BKG Professional NtripCaster Version 2.0.x

Operations Manual

Contents

1. Versions

The BKG Professional NtripCaster is meant for service providers handling several hundred incoming streams in support of thousand or more simultaneously listening clients. The BKG Professional NtripCaster software follows NTRIP Version 2. The main advantages over NTRIP Version 1.0 include

- Full HTTP compatibility, cleared and fixed design problems and protocol violations
- Replaced non standard directives
- Adds chunked transfer encoding
- Improves header records
- Provides for sourcetable filtering
- Optional support of RTSP/RTP and UDP

2. Installation instructions

Here is what you need to do to get the software up and running:

1. Copy it somewhere into an empty directory, run *bunzip2 ntripcaster-version.tar.bz2* and *tar xfv ntripcaster-version.tar* for un-compression.

2. Run *./configure* (if you do not want the server to be installed in */usr/local/ntripcaster* specify the desired path with *./configure --prefix=<path>*)

3. Run *make*

4. Run *make install*

5. After that, the server files will be installed. Binaries will be in */usr/sbin* and */usr/bin*, configuration files in */etc/ntripcaster*, logs in */var/log/ntripcaster* and html templates in */usr/local/ntripcaster/templates*.

6. Go to the configuration directory and copy clientmounts.aut.dist, sourcemounts.aut.dist, users.aut.dist, groups.aut.dist, sourcetable.dat.dist and ntripcaster.conf.dist to clientmounts.aut, users.aut, groups.aut, sourcetable.dat and ntripcaster.conf. Change the contents of these files according to your needs.

If you need detailed information about the program's setup and operation, have a look into files README and INSTALL as well.

For any other technical questions please contact igs-ip@bkg.bund.de.

3. Start and stop of NtripCaster

The recommended home directory for NtripCaster installation is `/usr/local/ntripcaster`. Below this home directory the sub-directories `bin`, `conf`, `log`, `templates`, and `var` can be found.

- Sub-directory `bin` contains the

  (1) NtripCaster executable `ntripdaemon`
  (2) shell script `casterwatch` to continuously watch the `ntripdaemon` process
  (3) start-script `ntripcaster.`

  You can start the NtripCaster using the command ./`ntripcaster start`
  You can re-start the NtripCaster using the command ./`ntripcaster restart`
  You can stop the NtripCaster using the command ./`ntripcaster stop`

- Note that `casterwatch` should never crash. In case `ntripdaemon` crashes or is shut down for re-configuration reasons, `casterwatch` shall re-start it within a few seconds.

4. The NtripCaster Web Interface

- The NtripCaster's home page is http://NtripCasterIP:Port/home.

- The NtripCaster's Administrator's Web Interface is located at http://NtripCasterIP:Port/admin. Note that access to the Admin Web Interface is password protected. If port 80 is used, make sure that no other web servers are running on port 80 on the same machine.

- Sub-directory `templates` contains templates for the Admin Web Interface http://NtripCasterIP:Port/admin as well as the NtripCaster's home page http://NtripCasterIP:Port/home.

5. Main NtripCaster configuration file

The essential parameters for running the NtripCaster are defined in ntripcaster.conf under sub-directory `conf`.
Most of the parameters are self-explanatory. Note that neither ntripcaster.conf nor any other configuration file of the NtripCaster should be edited on a MS Windows system because this adds an extra 'carriage return' at the end of each record that may cause a problem.

- The `encoder_password` is a generic password valid only for stream upload through NtripServer programs that follow the NTRIP Version 1.0 standard for the communication with the NtripCaster.

- The `admin_password` and `oper_password` are passwords for administrating and operating the NtripCaster through a Telnet session.

- The NtripCaster comes with an integrated NtripClient that lets you pull streams from other NtripCasters.

  ```
  relay pull -i user:pass –m /PADO1 147.162.229.36:2101/PADO0
  ```

  Explanation: A stream coming from mountpoint `PADO0` on the remote NtripCaster `147.162.229.36:2101` is pulled using the user ID `user` and the password `pass` and made available on the local NtripCaster on mountpoint `PADO1`.

  You might prefer to pull a stream directly from an IP address and port without using the NTRIP protocol. If this is the case, use the following command line in ntripcaster.conf:

  ```
  relay pull –m /SYDN0 133.160.129.22:8000
  ```

  Explanation: A stream from `133.160.129.22:8000` is pulled and made available on the local NtripCaster installation through mountpoint `SYDN0`.

  The integrated NtripClient is using Ntrip version 1.0 protocol for communication and data transfer. For Ntrip version 2.0 usage please add `-2` as option.

  You can specify a HTTP proxy with the parameter "-p host:port" (usually only for Ntrip version 2). This could be useful if your caster is running behind a firewall.

- Alias mountpoints may be used to mount streams from the same server with a different mountpoint name, e.g.

  ```
  alias /FFMJ1 /FFMJ00DEU0
  ```

  where FFMJ1 is the alias. Note that the alias mountpoints are neither visible in the sourcetable nor via web interface (*sources* or *listeners*), but can be seen in the casters log file ntripcaster-yymmdd.log. They can be used to smooth the renaming of mountpoints, e.g. for introducing the long RINEX3 mountpoint names, where the old mountpoint names should be kept for a while.

- You may want to configure more than one communication port for the NtripCaster. Because of problems clients may experience with proxy servers in front of their application, it is recommended to use ports 80 and 2101. Usually, there is one of those not blocked by proxies. Please note, using port 80 typically requires the program to be started by root.

  ```
  port 80
  port 2101
  ```

- Parameter `logfiledebuglevel` defines the amount of details in the output saved in the daily log files. Note that log files may become huge if this parameter is not set to `0`.

- Once `acl_policy 1` is defined in ntripcaster.conf, you may deny access to the NtripCaster from any specific IP address or network of IP addresses. This may be of interest when the NtripCaster is experiencing a hacker attack from a specific IP or IP network.

  ```
  deny all 69.15.204.66
  deny all 147.162.*.*
  ```

6. Client authentication/authorization

The NtripClient authorization is configured through the files users.aut, groups.aut, and clientmounts.aut under sub-directory `conf`.

- Each record in users.aut is dedicated to one user of the system. This record defines his/her user ID and password. Users not listed in users.aut do not have access to protected streams.

  ```
  soehne:wolfgang
  stuerze:andrea
  Yan:Thomas
  ntrip:password
  anderson:greg123
  ```

- Selections of users are then put together to form groups. Each record in groups.aut defines one group. To keep the caster usage under control it is possible to

  (a) specify a maximum number of streams which can be pulled simultaneously by all group members and/or
  (b) specify a maximum number of streams which can be pulled from a certain client IP address.

  For (a) you would have to add an integer number at the end of a record in groups.aut. This integer number defines the maximum number of simultaneously listening clients from that group. Any additional client from that specific group denies access. No integer at the end of a record allows an unlimited number of simultaneously listening clients for that group.

  For (b) the parameter specification `max_ip_connections` in ntripcaster.conf can be overruled by adding string “ip<n>” at the end of a record in groups.out, In this case <n> defines the maximum number of clients listening at a specific client IP address.

  Following, we give examples for the three possible configuration record setups in groups.aut. Please, note here we suppose a `max_ip_connections 5` configuration option in ntripcaster.conf.

  Example-1: `unavco:Anderson,stuerze`

  User group “unavco” (with members “Anderson” and “stuerze”) can pull simultaneously an unlimited number of streams from any client IP address.

Example-2: `bkg:soehne,stuerze:10`

User group “bkg” (with members “soehne” and “stuerze”) can pull a maximum number of 10 streams simultaneously. However, the group could only pull a maximum number of 5 streams from the same client IP address, since the `max_ip_connections` option is set to be 5 (cf. above).

Example-3: *gYan:Yan:ip3*

User group “gYan” (with the only member “Yan”) has access to an unlimited number of streams but can only pull 3 streams from the same client IP address.

- The file clientmounts.aut defines access of groups to specific mountpoints. Mountpoint strings in the file are preceded by a slash. Each mountpoint is accessible only to members of those groups listed after the mountpoint identifier followed by a colon. There is a need to give at least one group for each mountpoint. The length of a mountpoint string is limited to 8,192 characters. If this value is exceeded, an error messages appears in the log file saying “READ ERROR: too long line in mount authentication file (exceeding BUFSIZE)". A mountpoint that does not show up in clientmounts.aut remains unprotected.

```
/SYDNEY:bkg,gYan
/FFMJ0:unavco
```

- There are two pre-defined admin-mountpoints named `admin` and `oper` in clientmounts.aut which do not control access to mountpoints for streams but let you list groups whose members are enabled to carry out administrative and operational tasks. These admin-mountpoints are defined as

```
/admin:FirstAdmin,SecondAdmin,NTRIPAdmin
/oper:FirstAdmin,SecondAdmin
```

and thus mention `FirstAdmin`, `SecondAdmin`, and/or `NTRIPAdmin` as authorized groups to carry out administrative/operational tasks. Note that you need to have operator rights to execute all commands available for the NtripCaster operation because administrator rights are limited to a subset of administration commands.

Example:

A taxi company in Sydney wants to enable each of its 100 drivers to receive a specific DGPS corrections stream through individual accounts. Since no more than 30 drivers are ever on duty at the same time, the company applied (and most likely paid) for access rights for a maximum of 30 concurrent clients only.

(1) In users.aut we need to configure 100 user accounts:

```
taxi1:password1
taxi2:password2
taxi3:password3
…
taxi100:password100
```

(2) In groups.aut we then create a group `company`, listing all the 100 users but putting a limit of 30 concurrent users

```
company:taxi1,taxi2,taxi3,…,taxi100:30
```

(3) In clientmounts.aut we then allow the group `company` to access the DGPS corrections stream from mountpoint `SYDNEY`

```
/SYDNEY:company
```

7. Server authentication/authorization

- An NtripServer intending to occupy a mountpoint via NTRIP Version 1.0 must use the generic `encoder_password` as defined in ntripcaster.conf for stream upload.
- An NtripServer intending to occupy a mountpoint using the new NTRIP Version 2.0 protocol must have an account in users.aut defined as a member of a group of accounts in groups.aut which is authorized to use that mountpoint through a valid entry in sourcemounts.aut.

Example:
In users.aut we may have registered `provider1` (user ID) with password `password1` through:

```
provider1:password1
```

In groups.aut we may then configure `provider1` as a member of group `gupload` through:

```
gupload:provider1,provider2,provider3,…
```

In sourcemounts.aut we may then allow the members of group `gupload` to upload a stream to mountpoint `SYDNEY` through

```
/SYDNEY:gupload
```

Note that when using the generic `encoder_password` also as the password for `provider1`, this single password would be valid for *provider1* for both, stream upload through NTRIP Version 1.0 and NTRIP Version 2.0. However, stream upload through NTRIP Version 2.0 would in addition need the user ID `provider1`.

8. Administration through Web Interface

The NtripCaster knows administrator and operator groups whose members perform administrative and operational duties and responsibilities through the Admin Web Interface http://NtripCasterIP:Port/admin. These groups are defined in groups.aut and receive their specific responsibility through the `admin` and `oper` admin-mountpoints in clientmounts.aut. The groups may be named `FirstAdmin`, `SecondAdmin`, and `NTRIPAdmin`.

- Users from users.aut that are defined as a member of the `FirstAdmin` group or the `SecondAdmin` group in groups.aut through

  ```
  FirstAdmin:Yan
  SecondAdmin:soehne
  ```

  have unlimited rights on the NtripCaster administration through the Admin Web Interface if these groups are part of the admin-mountpoint `oper` in clientmounts.aut.

- Users from users.aut that are defined as a member of the *NTRIPAdmin* group in groups.aut through

  ```
  NTRIPAdmin:ntrip
  ```

  have limited rights concerning the NtripCaster administration through the Admin Web Interface if this group is only part of the admin-mountpoint `admin` in clientmounts.aut. They may use any Admin Web Interface command with the exception of command http://NtripCasterIP:Port/admin?mode=set.

Some basic monitoring and administration tasks can be accomplished through the various menus of the web interface:

- *home*: The homepage
- *statistics*: Some statistical information like total number of sources and listeners, total running time of caster software, et al.
- *sourcetable*: View the content of sourcetable.dat. Active data streams are displayed in black, missing in red.
- *listeners*: Overview of currently active users with details of host/IP, username, mountpoint name, Id, total time of connect, bytes written, errors, client software used, connection type
- *sources*: Overview of currently active sources with details of mountpoint name, Id, host, source agent, date of connect, IP, number of active client connections, kbytes read/written, total number of client connections, total time of connect
- *connections*: Lists all active connections (sources and listeners), gives the possibility to get rid of some listeners or sources. Clicking on the Id immediately starts the kick command and stops the connection.
- *admins*: Lists all currently connected admins

- *settings*: Lists several parameters from file ntripcaster.conf.

9. Administration through Telnet

The NtripCaster can be administrated/operated via Telnet. To establish a telnet session to the NtripCaster use a command window and enter the following two commands:

```
telnet NtripCasterIP 2101
ADMIN [admin_password]
```

where 2101 stands for the NtripCaster's listening port and `admin_password` is taken from ntripcaster.conf. Then press Enter twice. You may now use any of the administration/operation commands described in section 3.2.2 of the NtripCaster implementation document available from NtripImplementation.pdf, these commands are (see Appendix for more details):

```
help, admins, alias, allow, auth, deny, kick, list, listeners, oper, quit,
rehash, relay, set, sources, stats, tail, tell, uptime.
```

The execution of some of these commands requires operator rights. To become an operator, use the `oper` command: "*oper oper_password*". Note that neither the `admin_password` nor the `oper_password` has anything to do with the passwords listed in users.aut. These two passwords are defined in ntripcaster.conf. Note further that none of the configuration changes made through the Telnet admin commands are permanent. As soon as the NtripCaster is re-started, the contents of the basic configuration files becomes active again.

10. The Sourcetable

The NtripCaster's sourcetable is defined in the file sourcetable.dat under sub-directory `conf`. Note that the NtripCaster delivers a so-called dynamic Sourcetable to NtripClients on their request. This dynamic Sourcetable is generated from sourcetable.dat but comprises only those streams/mountpoints that are available for the NtripCaster at the time when the request is received. Streams that have an outage will not show up.

- See NtripDocumentation.pdf for a complete description of mandatory and optional record types and their data fields.

- The Sourcetable of www.igs-ip.net is an example for a complete Sourcetable setup.

- Although not mandatory it is recommended to define Sourcetable records of type NET and CAS in addition to the record type STR.

  (1) One NET-record should be defined for each network listed in data field number 8 of the STR records. Note that – although not explicitly defined in the NTRIP Version 1.0 standard - data field number 7 of a NET-record should contain an HTTP link to a RINEX skeleton files directory for the corresponding STR records.

  (2) One CAS-record should describe the operating NtripCaster installation. In addition it is recommended to include at least another CAS-record for the NtripCaster http://www.rtcm-ntrip.org/home.

  ```
  CAS;rtcm-ntrip.org;2101;NtripInfoCaster;BKG;0;DEU;
  50.12;8.69;http://www.rtcm-ntrip.org/home
  ```

- Data field number 16 of the STR-records carries information concerning stream protection. Note that this is meant only as an information for NtripClients downloading the Sourcetable. The stream protection mechanism of the NtripCaster is based solely on the information in clientmounts.aut. So, even if data field number 16 of an STR-record in the Sourcetable describes that a stream is unprotected, it remains protected for the system if defined as such in clientmounts.aut and vice-versa.

- Only streams/mountpoints listed in sourcetable.dat are visible and thus accessible for an ordinary NtripClient. However, the NtripCaster will accept any stream coming from an NtripServer (if properly authorized for stream upload), even if it is not configured in sourcetable.dat and clientmounts.aut. An unregistered stream/mountpoint is visible only through the Admin Web Interface command http://NtripCasterIP:Port/admin?mode=sources. Such stream is not part of the dynamic Sourcetable delivered on NtripClient's request, and is only accessible to users who are aware of its existence/mountpoint. If an unregistered mountpoint is used for stream upload by mistake, the provider of such a stream should be informed that he is using an incorrect mountpoint string and hence should cease the upload.

- File sourcetable.dat shall not contain a last record: `ENDSOURCETABLE`

11. Changes in NtripCaster configuration

Permanent configuration changes must be made in the configuration files ntripcaster.conf, users.aut, groups.aut, clientmounts.aut, sourcemounts.aut, and sourcetable.dat. They become active through Admin Web Interface commands.

(1) Command http://NtripCasterIP:Port/admin?mode=rehash activates additions to ntripcaster.conf, users.aut, groups.aut, clientounts.aut, sourcemounts.aut, and sourcetable.dat. Note that the NtripCaster process is kept alive during that procedure. Because of this re-configuration on-the-fly, incoming and outgoing streams remain undisturbed.

(2) Command http://NtripCasterIP:Port/admin?mode=resync causes a shut-down and re-start of the NtripCaster executable and thus activates changes as well as additions to ntripcaster.conf, users.aut, groups.aut, clientmounts.aut, sourcemounts.aut, and sourcetable.dat. Note that all incoming and outgoing streams are disconnected through that procedure for a short time (usually for 20 seconds up to one minute).

12. Log files

Daily log files are saved under sub-directory `logs`. The NtripCaster maintains three types of log files: access-yymmdd.log, ntripcaster-yymmdd.log, and usage-yymmdd.log.

- Note that files of type access-yymmdd.log follow the CSV format. You may like to upload these files to an external archive and by that rename its suffix to *.csv for read compatibility with the MS Excel program. The contents of these files may become the base for an accounting system.

```
Date,Time,User,IP,Station,Client,Seconds,Bytes
10/Aug/2007,20:42:31,hr,141.74.33.2,BURG0,NTRIP BNC 1.4,86,15166
10/Aug/2007,20:42:32,hr,141.74.33.2,MOBS1,NTRIP BNC 1.4,87,10873
```

The NtripCaster does not delete these daily log files. Check the disk space from time to time and delete them manually when necessary. Note that the size of the log files may become very large depending on parameter `logfiledebuglevel` defined in ntripcaster.conf.

13. Security Aspects

The recommended home directory for NtripCaster installation is `/usr/local/ntripcaster`. This implicitly requires running the NtripCaster software as root. However for security reasons it could be of interest to run the NtripCaster software as a normal user. If you want to do so, you have to take into account that a user application cannot open ports <= 1024. In order to allow the caster software to use port 80 although running it as a normal user, the following steps may have to be executed by your system administrator:

- Increase the number of open files:
  `/etc/security/limits.conf:`
  - `soft nofile 4096`
  - `hard nofile 10240`
- Redirect port 80 to port xxxx (with xxxx being 1080 in the following example):
  - Filter Rule
    ```
    iptables -t filter -A RH-Firewall-1-INPUT -p tcp -m state --state NEW
    -m tcp --dport 1080 -j ACCEPT
    ```
    (Note that "RH-Firewall-1-INPUT" stands for the name of the chain, such as for Red Hat)
  - Redirection rule
    ```
    iptables -t nat -A PREROUTING -p tcp --dport 80 -j REDIRECT --to-port
    1080
    ```
- Saving the configuration:
  - `iptables-save > /etc/sysconfig/iptables`

After that, in the NtripCaster configuration file `ntripcaster.conf` the port entry has to be changed from port 80 to xxxx (with xxxx being 1080 in this example). The NtripCaster software can now be started using a normal user account. After all, the NtripCaster would still be available through port 80.

14. Communication via SSL

This section has kindly been provided by Wim Aerts, Royal Observatory of Belgium. It guides you through setting up an Apache 2 HTTP Server instance to add SSL to your NtripCaster. It only deals with getting things to work, not with securing issues. For that the reader is referred to https://www.ssllabs.com/ssltest/index.html and relevant documents on that website.

Take following steps:

1. Find out on what port your NtripCaster is listening, e.g. at port 2101.
2. Select a port that you would like to use to offer a secure SSL access to your NtripCaster, e.g. port 443.
3. Configure Apache 2 HTTP Server to also listen on the port you selected in step 2.
   This is done by adding a statement 'Listen 443' to your global Apache 2 HTTP Server configuration file. On Ubuntu systems this should be added to file /etc/apache2/ports.conf.
4. Set up a new virtual host that will take care of the SSL and proxy to the NtripCaster. This is done by adding the following code. Please specify the correct file location for your certificate and key file.

   ```
   8<---8<---8<---
   <IfModule mod_ssl.c>
   <VirtualHost _default_:443>
        ProxyPreserveHost On
        ProxyRequests Off
        ProxyPass / http://localhost:2101/
        ProxyPassReverse / http://localhost:2101/
        SSLEngine On
        SSLProtocol -all +SSLv3 +TLSv1
        SSLCipherSuite TLSv1+HIGH:!SSLv2:!aNULL:!eNULL:
        SSLOptions +StrictRequire
        SSLCertificateFile    /etc/ssl/certs/...
        SSLCertificateKeyFile /etc/ssl/private/...
        <Directory />
                SSLRequireSSL
                Allow from all
        </Directory>
   </VirtualHost>
   </IfModule>
   8<---8<---8<---
   ```

   This should go somewhere in the Apache 2 HTTP Server configuration file(s). On Ubuntu it is placed in a separate file, e.g.: */etc/apache2/sites-available/mysite*. Command `a2ensite mysite` will then 'enable' this 'available' site.
5. Make sure that the Apache 2 HTTP Server is configured to load all the modules needed to do what you desire. You need at least modues mod_mime, mod_proxy, mod_proxy_http, and mod_ssl.
6. Start or restart the Apache 2 HTTP Server deamon. On Ubuntu this can be done through `/etc/init.d/apache2 restart`. Check the error log files.
7. Put on the URL bar https://ip_of_your_NtripCaster:443 in your browser to check the connection. You should now see the sourcetable.
8. If the test at step 7 fails, check firewalls.

Note that SSL support makes only sense for NTRIP Version 2. Stream transport through SSL and NTRIP Version 1 would have a problem because it is not fully HTTP compatible.

Also note that username and password are base64 encoded if Ntrip Version 2 is used.

15. LDAP (Lightweight Directory Access Protocol)

LDAP (Lightweight Directory Access Protocol) is a communication protocol for data exchange within a computer network that allows the request and modification of information provided by a directory service. Because LDAP is based on the client server principle it describes the communication between LDAP client and directory server. From such a directory that is realized as hierarchical database object-related data can be selected.

For data privacy reasons LDAP functionality is implemented for user authentication. Currently only simple LDAP access is supported. In case of LDAP usage usernames need to be added in normal configuration as well instead (e.g. simply add * as password in users.aut file), but for password checks LDAP is used: The NTRIP Caster, acting as LDAP client, formulates a request containing user name and password mentioned within the NTRIP client/server request and gets the answer "success" in case of congruence or "failure" otherwise from the LDAP server. LDAP authentication is normally disabled. It gets enabled when an LDAP server is specified in ntripcaster.conf:

```
ldap_server 127.0.0.1
```

Furthermore, the settings `ldap_uid` and `ldap_people_context` in ntripcaster.conf are required for the LDAP request formulation:

```
ldap_uid_prefix uid
ldap_people_context ou=people
```

The bind call is done with `{prefix}={user},{context}.`

Appendix

Commands used with telnet

alias

The alias command is used for two things. It can be used to add an alias for a local mountpoint so that a stream can be accessed from two mountpoints. Or it can be used to add an alias for a remote NtripCaster stream.

Syntax:

```
alias add <mountpoint> <newmountpoint>
alias del <mountpoint>
alias list
```

allow and deny

This adds a hostmask to an internal access lists (acl). It specifies that this hostmask should allow or deny access. A type for this acl can be specified, which can be either all, client, source or admin. It only has affect on the specified connection type.

Syntax:

```
allow|deny <client|source|admin|all> add <hostmask>
allow|deny <client|source|admin|all> del <hostmask>
allow|deny <client|source|admin|all> list
```

Examples:

```
allow client add *.se
allow all del *.netcom.com
deny admin add *.se
deny admin del ap.*.com
allow admin list
```

admins

The „admins“ command is used to list the admins connected to the server. It will display one admin connection per line, like:

```
Admin 341 <d66.ryd.student.liu.se> connected for 1 hours, 8 minutes and 14 seconds. 0
commands issued
```

A host-mask as an argument can be supplied, and only the admins who match the mask, will be shown.

Syntax:

```
admins [hostmask]
```

Example:

```
admins *.ifag.de
```

help

Without arguments, the command „help“ prints a list of all commands and what they are used for. With a command as an argument, a slightly longer description of what the command does, is printed.

Syntax:

```
help [command]
```

Examples:

```
help kick
help
```

kick

This command can be used to get rid of some listeners, admins, or sources. A single connection may be kicked out as well as all connections of a certain type matching a hostmask.

Syntax:

```
kick <id>
kick <-acs> <hostmask>
kick <hostmask>
```

Examples:

```
kick 314
kick -a *.com
kick *.badsource.com
```

listeners

This command displays a list of all connected listeners. The output can be restricted to those listeners matching a specific hostmask.

Syntax:

```
listeners [hostmask]
```

Example:

listeners *.ifag.de

oper

Most commands on the admin console are restricted to NtripCaster operators. To obtain operator privileges, the „oper“ command with the NtripCaster operator password as argument can be used.
Syntax:
oper <operator password>
Example:
oper cooloperpass

quit

The „quit“ command is used to leave the NtripCaster admin console. This can only be used, when the console is accessed through the remote interface (e.g. using telnet). An optional argument will be used as a signoff message, e.g. displayed to other connected admins.
Syntax:
quit [message]
Example:
quit Bye Bye

rehash

This command can be used after changing parameters in the NtripCaster configuration file (ntripcast.conf or source-table.dat). When used with an argument, it will understand this argument as a filename and parse this as a configuration file.
Syntax:
rehash [filename]
Example:
rehash /tmp/newrtcmast.conf

sources

„sources“ is used just like „listeners“ to view all the connected sources. It can be used with an optional argument to limit the list to only the sources matching the specified hostmask. It also accepts a large number of options. By default the output is defined by the variable default_sourceopts.
The options are as follows.
i - Connection id
s - Socket descriptor
t - Time of connect
p - Connection ip
h - Hostname
c - Connection state
y - Source type
c - Number of clients
d - Dumpfile
r - Mountpoint priority
M - Source mountpoint
R - Bytes read
W - Show bytes written
C - Total client connects
T - Time connected
Syntax:
sources [hostmask] [-options]
Example:
sources *.apan.com –aCW

set

This command is used without arguments. „set“ will display a list of all NtripCaster variables and their current values. It can be used also to change any of those variables if the name is specified and the new value is entered as arguments. Typing „help settings“ will give a short description of all the settings, and a longer description for the settings is available here.
Syntax:
set
set <variable name>
set <variable name> <new value>
Examples:

```
set client_timeout
set max_clients 580
```

stats

The „stats“ command will print some information about the current number of connections, averages on client connections, bytes read and written, etc. It looks a lot like the output available from the statistics file. As an argument, „hourly“ or „daily“ can be put. Then the averages and totals for the last hour or day will be printed.

Syntax:

```
stats [hourly|daily]
```

Example:

```
stats hourly
```

tail

The „tail“ command can be issued by an administrator to display messages written to the log file. It's similar to the unix tail command.

Syntax:

```
tail
```

Example:

```
Tail
```

tell

An admin can send a message to all other connected admins using the „tell“ command. Any argument will be send to all admins.

Syntax:

```
tell <message>
```

Example:

```
tell new NtripSource
```

uptime

The „uptime“ command is similar to the Unix uptime command. It prints the number of days, hours, minutes and seconds the NtripCaster is up and running.

Syntax:

```
uptime
```

Example:

```
Uptime
```

list

This command is used to list all connections, sources, admins, or clients.

Syntax:

```
list [hostmask]
```

Example:

```
list *.toomanyarguments.com
```

relay

remote server, the „relay“ command should be used.

Syntax:

```
relay pull [-m <localmount>] <url>
```

Example:

```
relay pull ifag.de:2101/BUCU0
```

auth

This command is used to display authorization users, mounts and groups.

Syntax:

```
auth <groups|users|mounts>
```

Example:

```
auth groups
auth users
```