

# *Manual for Bubabot An Anarchy Online Chatbot*

Written by Sebuda and Derroylo

**Index:**

1	Instruction to and Installation of the Bot.....	4
1.1	What is Budabot? .....	4
1.2	Version History .....	4
1.3	Some Basic Informations about the Bot itself.....	4
1.4	How to start? .....	4
1.5	How to change the basic settings of the bot? .....	5
1.6	How to enable/disable commands, events or complete modules and changing their access level? .....	7
1.7	How to setup the adminlist? .....	9
1.8	How to use the banlist? .....	9
1.9	Chatrelay between two guilds .....	10
2	List of Commands .....	10
2.1	Info .....	10
2.2	Online Commands.....	10
2.3	Timer commands.....	11
2.4	Citycloak commands .....	11
2.5	Bio-Material commands .....	12
2.6	Guildrelay.....	12
2.7	Raid commands .....	12
2.8	Guestchannel commands.....	13
2.9	Alternative Char commands.....	13
2.10	Logon Message .....	14
2.11	Loot commands .....	14
2.12	Bio Regrowth and Guardian commands .....	14
2.13	Items commands.....	15
2.14	Pocketboss and Symbiants .....	15
2.15	Team Commands.....	15
2.16	Tower commands .....	16
3	Developer Informations.....	17
3.1	Introduction .....	17
3.2	Filestructure of the bot .....	17
3.3	Structure for your own modules.....	17
3.4	Functions of the bot core .....	18
3.4.1	Create a blob, chatcmd or an itemlink.....	18
3.4.2	Send a message.....	18
3.4.3	Registering of commands.....	19
3.4.4	Registering of Events .....	21
3.4.5	Register a group of commands.....	21
3.4.6	Settings .....	22
3.4.7	Helpfiles .....	23
3.5	Variables of the botcore .....	24
3.5.1	Guildmembers .....	24
3.5.2	Admins .....	24
3.5.3	Vars .....	24
3.5.4	Settings of the bot.....	24
3.5.5	Sender.....	24
3.5.6	Chatlist .....	24
3.5.7	Buddylist .....	24
3.5.8	Type of the message.....	25

**Picture Index:**

Figure 1: Settings .....	5
Figure 2: Symbol Settings .....	6
Figure 3: Default tell color setting .....	6
Figure 4: Commando and Event config .....	7
Figure 5: Commands and Events on config .....	7
Figure 6: Config of the topic command .....	8
Figure 7: Config of the online group.....	8
Figure 8: Config of the online command .....	9
Figure 9: Adminlist .....	9
Figure 10: Banlist .....	10
Figure 11: My first module .....	17

# 1 Instruction to and Installation of the Bot

## 1.1 What is Budabot?

Budabot is an Anarchy Online Chatbot which offers many features to help to manage your guild and also your raids. This guide is meant as a help into your first steps with this bot and isn't meant to be a full list of all features of this bot since those can change by custom modules of other users. To see a full list of features you should check the in game help files of this bot. To look at them do "/tell botname help". If you can't find what you need we are there to help you out. Please don't send us in game tells, since we want to just play from times to times too☺. You can reach us always over our project Homepage (<http://sourceforge.net/projects/budabot>), our Forum (<http://budabot.sourceforge.net>) or the Hackersquest Forum ([www.hackersquest.com](http://www.hackersquest.com)).

We are hoping that you enjoy this bot since we have spent a lot of time to make it easy usable for everyone.

Greetings Derroylo and Sebuda

## 1.2 Version History

Version 0.5

- First Developer release

Version 0.6

- Second Developer release(see patchnotes.txt for more details)

## 1.3 Some Basic Information about the Bot itself

Coder:	Sebuda Derroylo
Programming Language:	PHP5
Operating Systems:	Windows (from Version ME) Unix System (not tested yet but works theoretically too)
Database Systems:	MySQL SQLite

## 1.4 How to start?

As first step you need to have a character, on which the bot should run later and if you want to use this bot as a guildbot or as an orgraidbot you need to invite that character in your guild. As rank he will need at least "Squad Commander" to be able to shows the Tower Attacks.

To start the bot simple click on the chatbot.bat. As you are first using this bot, you need to fill out some information about your Account and more, which will be explained at the given time.

The first questions that are appearing are according to your Account. These are your Account name, Password, dimension and the Character you want to use. Please be aware that you need to care about upper and lower cases. In the next step you will be asked for the Administrator of this bot. This user will be the one that has access to all commands of this bot.

If you are running the bot in your org you need to enter your Guild name in the next step, when you don't enter it the bot will not be able to get the Members of your org.

In the next setup parts you will be asked if you want to put other bots on ignore. This means if you are running other bots in your org you should enter the name of this bot here, otherwise it can happen that both bots can end in an endless loop sending each other "Unknown Command or access denied". When you want more then one bot on the ignore list you will need to separate them with a Semicolon.

Now we are coming to the Database settings. Answer here with "no" when you don't want to use your own setup for the Database. As short explanation the Database is the place where the bot saves for example you're Guild Members and many more things that are needed for the bot to run.

On the last step you will be asked if you want to enable all Modules by default. These Modules are holding all commands that are provided by the bot. When you enter yes the bot will start with all commands and events enable if no they will be disabled. You can disable or enable the commands later too in game, how to do that will be explained on a later place.

When all was entered correctly the bot should start and will appear online after he logged in.

You can reenter the setup every time with deleting the "delete me for new setup" file.

After the bot has started you will notice that the bot is creating a org roster list (when the bot is an member of the org and you have entered in the setup part the name of your guild). This process is needed to get an actual list of the members of the org. After the bot is done with this step, he will automatically restart.

### 1.5 How to change the basic settings of the bot?

After the bot is started up you may want to customize the bot settings for your personal use. The bot offers a few basic settings that can be changed. These are for example the prefix for the commands used in guild or orgchannel and also the different colors for tells, guild and privatechat text are changable.



Figure 1: Settings

In the screenshot above you can see an example of the bot settings. Behind the name you can see the current setting for this option. Depending on the setting there can be also a “Help” and “Change this” link. When you are following the help link you can see a specific help for this setting, as the name of the setting will maybe not give an clue what it effects, so you should read them too sometimes. If the Setting is changable you will see the “Change this” link, when not this setting will be used internal and is only there to give you information on which state this setting is currently.

I will explain the changing of an setting with 2 examples.

First we are going to change the setting of the prefix for commands. Open your settings window (with “/tell botname settings”) and you should see something similar with the following screenshot.

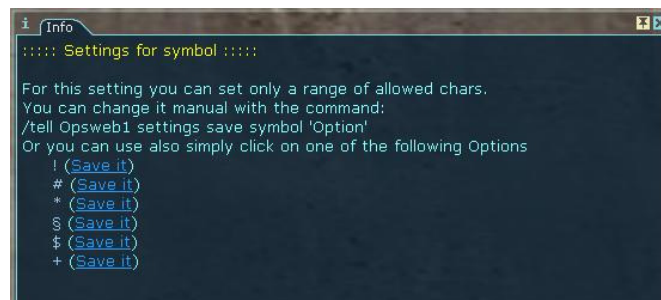


Figure 2: Symbol Settings

Click on the “Change this” link and a new window will be available with the options for this setting. For this setting the bot is giving us a few examples which you can use. Simple click on “Save this” and the bot will save it. At the case the bot isn’t showing a prefix you want to use, you need to change it manually then. For example you want to use the sign “?” as prefix for your commands then simple do “/tell botname settings save symbol ?” and he will save it.



Figure 3: Default tell color setting

As second example we are going to change the default color for tells. Open your settings window again with “/tell botname settings” and click on “Change this” behind the setting for “Default tell colors”. Same as on the first example the bot is showing you a few possible settings for it and you can directly save them with clicking on “Save this”.

## 1.6 How to enable/disable commands events or complete modules and changing their access level?

The bot allows you to enable/disable commands, events or complete modules over the “config” command. Open the config window with “/tell botname config”. Now you can see a list with all available modules with their commands and events.

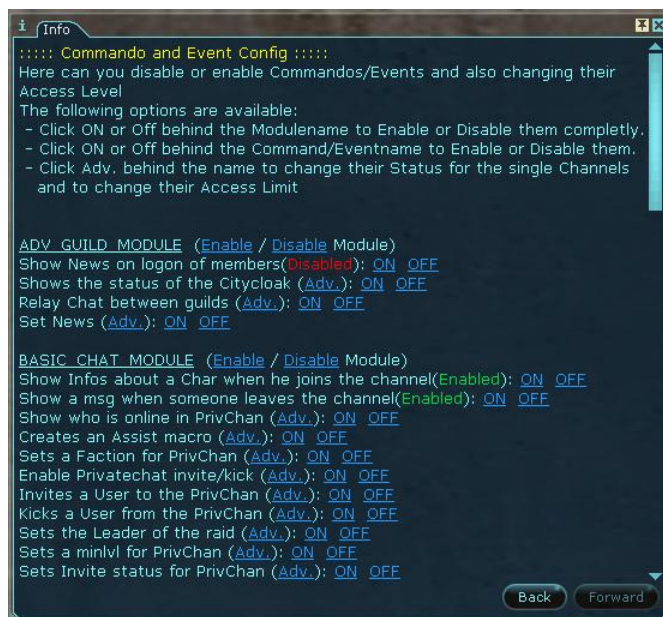


Figure 4: Commando and Event config

The screenshot above shows you an example how the config window looks like. You can identify the modules as their name is underlined. Behind the name you can find two links. One is for enabling and the other one is for disabling the whole module. Maybe you are asking yourself now what an module is, well it is a collection of commands and events that are gathered together to get an easy overview of what the bot offers. For example there is an Module called “BASIC\_GUILD\_MODULES”. These is holding together the basic modules that are needed to be able to use the bot for a guild. Commands and Events that are in there are the Online/Count/Sm commands, Logon/Logoff Events and so on.

To be able to identify if it is an command or event look at the following screenshot.

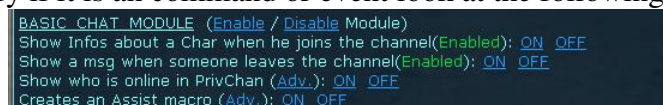


Figure 5: Commands and Events on config

You will notice that behind some names stands “Enabled” or “Disabled”. These are the events and can only be enabled or disabled, which you can do directly behind the name with the links “On” and “Off”. Commands have an “Adv.” Link behind their name which is just a short form for Advanced Settings.

Same as the Events you can directly enable or disable them with the links behind it but it is far not that what the bot is offering. Let’s take a look at the Advanced Setting for an command. As an example i am showing now the topic command, which can be found under the “BASIC\_CHAT\_MODULES”.





**Figure 6: Config of the topic command**

In this window you can see the current status of this command in all 3 channels. As you can see, you can enable or disable the command for every single channel. Under the channels you can switch also the required minimum access level that is needed to use this command for the channel. One more thing that you will be notice on the other commands is that behind a Channel the current status says “Unused”. This just means that the command is not available for this channel and can therefore not be enabled/disabled or their admin status changed.

Under the three channels for the command you will see one one more point which is called “subcommand for this command”. Easiest way to explain what subcommands are is an little example. Normally when you are looking at the topic command it allows you to show the current topic and also let you change it but maybe you don’t want to allow everyone to change it. The subcommands are allowing you to set different admin requirements to show and to change the topic.

There exists also another type of config. Open the advanced Config for the online command and you will notice a different view as you have seen before.



**Figure 7: Config of the online group**



What you see is a group of commands. The bot gathers all commands together that has nearly the same use. On this window you can change the admin level of the whole group or can enable/disable them all at once but you can set them manually too when you click on the “Adv.” Link behind the command. The window that appears then looks the same as from the other commands.



Figure 8: Config of the online command

## 1.7 How to setup the adminlist?

If you are using the bot as raidbot you will need to setup an adminlist. The bot differs between 4 adminlevels. The highest one is mostly you as the Administrator. This admin is set over the setup part of the bot. Next are the Moderators. They are taking care about what happens on your bot and for example they have access to the settings and the config of the bot too and can ban players from it if it is needed. The next lower adminlevel are the Raidleaders. They are leading raids that are hosted by the bot. Last adminlevel isn't a real level in its meaning since you don't need to add them manually. These access level is given about the leader command in the Private Group of the bot. Depending on your config of the bot the admins have different access to it's commands.

To see your current adminlist do “/tell botname adminlist”



Figure 9: Adminlist

To add or remove someone from the adminlist you need to use one of the following commands.

- Add a new Raidleader “/tell botname raidleader player\_name”
- To remove a Raidleader “/tell botname kickraidleader player\_name”
- To add a new Moderator “/tell botname addmod player\_name”
- To remove a Moderator “/tell botname kickmod player\_name”

## 1.8 How to use the banlist?

Incase you don't want to let a player be able to join the Raidbot you will need to ban him. To see the current banlist do “/tell botname banlist”.

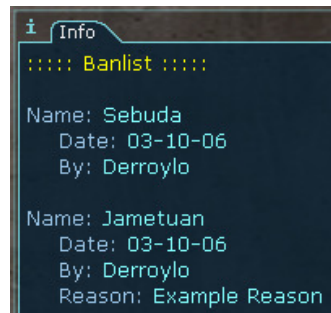


Figure 10: Banlist

When you want to add a player to this list you need to do “/tell botname ban player\_name”. The player will now appear on the banlist with info about who banned him and when. If that is not enough you can enter a reason for the ban too. Useful to remember why he got banned. “/tell botname ban player\_name reason enter\_reason\_for\_ban” is the command for it. To remove a ban from a player do “/tell botname unban player\_name”.

### 1.9 Chatrelay between two guilds

Let us take the following scenario to explain how the guildrelay system is working. You have 2 guilds, guild1 and guild2, between those you want to relay the chat. The bot for guild1 is bot1 and for guild2 it is bot2. As first step you need to make sure both bots are using the same Database. When you are using MySQL this is pretty easy, you just need to use to enter on the setup procedure the same Databaseconnection and Databasename. If you are using SQLite(which is it by default) the databasename and the database filelocation must be the same. Start both bots and enable on both the guildrelay commands (under ADVANCED\_GUILD\_MODULES) when you haven't done it already. On the last step you need to enable guildrelay now with the command “/tell botname guildrelay bot\_name”. In our scenario it is “/tell bot1 guildrelay bot2” and “/tell bot2 guildrelay bot1”. From now you should see all messages that are written in the guild1 also in the guild2.

## 2 List of Commands

### 2.1 Info

This list is only a part of all commands that are supported by this bot. For a full list of features check the helpsystem of the bot with “/tell botname help”.

### 2.2 Online Commands

General Description: Shows the current members that are online in your guild or who is in the privatechannel

Command: *online*

Description: Executed in the guildchannel it will show the current online orgmembers (if guildrelay is enabled from these guild too) and also the members that are in your guestchannel.

Executed in the privatechannel it will show the players in this channel.

Example: (in guildchannel) *!online all* Shows all online guildmembers  
 (in guildchannel) *!online nt* Shows all Nanotechnicians that are online

Command: *count*

Description: Executed in the guildchannel it will show the current online orgmembers (if guildrelay is enabled from these guild too).  
Executed in the privatechannel it will show the players in this channel.

Example: (in guildchannel) *!count all*      Shows all online guildmembers  
(in guildchannel) *!count nt*      Shows all Nanotechnicians that are online

Command: *sm* or *chatlist*

Description: Executed in the guildchannel it will show the current online orgmembers (if guildrelay is enabled from these guild too).  
Executed in the privatechannel it will show the players in this channel.

Example: (in guildchannel) *!sm*      Shows all online guildmembers  
(in privatechannel) *!sm*      Shows all players in the privatechannel

## 2.3 Timer commands

General Description: The timers are separated into two parts. One for tells and one for guild/priv channels. That means you can set timers in tells which aren't shown in the guildchannel and only the person who set it can check them.

Command: *timer time\_of\_running\_in\_min*

Description: Sets a timer that runs a specific amount of minutes.

Example: (in guildchannel) *!timer 10*      The timer will run for 10minutes

Command: *timer time\_of\_running\_in\_min timer\_name*

Description: Sets a named timer that runs a specific amount of minutes.

Example: (in guildchannel) *!timer 10 test*      The timer named **test** will run for 10minutes

Command: *timers*

Description: Shows all running timers.

Example: (in guildchannel) *!timers*      Shows all running timers from guild/privatechannel.  
(in tells) *!timers*      Shows your personal and guild/priv timers.

Command: *timer del timer\_name*

Description: Deletes a running timer.

Example: (in guildchannel) *!timer del test*      Deletes the timer **test**.

## 2.4 Citycloak commands

General Description: The citycloak command shows you if the cloak of your city is enabled or disabled and how long you can change it's status. It shows also a history of the citycloak.

Command: *city*

Description: Shows the current status of your citycloak.

## 2.5 Bio-Material commands

General Description: With this command you can identify a Bio Material that is dropping of Aliens at various places.

Command: *bio drop\_bio\_material\_here*

Description: Identifies the type of the Bio-Material

## 2.6 Guildrelay

Command: *guildrelay other\_botname*

Description: Enables Orgchat relay to another bot.

Example: (in guildchannel) *!guildrelay bot* Sends all orgmessages to **bot**.

Command: *guildrelay off*

Description: Disables guildrelay.

## 2.7 Raid commands

General Description: These commands are only usable in the raidchannel of the bot.

Command: *leader*

Description: Set someone as the leader of the raid.

Example: *!leader*                      Sets yourself as leader  
              *!leader player*        Sets **player** as the new raidleader

Command: *assist player*

Description: Creates an assistmacro for the player.

Example: *!assist Derroylo*    Creates an assistmacro for **Derroylo**

Command: *topic*

Description: Shows the current topic of the bot

Command: *topic text*

Description: Sets a new topic for the bot.

Example: *!topic bla bla*        Sets the text **bla bla** as new topic

Command: *topic clear*

Description: Clears the current topic of the bot

Command: *minlvl lvl*

Description: Sets a min level that is needed to join the raidbot.

Example: *!minlvl 140*            Only player with lvl140+ can now join.

Command: *open mode*

Description: Changes the mode of the raidbot.

Example: *!open all*                The bot is open for everyone now.  
              *!open org*            Only orgmembers can join now.

Command: *faction faction*

Description: Allows only one faction to join the raidbot.

Example: *!faction omni*        Allows only omnimembers to join the bot.  
              *!faction clan*        Allows only clanmembers to join the bot.

Command: *faction not* faction

Description: Allows only one faction not to join the raidbot.

Example: *!faction not omni* Allows only non omnimembers to join the bot.  
*!faction not clan* Allows only non clanmembers to join the bot.

## 2.8 Guestchannel commands

General Description: Allows the guild to invite non-org player to the guestchannel. You can allow them too to use the guildcommands but these needs to be enabled over the settings command. Note that the gueschannel relay to the org will be disabled if anyone else as a guest joins the Private Channel.

Command: *guest invite* player\_name

Description: Invites a player to the guestchannel.

Example: *!guest invite* Sebuda Invites Sebuda to the guestchannel.

Command: *guest kick* player\_name

Description: Kicks a player from the guestchannel.

Example: *!guest kick* Sebuda Kicks Sebuda from the guestchannel.

Command: *guest add* player\_name

Description: Adds a player to the guestlist. He will be auto-invited when he logs on.

Example: *!guest add* Derroylo Adds Derroylo to the guestlist.

Command: *guest rem* player\_name

Description: Removes a player from the guestchannel.

Example: *!guest rem* Derroylo Removes Derroylo from to the guestlist.

Command: *guests*

Description: Shows the current guest of the bot.

Command: *guestlist*

Description: Shows the current guestlist.

## 2.9 Alternative Char commands

Command: *alts*

Description: Shows your own alts.

Command: *alts* player\_name

Description: Shows an alt list of another player.

Example: *!alts* Sebuda Shows the alts of Sebuda.

Command: *alts add* player\_name

Description: Adds another char to your alt list.

Example: *!alts add* Sebuda Sets Sebuda as on of your alts.

Command: *alts rem* player\_name

Description: Removes a char from your alt list.

Example: *!alts rem* Sebuda Removes **Sebuda** as on of your alts.

## 2.10 Logon Message

General Description: Sets a message that is beeing shown on the logon message of an player.

Command: *logon* your\_logon\_message

Description: Sets your Logon message.

Example: *!logon* my new logon msg                Sets your logon message.

Command: *logon* clear

Description: Clears your Logon message.

## 2.11 Loot commands

Command: *loot* item\_name

Description: Adds an item to the loot list.

Example: *!loot* First-Aid kit                Adds **First-Aid kit** to the loot list.

Command: *list*

Description: Shows the loot list.

Command: *roll*

Description: Rolls the loot and shows the winners.

Command: *add* slot\_number

Description: Adds you to the slot.

Example: *!add* 1                Adds you Slot 1.

Command: *setminlvl* slot\_number level

Description: Sets a minlvl for a Slot.

Example: *!setminlvl* 1 150                Sets the minlvl from 150 to Slot 1.

Command: *pwinners*

Description: Shows the current status of the preservewinners modus.

Command: *pwinners* new\_status

Description: Sets the status for the preservewinners modus.

Example: *!pwinners* on                Enables the preservewinners.

*!pwinners* off                Disables the preservewinners.

Command: *pwinners* clear

Description: Clears the preservewinners list.

## 2.12 Bio Regrowth and Guardian commands

Command: *guard*

Description: Shows the list of players that should be able to cast Guardian. If you have set an assist on a soldier he will be always on the last place.

Command: *g*

Description: Shows the cast of the Guardian, when it is running out and when it is ready again.



Command: *bior*

Description: Shows the list of players that should be able to cast Bio Regrowth.

Command: *b*

Description: Shows the cast of the Bio Regrowth , when it is running out and when it is ready again.

## 2.13 Items commands

General Description: Searches for an item in the AO Itemsdatabase.

Command: *items* item\_name

Description: Search in the Itemsdatabase for the given item name.

Example: *!items* chosen nano                      Shows all items that matches **chosen nano**.

Command: *items* QL item\_name

Description: Search in the Itemsdatabase for the given item name with a specific QL.

Example: *!items* 235 chosen nano                      Shows all items that matches **chosen nano** and the **QL235**.

## 2.14 Pocketboss and Symbiants

Command: *pb* pocketboss\_name

Description: Searches for an pocketboss and shows the symbs he is dropping.

Example: *!pb* Brutal Rafter    Shows the symbs that drops from this pb.

Command: *symb* location type

Description: Searches for symbiants that matches the criteria and shows the pb that drops it.

Example: *!symb* eye infantry                      Shows all Infantry Eyes that drops from PB's.

## 2.15 Team Commands

Command: *teams*

Description: Shows the current team setup and also a Administration window for it.

Command: *teams clear*

Description: Shows Clears the current team setup.

Command: *teams 0* player\_name

Description: Removes a player from all teams.

Example: *!teams 0* Derroylo                      Removes Derroylo from all the team he was assigned to.

Command: *teams* team\_number player\_name

Description: Adds a player to a team.

Example: *!teams 1* Derroylo                      Adds Derroylo to team 1.

Command: *team* team\_name

Description: Creates a new team with the given name.

Example: *!team* Tank Team                      Creates a new team called Tank Team.

Command: *team* new\_team\_name team\_number

Description: Renames an existing team.

Example: *!team* Backup TTeam 2 Renames Team 2 to Backup TTeam.

## **2.16 Tower commands**

Command: *battle*

Description: Shows the last attacks on Notumfields.

Command: *victory*

Description: Shows the last winners of Notumfield battles.

## 3 Developer Informations

### 3.1 Introduction

This part is for peoples that want to develop their own modules for this bot. You will find here the needed file structure for your modules, functions that you may find helpful and also some important variables that are provided by the bot.

### 3.2 File structure of the bot

When you open your bot folder you will find several folders. Every one of them has a specific use. Which one it has will be shown in the following table.

Name	Type	Usage
cache	Folder	Cache folder for XML files(from whois or history cmds)
core	Folder	Includes most of the core modules of the bot like admin, settings and config stuff
data	Folder	Mostly the SQLite DB's are saved here
dll	Folder	Required Dll files for the bot
modules	Folder	Modules that are used by the bot
sql	Folder	SQL Files like the items database and Pocketboss db
chatbot.bat	File	Starts the Bot
config.php	File	Configurationfile for the bot
main.php	File	Includes DLL loading, creating bot/db object creation and the main loop

### 3.3 Structure for your own modules

To start with your own modules you will need to create a new folder in the module folder of the bot. Please use only uppercases names for them (just to use the same format as we do). Go into your new created folder and create a new PHP file with the same name as your new module folder. This file will include later all the registering of the commands/events/settings/helpfiles for your module. This file could look like this for example:

```
1 <?
2 $MODULE_NAME = "MY_OWN_MODULE";
3 $PLUGIN_VERSION = 0.1;
4
5 //Commands
6 bot::command("guild", "$MODULE_NAME/first_command.php", "test", "all", "My first module");
7
8 //Events
9 bot::event("logOn", "$MODULE_NAME/logon.php", "none", "Says welcome to all that logson");
10
11 //Helpfiles
12 bot::help("welcome", "$MODULE_NAME/welcome.txt", "guild", "Help for my first module");
13 ?>
```

Figure 11: My first module

To see what every bot function can be found too in this manual when you look at the next sites. Now you just need to write the files that are registered now(first\_command.php, logon.php and welcome.txt) and they will be available after the next reboot of the bot.

### 3.4 Functions of the bot core

#### 3.4.1 Create a blob, chatcmd or an itemlink

**Function name:** makeLink

**Description:** Creates a Link

Argument: \$name

Description: Name of this link

Argument: \$content

Description: Content of this link

Argument: \$type

Description: Type of this link

Default Value: blob

Allowed content: *blob, text, chatcmd*

Argument: \$style

Description: Style for this link

Default Value: NULL = no style

**Functionname:** makeItem

**Description:** Creates an Itemlink

Argument: \$lowID

Description: Low ID of the Item

Argument: \$hiID

Description: High ID of the Item

Argument: \$ql

Description: Quality of the Item

Argument: \$name

Description: Name of the Item

#### 3.4.2 Send a message

**Function name:** send

**Description:** Sends a message or add/rem/checks a buddy

Argument: \$message

Description: The message that should be send or a buddyadd/isbuddy/rembuddy

Allowed content: *buddyadd, isbuddy, rembuddy* or a text to send

Argument: \$who

Description: High ID of the Item

Default Value: NULL = privatechannel

Allowed content: Playername, *guild*, public channelname

### 3.4.3 Registering of commands

**Function name:** command

**Description:** Adds a command to the bot. This can be configured over the config cmd.

Argument: \$type

Description: Type of the command

Allowed content: *msg, priv, guild*

Argument: \$filename

Description: Name of the file that is assigned with the command

Argument: \$command

Description: The command itself

Argument: \$admin

Description: Access level for this command

Default Value: *all*

Allowed content: *leader, raidleader, mod, admin*

Argument: \$description

Description: Description of this command(shown on the config cmd)

Default Value: *none*

Allowed content: text

**Function name:** regcommand

**Description:** Adds a command to the bot. Will be active and not configurable

Argument: \$type

Description: Type of the command

Allowed content: *msg, priv, guild*

Argument: \$filename

Description: Name of the file that is assigned with the command

Argument: \$command

Description: The command itself

Argument: \$admin

Description: Access level for this command

Default Value: *all*

Allowed content: *leader, raidleader, mod, admin*

**Function name:** unregcommand

**Description:** Deactivates an command

Argument: \$type

Description: Type of the command

Allowed content: *msg, priv, guild*

Argument: \$filename

Description: Name of the file that is assigned with the command

Argument: \$command

Description: The command itself

**Function name:** subcommand

**Description:** Adds a subcommand to the bot. This can be configured over the config cmd.

Argument: \$type

Description: Type of the command

Allowed content: *msg, priv, guild*

Argument: \$filename

Description: Name of the file that is assigned with the command

Argument: \$command

Description: The command itself(must be in pregmatch pattern)

Argument: \$admin

Description: Access level for this command

Default Value: *all*

Allowed content: *leader, raidleader, mod, admin*

Argument: \$dependson

Description: The name of the command to which this subcommand belongs

Allowed content: any existing command

Argument: \$description

Description: Description of this command(shown on the config cmd)

Default Value: *none*

Allowed content: text



### 3.4.4 Registering of Events

**Function name:** event

**Description:** Adds an event to the bot. This can be configured over the config cmd.

Argument: \$type

Description: Type of the command

Allowed content: *towers, orgmsg, msg, priv, guild, joinPriv, leavePriv, logOn, logOff, 2sec, 1min, 15min, 1hour, 24hrs, setup*

Argument: \$filename

Description: Name of the file that is assigned with the event

Argument: \$dependson

Description: The name of the command to which this event belongs

Default Value: *none*

Allowed content: any existing command

Argument: \$desc

Description: Description of this event(shown on the config)

Default Value: *none*

Allowed content: text

**Function name:** regevent

**Description:** Adds an event to the bot. This will be active and not configurable.

Argument: \$type

Description: Type of the event

Allowed content: *towers, orgmsg, msg, priv, guild, joinPriv, leavePriv, logOn, logOff, 2sec, 1min, 15min, 1hour, 24hrs, setup*

Argument: \$filename

Description: Name of the file that is assigned with the event

**Function name:** unregevent

**Description:** Deactivates an event

Argument: \$type

Description: Type of the event

Allowed content: *towers, orgmsg, msg, priv, guild, joinPriv, leavePriv, logOn, logOff, 2sec, 1min, 15min, 1hour, 24hrs, setup*

Argument: \$filename

Description: Name of the file that is assigned with the event

### 3.4.5 Register a group of commands

**Function name:** regGroup

**Description:** Registers a group of command.

Argument: \$group

Description: Name of the group

Argument: \$module

Description: Name of the module where these files are located.

Argument: \$desc

Description: Text that is shown on the config as description.

Default Value: *none*

Allowed content: text

### 3.4.6 Settings

**Function name:** addsetting

**Description:** Adds an setting.

Argument: \$name

Description: Name of the setting

Argument: \$description

Description: Text that is shown on the settings as description.

Default Value: *none*

Allowed content: text

Argument: \$mode

Description: Mode of the setting

Allowed content: *edit, hide, noedit*

Argument: \$setting

Description: Default value for this setting

Default Value: *none*

Argument: \$options

Description: Values that the user can choose from.

Default Value: *none*

Argument: \$intoptions

Description: Values that are assigned to the option that user has choosen.

Default Value: *0*

Argument: \$admin

Description: Access level for this command

Default Value: *mod*

Allowed content: *leader, raidleader, mod, admin*

Argument: \$help

Description: Helpfile for this setting

Default Value: *NULL*

**Function name:** getsetting

**Description:** Gets the value of a current setting

Argument: \$name

Description: Name of the setting

**Function name:** savesetting

**Description:** Save changes to a setting

Argument: \$name

Description: Name of the setting

Argument: \$newsetting

Description: New Value for this setting.

### 3.4.7 Helpfiles

**Function name:** help

**Description:** Adds an helpfile.

Argument: \$command

Description: Name of the command that calls this helpfile

Argument: \$filename

Description: Name of the file that is assigned with the helpfile

Argument: \$admin

Description: Access level for this command

Default Value: *all*

Allowed content: *leader, raidleader, mod, admin*

Argument: \$info

Description: Text that is shown on the help command as description.

Default Value: *NULL*

Allowed content: text

### **3.5 Variables of the botcore**

A list of variables that are important for your modules

#### **3.5.1 Guildmembers**

Name: guildmembers

Description: Array of the guildmembers with their rank id's

Type: Array

Keys: Name of a guildmember

Values: Rank id of the member

#### **3.5.2 Admins**

Name: admins

Description: Array of the admins of this bot with their rank id's

Type: Array

Keys: Name of an admin

Values: Rank id of the admin

#### **3.5.3 Vars**

Name: vars

Description: Array with different values like guildname(my guild) and guild id(my guild id).

Type: Array

Keys: my guild, my guild id, name

#### **3.5.4 Settings of the bot**

Name: settings

Description: Array with all registered settings of the bot. Use the botcore functions to change them or the changes will not be saved!

Type: Array

#### **3.5.5 Sender**

Name: sender

Description: Name of the player that has send the message

Type: Variable

#### **3.5.6 Chatlist**

Name: chatlist

Description: Array with all players that joined the privategroup of the bot.

Type: Array

Keys: playernames

Values: True when he joined

#### **3.5.7 Buddylist**

Name: buddyList

Description: Array with all players from the friendslist of the bot.

Type: Array

Keys: playernames

Values: 1 => he is online, 0 => he is offline

### 3.5.8 Type of the message

Name: type

Description: Source from where the message came.

Type: Variable

Values: msg, priv, guild, joinPriv, leavePriv, logOn, logOff, towers, orgmsg