group you plan to bridge

1.1.2 Optional configuration changes

1. Set description or profile picture for your bot with @BotFather
2. Block your bot from being added to more groups (`/set joingroups`)

1.2 Configure IRC channel

There is no required configuration for an IRC channel. However, there are recommendations for best practices:

1. Register your channel
2. Give permanent voice to your bridge bot via ChanServ (most networks use the +V flag)
 - *Example:* On freenode, `/query ChanServ ACCESS #channel ADD my-teleirc-bot +V`

1.3 Configure and run TeleIRC

This section explains how to configure and install TeleIRC itself.

1.3.1 Requirements

- git
- nodejs (v8 and v10 supported)
- yarn

1.3.2 Install dependencies

1. Clone the repository (`git clone https://github.com/RITlug/teleirc.git`)
2. Install dependencies (`yarn`)

1.3.3 Configuration

TeleIRC uses `dotenv` to manage API keys and settings. The config file you use is a `.env` file. Copy the example file to a production file to get started (`cp env.example .env`). Edit the `.env` file with your API keys and settings.

See also:

See *Config file glossary* for detailed information.

Relay Telegram picture messages via Imgur

TeleIRC retrieves picture messages via the Telegram API. By default, picture messages from Telegram are sent to IRC through Imgur. See [context](#) for why Imgur is enabled by default.

Note: By default, TeleIRC uses the generic Imgur API key. Imgur highly recommends registering each TeleIRC bot.

To add Imgur support, follow these steps:

1. Create an Imgur account
2. **Register your bot with the Imgur API**
 - Select *OAuth2 without callback* option
3. Put client ID into `.env` file

How to deploy TeleIRC

There are several ways to deploy TeleIRC persistently. This page offers suggestions on possible deployment options.

2.1 systemd

The **recommended deployment method** is with [systemd](#). This method requires a basic understanding of systemd unit files. Create a unique systemd service for each TeleIRC instance.

A [provided systemd service](#) file is available. Add the systemd unit file to `/usr/lib/systemd/system/` to activate it. Now, `systemctl` can be used to control your TeleIRC instance.

Note the provided file makes two assumptions:

- Using a dedicated system user (e.g. `teleirc`)
- TeleIRC config files located at `/usr/lib64/teleirc/` (i.e. files inside TeleIRC repository)

2.2 pm2

`pm2` keeps NodeJS running in the background. If you run an application and it crashes, `pm2` restarts the process. `pm2` also restarts processes if the server reboots. Read the [pm2 documentation](#) for more information.

After `pm2` is installed, run these commands to start TeleIRC:

```
cd teleirc/  
pm2 start -n my-teleirc-bot teleirc.js
```

2.3 Arch Linux AUR

On ArchLinux, see [teleirc-git](#) in the AUR. The AUR package uses the `systemd` method to deploy TeleIRC. Place TeleIRC configuration files in the `/usr/lib/teleirc/` directory.

2.4 Docker

Docker is another way to deploy TeleIRC. Dockerfiles and images are available in `images/`.

2.4.1 Which image do I choose?

Node Alpine Linux and Fedora images are provided.

Image	File	Size
Node Alpine Linux (node:10-alpine)	Dockerfile.alpine	374 MB
Fedora latest	Dockerfile.fedora	569 MB

This guide uses alpine. If you wish to use fedora, replace alpine with fedora.

2.4.2 Building Docker image

You may see errors running yarn. You can safely ignore them. They are not fatal.

```
docker build . -f images/Dockerfile.alpine -t teleirc
docker run -d -u teleirc --name teleirc --restart always \
  -e TELEIRC_TOKEN="00000000:AAAAAaAAa2AaAAaoAAAA-a_aaAAaAaaaAA" \
  -e IRC_CHANNEL="#channel" \
  -e IRC_BOT_NAME="teleirc" \
  -e IRC_BLACKLIST="CowSayBot,AnotherNickToIgnore" \
  -e TELEGRAM_CHAT_ID="-00000000000000" \
  teleirc
```

2.4.3 Docker Compose

Optionally, you may use `docker-compose`. An example compose file is provided.

Run these commands to use Docker Compose:

```
cp images/docker-compose.yml.example docker-compose.yml
cp env.example .env
docker-compose up -d teleirc
```

Config file glossary

This page is a glossary of different settings in the `env.example` configuration file. The values shown for the settings are their defaults. This glossary is intended for advanced users.

3.1 IRC settings

IRC_BLACKLIST="" Comma-separated list of IRC nicks to ignore

IRC_BOT_NAME=teleirc IRC nickname for bot

IRC_CHANNEL=#channel IRC channel for bot to join

IRC_SEND_STICKER_EMOJI=true Send emojis associated with a sticker to IRC (when a Telegram user sends a sticker)

IRC_SEND_DOCUMENT=false Send documents and files from Telegram to IRC ([why is this false by default?](#))

IRC_PREFIX="<" Text displayed before Telegram name in IRC

IRC_SUFFIX=">" Text displayed after Telegram name in IRC

IRC_SERVER=chat.freenode.net IRC server to connect to

IRC_NICKSERV_SERVICE=NickServ IRC service used for authentication

IRC_NICKSERV_PASS="" IRC account password to complete IRC authentication

IRC_EDITED_PREFIX="[EDIT] " Prefix to include when a user edits a Telegram message and it is resent to IRC

IRC_MAX_MESSAGE_LENGTH=400 Maximum length of the message that can be sent to IRC. Longer messages will be split into multiple messages.

3.2 Telegram settings

TELEGRAM_CHAT_ID=-000000000000 Telegram chat ID of bridged group (how do I get this?)

TELEIRC_TOKEN=000000000:AAAAAAaAAa2AaAAaoAAAA-a_aaAAaAaaaAA Private API token for Telegram bot

MAX_MESSAGES_PER_MINUTE=20 Maximum number of messages sent to Telegram from IRC per minute

SHOW_ACTION_MESSAGE=true Relay action messages (e.g. /me thinks TeleIRC is cool!)

SHOW_JOIN_MESSAGE=false Send Telegram message when someone joins IRC channel

SHOW_KICK_MESSAGE=true Send Telegram message when someone is kicked from IRC channel

SHOW_LEAVE_MESSAGE=false Send Telegram message when someone leaves IRC channel

3.3 Imgur settings

USE_IMGUR_FOR_IMAGES=true Upload picture messages from Telegram to Imgur, send Imgur link to IRC

IMGUR_CLIENT_ID=0000000000 Imgur API client ID value to access Imgur API

3.4 Miscellaneous settings

NTBA_FIX_319=1 Required to fix a bug in a library used by TeleIRC. For context, see [yagop/node-telegram-bot-api#319](#).

Frequently asked questions

This page collects frequently asked scenarios or problems with TeleIRC. Did you find something confusing? Please let us know in our developer chat or open a new pull request with a suggestion!

4.1 Telegram

How do I find a chat ID for a Telegram group? The chat ID is found when viewing results from the Telegram API from a web browser. First, add your bot to the group. Then, open a browser and enter the Telegram API URL with your API token, as explained in [this post](#). Next, send a message in the group with the bot username and refresh the browser window. You will see the chat ID for the Telegram group along with other information.

I reinstalled TeleIRC after it was inactive for a while. But the bot doesn't work. Why? If a Telegram bot is not used for a while, it “goes to sleep”. Even if TeleIRC is configured and installed correctly, you need to “wake up” the bot. To fix this, *remove the bot from the group and add it again*. Restart TeleIRC and it should work again.

Who uses TeleIRC?

The following list of projects and communities use RITlug TeleIRC:

- BSD Argentina (@bsdar | #bsdar)
- CityZen app (@CityZenApp | #cityzen)
- **Fedora Project (@fedora | #fedora-telegram)**
 - Fedora Albania (@FedoraAlbania | #fedora-sq)
 - Fedora CommOps (@fedoracommops | #fedora-commops)
 - Fedora Diversity and Inclusion Team (@fedoradiversity | #fedora-diversity)
 - Fedora LATAM (@federalat | #fedora-latam)
 - Fedora Marketing Team (@fedoramktg | #fedora-mktg)
 - Flock to Fedora (@flocktofedora | #fedora-flock)
- FOSS@MAGIC at RIT (@fossrit | #rit-foss)
- Hamara Linux (@hamaraLinux | #hamara)
- **LibreOffice Community (@libreofficecommunity | #libreoffice-telegram)**
 - LibreLadies (*Telegram invite-only* | #libreladies)
 - LibreOffice AppImage (*no public invite link* | #libreoffice-appimage)
 - LibreOffice Design Team (*no public invite link* | #libreoffice-design)
 - LibreOffice QA Team (*no public invite link* | #libreoffice-qa)
- Linux Users Massa Carrara (Telegram | IRC)
- MINECON Agents (*no public invite link* | #MineconAgents)
- MusicBrainz (@musicbrainz | #musicbrainz-telegram)
- Pure Data (@puredata | #dataflow)
- **RITlug (@ritlug | #ritlug)**

– RITlug teleirc (@teleirc | #ritlug-teleirc)

5.1 How to get on this list

Want to have your community added to this page? Let us know you're using TeleIRC too! [Submit an issue](#) against this repo with the following info:

- Organization / group name and website
- Telegram group URL
- Your IRC channel

To be added, your group must not discuss illegal, illicit, or generally inappropriate content.

CHAPTER 6

Live demo

A public Telegram supergroup and IRC channel (on Freenode) are available for testing. Our developer community is found in these channels.

- Telegram: [@teleirc](#)
- IRC: [#ritlug-teleirc](#) (chat.freenode.net)

CHAPTER 7

Indices and tables

- github.com/RITlug/teleirc
- `genindex`
- `modindex`
- `search`