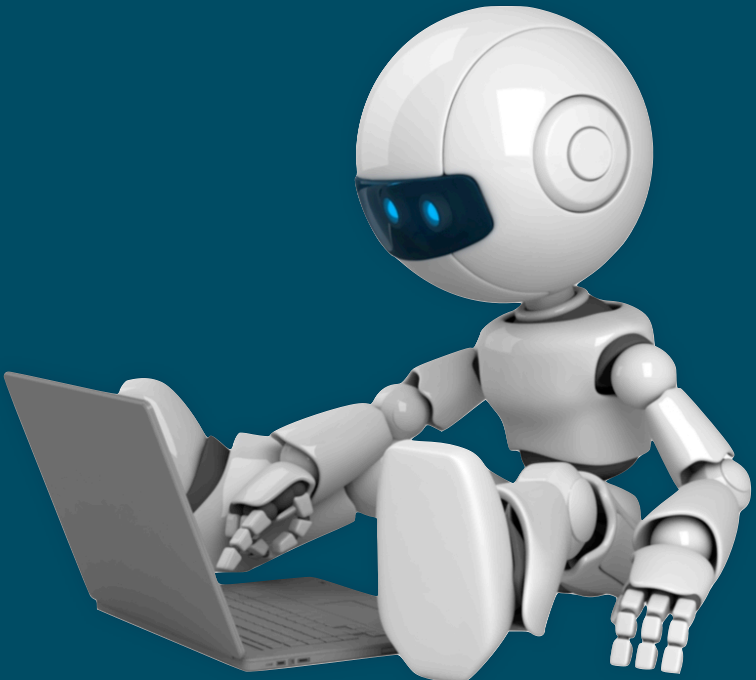
FurCode Press

Features:

- Extensive Preamble
- Source Code
- Fun Facts

Mastering **Snoonet Bots**

Learn about SubBot, MemeBot, and other bots in one book!



1st Edition

Created by, and designed by rod156
Edited by flotwig

Table of Contents

Changelog (for 1st Edition)

7/29/14

I. First revision of “Mastering Snoonet Bots”

Preamble

In this eBook, you will learn the basics of operating various bots available in the Snoonet network. This is also useful as a reference book for operating those bots in the future.

If this is your first time using a bot on Snoonet, we recommend trying out the exercises in a Private Message to the bot, or an empty channel, rather than using it in a busy channel, as you may cause disturbance and trigger spam protection.

Many of the bots mentioned in this book have their source code publicly available. While the author of this book has made sure that the sources for each bot are accurate, the source code may move or change, in which case links and source seen in this book may not be accurate.

Introduction

*Welcome to Mastering **Snoonet Bots**!*

Most of the lessons you will be using in this book have been made for use in an IRC client. If you do not have an IRC client handy, we recommend you get one set up before continuing with this book. In this book, most bots will use the prefix mentioned in each page for command detection, however other methods may also be used to detect commands, such as “Bot Name, <command>” and no prefix in PMs. See the source code of each bot for more on command prefixes. We assume each bot uses their default call prefix for the remainder of this book.

Conventions used in this Book

Within this book, we will use the following features to define different contexts within the text.

`Monospaced Text` on a green or blue background denote commands to use in an actual IRC channel or PM.

Red Underlined Text denotes an variable in a command that should be replaced when actually using it.

Orange Semibold Text denotes an user, if one is used in a command.

Blue Italic Text denotes a constant that can be used in a command, or a location on your filesystem.

`Purple Monospaced Text` denotes a response from a bot.

Book Organization

The book is organized by more popular bots first, then less popular bots and their administrative tools later. Some of the lessons can only be performed by an user with the right permissions, in which case you might have to query an Operator in order to see if such restrictions apply to you.

Getting Started

Registering your account

If this is your first time on the Snoonet network, we recommend you register your account, to make sure that your nick is reserved. Some of the bots in this book also require registering your nick.

If your current nick is unregistered, you can claim it and register it by using :

```
/msg NickServ register <password> <email>
```

Pick an easy to remember password, as well as a valid email.

The email you choose will be used to verify that you are a human, and to finish the registration of your nick. This email will also be used for receiving MemoServ messages.

You will receive the following response:

```
An email containing nickname activation instructions has
been sent to user@example.com.
If you do not complete registration within one day, your
nickname will expire.
<nick> is now registered to user@example.com, with the
password <password>.
```

You must now check the email used for registration in order to continue.

Once you receive your verification email, use the command specified in it to finish the registration.

SubBot (rAnnouncer)

A bot that announces new Reddit posts.

- SubBot allows your channel to integrate with the Reddit community, by announcing new posts from any subreddit within your channel. The announcements are in real time, and works great for Reddit communities connecting with their IRC counterparts.

- Source Code for SubBot can be found on Github at https://github.com/paradox460/subbot

Managing SubBot

After inviting SubBot into your channel, configuring SubBot is simple. To add a new channel to the list that SubBot should announce, use:

```
\subreddit add <subreddit>
```

In the above command, the back-slash tells SubBot that this is a command to detect. The subreddit you input should just be the name, without /r/.

To remove a subreddit you added, use:

```
\subreddit del <subreddit>
```

Note that the subreddit you remove should be spelled exactly as it is called. This command is case-sensitive.

A list of subreddits currently being announced can be obtained with:

```
\subreddit list
```

After configuring SubBot, new announcements should start appearing. These announcements looks somewhat similar to this:

```
/r/all: <purebishop> My Test Post ( http://redd.it/
2c1vhj ) [ self.all ]
```

Each announcement contains the subreddit it comes from, the user who posted it, the name of the post, a link to the post, and finally the type of post.

If you wish to make the bot leave the current channel, use:

```
\part
```

Configuring SubBot

If you are hosting your own SubBot instance, you can also configure the bot's prefix, bot name, channels to join, as well as users allowed to manage the bot. SubBot's configuration can be found in *conf/config.yml* on your server's filesystem.

Note that hosting your own SubBot instance requires a Reddit API account, as well as permission from your network to operate the bot.

Fun Fact: SubBot was written in Ruby.

MemeBot

A generator for memes.

- MemeBot allows you to create a "meme" within IRC. The text used in the meme is inputed as a command, and the results are returned as a link to imgur.

- Source Code for MemeBot can be found on Github, at https://github.com/paradox460/memebot

Creating Memes

After inviting MemeBot into your channel, you can start generating Memes by using:

```
!meme m:<MemeName> <Top Text>;<Bottom Text>
```

In the above command, you pick your meme, then specify the top and bottom text, separated by a colon.

After a few seconds, your meme will be returned, similar to this:

```
User: http://i.imgur.com/gTvYMDq.jpg
```

You can also have MemeBot choose the meme for you, using:

```
!meme <Top Text>;<Bottom Text>
```

This will return a meme using your text, but with a randomly picked background.

To list all the memes available to use within a request, use:

```
!memes
```

Fun Fact: MemeBot was also written in Ruby, in "a single nights bathtub coding session".

ImageBot

A bot for getting random subreddit images.

- ImageBot allows you to get an image from a subreddit's posts. This is great for looking at interesting content.

Retrieving an Image

After inviting ImageBot into your channel, you can get an image from a subreddit by using:

```
!image <subreddit>
```

Fun Fact: ImageBot is based on the Cinch framework.

WeedBot (weeedbot)

A bot by Nekosune.

- WeedBot is a SkyBot-based themed bot, with a sayings module, as well as the ability to generate comics from recent conversation.

- Source Code can be found at https://github.com/nekosune/WeedBot

Generating a Comic

After inviting Weedbot into your channel, you can get create a comic from the current channel by using:

```
!comic
```

If you wish for the comic not to be uploaded (such as in a private channel), you can use:

```
!comicnoup
```

To generate a comic from a Reddit comment thread, use:

```
!comicreddit <url>
```

Fun Fact: Weedbot uses the PIL library to generate comics.

MemoServ

A system to send messages to other users.

- MemoServ can be used to send memos to registered users within the Snoonet network. This is great for leaving notes to offline users, or reaching an user when a Private Message is not appropriate.

- Source Code can be found at https://github.com/atheme/atheme/

Getting Help

MemoServ has a built in Help system; you can trigger it by using:

```
/msg MemoServ help
```

To get more help on a specific command, use:

```
/msg MemoServ help <command>
```

Sending a Memo

Using MemoServ to send a memo is easy, simply use:

```
/msg MemoServ send <user> <message>
```

The memo you send will be stored for the user. When the user comes back online, they will be notified that they have a memo with a message similar to this:

```
You have 1 new memo.
To read them, type /msg MemoServ READ NEW
```

The user will also be notified via their registered NickServ email.

Reading a Memo

If you have received a memo, you can read it by using:

```
/msg MemoServ read new
```

Once you have read the message, the sender will be notified.

Fun Fact: MemoServ is part of Atheme's IRC services.

SnooStat

A bot that generates network, channel, and user statistics at https://stats.snoonet.org/.

● Maintained by flotwig

Getting SnooStat in your channel

SnooStat is a network service, and it can be added to channels by asking an admin in #help.

Using SnooStat in your channel

SnooStat has one channel command:

```
!stats
```

Which will show you your stats in the channel:

```
SnooStat- Channel stats for flotwig on #snoonet
SnooStat- letters: 2717, words: 451, lines: 89, smileys:
3, actions: 1
```

Fun Fact: flotwig created this bot.

Statistics

A bot that generates channel statistics.

- Source code can be found at: https://github.com/flotwig/StatsBot
- The Statistics bot can be used to generate extended statistics information on your channel.

Adding Statistics

To track your channel, invite the bot by using:

```
/invite Statistics
```

Retrieving Statistics

If your channel is already being tracked by the Statistics bot, you can obtain your user statistics by using:

```
!stats
```

in the channel. This will return the latest statistical information for your user on the current channel. The response looks similar to:

```
Stats for this channel can be found at https://
chanstats.snoonet.org/%23snoonet.html
```

This will appear as a notice from the bot, not in the channel.

Web Statistics

A summary page of your channel can also be seen on the web, using the bot's statistics site at https://chanstats.snoonet.org/

You can also retrieve a summary page for a specific channel within IRC by using:

```
!stats <channel>
```

Fun Fact: flotwig created this bot.

##Random

Join a random channel on the Snoonet Network.

- The ##random channel on Snoonet can be used to discover new channels offered on the network. This is great for growing a new community.

- Source code can be found at: https://gist.github.com/flotwig/601049c0b8710e99b0fd

Joining a Random Channel

To join a random channel, use:

```
/join ##random
```

When you join the channel, the bot will send you to another random channel instead, and you will be removed from ##random.

Fun Fact: flotwig also created this bot.

Flags

The part of ChanServ used to promote users.

- Managing flags is an important role for any operator, and helps to develop your IRC community.

Adding Flags to an user

After registering your channel, to promote an user on your channel, use:

```
!flags <user> <flags>
```

There are only three values that can be used for flags, *VOP*, *HOP*, and *AOP*. You will receive a response similar to the following:

```
Operator set flags +AHV on User
```

Removing Flags from an user

If you wish to demote an user, there are multiple methods. To remove every flag from an user, use:

```
!flags <flags> -*
```

in the channel. You will receive a response similar to:

```
Operator set flags -AHV on User
```

To remove only specific flags from an user, use:

```
!flags <flags> -<flags>
```

Note that in this command, the flags notation are the ones used by ChanServ.

Listing Flags

If you wish to see a list of flags, use:

```
!flags
```

ChanServ will send a notice with the list of flags for every user (if an Operator), or just you (if voiced or a half-op).

Fun Fact: These are all shortcut commands, they can be also executed by messaging ChanServ.