CENTRAVI

Centravi Bot

The one thing you need on IRC.

Berend de Groot

2/8/2013

This document is written in order to describe the functionality of the Centravi channel management bot. While working actively on this document new functionality is added and documented. Whether this is convenient or not is determined by the Administrators. It definitely does not describe any technical aspects or sensitive information regarding Centravi and its staff.

Contents

Contents

1. Introduction

This document is written in order to describe the functionality of the Centravi channel management bot. While working actively on this document new functionality is added and documented. Whether this is convenient or not is determined by the Administrators. It definitely does not describe any technical aspects or sensitive information regarding Centravi and its staff.

1.1. For who is this document?

It is written for anyone who has interest in knowing how Centravi globally works. It will not, and it shall not describe any technical aspects of this piece of (yet to be build) software. Even though this document is written for anyone, Centravi is the publisher and owner. Whether this document goes public or not, is their decision to make.

The purpose of this document is to get everyone familiar with the new bot even though it is not build yet. Developers may use this document as reference while building the bot; it helps with fast progress and prevents developers from losing their major goal, building and improving a channel management bot for all OnlineGamesNet users. It can however also be used to gain experience in the exact functionality of the bot, which does not include technical aspects, as said before.

1.2. Document Set-up

The first paragraphs will explain you what IRC is and the purpose of it. This might sound weird for most of the readers because mostly everyone involved has tons of IRC experience. However some people who read this document may not have any experience regarding to IRC. After having you told that, we give you a clear overview about what the Centravi bot is and what it does. If you got a clear view about this already, you can skip these paragraphs and continue reading the general requirements set up by the Centravi administrators. Based on these requirements we continue giving a close look into each category of the bot and why we divided the bot in these categories, and then describe commands regarding to this category.

1.3. After Reading

If you have read the entire document, you have all the knowledge you need for using Centravi and a lot more. You will know exactly, almost as good as the developers know it, how the bot works. For just using the bot it is definitely not necessary to know it all, however if you are willing to help anyone regarding their Centravi problems. You should have all the knowledge carried by this document. It is written with much care and took a lot of time get all the functionality described properly.

We are always open for feedback and suggestions, these helps us improving our service and staff, helps us to provide our users **what they ask for** and not what we think they ask for. We listen to everyone as long they suggestion is well-founded, Thank you in advance.

2. Internet Relay Chat

Internet Relay Chat, short as IRC is an internet protocol designed for text messaging (chat) in groups or in private. For more technical information check the Wikipedia page about IRC. There are a lot different networks which all have their own purpose. The network we are using and where the Centravi bot will be running is called OnlineGamesNet.

OnlineGamesNet now days in an IRC network which allows all players from Game Forge to have their own game channel, team channel and if necessary alliance/guild channels.

2.1. Channels

As said before; IRC uses channels, also called rooms. Each room has his own purpose, in these rooms are topics set which shows the user which subject to talk about. I take as example the channel #centravi. As you can see al channels start with # before the channel name. #centravi is our public project channel. This has the purpose to connect all our users with our staff. From here we offer support and inform users about any news.

Gameforge got a set of games which all have their own channels on this network. For example ogame; Ogame is a game written by the director itself. It's a famous browser game over a lot of languages. All these languages have their own channel; #ogame.nl, #ogame.es etc. These also have staff channels, administration channels, etc etc.

2.2. Operators

Each channel has his own operators. Operators are shown in the user list with a @ sign before their name. Operators are also called supporters or ops. These people watch over specific channels to make sure the channels are used for its purpose and may kick or ban people who break channel rules. These people are staff members of that specific game community, IRC Supporters or any other administrators.

2.3. Voice

Voice is something like a status in a specific channel, these users can not ban or kick people using the IRC protocol, still they are shown above the normal users and are probably part of the staff of that channel and have other task regarding that staff which not includes IRC tasks. Voice can also be used with specific modes to make them able to speak in the channel. However this is a more technical aspect we won’t go deeper into (Check the IRCu channel modes yourself).

2.4. Regular Users

Besides the above 2 type of users, we also have the regular users who use the network to speak about the games they play. Most of the people on IRC are placed in this group. They are not any less than the other users, however they are not working as volunteers to do such jobs described above, nevertheless this network needs users as much as it needs their staff and game communities.

2.5. IRC Network Staff

The network itself also got his own staff, these people help users out with their IRC problems, and some of them administrate the server. This is a small group of very experienced users who do mostly besides their support work in this staff other stuff on IRC as well. They make sure users get the support they deserve on this network and help them with any trouble the user might run into. For these people anyone but their self is a user, an administrator at for example the ogame.org game is a user to them as soon he comes to the support desk with any problems. Same counts for the Centravi team members; we might be users to the IRC network staff. If they come to us requesting the bot, they are our users.

2.6. Services

OnlineGamesNet uses services to make life easier on IRC. IRC itself allows users to have their own channels as operator however when they disconnect from the network everyone can take over their channels. Even if they give other people operator status they can take over the channel. Only IRC Operators could change this back to the rightful owner. Well how do you know someone is the rightful owner?

The services are a set of IRC bots, which helps with recognizing users using accounts. Based on these accounts users can have their own channels. Users register an account on the first bot called “Authserv” this bot will grant you access to your account and any channels you have so called “access”. Another bot called Chanserv is (like Centravi) a channel management bot. This bot holds a user list with accounts and stays in each channel which Is registered. When a user disconnects from IRC, comes back and logs in to his account. This bot makes sure access the user has will be assigned in all his channels (if he is in that channel). This service bot cannot be kicked or banned. Only IRC operators can remove its operator status. If you got this bot in your channel your channel is safe from overtakes and you can go away and come back whenever you want. All official channels (#centravi, etcetera) have Chanserv present in their channel. However Chanserv is only allowed in “Official channels”. Official channels are;

- Game channels, public and team (#ikariam.nl, #ikariam.nl-team).
- Alliance guild channels.
- Project channels.

Further details can be found here: http://www.onlinegamesnet.net/registrationGuidelines.php. If you fit in the requirements of registering a channel you can have Chanserv and your channel will be save from overtakes.

As you know now, IRC is for chatting with other people, meaning you control those bots by providing it with commands on IRC. For example the help command, you don’t just write “help” but those bots use triggers. A trigger is a character wrote before each commands, so the bot recognizes it and replies.

2.7. Centravi

This is where Centravi comes in, as said you can only have Chanserv if you fit the above requirements. If not, then what? Then Centravi! Centravi is an alternative bot for users who want to register a channel and will be recognized as owner of that channel. Centravi will do almost the same as Chanserv, the major difference is; Chanserv is a network service. Centravi is a bot build by users on the network itself. It will be a bit less safe, however still the best alternative there is on OnlineGamesNet.

As you already understood the bots holds a user list, on this list are all the users who has access in that channel, settings in this bot makes the owner able to say who is an operator, who gets voice and who may not enter the channel (banned). This bot helps a lot managing your channel. It is possible without, however for users who clearly lack experience on IRC will not be able to manage their channel without it.

3. Analysis

Centravi has a bot running, which is working – as far the users concern goes. It got a few bugs which developers are unable to fix because of the way it is built by the old developers. If something sudden happens, this bot might shut down and who knows if it will run another day. That’s why it's time to rebuild this bot with new developers and a new idea, a more advanced idea with hopefully a more advanced and better output to.

4. General

We have a global set of requirements provided by the Centravi project administrators. This list of requirements **must** be implemented in the new bot. New changes might come while developing; these will be added to this document as requirements later on.

- We shall be able to translate the bot in different languages.
- We shall be able to add new languages ourselves.
- Users shall be able to use commands in private, the bot should also check Authserv accounts if that happens.
- All commands should be dynamic. Including user access and staff access.
- If a new channel gets registered with Centravi, the bot should have a default set of settings set for that channels.
- We shall have the ability to change the default settings as well.
- We shall be able to change staff access.
- Security override should be enabled whenever you try to use staff commands.
- If a user lacks staff access the bots shall not reply on any staff commands.
- There shall be a possibility to bind commands.
- Centravi shall join on invite when the channel is registered.
- Centravi uses trigger “?”.

4.1. Bot Access

We got 2 kinds of access levels, the channel access and the oper access. Channel access is for users inside each channel. Oper access is for the Centravi staff members.

The Centravi staff members have the ability to register channels, and do several other tasks regarding managing channels and user accounts. These staff members can have different access levels. For example administrators need to have the ability to add new supporters, while supporters never need that ability. Either way, they need to register channels. So like the old bot we've given staff member’s access levels. Each (staff) command will be linked to an amount of access which can be changed in the bot configuration. Staff access goes from level 500-1000. This is because the normal channels access goes from 0-500 and we want to keep these completely separated. Each other command which requires access (besides the staff commands) will be linked to both a channel and staff amount of access. These are also both changeable in the bot configuration. The channel access is of course used in each channel separately. If a user got 100 accesses in one channel, it does not count in another. However staff members, who get 200 staff access, can use all commands in all channels linked to lower or equal then 200 staff access. This way supporters can manage channels and if they got reports of abuse, or need to set things right, they have the ability, of course is this not for abuse purposes.

5. Staff Commands

Note: Usage of the staff commands are not yet explained in detail.

As already shortly said in 4.1 staff commands each got their own access level. These access levels can be changed and configured as the administrators want it to. Each staff command will ignore users without staff access neither a reply “Lack access” is replied, users don't need to know about the commands, nor their access levels. Besides that, we got a special “Security Override”; each staff member (Users with staff access makes them immediately staff members) has the ability to turn this setting on. If this setting is turned on they can use staff commands below and equal to their level. If this setting is disabled. They can't use it.

This has security reasons with the “channel and staff access” inside a channel. A supporter might think he has access in that channel and kick someone, meanwhile he has not. The owner can be mad about that and reports abuse. We want to prevent this from happening with the security override. Staff members should disable this if they are not using it. They can enable and disable this setting using the **god** command.

All the oper commands can be seen in an overview with the command **OCMMANDS**. All the commands with their access can be changed using the **MODCMD,** this command also implements all the user commands and their access levels.

Commands can be bind as well, for example you got the “addusertogroup” command, which is way too long. We can bind it to for example adtg. Both “addusertogroup” and “adtg” can be used with the same functionality. For this we use the **BIND** and **UNBIND** commands.

Centravi administrators can also manage the staff members and their access. This can be done with the **OACCESS** command. Here is a list with all users and their staff access. Including the ability to add or delete staff members.

5.1. Channel Management

As mentioned earlier, staff members can register channels if they have the right amount of staff access. Channels can be registered with the **REGISTER** and **UNREGISTER** commands. Before channels get registered, the bot holds a special list called “do-not-register”. If a channel is in this list, it won’t be registered using the command. There is a force attribute available to override the do-not-register list. Otherwise it is not possible to register the channel until it is removed from this list. Channels can be added to the do-not-register list using **ADDDNR** and **DELDNR** and the list can be seen using **LISTDNR**.

It might be that you want the bot to leave a channel for a moment, in case operator status got lost, or any other reason. If you unregister a channel all settings including users get lost. That’s why we have a **PART** and **JOIN** command implemented in the bot. Bad behavior can also be punished by suspending a channel. In this case the bot leaves a channel with a message and won't come back nor join on invite until it is unsuspended. For this we use the **SUSPEND** and **UNSUSPEND** commands. If a user feels like moving a channel, which often happens because they just want to change the name. We also got a special command for that, the **MOVE** command.

Staff members also have the ability to check channels if there are enough users inside. Centravi has limited channels so only wants to keep active channels registered. Using the **EMPTY** command shows you all channels with less than a specific amount of users which we can manually configure ourselves. Besides checking for inactive channels, the bot also needs operator status to work properly, using the **NOOP** commands provides you with a list of all channels where the bot lacks operator status. This can be very helpful contacting users or even unregistering.

The bot of course should never have down-time or leave registered channels without good reason. However we can't give all guarantees this won't happen and even if this happens we can’t guarantee that the bot joins all channels either. However there is a command written which let the bot joins all channels currently registered with Centravi. This command is called **JARC**. Short for **J**oin **A**ll **R**egistered **C**hannels.

If a staff member is looking for a specific channel, it can use the **SEARCH** function to search for channels by its name (Or parts of its name). This command does not take any further explanation then searching channels.

5.2. Bot Management

The bot itself can be controlled by (probably) administrators. We are not talking about registering channels, or adding binds which is probably already an administrative function. Nevertheless the bot has commands to **QUIT** the entire code meaning the bot won’t work anymore. It also includes the ability to send **RAW** lines and let the bot do whatever the administrator provides. Simpler version of this command is the **SATY** and **EMOTE.** These commands can be used to let the bot use emote or say anything you provide.

We have spoken about settings which we can configure ourselves; these settings can be seen and changed with the **CONFIG** command. This command also includes for example the admin channel and the support channel so it works correctly with a support desk module. Statistics about the bot can be seen with the **STATS** command. These statistics show the following:

- Information about Centravi.
- Amount of channels.
- Current uptime.
- Amount of user accounts.
- Amount of users with access.
- Amount of accounts with staff access.
- Amount of commands and binds.
- Amount of Languages.

These statistics will be generated automatically by the bot itself, these statistics cannot be changed either.

5.3. Account Management

Centravi holds his own account management system. It uses the accounts from Authserv as verification a user is really that user. It saves those accounts with specific information. For example all channels the user has access to, email, language used by this user etc. Staff members can manually register accounts (even while its not a valid Authserv account) with **OREGISTER** and remove those accounts again with **OUNREGISTER**.

It might be that a user decides to change his Authserv account, this can be done by the network staff, and all network services will synchronize since they read from the same database. However Centravi does not, and cannot. Staff members have to rename those accounts manually using the **ORENAME** command.

If staff members want to check what a user did last, for example if they got abuse reports or anything familiar. Staff members can check the last 10 commands used by a specific user account using the **AEVENTS** command on any authserv account

Accounts can have different modes. For example users are “normal” bots are “bot” and users on vacation are “vacation”. These modes can be changed for each user at all times by the Centravi supporters. These modes itself can be modified as well.

6. Channel User Management

Before we explain the Channel Management on user level it is required for each user to have an Authserv account. Otherwise the user won't be able to use the Centravi service. It is of course possible to gain operator status and manage channel settings using the IRC protocol. However Centravi won't allow you to, and won't secure your account nor operator status.

Channel access varies from 1 to 500, it depends on the channel settings whether you are able to change for example settings or gain operator status. You can have 499 access in a specific channel, however the owner of that channel might decide that users with 499 still can't do anything.

500 Access is for the owner of the channel only, and he is not able to give this amount to anyone else. So we call 500 the "Owner access", different amount of access have different names:

- 1 – 199: Peon
- 200 – 299: Operator
- 300 – 399: Master
- 400 – 499 Co-Owner
- 500: Owner

These names are static and cannot be changed. However meaning of these accesses can be changed. A list of all users can be received by writing **USERS.** On that list you see all users in that specific channel, including their access, if they are in the channel and which use mode they have. Example is shown below.

Access	User	Mode	Seen
500	Berend	Normal	Here
450	Victor	Vacation	1d2h50m
300	Cow	Bot	Here

For all commands we are about to explain the following rule counts:

Any usage different then described in these paragraphs will trigger an error. Lacking one or more of the requirements assigned to a command will trigger an more advanced error with further explanation.

6.1. Channel Settings

We've spoken about the channel accesses shortly, access is assigned to users. Depending on the access a set of commands are open for them to use. However which commands? Well that is something the channel owner can manage himself to. By registering a channel a set of default settings is loaded to the channel. For example for adding new users you need 300, for gaining voice 100, for gaining operator 300, etcetera.

List of default settings:

<List of default settings here>

These settings can be change using the ?**SET** command. After the command we provide the setting and the value. Values can be of different types, numeric, Booleans and text. Any wrong usage will trigger an error with explanation about the correct usage. This explanation is more advanced compared to the other commands. Example:

- ?**SET** GiveOps 300
- ?**SET** DefaultTopic Welcome to #Centravi!

6.2. Adding Users

Obviously it is possible to add new users to the user list. The ability of adding new users depends on a few things.

- Both users their authserv account.
- Amount of access of the user using the command.
- Amount of access he is trying to give to the other user.
- Amount of access assigned to the **ADDUSER** command.
- Whether the account is suspended or not.

First of all the user using the commands needs to have **more** access then the amount of access assigned to the add user command. Second the user can't give equal or higher access to the other user then his own. Only the user using the commands needs to be logged in and recognized by Centravi. If the user is already on the list it won't be added. Last, neither the user trying to add nor the user who is being added can be suspended. Either way Centravi won't add the user. In case the user who is being added is suspended simply un-suspending would solve the issue. If however the user who is trying to add is suspended it's not possible.

Using the commands goes as follows: Me as owner, tries to add an account Victor to the user list with 100 access, we have two possibilities to do this. First is by providing the nickname and access level, second is by providing his authserv account and access.

- ?**ADDUSER** Victor 100
- ?**ADDUSER** *Victor 100

If we use the user his authserv account we provide a * before the name, this way Centravi knows we are providing an authserv account and not a nickname. If you provide a nickname Centravi will automatically track the user his authserv account.

6.3. Deleting Users

Deleting users is an easy task, but still necessary to explain a few. Also this depends on a few things.

- Authserv account of the user using the command.
- Amount of access of the user using the command.
- Amount of access of the user who is being deleted.
- Amount of access assigned to the **DELUSER** command.
- Whether the account is suspended or not.

Again only the user using the command needs to be logged in with Authserv and recognized with Centravi. The user using the command needs to have higher or equal access then the amount of access assigned to the **DELUSER** command. Also he needs to have more access then the users he is trying to delete. The user using the command cannot be suspended, the user who is being deleted can be suspended, and as long the access is right it does not matter.

Using the command goes as follows: I've just added Victor to the user list, now I want to delete him from there again. He is in the channel so I could simply use this command

- ?**DELUSER** Victor

If Victor is not in my channel, or not logged in to his authserv account, Centravi will reply an error and using this command won't delete Victor from my list. What I do is using the * we saw earlier. It goes as follows.

- ?**DELUSER** *Victor

It does not matter what his nickname is, or if he is on IRC, it does not even matter if his authserv accounts still exists (Could be deleted due inactivity). The bot will delete it from the list anyway.

Its also possible to delete ourselves from the user list. This cannot be done by simply writing ?**DELUSER**. For this we need to use the **?DELME** command. After we have done this, we'll get a confirmation code which we enter after writing the command once again. This has the same security reasons as giving away ownership.

6.4. Changing Access Level

Users most likely want to change the access level of specific users. You could delete the user, and re-add it afterwards. However you will need 2 commands for this. Changing the level using **CLVL** can be seen as a deletion and creation however I would not call it that way. Changing the user's level has the same requirements as deleting and adding both together. Only difference from deleting is that neither the user trying to change the level nor the user whose level is being changed can be suspended. With the command we only need to provide the new amount of access, not the old. Victor has 100 access, however he is a nice guy and I like him to have 200 so he can kick people. This is what I can do:

- ?**CLVL** Victor 200
- ?**CLVL** *Victor 200

Again the * in front of the name tells Centravi we are talking about Authserv Accounts.

6.5. Suspending Users

We've already spoken about it real quickly because it is in the requirements of both deleting and creating users. We could suspend users if you want to temporary "remove" their access instead of deleting it permanent. Using this command depends on a few things.

- Authserv account of the user using the command.
- Amount of access of the user using the command.
- Amount of access of the user who is being deleted.
- Amount of access assigned to the **SUSPEND** command.
- Whether the account is suspended or not.

The user using the command needs to be logged in to authserv and recognized with Centravi. Also the user needs to have higher or equal access then the access assigned to the

SUSPEND command. The User using the commands needs to have more access then the user he is trying to suspend. Suspending accounts goes as follows:

- ?**SUSPEND** Victor
- ?**SUSPEND** *Victor

If the user is already suspended you can of course unsuspend it as well.

- ?**UNSUSPEND** Victor
- ?**UNSUSPEND** *Victor

You'll probably get by know that the * in front of the name tells Centravi we are talking about Authserv Accounts.

6.6. Check Access

As you know we can use the **USERS** command to see all users and their access. What if we got 500 users and want to check 1 user his access. Receiving all other 499 would be a bit cumbersome would it not? Introducing the **ACCESS** or mostly wrote as **A** command. We can receive the amount of access of specific users by simply writing the following:

- ?**A** Victor
- ?**A** *Victor

No access is necessary for this command, if the channel owner allows any user to use commands like these in the channel (**PUBCMD** Setting). An more advanced command familiar to this one is the ?**MA** command. This shows our access we have in all channels. Centravi staff members can use this command on other people there account as well. Normal users can simply see this for their own account.

Besides checking access, it is possible to check how long the user has not been in the channel. Like access this can be found on the user list as well. However using the ?**SEEN** commands (again) prevents receiving an entire list with information you don't need.

6.7. Channel Ownership

As mentioned earlier, each channel has one owner. This owner has 500 access, and nobody else can have this amount of access. If the owner decides that he don't want to be owner of the channel anymore there is a possibility to give away his ownership to somebody. The only requirement is that the user gaining the ownership is a **valid** authserv account. We need to write 2 commands to be able to give away ownership. Since this command removes our access and makes some permanent changes. It asked for a confirmation after we wrote the following:

- ?**GIVEOWNERSHIP** Victor
- ?**GIVEOWNERSHIP** ***Victor

After running this command Centravi will give us some kind of code which you enter after the command we just wrote.

- ?**GIVEOWNERSHIP** Victor <code>
- ?**GIVEOWNERSHIP** ***Victor <code>

From that moment I'm not the owner of the channel anymore, Victor is. Centravi Staff members won't Un-Do this command since I wrote it myself and even confirm it afterwards.

6.8. Regain Access

It is possible that somehow our operator status in a channel is gone, even though we do have enough access for it (Might be someone just took it). We could rejoin the channel to let Centravi give it back to you. Or we just write **?UP**. This command gives back our Operator/Voice if our access is higher than access assigned to either one of these settings. We can do it the other way by writing **?DOWN**. This command removes our operator and/or voice in a channel if you have it.

This only affects one channel. Let's say we lost Operator in 5 channels, you would think we need to write this command in all 5 channels. For this we got the **?UPALL** command. This gives back our status in all channels where we lost it somehow. The other way around is the same as in a single channel by using **?DOWNALL**.

This should only work when we have enough access in a channel to gain either operator or voice. Suspended users are ignored.

6.9. Trim User List

It might be that a User List gets outdated after a while. In this case we could delete every user one by one. This takes time, which is wasted time. Using the **TRIM** commands helps. With this command we can provide an access range and "Last seen" (How long ago the user last where in the channel). Using this command depends on a few things.

- Authserv Account of the user using the command.
- Access of the user using the command.
- Access assigned to the **TRIM** command.
- Whether the account is suspended or not.

The user using the command needs to have more access than the access assigned to the **TRIM** command. Also this user needs to be logged in to Authserv and recognized by Centravi. And last the account of the user trying to trim the list cannot be suspended.

Trim Usage Here

6.10. Synchronize

Because the user list Centravi keeps is different from the real IRC protocol. Centravi protects against these differences as much as possible. It is however possible that some users don't have Operator status while they are able to have it. Might be they used **DOWN** or anything familiar. If a lot of users their channel statuses seem not equal to the access on the user list. Simply writing ?**RESYNC** would do the trick. This command does a few things to our channel:

- Removes Operator status from those who lack access.
- Removes Voice status from those who lack access.
- Gives Operator status to those who have enough access.
- Gives Voice status to those who have enough access.

The requirements for this commands is the same as with the **TRIM** Command. Except it used the access assigned to the **RESYNC** command.

6.11. Channel Invatations

IRC allows us to set a channel mode +i, which only let people join the channel on invite. There are a few kinds of invites to use for us. First we can easily invite users in our channel by using the **INVITE** command as follows:

- ?**INVITE** <User>

For using this command you need to be logged in to Authserv and recognized by Centravi. It is possible that we have access in specific channels where we are not in, and +i is set. In this case we can invite ourselves in 2 ways.

- ?**INVITEME** <channel>
  Invite to a specific channel
- ?**IA**
  Invite to all channels where you have auto invite enabled.

To be able to invite ourselves to a channel with the **INVITEME** command we need to have more access then the INVITEME setting on that channel. Also we need to be logged in to authserv and recognized with Centravi. For the IA command we need to have the just given to 2 requirements, including auto invite enabled in that channel.

6.12. Events

Each channel has events, such as commands used by users in that channel. It might be that the owner of a specific channel is not around for a while and when he gets back, some settings are changed. By using the **EVENTS** command, he'll see all events happened within the given criteria so he knows who used what command – changed the setting.

Events Usage Here

7. Channel Ban Management

We've already mentioned this earlier: The channel bans. IRC is for having fun, and if everyone would behave that way this part would not even been build. Sadly, it has to.

Banning users is used to prevent specific users or groups to be able to join the channel they are banned in. IRC uses hosts, each user got his own hostname with his name inside, this means those hosts are personal and as long we're logged in, we most likely have that host. Nevertheless users could still disable this feature and use the host of their internet provider. Either way the host is personal as long it's not used from public places and connections. Example, Victor who we recently added to the user list is being annoying and we want to get rid of him. We delete him from the list and ban his host _*!*@Victor.support.onlinegamesnet._ Now Victor lacks access to our channel and he can't join anymore. Victor is part of the OnlineGamesNet staff meaning he has a *.support.* host. Users on this network have a user's host like _*!*@Name.User.OnlineGamesNet._ The last one is the host from the internet provider or RDNS if enabled_: berend@b3r3nd.com_. If we would ban _*!*@*_ from our channel, it affects all hosts and nobody will be able to join. If we ban _*!*@*.support.*_ all staff members from the support staff like Victor are unable to join the channel.

The channel bans are stored in each specific channel. The Centravi bans are stored **separately** from this list. It could be that Centravi has bans set, and the channel has not synchronized them yet or anybody removed them from the channel manually without using Centravi. If the user joins, which is possible – he gets banned right away by Centravi. It is not hard to keep these 2 synchronized with each other as long the user uses Centravi to manage bans. Centravi will take care of the bans on the channel itself.

The bans can be seen by using the **BANS** command. This command shows the **Centravi** bans, not the channel bans. Using this command depends on a few things:

- Authserv Account.
- Access of the user using the command.
- Access assigned to the **BANS** command.
- Whether the account is suspended or not.

Of course the user need to be logged in with Authserv and recognized by Centravi. The user's access need to be higher or equal then the amount of access assigned to the **BANS** command and his account cannot be suspended. The ban list provides you with quite some information such as, banned host, banned by who, when is the ban triggered and why the host is banned.

Ban List Example

As earlier said, Centravi bans are different from channel bans. If we somehow manage to get all bans on a channel removed, we could manually set them all back or wait until that user tries to join and Centravi bans him automatically. Either way would be cumbersome. That's why there is a **BANALL** command. This commands synchronizes the bans on channel with the bans known by Centravi.

7.1. Banning Users

In order to ban users quickly all we need is a nickname from a user who is online in the channel. For this we could use the **KICKBAN** or **ADDBAN** command, the difference is that the **KICKBAN** command also kicks the user from the channel, the **ADDBAN** command simply bans the user. The **KICKBAN** command can only be used on users who are online. Otherwise the command will come up with an error. De **ADDBAN** commands is a bit more advanced an does different things depending on the given criteria.

If you provide the command with a host, (*!*@ inside the parameter) it will ban the host from the channel. If you provide a nickname from an online user, it automatically finds out the host name and bans the hostname, if you provide a nickname from a user which is not online, it bans the nickname from the channel and not the host.

Using these commands depends on a few things.

- Account of both users.
- Access of both users.
- Access assigned to the **ADDBAN** command.
- Whether the account is suspended or not.

The user using the commands needs to be logged in with his authserv account and recognized by Centravi. The account of the same user cannot be suspended. The user needs to have a higher or equal amount of access then the amount assigned to the **ADDBAN** command, also this user needs to have more access then the user he is trying to ban.

We use these commands as follows:

- ?**ADDBAN** Victor
- ?**KICKBAN** Victor
- ?**ADDBAN** *!*@Victor.*

7.1.1. Time Bans

There is a possibility that we want to ban users for a certain time, we could just set a band and keep track of the ban ourselves. However this would be cumbersome so we have the ability to set time bans with the **TIMEBAN** command. This command globally works the same as the **ADDBAN** command, only difference is that you provide a time as parameter. When that time is over, the ban is removed. We use this command as follows:

- ?**TIMEBAN** Victor 1d

As you can see we can use different time formats in the **TIMEBAN** command. If we would only provide minutes, and you want to ban someone for 1,5 month? Well, have fun counting. The following formats are available.

- 1y – 1 Year
- 1M – 1 Month
- 1w – 1 Week
- 1d – 1 Day

- 1h – 1 Hour
- 1m – 1 Minute
- 1s – 1 Second

We can use as much formats as we like, for example 1y2M4w1d1h10m1s. The use will be banned for a year, 2 months, 4 weeks, 1day, 1 hour, 10 minutes and 1 second. Useless but still possible.

7.2. Deleting Bans

Obviously it is possible to delete bans as well. This can be done ban by ban, matching criteria or all at once. Depending on what we want to do we should choose which way we will delete bans. Most simple version is just using **DELBAN**, provide the host as parameter and the ban will be removed.

- ?**DELBAN** *!*@Victor.*

If the user is online we could also specify the username, and it tracks down the host and (if it is banned) removes the ban on it.

We can also remove all bans at once with the **DELBANALL** command. This commands does not really need any further explanation. Simply write the commands and it removes all bans from the Centravi ban list.

Deleting bans means the ban is removed from both channel and Centravi. We might want to just delete it from the channel, but keep the ban in the Centravi ban list. We can use the **UNBAN** command for that, which works the same as the **DELBAN** command. Also we have a **UNBANALL** command which also works the same as the **DELBANALL** command.

All these commands above got the same requirements of usage as described with the **ADDBAN** command.

7.3. Unban Own Account

It is possible we got banned from a channel, if we got access in that channel, which is higher or equal then the amount of access assigned to the **UNBANME** command we can unban ourselves. We can send a private command to Centravi, telling him to unban us in that certain channel. Example:

- /msg Centravi **UNBANME** #Centravi

8. Channel Commands

We've described the channel management already, which manages the users. Along these commands are some other commands which make life easier doing the channel management. However these do (in our eyes) not belong to the channel management category. That's why we divided these in a separate category called: Channel Commands.

These commands also include help files. Because some users might not ready any guides, and especially won't read them totally. They can ask for help by each single command, with just a quick but clear answer. All commands can be found by the **COMMANDS** command. This command does not require any Authserv account or access levels. Neither does the **COMMAND** command. With this command (as just said) we can rather information about single commands:

- ?**COMMAND ADDUSER**

This will provide you with the usage of the **ADDUSER** command including some example and a little explanation. If we would like to know the members of the Centravi Staff: All current supporters, administrators and developers. We can use the **STAFF** command. To get more information about the Bot, such as version and its developers use **VERSION**. Both these 2 commands are combined in the **CREDITS.**

We have statistics for different parts of the bot, so we have the **INFO** which provides an user with information about a specific channel, the **STATS** command which shows statistics about the bot itself, **LANGUAGESTATS** which shows statistics about the bots languages.

Add

Examples

Here

8.1. Managing Channel Statussen

For all the following commands described below regarding the giving and taking from operator/voice, kicking users, changing modes, and changing the topic counts the following rule:

All commands require us to have an authserv account, logged in and recognized by Centravi. Also It requires us to have more access then the amount of access assigned to the command. For taking away operator status and voice you need to have more access then the person you are trying to de-op/de-voice.

As said it is possible to give and take operator status from users. It's simple and does not require that much explanation:

- ?**OP** Victor
- ?**OP** Victor Rena Berend
- ?**DEOP** Victor
- ?**DEOP** Victor, Rena, Berend

We can give and take operator status from multiple users at the same time if we want to.