Manderlbot, an erlang irc bot

Dimitri Fontaine

February 3, 2009

Contents

Chapter 1

Introduction

This document provides some helpfull information (or I hope so) about the manderlbot irc bot, on how to use and configure it.

A bit of history

But why did we wrote this software ?

Well, I wanted an irc bot to play with, in order to have it say some silly things automatically on answer to our own idioties. I did not want an eggdrop or whatever controlling channel bot. I saw that manderlbot[1] project existing, was already familiar with erlang developpment, so I began using it.

The existing project was on early stage of development, and I wanted the bot to do more and more things. So I wrote some code to make it fit my needs. As the original authors would not consider my patches, I forked the project, keeping the name (they seemed not to work on their version at all), and hosting it on the TuxFamily services[2].

Chapter 2

Using manderlbot

This is the user documentation of manderlbot.

2.1 Install

The better would be to run a debian GNU/Linux system. In that case, a simple `apt-get install manderlbot` get you with all the necessary stuff.

Otherwise, you'll have to get the last available manderlbot release on `manderlbot.tuxfamily.org`, and the necessary tools, that is:

- erlang system, see `erlang.org` or your distribution vendor for some packages

- `xmerl-0.15` library (which will allow for configuration file reading), available as a user contribution on the `erlang.org` web site.

If you are using `windows`, as erlang is known to work even on such a *system*, you should be abble to install and run `manderlbot`. But I did never try that, and do not plan to do it ever. Good luck!

So you have installed erlang system and placed `xmerl` library (version `0.15`) at the right place (on my debian system, it is found in `/usr/lib/erlang/lib/xmerl-0.15`). You now can type the usual following:

```
$ make
$ sudo make install
```

And manderlbot will be ready to be launched!

2.2 Launch and stop

As of the 0.9.1 release of manderlbot, you can control manderlbot using the given `/usr/bin/manderlbot` script:

```
$ manderlbot
manderlbot start|stop|restart|status
$ manderlbot start
```

The stop option will kill the running manderlbot, and if started the status will list all the running bots, indicating the servers and channels where it is connected.

If you want to redirect outputs, you'll then prefer that way:

2.3 Configure

The `manderlbot` configuration file is to be found in `/etc/manderlbot/config.xml`, and has to be edited to fit your needs. By default it contains some basic rules as examples.

2.3.1 Configuration file

The configuration file is an XML file containing the following elements:

manderlbot is the opening XML element of the configuration, and contains the properties `name` and `controler`. The name will be shown as the bot fullname, the controller property may contain one or more nicknames separated by spaces, only those people will then be allowed to operate on the running bot from irc channel.

All the following sections, otherwise stated, are to be found under this one.

dict is the section where to define the dictionnary server you may want to use. See `dict.org` for details about the protocol and servers. Be aware that you can run a dict server locally, and download some useful dictionnaries.

The properties to define here are the dict server `host`, the `port`, and the `default` dictionnary to use.

server allows you to define which servers manderlbot should connect to. The properties are `host` and `port`. You have to define a server section for each and every irc server you want `manderlbot` to connect to.

channel section is where to configure the `manderlbot` behaviour and name. This section has to be embedded in the server one. You have to define a channel section per channel you want `manderlbot` to join on a server.

The properties of channel section are `name`, the channel name, `botname`, the manderlbot nickname on that channel, and `behaviours`, a list of behaviours name you want to activate for that channel.

behaviours will just contain your behaviour list

behaviour have to be found under the `behaviours` section. You define here your behaviour, which properties are `name`, the name to use in the channel definition, the `action`, defining what will be done, and one or more of the followings pattern elements: `pattern`, `op`, `to,`, `option` and `from`. You can even prefix those properties with 'exl_' to get an exclude pattern match (see section 2.3.2).

This element contains data wich will be used as the *action* parameter, as explain in section 2.3.3.

2.3.2 Behaviour matching

So when you define a behaviour, you want `manderlbot` to react on some event on irc channel it is connected to, and take some action. Here we see how to define the event you want it to react to. As on irc all you do is sending lines of text, an event

You can define some *regexp* on the following parts of the received line (containing some server informations relative to IRC protocol[2]):

pattern will try to match the user input, that is what your ordinary irc client will show you

op will try to match the irc operation, see the RFC for complete list (op can be "kick" or "join" for example)

to irc protocol `to` field, will probably contain the channel name, so you won't need that...

option irc protocol `option` field

from the nickname of the one who typed the current line, on the form `nick!~user@host.domain.tld`

And in order to make it even more powerful and readable, you can define the same patterns with an 'exl_' prefix, this will prevent the `action` to being taken if it matches. So you can define the parameters `exl_pattern`, `exl_op`, `exl_to`, `exl_option` and `exl_from`.

Of course, you can use any combination of the listed parameters, thus being quite precise on what you want to react to.

2.3.3 Implemented actions

The `action` parameter of the behaviour configuration element defines the manderlbot behaviour on matching a line. Here is a list of provided actions you can use. If you want `manderlbot` to take an action not described here, you will have to write some erlang code to teach him what you want!

The argument of the `action` is the xml data given enclosed in the `behaviour` element.

action send the given argument as if manderlbot had typed it after the `/me` irc command.

answer send the line prefixed with the sender name and a colon.

bloto this will count the matched lines per user, and first obtaining 5 points has won the business loto game. Just define your buzzwords set and make it a regexp!

debian_file will search the irc given file using the debian web site cgi. The argument is not used.

debian_pkg will search the irc given package using the debian web site cgi. The argument is not used.

dict will ask your defined dict server for the given word. The argument may be the dictionnary name to use in the query, but defaults to the 'default' entry of the dict config element.

google will ask google for the rest of the irc line. The argument is not used.

mute will mute the bot, you have to be a controler to use that. The argument is not used.

pyramid is a game named after a french TV game. You have to make guess a world

random will say one of the sentences listed in the arguments randomly. The sentences have to be separated by '%' signs.

reconf will ask the bot to re-read its configuration. It allows you to handle dynamically your configuration, no need to restart the bot, and irc control! You have to be in the controler list to use this action.

rejoin allows you te rejoin a channel (useful on kick, just add a `op="kick"` parameter to the behaviour element definition).

say will say the arguments.

timer will say the first argument, then wait for a random time, and say the other arguments. The args have to be separated bu a '%' char.

2.4   Interacting with the running bot, from irc

You can use the `reconf` and `mute` actions (see section 2.3.3) to control the bot from irc, and you define who can do that in the first configuration element, with the `controler` property.

Chapter 3

Conclusion

You should now be abble to install, run and configure your manderlbot, and have it play with you on your prefered IRC channel.

If you miss some action, please consider playing with the code (in erlang) or sending us some feature request, we may or may not implement your ideas!

If you want to contribute, send a patch, and you may obtain a write access on the CVS (you may want to create a user account on tuxfamily.org services first).

Enjoy manderlbot, enjoy free software!