Oxygene 2.0 beta user's manual

Last updated: Apr 10, 2008

- an all-round ADC slot rules and trigger bot for BCDC++ -

Oxygene2: let's your hub to get some fresh air!

Table of Contents

1. Preface

1.1 What is this about?

Oxygene2.lua is a Lua script for BCDC++. The original Oxygene was made to check some very basic rules on a Direct Connect hub and kick/notify the user if something is wrong. Lately, many other options added, so now it can even handle triggers which can do actions like responding to main chat messages and other events. It is developed since 2004, and the current Oxygene2 is made for ADC hubs, so you will need to run one to use this script. If you're still running an NMDC hub (like Aquila, YnHub or Ptokax) get Oxygene 1.6 instead. To disallow confusing between versions, the first released version of Oxygene2 gets the version 2.0.

After the op-team sets it up, it eases your everyday work: you don't need to verify the slot settings of your users manually anymore, and it allows you to make automatic events based on chat messages, a user's settings or on the current time.

1.2 License and copyright

Oxygene2.lua is licensed under the terms of GNU General Public License version 3. See license.txt for terms and conditions. This documentation is licensed under GNU Free Documentation License. See Appendix D for the full text of license.

1.3 Document syntax

...

2. Features

1) Works only hubs which you want it to work, this way you can enter as many hubs as you want with the script, it won't kick the users on the others ;)
2) Checks if Download/Upload bandwidth is shown in the Description field and checks the slot rules and the upload limit amount for that
    1. Able to process the numbers, don't matter if it's integer or float
    2. Rules are fully customizable
3) Advanced trigger functions adding possibility to check for non-wanted clients, respond for main chat actions, executing commands based on time and so (for example it can check any text in description field, or version number, can notice OpChat, or send a content of a text file in pm to a user or to the hub's main chat, it can send a particular message to the chat in every 20 minutes, etc.)
4) You can set up the protected users' list using exact or partial nick match or with regular expressions.
5) Logging to files "oxygene2_log.log" and "oxygene2_kicks.log"
6) Statistics about online users and client types
7) Able to kick selected users (selected nicks or a given amount) if they're violating the rules
8) Easy to translate to your own language using the Language files
9) Sure there is much more, just don't get in my mind now :)

3. Requirements

The script requires **BCDC++ 0.699c or newer** to run. It works only when you're connected to ADC hubs. It won't respond your commands on NMDC hubs.
You can get the latest version of BCDC++ from http://bcdc.dccafe.org (click on "English pages" at the bottom of the menu if don't understand the language :)) or http://utrum.dyndns.org:8000.

4. How to install?

Here you will see a short guide to let your new and fantastic script work. So after following the next two sections, you can start to discover the functionality of your brand new script.

4.1 Setting up the script

1) Download and install the latest version of BCDC++. It's **important** to set up a new client for the bot, **never use your own one** to run this script, it will drive you crazy since it sends and receives tons of private messages (moreover, it doesn't react for commands arriving from itself)!
2) Register an account for the bot with OP privileges or higher. Again, never use your own account, register a different one for the bot (for reasons, see above).
3) Decompress the archive containing the script and other files then copy everything to the "scripts" directory of your bot's BCDC++.
4) Open the **startup.lua** file using Notepad or any other text editor:
   1. Locate the lines beginning with **`dofile(`** in **startup.lua** which are used to load the scripts at the start of BCDC++. For example, you will find some lines similar to this:
      `dofile( DC():GetAppPath() .. "scripts\\formatting.lua" )`
      When you've found those **`dofile`**s (usually at the end of the file), add your own line to load Oxygene2 the following way:
      `dofile( DC():GetAppPath() .. "scripts\\oxygene2.lua" )`
   2. Then save **startup.lua**
5) Restart the bot's BCDC++ to make the script load then login with the bot-client to the hub.
6) Since the script only works on the hubs you specify, someone has to add the hub's address to the *allowed hubs* list. To do this, send the **`-chaddhub`**[1] command as a private message to the bot (from *your* own client).
7) Reconnect the bot to your hub to let it build its internal database which needs for functioning.
8) You are ready. From now on, when you want to run the bot, all you need to do is to start the BCDC++ it's running in then connect to your hub. For communicating with the bot, you can send private messages to it from a different clients. A few commands work on main chat too.

To use the script in the future, the only thing you need to do is to start the BCDC++ of the bot then join to the hub.

4.2 Choosing language

By default, after installation the script sends English messages to the users. To use other language, ensure that the appropriate language file and help file are also located in the "scripts" directory too (for example: oxygene2.lang.HUN.lua and oxygene2_help.HUN.txt), then use the **`-chset language HUN`** command to change the language. Don't forget to use the **`-chreload`** command after this to load the language file. Note that the settings are automatically saved, so you don't need to repeat this step until you delete the settings file.

[1] Alternatively, you can add any hub (not only the current one) if you specify a host: -chaddhub [host]

5. Commands which are recognized by the script

5.1 General

-help [command] or **-? [command]**
Usage: **`-help [command]`**, **`-? [command]`**
Description: Displays the help. If you specify a parameter, it displays a short help for the given command.
Works in: main chat, private chat
Only for: Operators on the allowed hubs

-chgetinfo <nick>
Usage: **`-chgetinfo <nick>`**
Description: Shows some info about a user. Mostly includes things you can read using your client, but this is more convenient than checking the tag, since you get the user's status against the current slot rules. If the user violates them, you can see the verbatim copy of the message he gets from the bot.
Command example: **`-chgetinfo JohnSmith`**
Works in: private chat
Only for: Operators on the allowed hubs
Example:

```
[22:12:00] <[OP]FleetCommand> -chgetinfo [OP]FleetCommand
[22:12:00] <Oxygene>
         ----------------------------------------------------------------------------------------------------
         Userinfo [[OP]FleetCommand]:
         ----------------------------------------------------------------------------------------------------
         Current SID: NTDT
         CID: 6FDJZXHGFTR6UH7ZGOFLPWF2HVTT2SOX5TRZBHQ
         IPv4: 88.209.207.79
         Description: [4M/256K] Toxic ;)
         Virtual tag: <++ 0.699,M:A,H:1/1/12,S:4>
         Bandwidth: 4096K/256K
         Operator: yes
         Shared: 119,44 GiB (128254658829 B)
         Files: 14397 shared. Average file size: 8,49 MiB
         Added: 2007.05.16. 19:58:51
         Not protected
         Slotrules: OK
         Points for slotrules violation: 0
         ----------------------------------------------------------------------------------------------------
```

-chstat
Usage: **`-chstat [full]`**
Description: Shows the user statistics including the number of slotrules violators, the Active/Passive users and users grouped by their client version. If you use the **`-chstat full`** command, the table includes the percentages of the client distribution too.
Answers for: private chat
Only for: Operators on the allowed hubs
Example:

```
[10:35:18] <[OP]FleetCommand> -chstat
[10:35:19] <Oxygene>
         ----------------------------------------------------------------------------------------------------------------------------
         General hub statistics                                                                                  Oxygene 2.0beta
         ----------------------------------------------------------------------------------------------------------------------------
         Autochecking active: yes
         Usercount: 262 users
         Bad slotrules: 233 users
         Average points: 78,76
```

```
        Active: 171
        Passive: 91
        ----------------------------------------------------------------------------------------------------------------------
        Detailed client statistics
        ----------------------------------------------------------------------------------------------------------------------
        ++ 0.69: 1
        ++ 0.694: 1
        ++ 0.695: 2
        ++ 0.696: 6
        ++ 0.697: 1
        ++ 0.698: 192
        ++ 0.699: 57
        ADCH++ v2.0.86-Release: 1
        Z++ 2.06: 1
        ----------------------------------------------------------------------------------------------------------------------
        Average points statistics
        ----------------------------------------------------------------------------------------------------------------------
        ++: 78,6
        ADCH++: 100
        Z++: 100
        ----------------------------------------------------------------------------------------------------------------------
OK
```

-chsay <chat_message>
Usage: **`-chsay <chat_message>`**
Description: This command allows our bot to "talk" in main chat. It does nothing else but repeat the message we want in main chat.
Answers for: private chat
Action goes to: main chat
Only for: Operators on the allowed hubs

5.2 Installation

-chaddhub [host]
Usage: **`-chaddhub [host]`**
Description: Adds the current or specified hub to the enabled hubs' list.
Important: After enabling the script, you must reconnect with the bot client to the hub!
Answers for: private chat
Only for: Any operator on the hubs where the script's running
Example: -chaddhub

-chrmhub [host]
Usage: **`-chrmhub [host]`**
Description: Removes the current or specified hub from the enabled hubs' list.
Answers for: private chat
Only for: Any operator on the hubs where the script's running
Example: -chrmhub

-chlisthubs
Usage: **`-chlisthubs`**
Description: Lists the addresses of the hubs where the script is currently allowed.
Answers for: private chat
Only for: Any operator on the hubs where the script's running
Example:

```
[11:00:15] <[OP]FleetCommand> -chlisthubs
[11:00:16] <Oxygene>
    ---------------------------------------------------------
    Currently allowed hubs:
```

```
----------------------------------------------------
adcs://dcwatch.dyndns.org:1984
----------------------------------------------------
```

5.3 Getting and setting configuration

-chgetconfig
Usage: **`-chgetconfig`**
Description: Shows some settings you can change
Answers for: private chat
Only for: Operators on the allowed hubs
Example:

```
[20:11:08] <[OP]FleetCommand> -chgetconfig
[20:11:09] <Oxygene>     -----------------------------------------------------------------------------------------
[20:11:09] <Oxygene>     Configuration                                                        Oxygene 2.0beta
[20:11:09] <Oxygene>     -----------------------------------------------------------------------------------------
[20:11:09] <Oxygene>     language                                                                          US
[20:11:09] <Oxygene>     opchat_name                                                                   OpChat
[20:11:09] <Oxygene>     chbandwidth                                                                        pm
[20:11:09] <Oxygene>     ulimitcheck                                                                         1
[20:11:09] <Oxygene>     trigcase                                                                            0
[20:11:09] <Oxygene>     inactivetime                                                                      120
[20:11:09] <Oxygene>     rulesurl                                          http://www.myhub.com/slotrules.asp
[20:11:09] <Oxygene>     triggers                                                                            1
[20:11:09] <Oxygene>     -----------------------------------------------------------------------------------------
[20:11:09] <Oxygene> OK
```

Explanation of config values:

- **language**
  The language which the script uses for displaying help and the messages for users. See 6.2 if you want to make your own translation. Use **-chreload** command after changing this value.
- **opchat_name**
  It's the name of the OpChat. You can make the script report anything you want to the opchat using some triggers.
- **chbandwidth**
  Sets the way of slot-checking (based on bandwidth indicated in the description field). Possible values: **off**, **pm** and **disconnect**. If you set to **off**, the script doesn't check the slot rules. The **pm** value makes it send a private message to the users if their settings don't fit the slot rules. The **disconnect** value kicks the users with bad slot rules using the current kick profile (see 5.5).
- **ulimitcheck**
  If set to **1**, then the script checks the upload bandwidth rules together with the slot rules. The user will notified only when **chbandwidth** is set to other than **off**.
- **inactivetime**
  Sets how much time (in seconds) should the bot wait after connecting to the hub before sending any private messages or kicking anyone because of the slotrules. This setting doesn't affect the trigger actions! If you don't have problem with entering the hub then you don't have to modify this variable.
- **rulesurl**
  Specifies the URL of the slotrules of the hub. This URL is sent at the end of every private/kick message which is sent because of slotrules.
- **trigcase**
  Enables or disables case-sensitive comparing in triggers. Set it to 0 if you want your triggers to be non case-sensitive; other than 0 to enable case-sensitivity.
- **triggers**

Enables or disables the triggers. Set it to **0** if you want to disable triggers, or to **1** if you want to enable them.

-chset <config_var> <value>
Usage: `-chset <config_var> <value>`
Description: Modifies the config values. You can modify any value which can be found in the config list (see -chgetconfig above). Please be careful when modifying them, since the script doesn't verify the values.
Answers for: private chat
Only for: Operators on the allowed hus
Example:

```
[14:26:22] <[OP]FleetCommand> -chset chbandwidth pm
[14:26:23] <Oxygene> [string][off >> pm] OK
```

-chreload
Usage: -chreload
Description: Loads the specified language file, then updates all userdata in order to storing the kick-messages on the new language. You only need to use this command after changed the **language** config value using the -chset command.

> Please note that this may take a while, even on a quite fast computer, the update can take a minute or even more :)

5.4 Checking bandwidth

5.4.1 Kicking and noticing users

-chnotice <nick> <pm/profile_name>
Usage: `-chnotice <nick> <pm/profile_name>`
Description: Sends a private message or kicks the user if violating the rules. Recommended to use it when you want to kick a special user for violating the rules. -chnotice can be used to disconnect/pm the user without waiting for a new $MyINFO from him/her.
If the second parameter is **pm**, the script sends a private message to the user. If you provide a profile name, the user will be kicked with that profile. If you send **default** as profile name, it kicks with the default kick profile.
Examples: `-chnotice [HUN]Sandor_vagyok default`
Answers for: private chat
Only for: Operators on the allowed hubs

-chcheckhub <max_num> [profile]
Usage: `-chcheckhub <max_num> [profile]`
Description: Checks the hub against slot rules and kicks at most **max_num** users using the given kick-profile. **profile** parameter is optional. If missing, the default kick-profile will be used (see 5.5).
Examples:

```
    -chcheckhub 200 ehub
    -chcheckhub 500
```

Answers for: private chat
Only for: Operators on the allowed hubs

If you want the users to be automatically notified or kicked for violating the rules, then set the **chbandwidth** config value to **pm** (for private messaging) or **disconnect** (for kicking). See 5.3.

5.4.2 Customizing slot rules

You can view and modify the slot rules using the **-chrules** command. There are various parameters you can use: **-chrules <list/add/rm/ulimit/clearall/restore> [params]**

Listing rules

To view the current slot rules, use the `-chrules list` command. You'll get a table similar to this:

```
[15:45:37] <[OP]FleetCommand> -chrules list
[15:45:38] <Oxygene>    ---------------------------------------------------------------------------------------------
[15:45:38] <Oxygene>    Bandwidth            min. slot   max. slot   rec. slot   max. hub   max-hub kick
[15:45:38] <Oxygene>    ---------------------------------------------------------------------------------------------
[15:45:38] <Oxygene>     Up to 56 kbps:      1           1           1           1          1
[15:45:38] <Oxygene>     Up to 64 kbps:      2           2           2           2          3
[15:45:38] <Oxygene>     Up to 96 kbps:      2           3           2           2          3
[15:45:38] <Oxygene>     Up to 128 kbps:     3           3           3           3          4
[15:45:38] <Oxygene>     Up to 160 kbps:     3           4           3           4          5
[15:45:38] <Oxygene>     Up to 256 kbps:     4           4           4           4          5
[15:45:38] <Oxygene>     Up to 384 kbps:     4           5           4           5          6
[15:45:38] <Oxygene>     Up to 512 kbps:     5           6           6           5          6
[15:45:38] <Oxygene>     Up to 768 kbps:     6           8           7           6          7
[15:45:38] <Oxygene>     Up to 1024 kbps:    6           8           8           6          7
[15:45:38] <Oxygene>     Up to 2048 kbps:    8           10          10          7          8
[15:45:38] <Oxygene>     Up to 4096 kbps:    11          20          20          8          9
[15:45:38] <Oxygene>     Up to 10240 kbps:   18          35          30          10         11
[15:45:38] <Oxygene>     Above that:         31          50          40          13         14
[15:45:38] <Oxygene>    ---------------------------------------------------------------------------------------------
[15:45:38] <Oxygene>    Upload limit config:
[15:45:38] <Oxygene>    Threshold: 60 % - Min. limit: 12 KiB/sec - Applied for: 128 kbps and above
[15:45:38] <Oxygene>    Total: 14 items
[15:45:38] <Oxygene>    ---------------------------------------------------------------------------------------------
[15:45:38] <Oxygene> OK
```

The **min. slot** and **max. slot** column shows how many slots should be opened for the given upload bandwidth. The **rec. slot** column is a recommendation for the user. If the slot number is out of the min-max interval, then the user gets notified (if bandwidth-checking is set to pm or disconnect. See 5.4.1 and 5.3) that it's recommended to open <rec. slot> slots according to the hub rules, so the bot asks him/her to open at least <min. slot> and at most <max. slot> slots to ensure he/she can stay in the hub.
The **max. hub** column contains the maximum number of hubs according to the rules. The latest, **max-hub kick** column provides some allowance for that since the script kicks/pms the user only after he/she overrun this number of hubs. The difference between the two columns is that the user will be notified after the **max-hub kick** column number of hubs, but the script will tell him/her to be only **max. hub** number of hubs.

The later part of the table shows the current upload limit rules. The users should set their upload limit at least <Threshold> percent of their nominal upload bandwidth, but minimum <Min. limit> KiB/sec.
The upload rules not applied to everyone but the users with the indicated minimum upload bandwidth. By default, it's set to 64 kbps. It means that users below 64 kbps upload bandwidth are not checked against rules. This protects the dial-in modem users and users with extremely bad

connection.

> The slot rules are only applied if **chbandwidth** config value is set to pm or disconnect. See 5.3 for details.

Adding and modifying a rule

For adding a new bandwidth-rule or modifying an old one, you should use the following command:
`-chrules add <upper_bandwidth> <minslot> <maxslot> <slotrec> <maxhub> <maxhub_kick>`
Where:
<upper_bandwidth> is the maximum bandwidth in kbps which the rule is applied for. Specify -1 if you want to add/modify the rule called "Above that". The -1 is special, because that will be applied to everyone who specifies larger upload bandwidth that the numbered element of the list.
<minslot> and **<maxslot>** is the minimum and maximum number of slots which the user should open not to be noticed/kicked. The **<slotrec>** is a slot recommendation (between the boundaries of min and max-slot). The script states the user that it's recommended to open <slotrec> slots, but he/she should keep the slot number between <minslot> and <maxslot> to be not kicked.
<maxhub> is the maximum numbers of hub where the user can stay at the same time.. but he/she won't be kicked until overruns the **<maxhub_kick>** number of hubs.

> Only the plain hub number and registered hub number is counted into the <maxhub> number, the hubs where the user is op are not.

You should use the same command to add a new bandwidth and to overwrite an existing one. In the latter case you should specify a bandwidth which already added to the rules.

Example:

```
[18:53:19] <[OP]FleetCommand> -chrules add 512 5 6 6 5 6
[18:53:19] <Oxygene> [overwrite] OK
```

Removing rule or rules

You have two options to remove rules. You can remove rules one by one, by using the **-chrules rm <bandwidth>** command, where the bandwidth is the upper bandwidth which is listed in the rules table.
You may remove all rule by clearing the whole table. For this, you can use the **-chrules clearall** command.

Examples:

```
[18:56:04] <[OP]FleetCommand> -chrules rm 512
[18:56:04] <Oxygene> OK

[18:56:13] <[OP]FleetCommand> -chrules clearall
[18:56:14] <Oxygene> OK
```

Restoring default rules

To restore the default slot rules table use the **`-chrules restore`** command. That will clear and recreate your existing slot rules table with the default values including the bandwidth rules.

Example:

```
[18:59:13] <[OP]FleetCommand> -chrules restore
[18:59:14] <Oxygene> OK
```

> When you add/modify/remove a rule or clear/restore the slot rules table, the script automatically disables the automatic bandwidth checking for the users' protection (you'll be notified). To re-enable, use the -chset command. See 5.3

Setting upload limit rule

You can set the upload limit rule with the **`-chrules ulimit <percent> <min_limit> <applied_from>`** command. The users should set their upload limit to <percent> percent of their upload bandwidth, but at least **`<min_limit>`** KiB/sec. It's only checked if the upload bandwidth of the user is equal or greater than **`<applied_from>`** kbps.

> The upload limit is checked only if **ulimitcheck** config variable is set to 1. See 5.3 for details.

5.5 Kick profiles

Kick profiles provides different modes for kicking. The script includes some preset for different types of hub software. For instance, you can use them to kick users for different interval than the default one or simply just disconnect them and more.
Two different profiles are known. The **chat**-type profile is a command which is sent to the mainchat of the hub. You can use it when your hub contains a mainchat-command for kicking. The **raw**-type profiles are commands which fits the NMDC protocol. The default profile named RawKick is equivalent with the right-click Kick-command included in DC++. You should use it if your hub doesn't contain any chat-command for kicking or banning. Of course, if you know NMDC protocol, you can add more raw-profiles by yourself.
The **`-chprofiles`** command handles the profiles: **`-chprofiles <list/add/rm/setdefault>`**. Discussed below.

5.5.1 Listing existing profiles

For listing the existing profiles, use the **`-chprofiles list`** command. You get a table like this:

```
[08:23:02] <[OP]FleetCommand> -chprofiles list
[08:23:03] <Oxygene>        ------------------------------------------------------------------------------------------------------------------------
[08:23:03] <Oxygene>        Profile name [type] (Command)
[08:23:03] <Oxygene>        ------------------------------------------------------------------------------------------------------------------------
[08:23:03] <Oxygene>         (1) rawexample [r] (BMSG %[mySID] +kick\\s%[userNI]\\s%[reason])
[08:23:03] <Oxygene>         (2) adchpp [c] (+kick %[userNI] %[reason])
[08:23:03] <Oxygene>        ------------------------------------------------------------------------------------------------------------------------
[08:23:03] <Oxygene>        Default profile: adchpp
[08:23:03] <Oxygene>        Total: 2 items
[08:23:03] <Oxygene>        ------------------------------------------------------------------------------------------------------------------------
[08:23:03] <Oxygene> OK
```

The profile type **r** means raw-kick profile, type **c** means chat-kick profile. See 5.5 for details.

5.5.2 Adding/removing profiles

To add a new profile, use the **-chprofiles add <profile_name> <profile_type> <command>** command where **<profile_name>** is the desired profile name, **<profile_type>** is the type. Use **c** for chat-kick profile, or **r** for raw-kick profile. **<command>** is the command used for kicking in that profile. It will be sent to main chat when the type is **c** or sent directly to the hub when it's **r**. You should know some ADC specification when using **r** profiles. In **<command>** parameter you can use the variables described in Appendix B.

> When adding a raw profile, you can use \n for command separator and should use the escaped form of ADC escape sequences like \\s for \s, \\n for \n. You don't have to add \n after the last command.

Example:

```
[08:30:41] <[OP]FleetCommand> -chprofiles add ehub4h c +kick %[userNI] %[reason] _BAN_4h
[08:30:42] <Oxygene> OK
```

To remove a profile, use the rm parameter for the -chprofiles command the following way: **-chprofiles rm <profile_name>** where **<profile_name>** is the profile name which you want to remove.

Example:

```
[08:31:28] <[OP]FleetCommand> -chprofiles rm ehub4h
[08:31:29] <Oxygene> OK
```

5.5.3 Selecting default profile

If you select a profile for default profile, that will be used everywhere where indicated (for example if you set the **chbandwidth** config value to **disconnect**). To do this, use the **-chprofiles setdefault <profile_name>** command.

Example:

```
[08:39:52] <[OP]FleetCommand> -chprofiles setdefault ehub4h
[08:39:52] <Oxygene> Profile 'ehub4h' selected
```

5.6 Triggers

With triggers, the bot can automatically react for any given event. Triggers can be activated by main chat messages/commands, by any incoming userdata (for example, e-mail-address, description, nick, etc.) or by timer.

It offers advanced possibility to be used for checking non-wanted clients on the hub, or reacting to any user action – for example – by kicking, private messaging (a custom message or a content of a file), noticing OpChat, or simply answering in main chat.

Trigger commands which are recognized by the script:

```
-chtrigs <list/addtrig/rmtrig/addc/rmc/adda/rma/enable/settype>
[params]

-chtrigs list [all/tirgger_name]
-chtrigs addtrig <trigger_name> [type] [interval]
-chtrigs rmtrig <trigger_name>
-chtrigs addc <trigger_name> <variable> <condition> <value>
```

```
-chtrigs addcondition <trigger_name> <variable> <condition>
<value>
-chtrigs rmc <trigger_name> <C_num/all>
-chtrigs rmcondition <trigger_name> <C_num/all>
-chtrigs adda <trigger_name> <action> <parameters>
-chtrigs addaction <trigger_name> <action> <parameters>
-chtrigs rma <trigger_name> <A_num/all>
-chtrigs rmaction <trigger_name> <A_num/all>
-chtrigs enable <trigger_name> <+/->
-chtrigs settype <trigger_name> <and/or>
-chtrigs setinterval <trigger_name> <interval>
-chtrigs reset <trigger_name>
```

All of them are discussed in detail below.

5.6.1 Listing triggers

You can list the currently added triggers by using the **`-chtrigs list [all/trigger_name]`** command. If you don't specify the trigger's name, you'll get a table containing all of their names. It looks like this:

```
[08.58:11] <[OP]FleetCommand> -chtrigs list
[08.58:12] <Oxygene>     ------------------------------------------------------------------------------------------------------------------
[08.58:12] <Oxygene>     [Enabled] (Cnt) Name
[08.58:12] <Oxygene>     [Conditions/Actions]
[08.58:12] <Oxygene>     ------------------------------------------------------------------------------------------------------------------
[08.58:12] <Oxygene>      [+] (1) UserCount
[08.58:12] <Oxygene>      [-] (2) Faq
[08.58:12] <Oxygene>     ------------------------------------------------------------------------------------------------------------------
[08.58:12] <Oxygene>     Total: 2 (of 2) triggers listed
[08.58:12] <Oxygene>     ------------------------------------------------------------------------------------------------------------------
[08.58:12] <Oxygene>     Note: Use -chtrigs list all command to list all triggers detailed, or -chtrigs list <trigger_name> to see a specified trigger
[08.58:12] <Oxygene>     ------------------------------------------------------------------------------------------------------------------
[08.58:12] <Oxygene> OK
```

You can see two triggers in the above example. If you want to see what the triggers do, you should list them individually (**`-chtrigs list UserCount`** for example) or you can list all triggers detailed by one command (using **`-chtrigs list all`**). To see the UserCount trigger, you should try the following command:

```
[09:09:00] <[OP]FleetCommand> -chtrigs list UserCount
[09:09:01] <Oxygene>     ------------------------------------------------------------------------------------------------------------------
[09:09:01] <Oxygene>     [Enabled] (Cnt) Name
[09:09:01] <Oxygene>     [Conditions/Actions]
[09:09:01] <Oxygene>     ------------------------------------------------------------------------------------------------------------------
[09:09:01] <Oxygene>      [+] (1) UserCount
[09:09:01] <Oxygene>      [C1] user is op
[09:09:01] <Oxygene>      [C2] %[chat] is -usercount
[09:09:01] <Oxygene>      [A1] mainchat: oxygene.lua %[version]: %[usercount] users (%[opcount] operators)
[09:09:01] <Oxygene>          C: 2, A: 1, T: and, Interval: 0, Activated 9 times since 07. 09. 2006. - 10:46:06
[09:09:01] <Oxygene>     ------------------------------------------------------------------------------------------------------------------
[09:09:01] <Oxygene>     Total: 1 (of 2) triggers listed
[09:09:01] <Oxygene>     ------------------------------------------------------------------------------------------------------------------
[09:09:01] <Oxygene> OK
```

The trigger names are highlighted with blue. The [+] and [-] signs before them indicates whether they are enabled or disabled. A trigger consists of conditions (marked with [C1], [C2], ...) and actions ([A1], [A2], ...). If the trigger's type (T:) is “and-trigger” and **all** conditions are met (and the trigger is enabled), the script executes the actions for that trigger. If it's an “or-trigger” then the actions are executed if **any** of the conditions met. If you have the UserCount trigger, as in the example above, you can execute it if you are an operator (Condition2: user is op) and enter the **-usercount** command to main chat (Condition1: %[chat] is -usercount):

```
[19:58:36] <[OP]FleetCommand> -usercount
[19:58:36] <Oxygene> oxygene.lua 1.5a: 4108 users (9 operators)
```

It works this simply :) As you can see, variables like %[usercount] can be used. Discussed later. You can also see the Interval property in the trigger's detailed view. That tells you how much time (in seconds) must be spent between two execution of the trigger. It's useful if the trigger is intended to do main chat actions and you want to limit the number of answers. For example, if you make a trigger which posts some URL to the main chat, you may want to set it to 60 or 120 seconds, so the users can't play with the bot. This also can protect the hub against flooding.

5.6.2 Enabling/Disabling triggers

Enabling/Disabling all triggers at once

You can enable/disable all triggers generally by setting the triggers config value using the **`-chset`** command (see 5.3). If you want to enable the triggers, just say **`-chset triggers 1`**, if you want to disable them, use the **`-chset triggers 0`** command to private chat. You can check the current state by using the **`-chgetconfig`** command. See 5.3 for more details. Note that this won't enable the triggers which are disabled individually, just enables the trigger processing itself.

Enabling/Disabling triggers one by one

You can enable/disable triggers one by one if you want. Use the **`-chtrigs enable <trigger_name> <+/->`** command for that. If you want to enable the trigger “Faq” mentioned in the 5.6.1 example, then you can do it with the **`-chtrigs enable Faq +`** command.

5.6.3 Adding a new trigger, condition and action

Adding and setting up a trigger consist of three steps. First, you have to add the trigger itself. Then you have to add the conditions and actions. When you're ready, just simply enable the trigger using the **`-chtrigs enable`** command, because all trigger is disabled by default to avoid non-wanted reactions while it's not set up correctly. We'll set up the UserCount trigger through the following example.

Adding a trigger

To add a trigger, you should use the **`-chtrigs addtrig <trigger_name> [type] [interval]`** command, where the name of the trigger can't contains spaces. The **`[type]`** parameter is optional. It specifies whether the trigger should be an “or-trigger” or an “and-trigger”. Possible values: **`and`**, **`or`**. If not given, **`and`** is applied. The **`[interval]`** is also optional, it sets the minimum time in seconds between two execution of the trigger (recommended to use it if the trigger posts messages to main chat). Example:

```
[08:52:00] <[OP]FleetCommand> -chtrigs addtrig UserCount and
[08:52:00] <Oxygene> "UserCount" added. Use -chtrigs addc or -chtrigs addcondition to add a condition then -chtrigs adda or -chtrigs addaction to add an action to it.
[08:52:00] <Oxygene> OK
```

Adding a condition

Now we have an empty trigger, which can do nothing. To make it do something, we have to add some condition using the **`-chtrigs addcondition`**[2] **`<trigger_name> <variable> <condition> <value>`** command, where **`<trigger_name>`** is the name of the trigger which you want to add the condition to. The condition is built from the remaining three parameters. The possible values for the last three parameters are:

Possible <variables>	***Possible <conditions>***	***Possible <values>***	***Trigger won't be activated by***[3]
Regular variables and values			
%[userNI], %[userNIshort], %[userDE], %[userEM], %[client_type], %[tagM], %[tagVE], %[connection], %[userSSshort]	is, isnot, contains, ncontains, similar_to, nsimilar_to, begins_with, nbegins_with, ends_with, nends_with	<any string value>	Timer, if any present; INF if all is missing (along with %[tagV]...);
%[chat]			INF, Timer if present; MSG if missing
%[date], %[time], %[fulldate]			Timer if missing (along with %[hour], %[min], %[sec])
%[myNI], %[mySID], %[myCID], %[version], %[usercount], %[opcount]			No limit
%[tagV], %[tagHN], %[tagHR], %[tagHO], %[tagSL], %[tagO], %[tagB], %[userSS], %[upBW], %[downBW], %[points], %[pointsint]	==, ~=, <, <=, >, >=, between, outof[4]	<any num value>, <min>-<max> for a "between" and "outof" condition	Timer if any present; INF if all is missing (along with %[userNI], etc..)
%[hour], %[min], %[sec]			Timer if missing (along with %[date], %[time], %[fulldate])
Special variables and values			
user	is, isnot	op	Timer if present

The **`<variable>`** must be chosen from the first column, then a condition follows from the second one, and finally you should provide a value. For example: **%[chat] is hello world!**

> You can find the meaning of all regular variables in **Appendix A.**

Regular conditions for numeric variables:

- ==: (that is, a double equal mark) means equal; example: **%[tagV] == 0.674**
- **~=:** not equal; example: **%[hour] ~= 11**
- **<, <=, >, >=:** quite obvious; example: **%[userSS] > 1000000000**
- **between**: matches if the content of the variable is between the two given values (boundaries also matches); example: **%[min] between 10-15**
- **outof:** the opposite of the between condition. The boundaries don't match; example: **%[hour] outof 20-04**

[2] A shorter version of this command is: -chtrigs addc

[3] INF = incoming userinfo, MSG = main chat message, Timer = timer

[4] The old version of the script used "smaller_than", "larger_than", etc. conditions. The old triggers still work with this version, the only restriction that you have to use the new syntax when you add a new trigger/condition.

Regular conditions for string variables:

- **is, isnot**: plain text match; true, if the **<variable>** is the same, as your condition (or differs, for **isnot**)
- **contains:** partial plain text match; true if the **<variable>** contains your condition (**ncontains** would be true if it doesn't contain it)
- **begins_with, ends_with:** plain text match; true if the **<variable>** begins/ends with your condition (**nbegins_with, nends_with** would be true if the beginning/end of text don't match your condition)
- **similar_to, nsimilar_to:** regex match; true if your condition (a search pattern) matches (or doesn't match for nsimilar_to) to the content of the variable

> The string comparison is case-sensitive if you set the **trigcase** config variable to other than 0. See 5.3.

> The value also can include any regular variable. For example: %[upBW] > %[downBW]; See **Appendix A** for the meaning of the variables.

Special conditions and values:

- **user is op** is true when the user has operator rights on the hub.

Example:
For our previous UserCount trigger, we should set two conditions. We'll set it to answer only for operators, and we will set the command. For this, we should use the following two commands:

```
[09:05:17] <[OP]FleetCommand> -chtrigs addcondition UserCount user is op
[09:05:18] <Oxygene> OK
[09:05:29] <[OP]FleetCommand> -chtrigs addcondition UserCount %[chat] is -usercount
[09:05:29] <Oxygene> OK
```

Adding an action

Now we have conditions, but what our trigger's gonna doing? At this moment, it can do nothing usable. So we should set at least one action what should be executed when all the conditions are met. For this, we use the **-chtrigs addaction**[5] **<trigger_name> <action> <parameter>** command. **<trigger_name>** is simply means the name of the trigger we would like to add the action to. The parameter **<action>** and **<parameter>** tell the trigger what to do exactly.
The possible actions are:

<action>	***<parameter>***
kick	<kick reason>
redirect	<new hub> <reason>
command	<command to execute>
mainchat	<mainchat message>
mainchatfile	<filename, relative to the application path>
pm	<private message>
pmfile	<filename, relative to the application path>
rxmainchat/rxpm/rxopchat	"<Txt>"; "<SearchPattern>"; "<ReplaceString>"
opchat	<the message to OpChat>

[5] A shorter version of this command is: -chtrigs adda

Explanation of actions:

- **kick** kicks the user, using the current kick-profile.
- **redirect** will redirect the user to the given hub with the provided message.
- **command** will execute any command which the script can parse (for example: "-chcheckhub 200"). The status messages will be posted to OpChat (if set in config)
- The **mainchat** and **pm** actions send a message to main chat or to the user in pm.
- **pmfile** sends the content of a file in pm (like the help file).
- **mainchatfile** sends the content of a file to the hub mainchat
- **opchat** action sends a message to the OpChat. OpChat's nick can be set with the OpChat config variable by using the `-chset` command (see 5.3.).
- **rxmainchat** sends message to mainchat (like **mainchat** action) with the difference it can use regexes to modify the original message (mostly the original chat message). It needs three parameters, all of them in quotes, followed by a ";". The first parameter is the text to modify. The second is the Search pattern, then, the third is the Replace String. It replaces the matches according to the provided rules. For example you can add an action like this: "%[chat]"; ".*give me an? (.*)"; "Hi %%[userNIshort], do you really need a %1?". Notice that you have to use double percent-marks in ReplaceString, because % has a special meaning there.
- **rxpm** sends a pm using regexes. See **rxmainchat** for more details.
- **rxopchat** sends a message to the OpChat using regexes. See **rxmainchat** for more details.

> The parameters and even the files can include regular variables. For example: **mainchat Hi %[userNIshort], you have %[userSS] Bytes in your share!** See Appendix A for the meaning of the variables.

So now, we add the action to our UserCount trigger:

```
[17:53:23] <[OP]FleetCommand> -chtrigs addaction UserCount mainchat oxygene.lua %[version]: %[usercount] users (%[opcount] operators)
[17:53:24] <Oxygene> OK
```

We are ready. But remember, all triggers are disabled by default. So the only thing we need to do is to enable the trigger using the following command:

```
[17:54:29] <[OP]FleetCommand> -chtrigs enable UserCount +
[17:54:30] <Oxygene> OK
```

5.6.4 Modifying trigger type

Although you could decide whether to create an or-trigger or an and-trigger when running the **-chtrigs add** command, but sometimes you can change your mind later. This is what **-chtrigs settype** command's for. It changes the type of the trigger. You have to provide the name of the trigger as first parameter, and the desired type as second one. For example, if you have a trigger "Antiadv" and you want to change its type to an "or-trigger", you could simply use the **-chtrigs settype Antiadv or** command.

5.6.5 Modifying the minimal execution frequency

The Interval property of the trigger (see 5.6.1) tells the program the minimal time interval between two execution of the trigger in seconds. If it's set to 0, then there's no limit in the execution frequency. You can modify it using the -chtrigs setinterval <trigger_name> <interval> command. The first parameter is the name of the trigger, the second parameter provides the interval.

For example, if you have a **Hublist** trigger, you can set the minimal execution frequency to 2 minutes using the **-chtrigs setinterval Hublist 120** command.

> Note that program ignores the Interval property if the trigger is activated by an operator. This means that an operator can execute the script as often as (s)he likes.

5.6.6 Reset trigger execution counter

5.6.7 Removing a trigger

You can simply remove a trigger by using the **-chtrigs rmtrig <trigger_name>** command.
Example:

```
[18:45:33] <[OP]FleetCommand> -chtrigs rmtrig TestTrigger
[18:45:34] <Oxygene> OK
```

5.6.6 Removing conditions/actions

You can remove any condition and action with the **-chtrigs rmcondition**[6] **<trigger_name> <C_num/all>** and **-chtrigs rmaction**[7] **<trigger_name> <A_num/all>** commands.
Both of them work similar. The first parameter for the -chtrigs command is rmcondition or rmaction depending on the desired action. The second parameter is the name of the trigger which the C/A is expected to be removed from. The last parameter is the counter of the removable condition:

```
[13:50:38] <Oxygene>         [+] (1) UserCount
[13:50:38] <Oxygene>         [C1] %[chat] is -usercount
[13:50:38] <Oxygene>         [C2] user is op
[13:50:38] <Oxygene>         [A1] mainchat: oxygene.lua %[version]: %[usercount] users (%[opcount] operators)
[13:50:38] <Oxygene>             C: 2, A: 1, T: and, Interval: 0
```

In the UserCount trigger there's two conditions (C1 and C2), and 1 action (A1). To remove Condition2, use the **-chtrigs rmcondition UserCount 2** command. If you want to remove all condition/action, provide `all` as condition/action counter.
Examples:

```
[10:19:21] <[OP]FleetCommand> -chtrigs rmaction UserCount 1
[10:19:22] <Oxygene> 1 action(s) removed successfully.
[10:19:22] <Oxygene> OK
[10:19:30] <[OP]FleetCommand> -chtrigs rmcondition UserCount all
[10:19:30] <Oxygene> 2 condition(s) removed successfully.
[10:19:30] <Oxygene> OK
```

5.7 Set up user exceptions (protected users)

Maybe you want to protect a user or a group of users against kicking (because of slotrules) for example, or maybe you want to provide the right to some users to ignore the Trigger Intervals, so they can activate your triggers as often as they'd like to. This is what the exception feature is for: to modify the default user rights.

You can add the following rights:

- **Kick protection:** The bot won't kick the users if they are protected (no matter if it's because of the slot rules or the triggers)
- **Ignoring slot rules:** The bot won't kick or pm those users who have this right because of any slot rules (including the bandwidth indication in the Description field, the number of slots, hubs, upload limit...).

[6] A shorter version of this command is: -chtrigs rmc

[7] A shorter version of this command is: -chtrigs rma

- **Ignoring all triggers:** If any user have this right, they can't activate any triggers. This can be both good and bad. It depends on what kind of triggers do you have.
- **Ignoring trigger time intervals:** The users who have this right can activate the triggers anytime, and doesn't matter whether the trigger have any limit in the time between two execution or not.

To add/modify/remove or list exceptions, you can use the **-chprotect <list/add/rm>** command

5.7.1 Listing exceptions

To list the table of exceptions, use the **-chprotect list** command:

```
[14:43:32] <[OP]FleetCommand> -chprotect list
[14:43:33] <Oxygene>      -------------------------------------------------------------------------------------------------------------------
[14:43:33] <Oxygene>      Nick patterns
[14:43:33] <Oxygene>      [M:Mode][K:Protect against kick][S:Ignore Slotrules][A:Ignore all triggers][T:Ignore TrigInterval]
[14:43:33] <Oxygene>      -------------------------------------------------------------------------------------------------------------------
[14:43:33] <Oxygene>      OpChat [M:2][K:1][S:1][A:0][T:0]
[14:43:33] <Oxygene>      ^%[VIP%] [M:0][K:1][S:1][A:0][T:1]
[14:43:33] <Oxygene>      -------------------------------------------------------------------------------------------------------------------
[14:43:33] <Oxygene>      Modes: [0: regex][1: partial match][2: exact match][3: CID]
[14:43:33] <Oxygene>      Total: 2 items
[14:43:33] <Oxygene>      -------------------------------------------------------------------------------------------------------------------
```

The above table contains two examples. First, you can see the nick pattern. Its behavior is defined by the mode property. It has the following meaning:

- mode0: The nick pattern is a regular expression (yes, if you use this, you need to have some knowledge of it). If that regex matches to the nick, the properties will be applied to the user. I recommend to use it to provide rights for a group of users.
- mode1: Partial match (plain text): This matches when the pattern can be found in the user's nick. You can also use this to provide rights for a group, but be careful, you can't use any modification character, and keep in mind that the match can happen **anywhere** in the nick!
- mode2: Exact match (plain text): This only matches when the nick and the pattern are exactly the same. You can use it to modify only one user's rights.
- mode3: CID

The other properties (K, S, A, T) can have two values: 0 means disabled, 1 means enabled.

- [K: Protect against kick]: When enabled, the script won't kick the user if violating the slot rules or if some trigger wants to do so.

  > Note that the kick-protection only protects against the kicking from **inside** the script. If any other Operator wants to do so, he or she can kick the user from his/her own client's Kick user(s) command.

- [S: Ignore Slotrules]: When enabled, the script won't kick and notify the user if violating the slotrules.
- [A: Ignore all triggers]: When enabled, no script will be activated by the user
- [T: Ignore TrigInterval]: When enabled, the minimal time between two execution of a trigger will be ignored for that user. This means, that anyone who has this right can activate any trigger as often as he or she wants.

  > Note that all Operators have this right by default, so you don't have to set it up for them.

5.7.2 Adding an exception

To add an exception, use the **-chprotect add <pattern> <mode> <kickprotection> <ignore slot rules> <ignore all triggers> <ignore trigger timing>** command where:

- **<pattern>** is the nick pattern, can't contain spaces
- **<mode>** can be 0 for regex, 1 for partial matches or 2 for exact matches, 3 for protecting a CID
- the other parameters can be 0 for disabled, or 1 for enabled

See 5.7.1 for more details.

Example:

```
[14:43:28] <[OP]FleetCommand> -chprotect add ^%[VIP%] 0 1 1 0 1
[14:43:29] <Oxygene> OK
```

5.7.3 Removing an exception

To remove an exception, simply use the -chprotect rm <pattern> command. To list them, use -chprotect list (see 5.7.1).

6. Examples

6.1 Trigger examples

Here you can find some examples to see how triggers work. If you made a nice one which can show any new to the readers, please feel free to send me. See Section 7.

6.1.1 Question/Exclamation mark quote

```
[16:41:00] <Oxygene>        [+] (1) QuestionMark
[16:41:00] <Oxygene>        [C1] %[chat] similar_to [?!][?!][?!][?!][?!][?!][?!]
[16:41:00] <Oxygene>        [C2] user isnot op
[16:41:00] <Oxygene>        [A1] kick: %[userNIshort] I think you ran out of your exclamation/question-mark quote...
[16:41:00] <Oxygene>            C: 2, A: 1, T: and, Interval: 0
```

Description: This trigger kicks every user who writes at least seven Q/E mark into his/her post.

Explanation: similar_to do a regex-check in the string (chat). The square-brackets are character-classes. So [?!] matches both question mark and exclamation mark. %[userNIshort] is the nick of the user without any [pre](fix)inHisNick

6.1.2 Hublist-trigger

```
[15:48:52] <Oxygene>        [+] (1) Hublist
[15:48:52] <Oxygene>        [C1] %[chat] similar_to ^.hublist
[15:48:52] <Oxygene>        [A1] mainchatfile: scripts/hublist.txt
[15:48:52] <Oxygene>            C: 1, A: 1, T: and, Interval: 120
```

Description: This is a useful trigger for the users asking for hublists. You should create a file named hublist.txt, then put it in your scripts directory (this is what the scripts send: “mainchatfile: scripts/hublist.txt”). If someone enters the ?hublist, -hublist, #hublist, etc. command into main chat, the trigger sends the content of hublists.txt to him/her as a mainchat message.

The interval: 120 property tells the program not to execute the trigger within two minutes. This protects against flooding. If someone would ask so often for a hublist, she/he may should read back the main chat.

Explanation: The “^.hublist” matches for any command which starts with any char followed by the “hublist” text since the dot character can be substituted with any character. The ^ means that it should state at the beginning of the line.

6.1.3 Time-based answer

```
[23:38:56] <Oxygene>        [+] (1) test
[23:38:56] <Oxygene>        [C1] user is op
[23:38:56] <Oxygene>        [C2] %[hour] outof 20-06
[23:38:56] <Oxygene>        [C3] %[chat] contains bier
[23:38:56] <Oxygene>        [A1] mainchat: We don't serve bier at daylight...
[23:38:56] <Oxygene>            C: 3, A: 1, T: and, Interval: 0
```

Description: This example shows a time-based trigger. If some operator would ask for a bier in the morning, the trigger answers him/her “We don't serve bier at daylight...”

Hour outof 20-06: matches if the current time if between 7:00-19:59. Note that 6:55 is still **between** 20 and 6. since the condition only examines the current hour, which is “six” and not “six o'clock”.

6.1.4 Timer-based execution

```
[08:14:22] <Oxygene>        [+] (1) Autokick
[08:14:22] <Oxygene>        [C1] %[time] similar_to ..0000
[08:14:22] <Oxygene>        [A1] command: -chcheckhub 200 verli_10m
[08:14:22] <Oxygene>            C: 1, A: 1, T: and, Interval: 0
```

Description: This trigger tries to kick 200 users in every hour.

Time similar_to ..0000: the “time” contains the current time in hhmmss format. The similar_to makes a regular expression match for it. The point can be substituted with any character so it will match when the minute and second are both 00. Alternatively you could use the “ends_with 0000” condition.

Optionally, you can add a “hour between 10-20” condition to ensure that the trigger won't kick users at night.

6.2 How to translate the messages to your own language?

You can translate two files at the moment to your own language: the help, and another file which contains the messages which the users get.

You can find the original messages in the **oxygene2.help.US.txt** and **oxygene2.lang.US.lua** file. You should make a copy of them named **oxygene2.help.*LNG*.txt** and **oxygene2.lang.*LNG*.txt**, where *LNG* should be your language (language code or any other name which you'd like to use).

I recommend to use Notepad2 or Notepad to make the translation.

Translating help is simple. Just open it, and change the strings. Although the message-file has its own format. It's a Lua-table in fact, but if you don't know what's that, you don't have to care about it :).

The first lines are comments:

```
--// Language: English (Original)
```

You can change them as you like, but be careful, that the “--” must be remain at the beginning of the line since that's what makes it comment.

The latter lines contains the strings which you can translate:

```
Oxygene._LANG = {
        AsWellAs = "as well as",
        BadSlots = "Our rules ask for %1 slots. Please set the slot number to %2 instead of %3 to provide your long staying on the hub"
}
```

1. **Don't** change the names of the strings which stays on the left side of the equal mark. If you do, the language file won't work.
2. The strings must be between quotation marks. If you want to put a quotation mark **inside** the text, use \” to do that (a backslash character followed by a quotation mark)
3. You should put a comma to the end of every line except the last one.
4. You can freely change the text between the quotation marks. The numbered parameters will be substituted with other values. I think, this is obvious. You can even change the order of them if you want. Note that if you want to put a percent mark to your string, you should use %% (double) since it has a special meaning and it's used for parameters.
5. After finishing the translation, you can load it with the **-chset language LNG** command. Then, you should use the **-chreload** command.

Feel free to send me to your translation so I can publish it.

7. Contacts and upgrades

E-mail: fleet@elitemail.hu
Updated versions of the script: http://bcdc.dccafe.org
Sourceforge project page and svn access: http://sourceforge.net/projects/elitedc

Appendix A: Variables for trigger actions

- **%[userNI]:** the nick of the user [string]
- **%[userNIshort]:** the user's nick without prefixes [string]
- **%[userSID]:** the user's SID [string]
- **%[userCID]:** the user's CID [string]
- **%[userDE]:** the user's description [string]
- **%[userEM]:** the user's e-mail address [string]
- **%[connection]:** the specified connection mode (“DSL”, “Cable”, “0.005”) [string]
- **%[chat]:** the latest chat message from the user [string]
- **%[myNI]:** the bot's own nick on the current hub [string]
- **%[mySID]:** the SID of the bot on the current hub [string]
- **%[myCID]:** the CID of the bot on the current hub [string]
- **%[version]:** the version of Oxygene.lua [string]
- **%[usercount]**: the number of online users on the hub [integer]
- **%[opcount]**: the number of online operators on the hub [integer]

- **%[userSS]:** the user's sharesize in Bytes [integer]
- **%[userSSshort]**: formatted share size (for example: 29,37 GiB) [string]
- **%[client_type]:** a string contains the client type from the tag (like “++”, “DCGUI”) [string]
- **%[tagV]**: the client version (for example: 0.689) [number]
- **%[tagVE]**: the client type and the version (“++ 0.689”) [string]
- **%[tagM]:** the stated connection mode from the tag (use to be “A”, “P”) [string]
- **%[tagHN], %[tagHR], %[tagHO]**: the number of hubs where the user is normal/registered/operator [integer]
- **%[tagSL]**: Number of upload slots [integer]
- **%[tagO]**: Auto upload speed limit, KiB/sec. The client opens upload slots automatically if its upload speed is below this speed (if set to 0, disabled) [integer]
- **%[downBW], %[upBW]**: The Down/Upload bandwidth of the user in kbps. 0 if not set [number]
- **%[points]:** The points got for slotrules violation [number]
- **%[pointsint]:** The same as above, but integer [number]
- **%[date]:** string containing the current date in “yyyymmdd” format (“20060130”) [string]
- **%[time]:** string containing the current time in “hhmmss” format (“114232”) [string]
- **%[fulldate]:** string containing both the current date and time in one string in “yyyymmddhhmmss” format (“20060130114232”) [string]
- **%[hour]:** a number with the hours from the current time (0-23) [integer]
- **%[min]:** a number with the minutes from the current time (0-59) [integer]
- **%[sec]:** a number with the seconds from the current time (0-59) [integer]

Appendix B: Kick profile variables

- %[userNI]
- %[userSID]
- %[userCID]
- %[myNI]
- %[mySID]
- %[myCID]
- %[reason]

Appendix C: Changelog

006/20070517: Added %[userSID], %[userCID], %[mySID], %[myCID] to trigger variables, cid protection added for -chprotect
005/20070516: Kick profiles work, new variables added for them (%[mySID], %[myCID], %[userSID], %[userCID])

004/20070516: More trigger fixes, upgraded to work with startup.lua 1.1.9
003/20070516: Fixed some bugs
002/20070516: Synced to Oxygene.lua R095
001/20061029: Initial modifications to get this script work on ADC hubs, building userdata, -chgetinfo works

Appendix D: GNU Free Documentation License

```
                GNU Free Documentation License
                  Version 1.2, November 2002


 Copyright (C) 2000,2001,2002  Free Software Foundation, Inc.
     51 Franklin St, Fifth Floor, Boston, MA  02110-1301  USA
 Everyone is permitted to copy and distribute verbatim copies
 of this license document, but changing it is not allowed.
```

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.

C. State on the Title page the name of the publisher of the Modified Version, as the publisher.

D. Preserve all the copyright notices of the Document.

E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.

F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.

G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.

H. Include an unaltered copy of this License.

I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.

N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.

O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties--for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements".

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See http://www.gnu.org/copyleft/.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

> Copyright (c) YEAR YOUR NAME.
> Permission is granted to copy, distribute and/or modify this document
> under the terms of the GNU Free Documentation License, Version 1.2
> or any later version published by the Free Software Foundation;
> with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.
> A copy of the license is included in the section entitled "GNU
> Free Documentation License".

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the "with...Texts." line with this:

with the Invariant Sections being LIST THEIR TITLES, with the
Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.