WeeChat quick start guide

Sébastien Helleu – flashcode@flashtux.org – version 3.8

Table of Contents

1. Start WeeChat

A recommended (but not mandatory) X terminal emulator for running WeeChat is rxvt-unicode. It has good UTF-8 support and works well with WeeChat's default key bindings.

Run from your shell:

```
$ weechat
```

§ 2. Online help

WeeChat has help for all commands, just issue:

```
/help
```

To get help on a specific command, issue:

```
/help command
```

Help is available for options as well:

```
/help config.section.option
```

(where `config` is configuration name (`weechat` for core, or a plugin name), `section` the section of this configuration and `option` the option name).

3. Set options

To set options, issue:

```
/set config.section.option value
```

WeeChat immediately uses the new value (you **never** need to restart WeeChat after changes to configuration).

All settings are saved when WeeChat ends (or with the `/save` command to force a write of the options).

> It is **not recommended** to edit configuration files by hand because WeeChat may write them at any time (for example on `/quit`) and after any change you must run the command `/reload` (with the risk of losing other changes that were not yet saved with `/save`).
> You can use the command `/set`, which checks the value and applies immediately the changes.

The plugin *fset* allows you to easily browse options and change them.

For example to display WeeChat options:

```
/fset weechat.*
```

IRC options:

```
/fset irc.*
```

The `/fset` command has completion on part of option names, so for example if you type `/fset hot` and press `Tab` this is completed as `/fset hotlist`. If you press `Enter`, options about the hotlist are displayed.

For more information about the `/fset` command and keys, see `/help fset`.

4. Core vs plugins

WeeChat "core" is only used to display data on screen and interact with the user, that means weechat core without plugins is useless (faithful users: IRC was part of core for versions ≤ 0.2.6).

All network protocols like IRC are provided in separate plugins.

Use the `/plugin` command to list loaded plugins, you should see "irc" and other plugins in the list.

5. Add an IRC server

You can add an IRC server with the `/server` command, for example:

```
/server add libera irc.libera.chat/6697 -ssl
```

In this command, `libera` is the internal server name used by WeeChat: you’ll be able to connect with `/connect libera` and the server options are *irc.server.libera.xxx*.

As usual, help is available if you’re lost:

```
/help server
```

6. Set custom IRC server options

WeeChat uses default values for all servers ("fall backs"), if you don’t specify a specific value for a server option. These default options are "irc.server_default.*".

For each server option, WeeChat uses its value if it is defined (not "null"). Otherwise WeeChat uses default value ("irc.server_default.xxx").

For example there are default nicks (based on your un*x login), and you can override them for the libera server with following command:

```
/set irc.server.libera.nicks "mynick,mynick2,mynick3,mynick4,mynick5"
```

To set the user and real names:

```
/set irc.server.libera.username "My user name"
/set irc.server.libera.realname "My real name"
```

To enable auto-connect to server at startup:

```
/set irc.server.libera.autoconnect on
```

If SASL is available on the server, you can use it for authentication (you will be identified before you join channels):

```
/set irc.server.libera.sasl_username "mynick"
/set irc.server.libera.sasl_password "xxxxxxx"
```

To run a command after connection to server, for example to authenticate with nickserv (only if you don’t use SASL for authentication):

```
/set irc.server.libera.command "/msg nickserv identify xxxxxxx"
```

ℹ Many commands in option *command* can be separated by `;` (semi-colon).

If you want to protect your password in configuration files, you can use secured data.

First setup a passphrase:

```
/secure passphrase this is my secret passphrase
```

Then add a secured data with your libera password:

```
/secure set libera_password xxxxxxx
```

Then you can use `${sec.data.libera_password}` instead of your password in the IRC options mentioned above, for example:

```
/set irc.server.libera.sasl_password "${sec.data.libera_password}"
```

To auto-join some channels when connecting to server:

```
/set irc.server.libera.autojoin "#channel1,#channel2"
```

The `/autojoin` command lets you configure the *autojoin* option easily (see `/help autojoin`).

You can also configure WeeChat to automatically update the *autojoin* option when you join or leave channels:

```
/set irc.server_default.autojoin_dynamic on
```

To remove a value of a server option, and use the default value instead, for example to use default nicks (irc.server_default.nicks):

```
/unset irc.server.libera.nicks
```

Other options: you can setup other options with the following command ("xxx" is option name):

```
/set irc.server.libera.xxx value
```

You can complete name and value of options with the Tab key and Shift + Tab for a partial completion (useful for long words like the name of option).

7. Connect to IRC server

```
/connect libera
```

With this command, WeeChat connects to the libera server and auto-joins the channels configured in the "autojoin" server option.

This command can be used to create and connect to a new server without using the `/server` command (see `/help connect`).

By default, server buffers are merged with WeeChat *core* buffer. To switch between the *core* buffer and server buffers, you can use `Ctrl` + `x`.

It is possible to disable auto merge of server buffers to have independent server buffers:

```
/set irc.look.server_buffer independent
```

8. Join/part IRC channels

Join a channel:

```
/join #channel
```

Part a channel (keeping the buffer open):

```
/part [quit message]
```

Close a server, channel or private buffer (`/close` is an alias for `/buffer close`):

```
/close
```

Closing the server buffer will close all channel/private buffers.

Disconnect from server, on the server buffer:

```
/disconnect
```

9. IRC private messages

Open a buffer and send a message to another user (nick *foo*):

```
/query foo this is a message
```

Close the private buffer:

```
/close
```

10. Buffer/window management

A buffer is a component linked to a plugin with a number, a category, and a name. A buffer contains the data displayed on the screen.

A window is a view on a buffer. By default there's only one window displaying one buffer. If you split the screen, you will see many windows with many buffers at same time.

Commands to manage buffers and windows:

```
/buffer
/window
```

For example, to vertically split your screen into a small window (1/3 width), and a large window (2/3), use command:

```
/window splitv 33
```

To remove the split:

```
/window merge
```

11. Key bindings

WeeChat uses many keys by default. All these keys are in the documentation, but you should know at least some vital keys:

- `Alt` + `←` / `Alt` + `→` or `F5` / `F6` : switch to previous/next buffer
- `F1` / `F2` : scroll bar with list of buffers ("buflist")
- `F7` / `F8` : switch to previous/next window (when screen is split)
- `F9` / `F10` : scroll title bar
- `F11` / `F12` : scroll nicklist
- `Tab` : complete text in input bar, like in your shell
- `PgUp` / `PgDn` : scroll text in current buffer
- `Alt` + `a` : jump to buffer with activity (in hotlist)

According to your keyboard and/or your needs, you can rebind any key to a command with the `/key` command. A useful key is `Alt` + `k` to find key codes.

For example, to bind `Alt` + `!` to the command `/buffer close` :

```
/key bind (press alt-k) (press alt-!) /buffer close
```

You’ll have a command line like:

```
/key bind meta-! /buffer close
```

To remove key:

```
/key unbind meta-!
```

12. Plugins/scripts

On some distros like Debian, plugins are available via a separate package (like weechat-plugins). Plugins are automatically loaded when found (please look at the WeeChat documentation to load/unload plugins or scripts).

Many external scripts (from contributors) are available for WeeChat, you can download and install scripts from the repository with the `/script` command, for example:

```
/script install go.py
```

See `/help script` for more info.

A list of scripts is available in WeeChat with `/script` or on this page ↗ (https://weechat.org/scripts/).

13. More documentation

You can now use WeeChat and read FAQ/documentation ↗ (https://weechat.org/doc/) for any other questions.

Enjoy using WeeChat!

Version 3.8