

BKG Ntrip Client (BNC) Version 2.5

Manual

January 2011

Federal Agency for Cartography and Geodesy, Frankfurt, Germany

BKG Ntrip Client (BNC) Version 2.5 Manual

The BKG Ntrip Client (BNC) is a program for simultaneously retrieving, decoding and converting real-time GNSS data streams from NTRIP broadcasters like <http://www.euref-ip.net/home> or <http://www.igs-ip.net/home> or <http://products.igs-ip.net/home>.

BNC has been developed for the Federal Agency for Cartography and Geodesy (BKG) within the framework of EUREF's Real-time GNSS Project (EUREF-IP, IP for Internet Protocol) and the Real-Time IGS Pilot Project (RTIGS).

BNC has been written under GNU General Public License (GPL). Binaries for BNC are available for Windows, 32-bit Linux, 64-bit Linux (compiled under -m32 32-bit compatibility mode), Solaris, and Mac systems. We used the MinGW Version 5.1.3 compiler to create the Windows binary. It is likely that BNC can be compiled on other systems where a GNU compiler and Qt Version 4.5.2 are installed.

Please ensure that you have installed the latest version of BNC available from <http://igs.bkg.bund.de/ntrip/download>. We are continuously working on the program and would appreciate if you could send any comments, suggestions, or bug reports to [euref-ip@bkg.bund.de] or [igs-ip@bkg.bund.de].

Contents

[1. Purpose](#)

[2. Modes & Resources](#)

[3. Settings & Handling](#)

[4. Limitations & Known Bugs](#)

[5. Authors](#)

[6. Annex](#)

1. Purpose

The purpose of BNC is to

- retrieve real-time GNSS data streams available through NTRIP transport protocol,
- retrieve real-time GNSS data streams via TCP directly from an IP address without using the NTRIP transport protocol, and/or
- retrieve real-time GNSS data streams from a local UDP or serial port without using the NTRIP transport protocol, and/or
- generate high-rate RINEX Observation and Navigation files to support near real-time GNSS post-processing applications, and/or
- generate ephemeris and synchronized or unsynchronized observations epoch by epoch through an IP port to support real-time GNSS network engines, and/or
- generate clock and orbit corrections to broadcast ephemeris through an IP port to support real-time Precise Point Positioning on GNSS rovers, and/or
- generate synchronized clock and orbit corrections to broadcast ephemeris epoch by epoch through an IP port to support the combination of such streams as coming simultaneously from various correction providers, and/or
- monitor the performance of a network of real-time GNSS data streams to generate advisory notes in case of outages or corrupted streams, and/or
- scan RTCM streams for incoming antenna information as well as message types and their repetition rates, and/or
- feed a stream into a GNSS receiver via serial communication link, and/or
- carry out a real-time Precise Point Positioning to determine a GNSS rover position.

BNC mainly supports decoding the following GNSS stream formats and message types:

- RTCM Version 2.x containing message types 18 and 19 or 20 and 21 together with 3 and 22 (GPS and GLONASS),
- RTCM Version 3.x containing message types
 - 1002, 1004 (GPS, SBAS, observations)
 - 1010, 1012 (GLONASS, observations)
 - 1019, 1020, 1045 (GPS, GLONASS, and proposed Galileo ephemeris)
 - 1057-1068 (proposed State Space Representation messages for GPS and GLONASS ephemeris correctors)
 - 1071-1077, 1081-1087, 1091-1097 (proposed 'Multiple Signal Messages' (MSM) for GPS, GLONASS and Galileo observations).
- RTIGS containing GPS record types 200 (observations) and 300 (ephemeris).

BNC allows to by-pass its decoding and conversion algorithms, leave whatever is received untouched and save it in files.

The first of the following figures shows a flow chart of BNC connected to a GNSS receiver via serial or TCP communication link for the purpose of Precise Point Positioning. The second figure shows the conversion of RTCM streams to RINEX batches. The third figure shows a flow chart of BNC feeding a real-time GNSS engine. The engine then estimates satellite orbit and clock correctors. The 'BKG Ntrip Server' (BNS) is used in this scenario to encode correctors to RTCMv3.

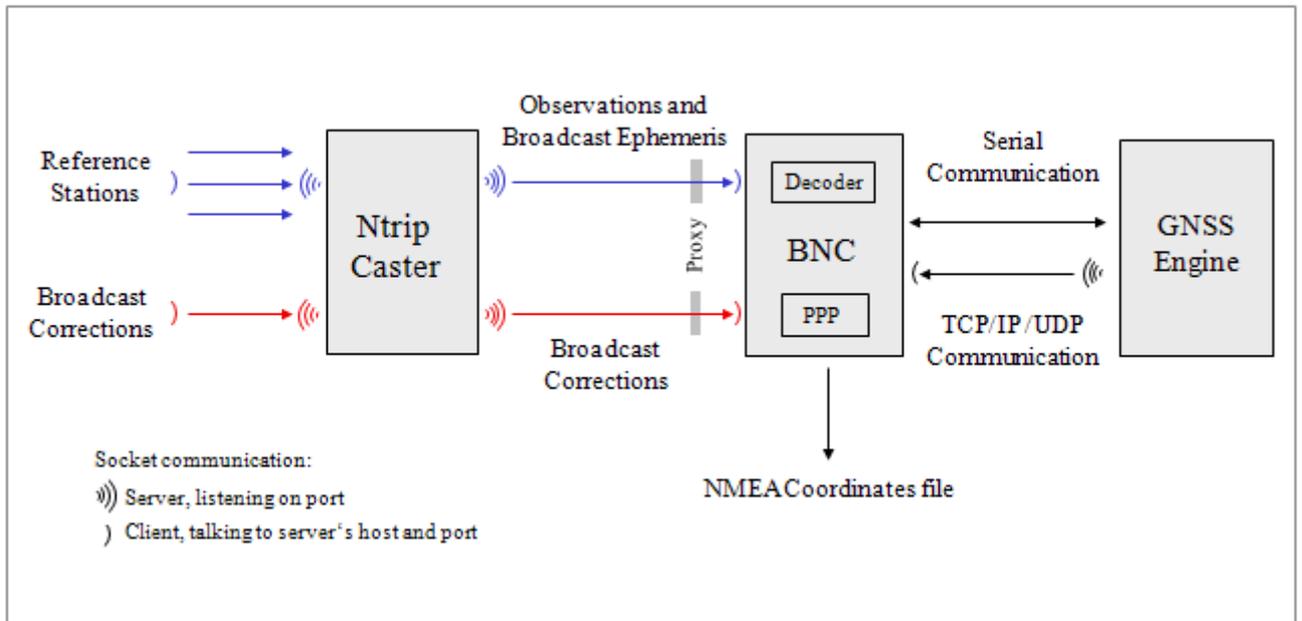


Figure: Flowchart, BNC connected to a GNSS receiver for Precise Point Positioning.

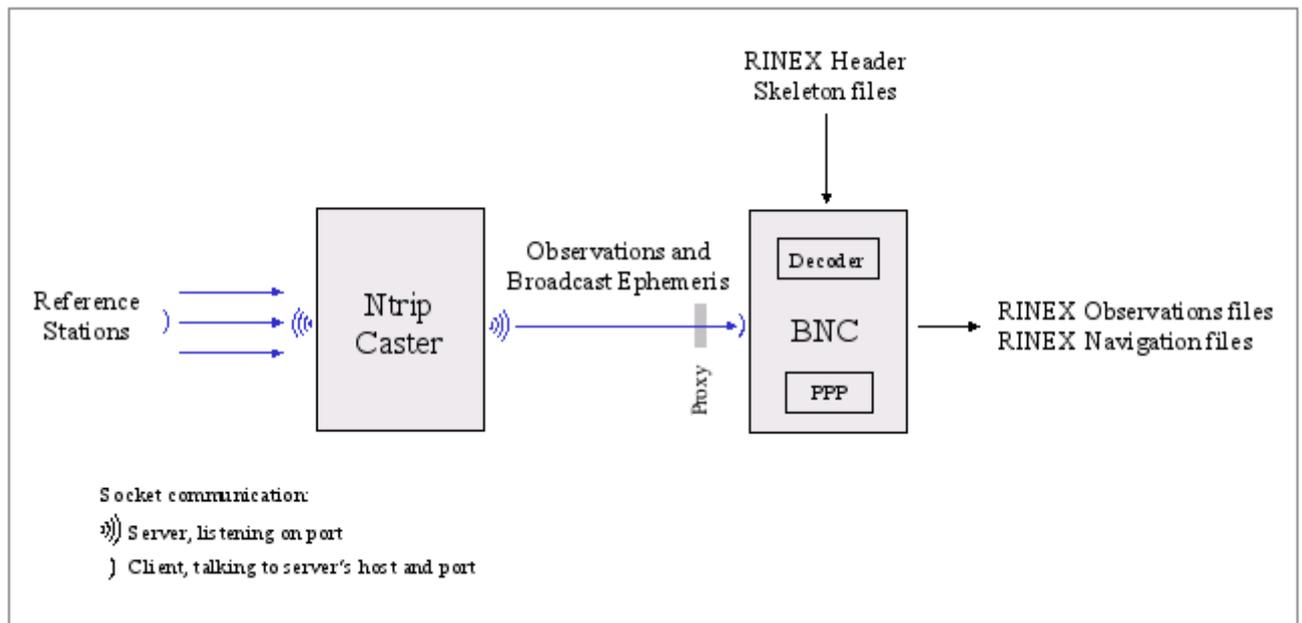


Figure: Flowchart, BNC converting RTCM streams to RINEX batches.

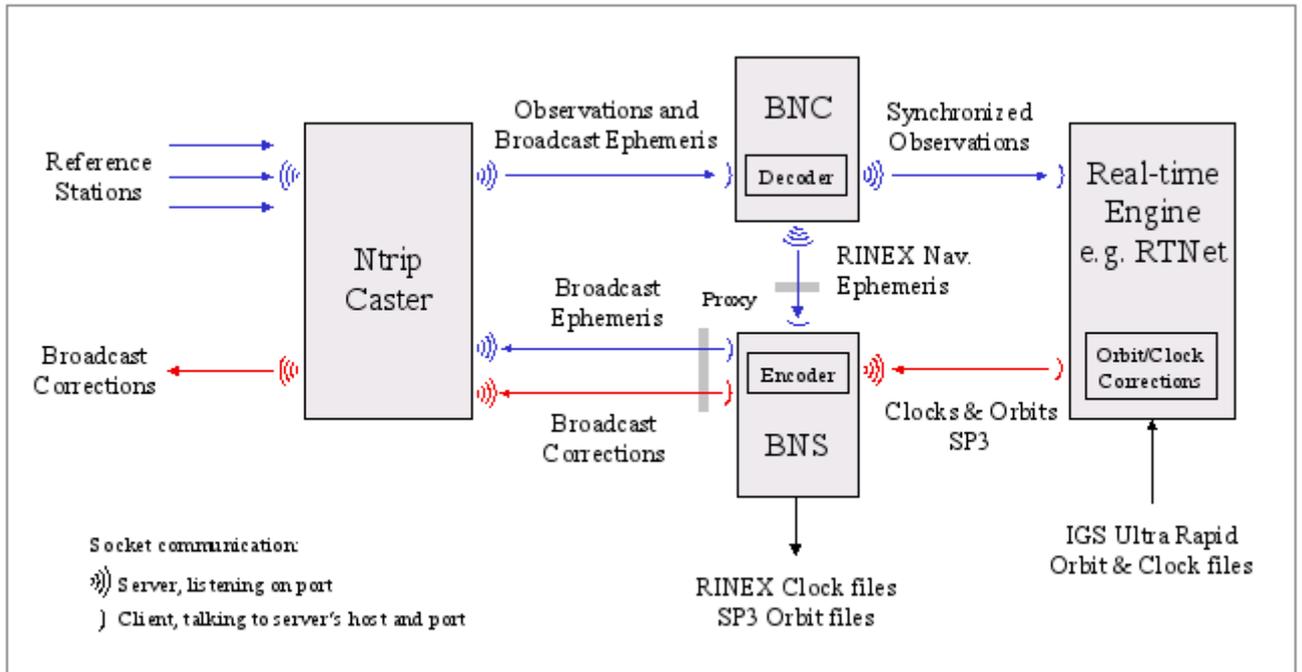


Figure: Flowchart, BNC feeding a real-time GNSS engine.

2. Modes & Resources

Although BNC is a real-time tool to be operated in online mode, it can be run offline for post-processing of data made available from a single file. Furthermore, apart from its regular window mode, BNC can be run as a batch/background job in a 'no window' mode using processing options from a previously saved configuration.

Unless in offline mode, BNC

- requires access to the Internet with a minimum of about 2 to 6 kbits/sec per stream depending on the stream format and the number of visible satellites. You need to make sure that the connection can sustain the required bandwidth.
- requires the clock of the host computer to be properly synchronized.
- has the capacity to retrieve hundreds of GNSS data streams simultaneously. Please be aware that such usage may incur a heavy load on the NTRIP broadcaster side depending on the number of streams requested. We recommend limiting the number of streams where possible to avoid unnecessary workload.

The main window of BNC shows a top menu bar section, a sections for tabs to set processing options, a 'Streams' section and a section for 'Log' tabs, and a bottom menu bar section, see figure below.

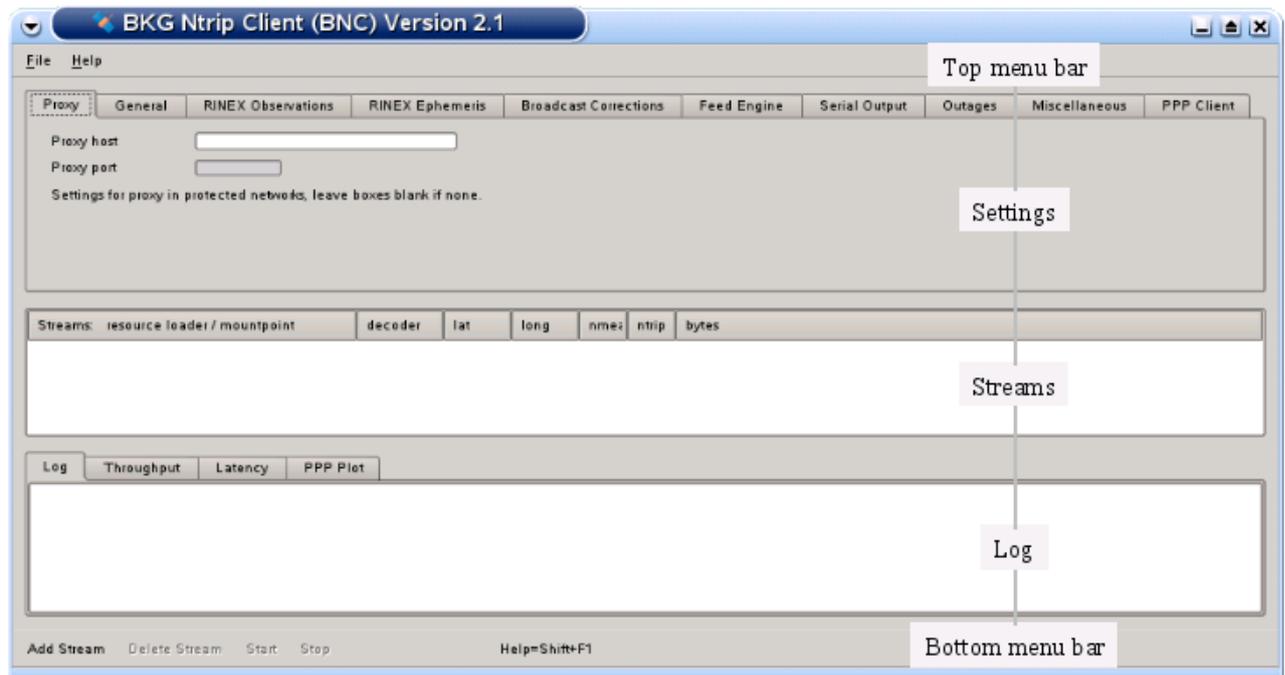


Figure: Sections on BNC's main window.

3. Settings & Handling

This chapter describes BNC's settings and how to handle the program. It explains the top menu bar, the processing options, the 'Streams' and 'Log' sections, and the bottom menu bar.

The usual handling of BNC is that you first select a number of streams ('Add Stream'). Any stream configured to BNC shows up on the 'Streams' canvas in the middle of BNC's main

window. You then go through BNC's various configuration tabs to select a combination of input, processing and output options before you start the program ('Start'). Records of BNC's activities are shown in the 'Log' tab. The bandwidth consumption per stream, the latency of incoming observations and PPP time series for coordinate components are shown in the 'Throughput', 'Latency' and 'PPP Plot' tabs of the main window.

As a default, configuration files for running BNC on Unix/Linux/Mac systems are saved in directory '\${HOME}/.config/BKG'. On Windows systems, they are typically saved in directory 'C:/Documents and Settings/Username/.config/BKG'. The default configuration file name is 'BNC.ini'.

The default file name 'BNC.ini' can be changed and the file contents can easily be edited. On graphical user interfaces it is possible to Drag & Drop a configuration file icon to start BNC (not on Mac systems). Some configuration options can be changed on-the-fly. See annexed 'Configuration Example' for a complete set of configuration options.

3.1. [Top Menu Bar](#)

3.1.1 [File](#)

3.1.2 [Help](#)

3.2. [Proxy](#)

3.3. [General](#)

3.3.1. [Logfile](#)

3.3.2. [Append Files](#)

3.3.3. [Reread Configuration](#)

3.3.4. [Auto Start](#)

3.3.5. [Raw Output File](#)

3.4. [RINEX Observations](#)

3.4.1. [File Names](#)

3.4.2. [Directory](#)

3.4.3. [File Interval](#)

3.4.4. [Sampling](#)

3.4.5. [Skeleton Extension](#)

3.4.6. [Script](#)

3.4.7. [Version](#)

3.5. [RINEX Ephemeris](#)

3.5.1. [Directory](#)

3.5.2. [Interval](#)

3.5.3. [Port](#)

3.5.4. [Version](#)

3.6. [Broadcast Corrections](#)

3.6.1. [Directory](#)

3.6.2. [Interval](#)

3.6.3. [Port](#)

3.6.4. [Wait for Full Epoch](#)

3.7. [Feed Engine](#)

3.7.1. [Port](#)

3.7.2. [Wait for Full Epoch](#)

3.7.3. [Sampling](#)

3.7.4. [File](#)

3.7.5. [Port \(unsynchronized\)](#)

- 3.8. [Serial Output](#)
 - 3.8.1. [Mountpoint](#)
 - 3.8.2. [Port Name](#)
 - 3.8.3. [Baud Rate](#)
 - 3.8.4. [Flow Control](#)
 - 3.8.5. [Parity](#)
 - 3.8.6. [Data Bits](#)
 - 3.8.7. [Stop Bits](#)
 - 3.8.8. [NMEA](#)
 - 3.8.9. [File](#)
 - 3.8.10. [Height](#)
- 3.9. [Outages](#)
 - 3.9.1. [Observation Rate](#)
 - 3.9.2. [Failure Threshold](#)
 - 3.9.3. [Recovery Threshold](#)
 - 3.9.4. [Script](#)
- 3.10. [Miscellaneous](#)
 - 3.10.1. [Mountpoint](#)
 - 3.10.2. [Log Latency](#)
 - 3.10.3. [Scan RTCM](#)
- 3.11. [PPP Client](#)
 - 3.11.1. [Obs Mountpoint](#)
 - 3.11.1.1. [XYZ](#)
 - 3.11.2. [Corr Mountpoint](#)
 - 3.11.3. [Options](#)
 - 3.11.3.1. [Use Phase Obs](#)
 - 3.11.3.2. [Estimate Tropo](#)
 - 3.11.3.3. [Use GLONASS](#)
 - 3.11.3.4. [Use Galileo](#)
 - 3.11.4. [Options cont'd](#)
 - 3.11.4.1. [XYZ Init](#)
 - 3.11.4.2. [XYZ White Noise](#)
 - 3.11.4.3. [Quick-Start](#)
 - 3.11.4.4. [PPP Plot](#)
 - 3.11.5. [NMEA](#)
 - 3.11.5.1. [NMEA File](#)
 - 3.11.5.2. [NMEA Port](#)
 - 3.11.6. [Receiver Antenna](#)
 - 3.11.6.1. [ANTEX File](#)
 - 3.11.6.2. [Antenna Name](#)
 - 3.11.7. [Satellite Antenna](#)
 - 3.11.7.1. [Ignore Offsets](#)
 - 3.11.8. [Sigmas](#)
 - 3.11.8.1. [Code](#)
 - 3.11.8.2. [Phase](#)
 - 3.11.8.3. [Tropo Init](#)
 - 3.11.8.4. [Tropo White Noise](#)
 - 3.11.9. [Options cont'd](#)
 - 3.11.9.1. [Sync Corr](#)
 - 3.11.9.2. [Averaging](#)

3.12. [Streams](#)

3.12.1 [Edit Streams](#)

3.12.2 [Delete Stream](#)

3.12.3 [Reconfigure Streams On-the-fly](#)

3.13. [Logging](#)

3.13.1 [Log](#)

3.13.2 [Throughput](#)

3.13.3 [Latency](#)

3.13.4 [PPP Plot](#)

3.14. [Bottom Menu Bar](#)

3.14.1. [Add Stream - Coming from Caster](#)

3.14.1.1 [Caster Host and Port](#)

3.14.1.2 [Casters Table](#)

3.14.1.3 [User and Password](#)

3.14.1.4 [Get Table](#)

3.14.1.5 [NTRIP Version](#)

3.14.1.6 [Map](#)

3.14.2 [Add Stream - Coming from TCP/IP Port](#)

3.14.3 [Add Stream - Coming from UDP Port](#)

3.14.4 [Add Stream - Coming from Serial Port](#)

3.14.5 [Start](#)

3.14.6 [Stop](#)

3.15. [Command Line Options](#)

3.15.1. [No Window Mode](#)

3.15.2. [Offline Mode](#)

3.15.3. [Configuration File](#)

3.1. Top Menu Bar

The top menu bar allows to select a font for the BNC windows, save configured options or quit the program execution. It also provides access to a program documentation.

3.1.1 File

The 'File' button lets you

- select an appropriate font.
Use smaller font size if the BNC main window exceeds the size of your screen.
- save selected options in configuration file.
When using 'Save & Reread Configuration' while BNC is already processing data, some configuration options become immediately effective on-the-fly without interrupting uninvolved threads. See annexed section 'Configuration Example' for a list of on-the-fly changeable configuration options.
- quit the BNC program.

3.1.2 Help

The 'Help' button provides access to

- help contents.
You may keep the 'Help Contents' window open while configuring BNC.
- a 'Flow Chart' showing BNC linked to a real-time GNSS network engine such as RTNet.
- general information about BNC.
Close the 'About BNC' window to continue working with BNC.

BNC comes with a help system providing online information about its functionality and usage. Short descriptions are available for any widget. Focus to the relevant widget and press Shift+F1 to request help information. A help text appears immediately; it disappears as soon as the user does something else. The dialogs on some operating systems may provide a "?" button that users can click; they then click the relevant widget to pop up the help text.

3.2. Proxy - for usage in a protected LAN

If you are running BNC within a protected Local Area Network (LAN), you might need to use a proxy server to access the Internet. Enter your proxy server IP and port number in case one is operated in front of BNC. If you don't know the IP and port of your proxy server, check the proxy server settings in your Internet browser or ask your network administrator.

Note that IP streaming is often not allowed in a LAN. In this case you need to ask your network administrator for an appropriate modification of the local security policy or for the installation of a TCP relay to the NTRIP broadcasters. If these are not possible, you might need to run BNC outside your LAN on a host that has unobstructed connection to the Internet.

3.3. General

The following defines general settings for BNC's logfile, file handling, reconfiguration on-the-fly, and auto-start.

3.3.1 Logfile - optional

Records of BNC's activities are shown in the 'Log' tab on the bottom of the main window. These logs can be saved into a file when a valid path is specified in the 'Logfile (full path)' field. The logfile name will automatically be extended by a string '_YYMMDD' carrying the current date. This leads to series of daily logfiles when running BNC continuously for extended. Message logs cover the communication status between BNC and the NTRIP broadcaster as well as problems that may occur in the communication link, stream availability, stream delay, stream conversion etc. All times are given in UTC. The default value for 'Logfile (full path)' is an empty option field, meaning that BNC logs will not saved into a file.

3.3.2 Append Files - optional

When BNC is started, new files are created by default and any existing files with the same name will be overwritten. However, users might want to append existing files following a restart of BNC, a system crash or when BNC crashed. Tick 'Append files' to continue with existing files and keep what has been recorded so far. Note that option 'Append files' affects all types of files created by BNC.

3.3.3 Reread Configuration - optional

When operating BNC online in 'no window' mode (command line option -nw), some configuration options can nevertheless be changed on-the-fly without interrupting the running process. For that you force the program to reread parts of its configuration in pre-defined intervals from the disk. Select '1 min', '1 hour', or '1 day' to let BNC reread on-the-fly changeable configuration options every full minute, hour, or day. This lets in between edited options become effective without interrupting uninvolved threads. See annexed section 'Configuration Example' for a configuration file example and a list of on-the-fly changeable options.

3.3.4 Auto Start - optional

You may like to auto-start BNC at startup time in window mode with pre-assigned configuration options. This may be required i.e. immediately after booting your system. Tick 'Auto start' to supersede the usage of the 'Start' button. Make sure that you maintain a link to BNC for that in your Autostart directory (Windows systems) or call BNC in a script below directory /etc/init.d (Unix/Linux/Mac systems).

See BNC's command line option -nw for an auto-start of BNC in 'no window' mode.

3.3.5 Raw Output File - optional

BNC can save all data coming in through various streams in the received order and format together in one single daily file. This is of importance i.e. when using the PPP option in offline mode where the contents of different streams carrying observations, orbit/clock correctors, and broadcast ephemeris are to be read from one file. Data will be saved in blocks in the received format seperated by ASCII records like (example):

```
2010-08-03T18:05:28 RTCM3EPH RTCM_3 67
```

This example block header tells you that 67 bytes are saved in the data block following this record. The information in this block is encoded in RTCM Version 3.x format, comes from Mountpoint RTCM3EPH and was received at 18:05:29 UTC on 2010-08-03. BNC adds its own time stamps because a complete time reference may not be provided for all incoming observations and epochs.

Note that streams in a 'Raw output file' which shall later be used in an offline PPP calculation must all be encoded in the same format.

The default value for 'Raw output file (full path)' is an empty option field, meaning that BNC will not save raw data into a daily file.

3.4. RINEX Observations

Observations will be converted to RINEX if they come in either RTCM Version 2.x, RTCM Version 3.x, or RTIGS format. BNC's RINEX Version 2 observation files generally contain C1, P1, L1, S1, C2, P2, L2 and S2 observations. RINEX Version 3 observation files generally contain the following observation types:

- For GPS satellites, 'G': C1C L1C D1C S1C C1W L1W D1W S1W C2P L2P D2P S2P C2X L2X D2X S2X C5 L5 D5 S5
- For GLONASS satellites, 'R': C1C L1C D1C S1C C1P L1P D1P S1P C2P L2P D2P S2P C2C L2C D2C S2C
- For Geostationary signal payloads, 'S': C1C L1C D1C S1C C1W L1W D1W S1W
- For Galileo satellites, 'E': C1 L1 D1 S1 C5 L5 D5 S5

In case an observation is unavailable, its value is set to zero '0.000'. Note that the 'RINEX TYPE' field in the RINEX Observation file header is always set to 'M(MIXED)' even if the file only contains data from one system.

The screenshot below shows an example setup of BNC when converting streams to RINEX. Streams are coming in from various NTRIP broadcasters as well as via a plain UDP and a serial communication link. Decoder 'ZERO' has been selected for one stream to not convert its contents but save it in original format.

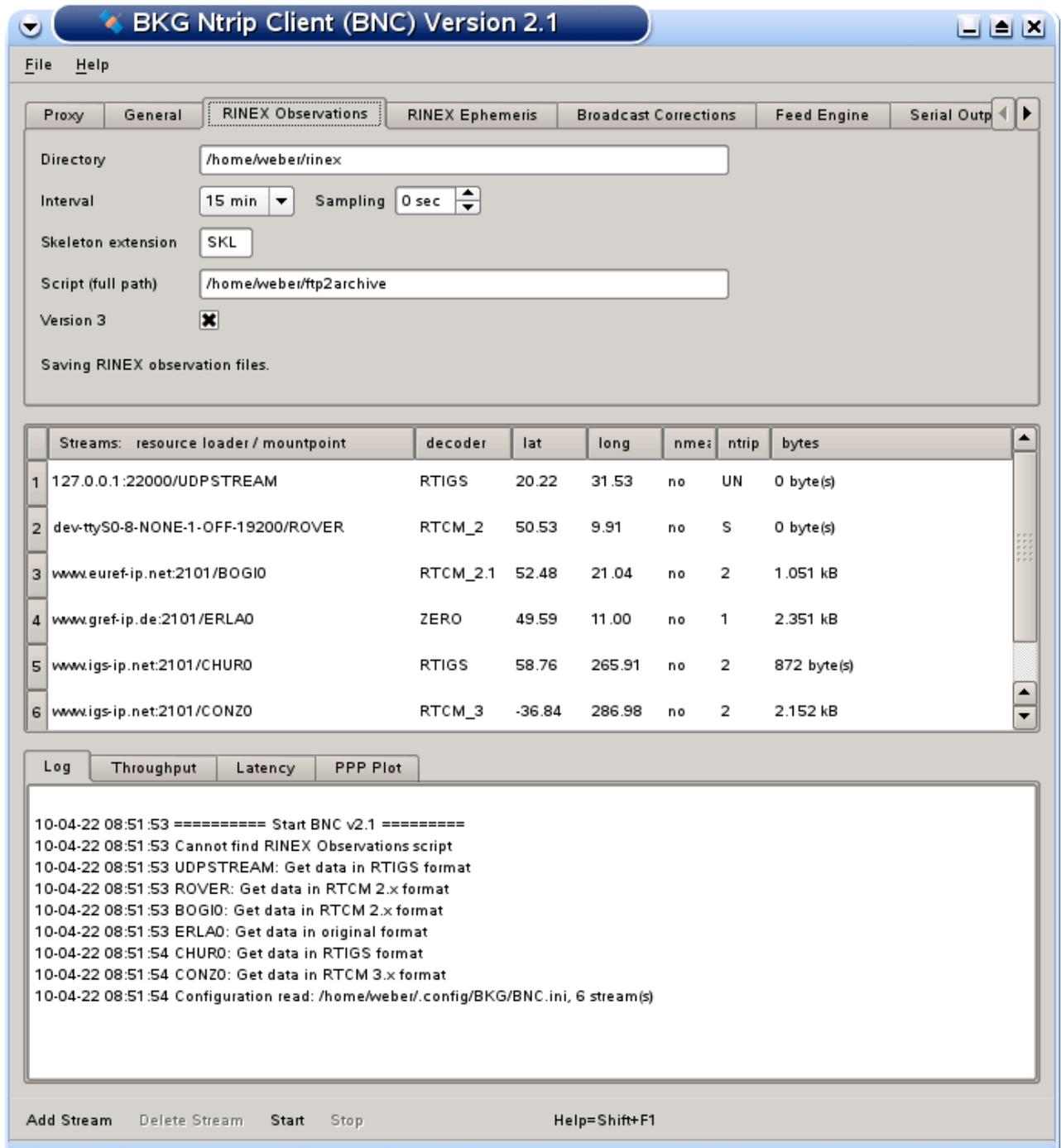


Figure: BNC translating incoming streams to 15 min RINEX Version 3 files.

3.4.1 RINEX File Names

RINEX file names are derived by BNC from the first 4 characters of the corresponding stream's mountpoint (4Char Station ID). For example, data from mountpoints FRANKFURT and WETTZELL will have hourly RINEX Observation files named

FRAN{ddd}{h}.{yy}O
 WETT{ddd}{h}.{yy}O

where 'ddd' is the day of year, 'h' is a letter which corresponds to an hour long UTC time block and 'yy' is the year.

If there are more than one stream with identical 4Char Station ID (same first 4 characters for their mountpoints), the mountpoint strings are split into two sub-strings and both become part of the RINEX file name. For example, when simultaneously retrieving data from mountpoints FRANKFURT and FRANCE, their hourly RINEX Observation files are named as

```
FRAN{ddd}{h}_KFURT.{yy}O  
FRAN{ddd}{h}_CE.{yy}O.
```

If several streams show exactly the same mountpoint name (example: BRUS0 from www.euref-ip.net and BRUS0 from www.igs-ip.net), BNC adds an integer number to the file name leading i.e. to hourly RINEX Observation files like

```
BRUS{ddd}{h}_0.{yy}O  
BRUS{ddd}{h}_1.{yy}O.
```

Note that RINEX file names for all intervals less than 1 hour follow the file name convention for 15 minutes RINEX Observation files i.e.

```
FRAN{ddd}{h}{mm}.{yy}O
```

where 'mm' is the starting minute within the hour.

3.4.2 Directory - optional

Here you can specify the path to where the RINEX Observation files will be stored. If the specified directory does not exist, BNC will not create RINEX Observation files. Default value for 'Directory' is an empty option field, meaning that no RINEX Observation files will be written.

3.4.3 File Interval - mandatory if 'Directory' is set

Select the length of the RINEX Observation file generated. The default value is 15 minutes.

3.4.4 Sampling - mandatory if 'Directory' is set

Select the RINEX Observation sampling interval in seconds. A value of zero '0' tells BNC to store all received epochs into RINEX. This is the default value.

3.4.5 Skeleton Extension - optional

Whenever BNC starts generating RINEX Observation files (and then once every day at midnight), it first tries to retrieve information needed for RINEX headers from so-called public RINEX header skeleton files which are derived from sitelogs. A HTTP link to a directory containing these skeleton files may be available through data field number 7 of the affected NET record in the source-table. See <http://www.epncb.oma.be:80/stations/log/skl/brus.skl> for an example of a public RINEX header skeleton file for the Brussels EPN station.

However, sometimes public RINEX header skeleton files are not available, its contents is not up to date, or you need to put additional/optional records in the RINEX header. For that BNC allows using personal skeleton files that contain the header records you would like to include. You can derive a personal RINEX header skeleton file from the information given in an up to date sitelog. A file in the RINEX Observations 'Directory' with a 'Skeleton extension' suffix is interpreted by BNC as a personal RINEX header skeleton file for the corresponding stream.

Examples for personal skeleton file name convention: RINEX Observation files for mountpoints WETTZELL, FRANKFURT and FRANCE (same 4Char Station ID), BRUS0 from www.euref-ip.net and BRUS0 from www.igs-ip.net (same 4Char Station ID, identical mountpoint stings) would accept personal skeleton files named

WETT.sk1
FRAN_KFURT.sk1
FRAN_CE.sk1
BRUS_0.sk1
BRUS_1.sk1

if 'Skeleton extension' is set to 'sk1'.

Note the following regulations regarding personal RINEX header skeleton files:

- If such a file exists in the 'RINEX directory', the corresponding public RINEX header skeleton file is ignored. The RINEX header is generated solely from the contents of the personal skeleton.
- Personal skeletons should contain a complete first header record of type
 - RINEX VERSION / TYPE
- They should then contain an empty header record of type
 - PGM / RUN BY / DATEBNC will complete this line and include it in the actual RINEX file header.
- They should further contain complete header records of type
 - MARKER NAME
 - OBSERVER / AGENCY
 - REC # / TYPE / VERS
 - ANT # / TYPE
 - APPROX POSITION XYZ
 - ANTENNA: DELTA H/E/N
 - WAVELENGTH FACT L1/2
- They may contain any other optional complete header record as defined in the RINEX documentation.
- They should then contain empty header records of type
 - # / TYPES OF OBSERV
 - TIME OF FIRST OBSBNC will include these lines in the final RINEX file header together with an additional
 - COMMENTline describing the source of the stream.
- They should finally contain an empty header record of type
 - END OF HEADER (last record)

If neither a public nor a personal RINEX header skeleton file is available for BNC, a default header will be used.

3.4.6 Script - optional

Whenever a RINEX Observation file is saved, you might want to compress, copy or upload it immediately via FTP. BNC allows you to execute a script/batch file to carry out these operations. To do that specify the full path of the script/batch file here. BNC will pass the RINEX Observation file path to the script as a command line parameter (%1 on Windows systems, \$1 on Unix/Linux/Mac systems).

The triggering event for calling the script or batch file is the end of a RINEX Observation file 'Interval'. If that is overridden by a stream outage, the triggering event is the stream reconnection.

As an alternative to initiating file uploads through BNC, you may like to call an upload script or batch file through your crontable or Task Scheduler (independent from BNC) once every 2 or 3 minutes after the end of each RINEX file 'Interval'.

3.4.7 Version - optional

The default format for RINEX Observation files is RINEX Version 2.11. Select 'Version 3' if you would like to save observations in RINEX Version 3 format.

3.5. RINEX Ephemeris

Broadcast ephemeris can be saved as RINEX Navigation files when received via RTCM Version 3.x as message types 1019 (GPS) or 1020 (GLONASS) or 1045 (proposed, Galileo) or via RTIGS records type 300. The file name convention follows the details given in section 'RINEX File Names' except that the first four characters are 'BRDC' and the last character is

- 'N' or 'G' for GPS or GLONASS ephemeris in two separate RINEX Version 2.11 Navigation files, or
- 'P' for GPS plus GLONASS plus Galileo ephemeris saved together in one RINEX Version 3 Navigation file.

Note that streams dedicated to carry Broadcast Ephemeris messages in RTCM v3 format in high repetition rates are listed on <http://igs.bkg.bund.de/ntrip/ephemeris>.

3.5.1 Directory - optional

Specify the path for saving broadcast ephemeris data as RINEX Navigation files. If the specified directory does not exist, BNC will not create RINEX Navigation files. Default value for Ephemeris 'Directory' is an empty option field, meaning that no RINEX Navigation files will be created.

3.5.2 Interval - mandatory if 'Directory' is set

Select the length of the RINEX Navigation file generated. The default value is 1 day.

3.5.3 Port - optional

BNC can output broadcast ephemeris in RINEX Version 3 ASCII format on your local host (IP 127.0.0.1) through an IP 'Port'. This function is introduced in order to support i.e. the 'BKG Ntrip Sate Space Server' (BNS) which transforms IGS clocks and orbits into corrections to broadcast ephemeris. Specify an IP port number to activate this function. The default is an empty option field, meaning that no ASCII ephemeris output via IP port is generated.

The source code for BNC comes with an example perl script 'test_tcpip_client.pl' that allows you to read BNC's ASCII ephemeris output from the IP port.

3.5.4 Version - optional

Default format for RINEX Navigation files containing broadcast ephemeris is RINEX Version 2.11. Select 'Version 3' if you want to save the ephemeris in RINEX Version 3 format.

Note that this does not concern the broadcast ephemeris output through IP port which is always in RINEX Version 3 format.

3.6. Broadcast Corrections

RTCM is in the process of developing new Version 3 messages to transport satellite clock and orbit corrections in real-time. Based on the latest available proposal, the following premature so-called 'State Space Representation' (SSR) messages currently under discussion have been implemented in BNC. The information below should not be misunderstood as a programmers guide. Programming efforts would definitely require access to the RTCM documentation of SSR messages.

- Message type 1057: GPS orbit corrections to Broadcast Ephemeris
- Message type 1058: GPS clock corrections to Broadcast Ephemeris
- Message type 1059: GPS code biases
- Message type 1060: Combined orbit and clock corrections to GPS Broadcast Ephemeris
- Message type 1061: GPS User Range Accuracy (URA)
- Message type 1062: High-rate GPS clock corrections to Broadcast Ephemeris
- Message type 1063: GLONASS orbit corrections to Broadcast Ephemeris
- Message type 1064: GLONASS clock corrections to Broadcast Ephemeris
- Message type 1065: GLONASS code biases
- Message type 1066: Combined orbit and clock corrections to GLONASS Broadcast Ephemeris
- Message type 1067: GLONASS User Range Accuracy (URA)
- Message type 1068: High-rate GLONASS clock corrections to Broadcast Ephemeris

RTCM Version 3 streams carrying these messages may be used i.e. to support real-time Precise Point Positioning (PPP) applications.

When using clocks from Broadcast Ephemeris (with or without applied corrections) or clocks from SP3 files, it may be important to understand that they are not corrected for the conventional periodic relativistic effect. Chapter 10 of the IERS Conventions 2003 mentions that the conventional periodic relativistic correction to the satellite clock (to be added to the

broadcast clock) is computed as $dt = -2 (R * V) / c^2$ where $R * V$ is the scalar product of the satellite position and velocity and c is the speed of light. This can also be found in the GPS Interface Specification, IS-GPS-200, Revision D, 7 March 2006.

Orbit corrections are provided in along-track, cross-track and radial components. These components are defined in the Earth-centered, Earth-fixed reference frame of the broadcast ephemerides. For an observer in this frame, the along-track component is aligned in both direction and sign with the velocity vector, the cross-track component is perpendicular to the plane defined by the satellite position and velocity vectors, and the radial direction is perpendicular to the along track and cross-track ones. The three components form a right-handed orthogonal system.

After applying corrections, the satellite position and clock is referred to the 'ionospheric free' phase center of the antenna which is compatible with the broadcast orbit reference.

The orbit and clock corrections do not include local effects (like Ocean Loading or Solid Earth Tides) or atmospheric effects (Ionosphere and/or troposphere). Depending on the accuracy of your application you should correct for such effects by other means. There is currently no RTCM SSR message for ionospheric state parameters. Such messages are needed for accurate single frequency applications. The development of Iono messages will be the next step in the schedule of the RTCM State Space Representation Working Group.

Broadcast Corrections can be saved by BNC in files. The file name convention for Broadcast Correction files follows the convention for RINEX files except for the last character of the file name suffix which is set to "C".

Saved files contain blocks of records in plain ASCII format where - separate for GPS, GLONASS, message types, streams, and epochs - the begin of a block is indicated by a line like (examples):

```
! Orbits/Clocks: 30 GPS 0 Glonass CLK11
```

or

```
! Orbits/Clocks: 0 GPS 19 Glonass CLK11
```

Such line informs you about the number of records (here 30 and 19) carrying GPS or GLONASS related parameters you should receive next as part of a certain stream.

The first five parameters in each broadcast corrections record are:

- RTCMv3 message type number
- SSR message update interval indicator
 - 0 = 1 sec
 - 1 = 2 sec
 - 2 = 5 sec
 - 3 = 10 sec
 - 4 = 15 sec
 - 5 = 30 sec
 - 6 = 60 sec
 - 7 = 120 sec
 - 8 = 240 sec
 - 9 = 300 sec

- 10 = 600 sec
- 11 = 900 sec
- 12 = 1800 sec
- 13 = 3600 sec
- 14 = 7200 sec
- 15 = 10800 sec
- GPS Week
- Second in GPS Week
- GNSS Indicator and Satellite Vehicle Pseudo Random Number

In case of RTCM message types 1057 or 1063 these parameters are followed by

- IOD referring to Broadcast Ephemeris set
- Radial Component of Orbit Correction to Broadcast Ephemeris [m]
- Along-track Component of Orbit Correction to Broadcast Ephemeris [m]
- Cross-track Component of Orbit Correction to Broadcast Ephemeris [m]
- Velocity of Radial Component of Orbit Correction to Broadcast Ephemeris [m/s]
- Velocity of Along-track Component of Orbit Correction to Broadcast Ephemeris [m/s]
- Velocity of Cross-track Component of Orbit Correction to Broadcast Ephemeris [m/s]

Undefined parameters are set to zero "0.000".

Example:

```

...
1057 0 1538 211151.0 G18      1      0.034   0.011  -0.064   0.000
0.000   0.000
1057 0 1538 211151.0 G16     33     -0.005   0.194  -0.091   0.000
0.000   0.000
1057 0 1538 211151.0 G22     50      0.008  -0.082  -0.001   0.000
0.000   0.000
...
1063 0 1538 211151.0 R09    111     -0.011  -0.014   0.005   0.000
0.000   0.000
1063 0 1538 211151.0 R10     43      0.000  -0.009  -0.002   0.000
0.000   0.000
1063 0 1538 211151.0 R21     75     -0.029   0.108   0.107   0.000
0.000   0.000
...

```

In case of RTCM message types 1058 or 1064 the first five parameters are followed by

- IOD set to zero "0"
- C0 polynomial coefficient for Clock Correction to Broadcast Ephemeris [m]
- C1 polynomial coefficient for Clock Correction to Broadcast Ephemeris [m/s]
- C2 polynomial coefficient for Clock Correction to Broadcast Ephemeris [m/s**2]

Example:

```

...
1058 0 1538 211151.0 G18      0      1.846   0.000   0.000
1058 0 1538 211151.0 G16      0      0.376   0.000   0.000
1058 0 1538 211151.0 G22      0      2.727   0.000   0.000
...
1064 0 1538 211151.0 R08      0      8.956   0.000   0.000
1064 0 1538 211151.0 R07      0     14.457   0.000   0.000
1064 0 1538 211151.0 R23      0      6.436   0.000   0.000

```

...

In case of RTCM message types 1060 or 1066 the first five parameters are followed by

- IOD referring to Broadcast Ephemeris set
- C0 polynomial coefficient for Clock Correction to Broadcast Ephemeris [m]
- Radial Component of Orbit Correction to Broadcast Ephemeris [m]
- Along-track Component of Orbit Correction to Broadcast Ephemeris [m]
- Cross-track Component of Orbit Correction to Broadcast Ephemeris [m]
- C1 polynomial coefficient for Clock Correction to Broadcast Ephemeris [m]
- Velocity of Radial Component of Orbit Correction to Broadcast Ephemeris [m/s]
- Velocity of Along-track Component of Orbit Correction to Broadcast Ephemeris [m/s]
- Velocity of Cross-track Component of Orbit Correction to Broadcast Ephemeris [m/s]
- C2 polynomial coefficient for Clock Correction to Broadcast Ephemeris [m]

Example:

```

...
1060 0 1538 211610.0 G30      82      2.533      0.635      -0.359      -0.598
0.000      0.000      0.000      0.000      0.000
1060 0 1538 211610.0 G31       5      -4.218      -0.208      0.022      0.002
0.000      0.000      0.000      0.000      0.000
1060 0 1538 211610.0 G32      28      -2.326      0.977      -0.576      0.142
0.000      0.000      0.000      0.000      0.000
...
1066 0 1538 211610.0 R22      27      1.585      2.024      2.615      -2.080
0.000      0.000      0.000      0.000      0.000
1066 0 1538 211610.0 R23      27      6.277      2.853      4.181      1.304
0.000      0.000      0.000      0.000      0.000
1066 0 1538 211610.0 R24      27      0.846      1.805      13.095      6.102
0.000      0.000      0.000      0.000      0.000
...

```

In case of RTCM message types 1059 or 1065 the first five parameters are followed by

- Number of Code Biases
- Indicator to specify the signal and tracking mode
- Code Bias
- Indicator to specify the signal and tracking mode
- Code Bias
- etc.

Example:

```

...
1059 0 1538 211151.0 G18 2 0   -0.010 11   -0.750
1059 0 1538 211151.0 G16 2 0   -0.040 11   -0.430
1059 0 1538 211151.0 G22 2 0   -0.630 11   -2.400
...

```

3.6.1 Directory - optional

Specify a directory for saving Broadcast Corrections in files. If the specified directory does not exist, BNC will not create Broadcast Correction files. Default value for Broadcast Corrections 'Directory' is an empty option field, meaning that no Broadcast Correction files will be created.

3.6.2 Interval - mandatory if 'Directory' is set

Select the length of the Broadcast Correction files. The default value is 1 day.

3.6.3 Port - optional

BNC can output epoch by epoch synchronized Broadcast Corrections in ASCII format on your local host (IP 127.0.0.1) through an IP 'Port'. Specify an IP port number to activate this function. The default is an empty option field, meaning that no Broadcast Correction output via IP port is generated.

The output format equals the format used for saving Broadcast Corrections in a file with the exception that the Mountpoint is added at each line's end.

The following is an example output for streams from Mountpoints RTCMSSR, CLK10 and CLK11:

```

...
1057 0 1538 211151.0 G18      1      0.034      0.011      -0.064      0.000
0.000      0.000 RTCMSSR
1057 0 1538 211151.0 G16     33     -0.005      0.194      -0.091      0.000
0.000      0.000 RTCMSSR
1057 0 1538 211151.0 G22     50      0.008     -0.082     -0.001      0.000
0.000      0.000 RTCMSSR
...
1058 0 1538 211151.0 G18      0      1.846      0.000 RTCMSSR
1058 0 1538 211151.0 G16      0      0.376      0.000 RTCMSSR
1058 0 1538 211151.0 G22      0      2.727      0.000 RTCMSSR
...
1059 0 1538 211151.0 G18  2  0  -0.010  11  -0.750 RTCMSSR
1059 0 1538 211151.0 G16  2  0  -0.040  11  -0.430 RTCMSSR
1059 0 1538 211151.0 G22  2  0  -0.630  11  -2.400 RTCMSSR
...
1063 0 1538 211151.0 R09    111    -0.011    -0.014     0.005     0.0000
0.000      0.000 RTCMSSR
1063 0 1538 211151.0 R10     43     0.000    -0.009    -0.002     0.0000
0.000      0.000 RTCMSSR
1063 0 1538 211151.0 R21     75    -0.029     0.108     0.107     0.0000
0.000      0.000 RTCMSSR
...
1064 0 1538 211151.0 R08      0      8.956      0.000 RTCMSSR
1064 0 1538 211151.0 R07      0     14.457      0.000 RTCMSSR
1064 0 1538 211151.0 R23      0      6.436      0.000 RTCMSSR
...
1066 0 1538 211610.0 R24     27     0.846      1.805     13.095     6.102
0.000      0.000      0.000      0.000      0.000 CLK11
1066 0 1538 211610.0 R23     27     6.277      2.853      4.181      1.304
0.000      0.000      0.000      0.000      0.000 CLK11
1066 0 1538 211610.0 R22     27     1.585      2.024      2.615     -2.080
0.000      0.000      0.000      0.000      0.000 CLK11
...
1060 0 1538 211610.0 G32     28    -2.326      0.977     -0.576      0.142
0.000      0.000      0.000      0.000      0.000 CLK10
1060 0 1538 211610.0 G31      5    -4.218     -0.208      0.022      0.002
0.000      0.000      0.000      0.000      0.000 CLK10
1060 0 1538 211610.0 G30     82     2.533      0.635     -0.359     -0.598
0.000      0.000      0.000      0.000      0.000 CLK10
...

```

The source code for BNC comes with an example perl script 'test_tcpip_client.pl' that allows you to read BNC's Broadcast Corrections from the IP port.

3.6.4 Wait for Full Epoch - mandatory if 'Port' is set

When feeding a real-time GNSS network engine waiting epoch by epoch for synchronized Broadcast Corrections, BNC drops (only concerning IP port output) whatever is received later than 'Wait for full epoch' seconds. A value of 2 to 5 seconds could be an appropriate choice for that, depending on the latency of the incoming Broadcast Corrections stream and the delay acceptable by your application. A message such as "COCK1: Correction overaged by 5 sec" shows up in BNC's logfile if 'Wait for full epoch' is exceeded.

3.7. Feed Engine

BNC can generate synchronized or unsynchronized observations epoch by epoch from all stations and satellites to feed a real-time GNSS network engine. Observations can be streamed out through an IP port and/or saved in a local file. The output is always in plain ASCII format and comprises the following parameters:

- For GPS satellites:
StationID GPSWeek GPSWeekSec 'G'PRN C1C L1C SlipCountL1 D1C S1C C1W
L1W SlipCountL1 D1W S1W C2P L2P SlipCountL2 D2P S2P C2X L2X
SlipCountL2 D2X S2X C5 L5 SlipCountL5 D5 S5
- For GLONASS satellites:
StationID GPSWeek GPSWeekSec 'R'PRN SlotNumber C1C L1C SlipCountL1 D1C
S1C C1P L1P SlipCountL1 D1P S1P C2P L2P SlipCountL2 D2P S2P C2C L2C
SlipCountL2 D2C S2C
- For Geostationary signal payloads:
StationID GPSWeek GPSWeekSec 'S'PRN C1C L1C SlipCountL1 D1C S1C C1W
L1W SlipCountL1 D1W S1W
- For Galileo satellites:
StationID GPSWeek GPSWeekSec 'E'PRN C1 L1 SlipCountL1 D1 S1 C5 L5
SlipCountL5 D5 S5

In case an observation is not available, its value is set to zero '0.000'.

Note on 'SlipCount':

It is the current understanding of BNC's authors that different slip counts could be referred to different phase measurements (i.e. L1C and L1P). The 'loss-of-lock' flags in RINEX are an example for making such kind of information available per phase measurement. However, it looks like we do have only one slip count in RTCM Version 3 for all phase measurements. As it could be that a receiver generates different slip counts for different phase measurements, we output one slip count per phase measurement to a listening real-time GNSS network engine.

The following is an output example for GPS, GLONASS and Galileo observations and observations obtained from a geostationary payload signal:

BKG Ntrip Client (BNC) Version 2.5

```
...
WTZX3 1616 149732.0000000 E52 27089285.092 142354765.663 -1
2212.322 45.500 27089287.942 106304461.365 -1 2212.404
42.300
...
WTZX3 1616 149732.0000000 G10 22608910.719 118810687.059 -1
2965.339 49.300 22608909.593 118810311.312 -1 2965.339
36.000 22608915.003 92579465.057 -1 2966.012 36.000
0.000 0.000 -1 0.000 0.000 0.000
0.000 -1 0.000 0.000
...
WTZX3 1616 149732.0000000 G07 23633028.684 124192961.644 -1
3686.418 48.800 23633026.847 124192961.885 -1 3686.418
35.000 23633032.480 96773737.419 -1 3685.139 35.000
23633033.547 96773738.190 -1 3685.172 43.500
0.000 0.000 -1 0.000 0.000
...
WTZX3 1616 149732.0000000 R20 2 24149338.926 129137949.211 48
2950.111 42.800 24149340.305 129137949.481 48 2950.111
41.800 24149356.146 100440627.082 48 2949.895 39.500
24149356.702 100440626.859 48 2949.896 40.000
...
```

The source code for BNC comes with a perl script called 'test_tcpip_client.pl' that allows you to read BNC's (synchronized or unsynchronized) ASCII observation output from the IP port and print it on standard output.

Note that any socket connection of an application to BNC's synchronized or unsynchronized observations ports is recorded in the 'Log' tab on the bottom of the main window together with a connection counter, resulting in log records like 'New client connection on sync/usync port: # 1'.

The following figure shows the screenshot of a BNC configuration where a number of streams is pulled from different NTRIP broadcasters to feed a GNSS engine via IP port output.

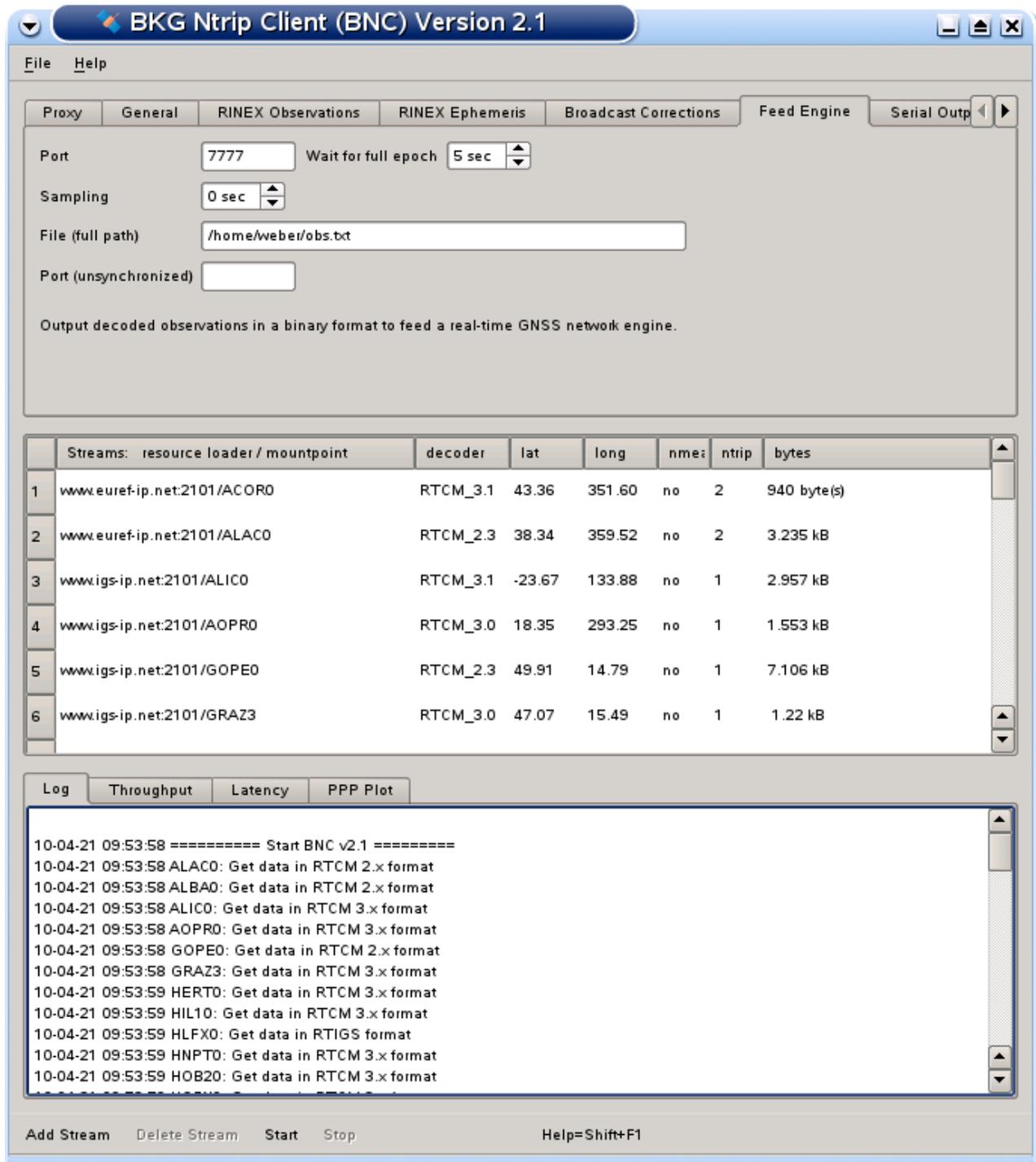


Figure: Synchronized BNC output via IP port to feed a GNSS real-time engine.

3.7.1 Port - optional

BNC can produce synchronized observations in ASCII format on your local host (IP 127.0.0.1) through an IP 'Port'. Synchronized means that BNC collects all data for any specific epoch which become available within a certain number of latency seconds (see 'Wait for Full Epoch' option). It then - epoch by epoch - outputs whatever has been received. Specify an IP port number here to activate this function. The default is an empty option field, meaning that no binary synchronized output is generated.

3.7.2 Wait for Full Epoch - mandatory if 'Port' is set

When feeding a real-time GNSS network engine waiting for synchronized input epoch by epoch, BNC drops whatever is received later than 'Wait for full epoch' seconds. A value of 3 to 5 seconds could be an appropriate choice for that, depending on the latency of the incoming streams and the delay acceptable for your real-time GNSS product. Default value for 'Wait for full epoch' is 5 seconds.

Note that 'Wait for full epoch' does not effect the RINEX Observation file content. Observations received later than 'Wait for full epoch' seconds will still be included in the RINEX Observation files.

3.7.3 Sampling - mandatory if 'File' or 'Port' is set

Select the synchronized observation output sampling interval in seconds. A value of zero '0' tells BNC to send/store all received epochs. This is the default value.

3.7.4 File - optional

Specifies the full path to a 'File' where synchronized observations are saved in plain ASCII format. The default value is an empty option field, meaning that no ASCII output file is created.

Beware that the size of this file can rapidly increase depending on the number of incoming streams. This option is primarily meant for testing and evaluation.

3.7.5 Port (unsynchronized) - optional

BNC can produce unsynchronized observations from all configured streams in ASCII format on your local host (IP 127.0.0.1) through an IP 'Port'. Unsynchronized means that BNC immediately forwards any received observation to the port. Specify an IP port number here to activate this function. The default is an empty option field, meaning that no binary unsynchronized output is generated.

3.8. Serial Output

You may use BNC to feed a serial connected device like an GNSS receiver. For that one of the incoming streams can be forwarded to a serial port. The following figure shows the screenshot of an example situation where BNC pulls a VRS stream from an NTRIP broadcaster to feed a serial connected rover.

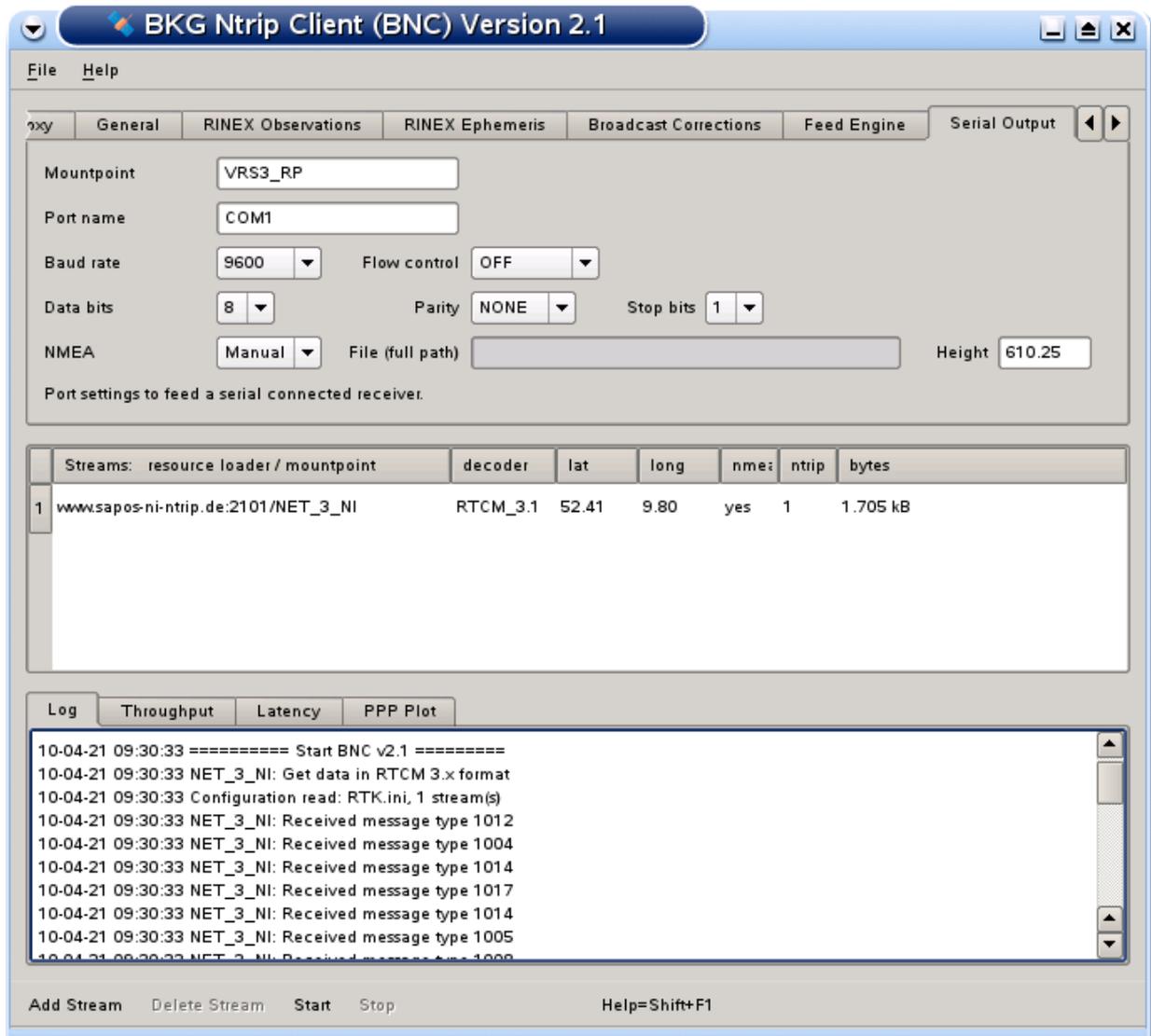


Figure: BNC pulling a VRS stream to feed a serial connected rover.

3.8.1 Mountpoint - optional

Enter a 'Mountpoint' to forward its corresponding stream to a serial connected GNSS receiver.

When selecting the serial communication options listed below, make sure that you pick those configured to the serial connected receiver.

3.8.2 Port Name - mandatory if 'Mountpoint' is set

Enter the serial 'Port name' selected on your host for communication with the serial connected receiver. Valid port names are

```
Windows:      COM1, COM2
Linux:        /dev/ttyS0, /dev/ttyS1
FreeBSD:      /dev/ttyd0, /dev/ttyd1
Digital Unix: /dev/tty01, /dev/tty02
HP-UX:        /dev/ttylp0, /dev/tty2p0
SGI/IRIX:     /dev/ttyf1, /dev/ttyf2
SunOS/Solaris: /dev/ttya, /dev/ttyb
```

Note that you must plug a serial cable in the port defined here before you start BNC.

3.8.3 Baud Rate - mandatory if 'Mountpoint' is set

Select a 'Baud rate' for the serial output link. Note that using a high baud rate is recommended.

3.8.4 Flow Control - mandatory if 'Mountpoint' is set

Select a 'Flow control' for the serial output link. Note that your selection must equal the flow control configured to the serial connected device. Select 'OFF' if you don't know better.

3.8.5 Parity - mandatory if 'Mountpoint' is set

Select the 'Parity' for the serial output link. Note that parity is often set to 'NONE'.

3.8.6 Data Bits - mandatory if 'Mountpoint' is set

Select the number of 'Data bits' for the serial output link. Note that often '8' data bits are used.

3.8.7 Stop Bits - mandatory if 'Mountpoint' is set

Select the number of 'Stop bits' for the serial output link. Note that often '1' stop bit is used.

3.8.8 NMEA - mandatory for VRS streams

Select 'Auto' to automatically forward all NMEA-GGA messages coming from your serial connected GNSS receiver to the NTRIP broadcaster and/or save them in a file.

Forwarding valid NMEA-GGA messages to the NTRIP broadcaster is required for receiving 'Virtual Reference Station' (VRS) streams. Thus, in case your serial connected receiver is not capable to provide them, the alternative for VRS streams is a 'Manual' simulation of an initial NMEA-GGA message. Its contents is based on the approximate (editable) latitude/longitude from the broadcaster's source-table and an approximate VRS height to be specified.

In summary: select 'Manual' only when handling a VRS stream and your serial connected GNSS receiver doesn't generate NMEA-GGA messages. Select 'Auto' otherwise.

3.8.9 File - optional if 'Auto' NMEA is set

Specify the full path to a file where NMEA messages coming from your serial connected receiver are saved.

3.8.10 Height - mandatory if 'Manual' NMEA is set

Specify an approximate 'Height' above mean sea level in meter for your VRS to simulate an initial NMEA-GGA message. Latitude and longitude for that (editable) are taken from the broadcaster's source-table.

This option concerns only 'Virtual Reference Stations' (VRS). Its setting is ignored in case of streams coming from physical reference stations.

3.9. Outages

At various times, the incoming stream might become unavailable or corrupted. In such cases, it is important that the BNC operator and/or the stream providers become aware of the situation so that necessary measures can be taken to restore the stream. Furthermore, continuous attempts to decode corrupted stream(s) can generate unnecessary workload for BNC. Outages and corruptions are handled by BNC as follows:

Stream outages: BNC considers a connection to be broken when there are no incoming data detected for more than 20 seconds. When this occurs, BNC will attempt to reconnect at a decreasing rate. It will first try to reconnect with 1 second delay, and again in 2 seconds if the previous attempt failed. If the attempt is still unsuccessful, it will try to reconnect within 4 seconds after the previous attempt and so on. The wait time doubles each time with a maximum wait time of 256 seconds.

Stream corruption: Not all bits chunk transfers to BNC's internal decoders return valid observations. Sometimes several chunks might be needed before the next observation can be properly decoded. BNC buffers all the outputs (both valid and invalid) from the decoder for a short time span (size derived from the expected 'Observation rate') and then determines whether a stream is valid or corrupted.

Outage and corruption events are reported in the 'Log' tab. They can also be passed on as parameters to a shell script or batch file to generate an advisory note to BNC operator or affected stream providers. This functionality lets users utilize BNC as a real-time performance monitor and alarm system for a network of GNSS reference stations.

3.9.1 Observation Rate - mandatory if 'Failure threshold', 'Recovery threshold', and 'Script' is set

BNC can collect all returns (success or failure) coming from a decoder within a certain short time span to then decide whether a stream has an outage or its content is corrupted. This procedure needs a rough a priori estimate of the expected observation rate of the incoming streams.

An empty option field (default) means that you don't want an explicit information from BNC about stream outages and incoming streams that cannot be decoded.

3.9.2 Failure Threshold - optional

Event 'Begin_Failure' will be reported if no data is received continuously for longer than the 'Failure threshold' time. Similarly, event 'Begin_Corrupted' will be reported when corrupted data is detected by the decoder continuously for longer than this 'Failure threshold' time. The default value is set to 15 minutes and is recommended so not to inundate user with too many event reports.

Note that specifying a value of zero '0' for the 'Failure threshold' will force BNC to report any stream failure immediately. Note also that for using this function you need to specify the 'Observation rate'.

3.9.3 Recovery Threshold - optional

Once a 'Begin_Failure' or 'Begin_Corrupted' event has been reported, BNC will check for when the stream again becomes available or uncorrupted. Event 'End_Failure' or 'End_Corrupted' will be reported as soon as valid observations are again detected continuously throughout the 'Recovery threshold' time span. The default value is set to 5 minutes and is recommended so not to inundate users with too many event reports.

Note that specifying a value of zero '0' for the 'Recovery threshold' will force BNC to report any stream recovery immediately. Note also that for using this function you need to specify the 'Observation rate'.

3.9.4 Script - optional

As mentioned previously, BNC can trigger a shell script or a batch file to be executed when one of the events described are reported. This script can be used to email an advisory note to network operator or stream providers. To enable this feature, specify the full path to the script or batch file in the 'Script' field. The affected stream's mountpoint and type of event reported ('Begin_Outage', 'End_Outage', 'Begin_Corrupted' or 'End_Corrupted') will then be passed on to the script as command line parameters (%1 and %2 on Windows systems or \$1 and \$2 on Unix/Linux/Mac systems) together with date and time information.

Leave the 'Script' field empty if you do not wish to use this option. An invalid path will also disable this option.

Examples for command line parameter strings passed on to the advisory 'Script' are:

```
FFMJ0 Begin_Outage 08-02-21 09:25:59
FFMJ0 End_Outage 08-02-21 11:36:02 Begin was 08-02-21 09:25:59
```

Sample script for Unix/Linux/Mac systems:

```
#!/bin/bash
sleep $((60*RANDOM/32767))
cat | mail -s "NABU: $1" email@address <<!
Advisory Note to BNC User,
Please note the following advisory received from BNC.
Stream: $*
Regards, BNC
!
```

Note the sleep command in this script which causes the system to wait for a random period of up to 60 seconds before sending the email. This should avoid overloading your mail server in case of a simultaneous failure of many streams.

3.10. Miscellaneous

This section describes a number of miscellaneous options which can be applied for a single stream (mountpoint) or for all configured streams.

The following figure shows RTCM message numbers contained in stream 'CONZ0' and the message latencies recorded every 10 seconds.

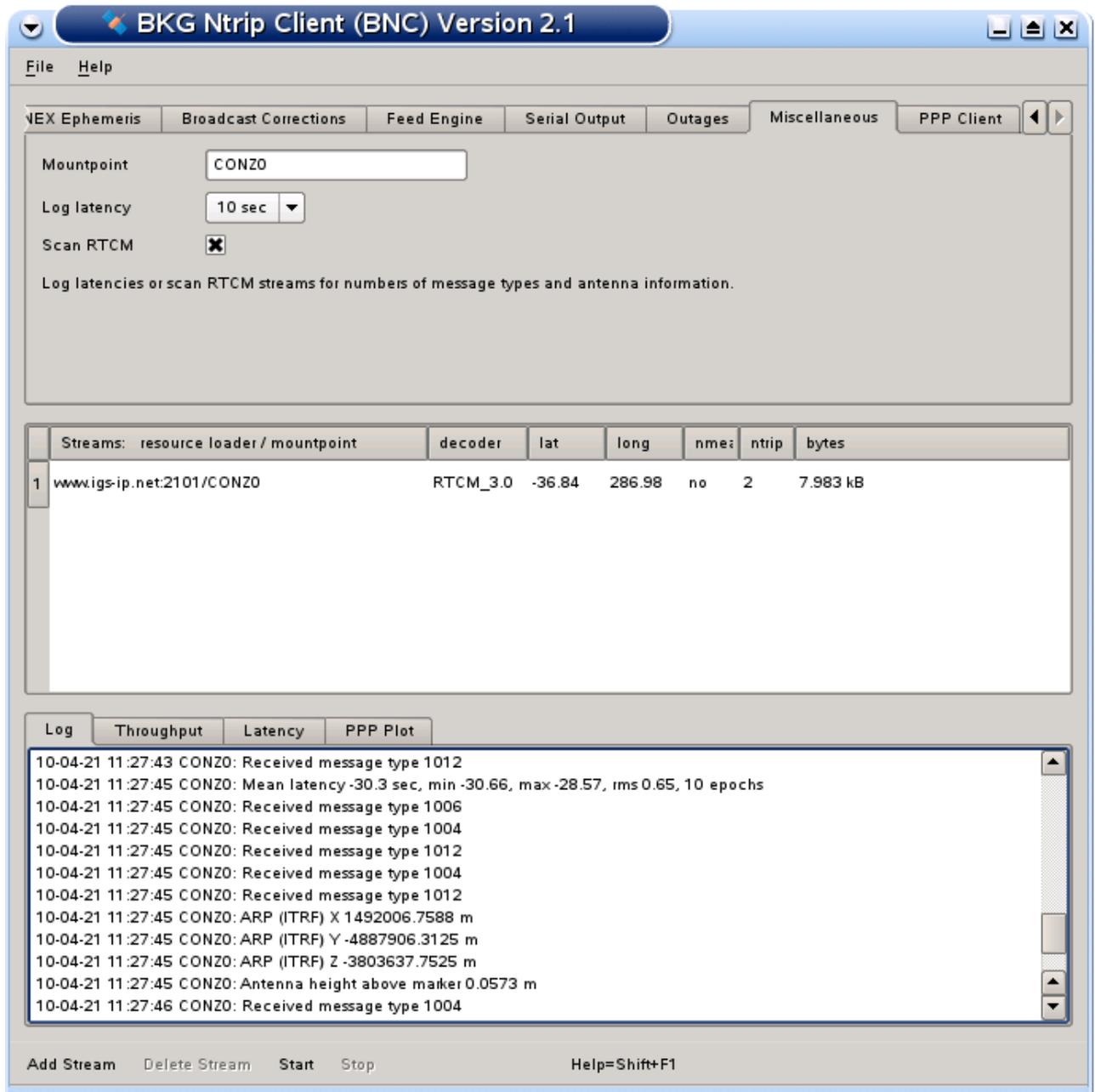


Figure: RTCM message numbers and latencies.

3.10.1 Mountpoint - optional

Specify a mountpoint to apply one or several of the 'Miscellaneous' options to the corresponding stream. Enter 'ALL' if you want to apply these options to all configured streams. An empty option field (default) means that you don't want BNC to apply any of these options.

3.10.2 Log Latency - optional

BNC can average latencies per stream over a certain period of GPS time, the 'Log latency' interval. Mean latencies are calculated from the individual latencies of at most one (first incoming) observation or correction to Broadcast Ephemeris per second. The mean latencies are then saved in BNC's logfile. Note that computing correct latencies requires the clock of

the host computer to be properly synchronized. Note further that the latencies available from the 'Latency' tab on the bottom of the main window represent individual latencies and not the mean latencies for the logfile.

Latency: Latency is defined in BNC by the following equation:

$$\begin{array}{r}
 \text{UTC time provided by BNC's host} \\
 - \text{GPS time of currently processed epoch} \\
 + \text{Leap seconds between UTC and GPS time} \\
 \hline
 = \text{Latency}
 \end{array}$$

Statistics: BNC counts the number of GPS seconds covered by at least one observation. It also estimates an observation rate (independent from the a priori specified 'Observation rate') from all observations received throughout the first full 'Log latency' interval. Based on this rate, BNC estimates the number of data gaps when appearing in subsequent intervals.

Latencies of observations or corrections to Broadcast Ephemeris and statistical information can be recorded in the 'Log' tab at the end of each 'Log latency' interval. A typical output from a 1 hour 'Log latency' interval would be:

```
08-03-17 15:59:47 BRUS0: Mean latency 1.47 sec, min 0.66, max 3.02, rms
0.35, 3585 epochs, 15 gaps
```

Select a 'Log latency' interval to activate this function or select the empty option field if you do not want BNC to log latencies and statistical information.

3.10.3 Scan RTCM - optional

When configuring a GNSS receiver for RTCM stream generation, the setup interface may not provide details about RTCM message types. As reliable information concerning stream contents should be available i.e. for NTRIP broadcaster operators to maintain the broadcaster's source-table, BNC allows to scan RTCM streams for incoming message types and printout some of the contained meta-data. The idea for this option arose from 'InspectRTCM', a comprehensive stream analyzing tool written by D. Stoecker.

Tick 'Scan RTCM' to scan RTCM Version 2.x or 3.x streams and log all contained

- numbers of incoming message types
- Antenna Reference Point (ARP) coordinates
- Antenna Phase Center (APC) coordinates
- antenna height above marker
- antenna descriptor.

Note that in RTCM Version 2.x the message types 18 and 19 carry only the observables of one frequency. Hence it needs two type 18 and 19 messages per epoch to transport the observations from dual frequency receivers.

Logged time stamps refer to message reception time and allow to understand repetition rates. Enter 'ALL' if you want to log this information from all configured streams. Beware that the size of the logfile can rapidly increase depending on the number of incoming RTCM streams.

This option is primarily meant for testing and evaluation. Use it to figure out what exactly is produced by a specific GNSS receiver's configuration. An empty option field (default) means that you don't want BNC to print the message type numbers and antenna information carried in RTCM streams.

3.11. PPP Client

BNC can derive coordinates for a rover position following the Precise Point Positioning (PPP) approach. It uses either code or code plus phase data in ionosphere free linear combinations P3 or L3. Besides pulling a stream of observations from a dual frequency receiver, this also requires pulling in addition

- a stream carrying satellite orbit and clock corrections to Broadcast Ephemeris in the form of 'State Space Representation' (SSR) messages as proposed by RTCM (i.e. premature message type 1060). Note that for BNC these correctors need to be referred to the satellite's Antenna Phase Center (APC). Streams providing such messages are listed on <http://igs.bkg.bund.de/ntrip/orbits>. Stream products.igs-ip.net:2101/CLK11 is an example.
- a stream carrying Broadcast Ephemeris available as RTCM Version 3 message types 1019, 1020, and (proposed) 1045. This is a must only when the stream coming from the receiver does not contain Broadcast Ephemeris or provides them only at very low repetition rate. Streams providing such messages are listed on <http://igs.bkg.bund.de/ntrip/ephemeris>. Stream RTCM3EPH on caster products.igs-ip.net:2101 is an example.

The following figure provides the screenshot of an example PPP session with BNC.

BKG Ntrip Client (BNC) Version 2.5

File Help

Observations RINEX Ephemeris Broadcast Corrections Feed Engine Serial Output Outages Miscellaneous PPP (1)

Obs Mountpoint FFMJ1 PPP X Y Z

Corr Mountpoint CLK11

Options Use phase obs Estimate tropo Use GLONASS Use Galileo

Options conf'd 100.0 Sigma XYZ Init 100.0 Sigma XYZ Noise Quick-Start (sec) PPP Plot

NMEA File Port

Coordinates from Precise Point Positioning (PPP).

Streams	resource loader / mountpoint	decoder	lat	long	nmea	ntrip	bytes
1	products.igs-ip.net:2101/CLK11	RTCM_3.0	50.00	10.00	no	1	5.457 kB
2	products.igs-ip.net:2101/RTCM3EPH	RTCM_3	50.09	8.66	no	1	40.923 kB
3	www.igs-ip.net:2101/FFMJ1	RTCM_3.0	50.09	8.66	no	1	43.802 kB

Log Throughput Latency PPP Plot

```

11-02-06 15:05:23 ===== Start BNC v2.5 =====
11-02-06 15:05:23 CLK11: Get data in RTCM 3.x format
11-02-06 15:05:23 RTCM3EPH: Get data in RTCM 3.x format
11-02-06 15:05:23 FFMJ1: Get data in RTCM 3.x format
11-02-06 15:05:24 Configuration read: /home/weber/.config/BKG/BNC.ini, 3 stream(s)
11-02-06 15:07:49 FFMJ1 PPP 15:08:02.0 13 4053458.795 +- 5.144 617729.872 +- 3.834 4869395.594 +- 7.167
11-02-06 15:07:49 FFMJ1 PPP 15:08:03.0 13 4053457.842 +- 3.644 617729.655 +- 2.711 4869394.988 +- 5.068
11-02-06 15:07:51 FFMJ1 PPP 15:08:04.0 13 4053458.025 +- 2.978 617729.469 +- 2.211 4869394.913 +- 4.131
11-02-06 15:07:51 FFMJ1 PPP 15:08:05.0 13 4053458.292 +- 2.580 617729.491 +- 1.912 4869394.699 +- 3.568
11-02-06 15:07:54 FFMJ1 PPP 15:08:06.0 13 4053457.989 +- 2.308 617729.667 +- 1.707 4869394.554 +- 3.180
    
```

Add Stream Delete Stream Start Stop Help ?=Shift+F1

Figure: Precise Point Positioning (PPP, tab 1) with BNC.

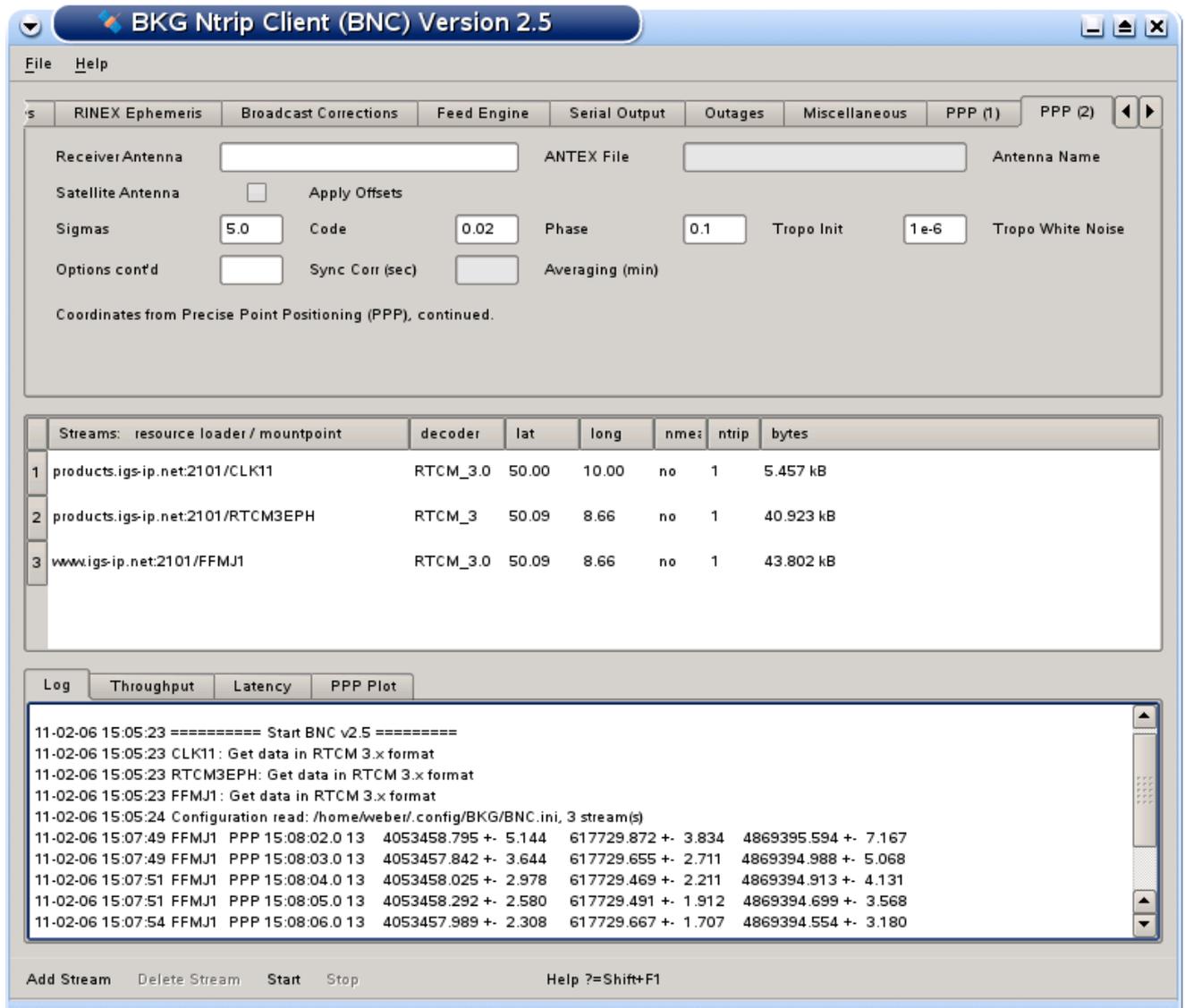


Figure: Precise Point Positioning (PPP, tab 2) with BNC.

PPP results are shown in the 'Log' tab on the bottom of BNC's main window. Depending on the processing options, the following values are shown about once per second (example):

```

10-09-08 09:14:06 FFMJ1 PPP 09:14:04.0 12 4053457.429 +- 2.323
617730.551 +- 1.630 4869395.266 +- 2.951
    
```

The 'PPP' string in that is followed by the selected mounpoint, a PPP time stamp in GPS Time, the number of processed satellites, and XYZ coordinates with their formal errors as derived from the implemented filter in [m]. The implemented algorithm includes an outlier and cycle slip detection. The maximum for accepted residuals is hard coded to 10 meters for code observations and 10 centimeters for phase observations.

More detailed PPP results are saved in BNC's logfile. Depending on the selected processing options you find

- code and phase residuals for GPS and GLONASS and Galileo in [m],
- receiver clock errors in [m],

- a-priori and correction values of tropospheric zenith delay in [m],
- time offset between GPS time and Galileo time in [m],
- L3 biases, also known as 'float ambiguities', given per satellite.

These parameters are saved together with their standard deviation. The following is an example extract from a log file when BNC was in 'Single Point Positioning' (SPP) mode:

10-12-06 18:10:50 Single Point Positioning of Epoch 18:10:48.0

```
-----
18:10:48.0 RES G04 L3 0.0165 P3 -0.1250
18:10:48.0 RES G11 L3 0.0150 P3 0.7904
18:10:48.0 RES G13 L3 0.0533 P3 0.4854
18:10:48.0 RES G17 L3 -0.0277 P3 1.2920
18:10:48.0 RES G20 L3 -0.0860 P3 -0.1186
18:10:48.0 RES G23 L3 0.0491 P3 -0.1052
18:10:48.0 RES G31 L3 0.0095 P3 -3.2929
18:10:48.0 RES G32 L3 0.0183 P3 -3.8800
18:10:48.0 RES R05 L3 -0.0077
18:10:48.0 RES R06 L3 0.0223
18:10:48.0 RES R15 L3 -0.0020
18:10:48.0 RES R16 L3 0.0156
18:10:48.0 RES R20 L3 -0.0247
18:10:48.0 RES R21 L3 0.0014
18:10:48.0 RES R22 L3 -0.0072
18:10:48.0 RES E52 L3 -0.0475 P3 -0.1628
18:10:48.0 RES G04 L3 0.0166 P3 -0.1250
18:10:48.0 RES G11 L3 0.0154 P3 0.7910
18:10:48.0 RES G13 L3 0.0535 P3 0.4855
18:10:48.0 RES G17 L3 -0.0272 P3 1.2925
18:10:48.0 RES G20 L3 -0.0861 P3 -0.1188
18:10:48.0 RES G23 L3 0.0489 P3 -0.1055
18:10:48.0 RES G31 L3 0.0094 P3 -3.2930
18:10:48.0 RES G32 L3 0.0183 P3 -3.8800
18:10:48.0 RES R05 L3 -0.0079
18:10:48.0 RES R06 L3 0.0223
18:10:48.0 RES R15 L3 -0.0020
18:10:48.0 RES R16 L3 0.0160
18:10:48.0 RES R20 L3 -0.0242
18:10:48.0 RES R21 L3 0.0016
18:10:48.0 RES R22 L3 -0.0072
18:10:48.0 RES E52 L3 -0.0474 P3 0.1385
```

```
clk      = 64394.754 +- 0.045
trp      = 2.185 +0.391 +- 0.001
offset   = -415.400 +- 0.137
amb G17  = 11.942 +- 0.045
amb G23  = 248.892 +- 0.044
amb G31  = 254.200 +- 0.045
amb G11  = -12.098 +- 0.044
amb G20  = -367.765 +- 0.044
amb G04  = 259.588 +- 0.044
amb E52  = 6.124 +- 0.130
amb G32  = 201.496 +- 0.045
amb G13  = -265.658 +- 0.044
amb R22  = -106.246 +- 0.044
amb R21  = -119.605 +- 0.045
amb R06  = 41.328 +- 0.044
amb R15  = 163.453 +- 0.044
amb R20  = -532.746 +- 0.045
amb R05  = -106.603 +- 0.044
amb R16  = -107.830 +- 0.044
```

Note that BNC's 'PPP Client' option can also be used in 'Offline Mode'. Apply the 'Offline Mode' command line options for that to read a file containing synchronized observations, orbit and clock correctors, and broadcast ephemeris. Such a file can be generated using BNC's 'Raw output file' option. The first five characters of the file name read in 'Offline Mode' must then be the same as the specified PPP 'Mountpoint': If you produce a 'Raw output file' named 'FFMJ1' then the PPP 'Mountpoint' needs to be also specified as 'FFMJ1' and the command line to execute BNC on a Windows system in 'Offline Mode' could look like:

```
bnc.exe --conf c:\temp\BNC.ppp --file c:\temp\FFMJ1 --format RTCM_3
```

Streams in a 'Raw output file' which shall later be used in an offline PPP calculation must all be encoded in the same format.

When using the PPP option, it is important to understand which effects are corrected by BNC.

- BNC does correct for Solid Earth Tides and Phase Windup.
- Satellite Antenna Phase Center Offsets are not corrected because applied orbit/clock correctors are referred to the satellite's antenna phase center.
- Satellite Antenna Phase Center Variations are neglected because this is a small effect usually less than 2 centimeters.
- Observations can be corrected for a Receiver Antenna Offset. Depending on whether or not this correction is applied, the estimated position is either that of the receiver's antenna phase center or that of the receiver's Antenna Reference Point.
- Receiver Antenna Phase Center Variations are not included in the model. The bias caused by this neglect depends on the receiver antenna type. For most antennas it is smaller than a few centimeters.
- Ocean and atmospheric loading is neglected. Atmospheric loading is pretty small. Ocean loading is usually also a small effect but may reach up to about 10 centimeters for coastal stations.
- Rotational deformation due to polar motion (Polar Tides) is not corrected because this is a small effect usually less than 2 centimeters.

3.11.1 Obs Mountpoint - optional

Specify an 'Observations Mountpoint' from the list of selected 'Streams' you are pulling if you want BNC to derive coordinates for the affected rover position through a Point Positioning solution.

Furthermore, specify the Point Positioning method you want to apply. Options are

- Precise Point Positioning (PPP, default), and
- Single Point Positioning (SPP).

3.11.1.1 XYZ - optional

Enter the reference coordinate components X,Y,Z of the receiver's position in meters if known. Default are empty option fields, meaning that the antenna's XYZ position is unknown.

Once XYZ coordinate components are defined, the 'PPP' line in BNC's logfile is extended by North, East and Up displacements to (example):

```
10-08-09 06:01:56 FFMJ1 PPP 06:02:09.0 11 4053457.628 +- 2.639
617729.438 +- 1.180 4869396.447 +- 1.921 NEU -0.908 -0.571
1.629
```

The parameters following the 'NEU' string provide North, East and Up components of the current coordinate displacement in meters.

3.11.2 Corr Mountpoint - optional

Specify an orbit/clock 'Corrections Mountpoint' from the list of selected 'Streams' you are pulling if you want BNC to correct your positioning solution accordingly.

3.11.3 Options

BNC allows to use different Point Positioning processing options depending on the capability of the involved receiver and the application in mind.

3.11.3.1 Use Phase Obs - optional

By default BNC applies a Point Positioning solution using an ionosphere free P3 linear combination of code observations. Tick 'Use phase obs' for an ionosphere free L3 linear combination of phase observations.

3.11.3.2 Estimate Tropo - optional

BNC estimates the tropospheric delay according to equation

$$T(z) = T_{\text{apr}}(z) + dT / \cos(z)$$

where T_{apr} is the a-priori tropospheric delay derived from Saastamoinen model.

By default BNC does not estimate troposphere parameters. Tick 'Estimate tropo' to estimate troposphere parameters together with the coordinates and save T_{apr} and dT in BNC's log file.

3.11.3.3 Use GLONASS - optional

By default BNC does not process GLONASS but only GPS observations when in Point Positioning mode. Tick 'Use GLONASS' to use GLONASS observations in addition to GPS (and Galileo if specified) for estimating coordinates in Point Positioning mode.

3.11.3.4 Use Galileo - optional

By default BNC does not process Galileo but only GPS observations when in Point Positioning mode. Tick 'Use Galileo' to use Galileo observations in addition to GPS (and GLONASS if specified) for estimating coordinates in Point Positioning mode.

3.11.4 Options cont'd

You may want to introduce specific sigmas for code and phase observations. You may also like to carry out your PPP solution in Quick-Start mode or output a time series of displacement components.

3.11.4.1 XYZ Init - mandatory

Enter a sigma in meters for the initial XYZ coordinate components. A value of 100.0 (default) may be an appropriate choice. However, this value may be significantly smaller (i.e. 0.01) when starting for example from a station with known XZY position in Quick-Start mode.

3.11.4.2 XYZ White Noise - mandatory

Enter a sigma in meters for the 'White Noise' of estimated XYZ coordinate components. A value of 100.0 (default) may be appropriate considering the potential movement of a rover.

3.11.4.3 Quick-Start - optional if XYZ is set

Enter the length of a startup period in seconds for which you want to fix the PPP solution to a known XYZ coordinate. Constraining coordinate components is done in BNC through setting the 'XYZ White Noise' temporarily to zero.

This so-called Quick-Start option allows the PPP solutions to rapidly converge after startup. It requires that the antenna remains unmoved on the known position throughout the defined period. A value of 120 (default) is likely to be an appropriate choice for 'Quick-Start'

You may need to create your own reference coordinate through running BNC for an hour in normal mode before applying the Quick-Start option. Don't forget to introduce a realistic sigma 'XYZ Ini' according to the coordinate's precision.

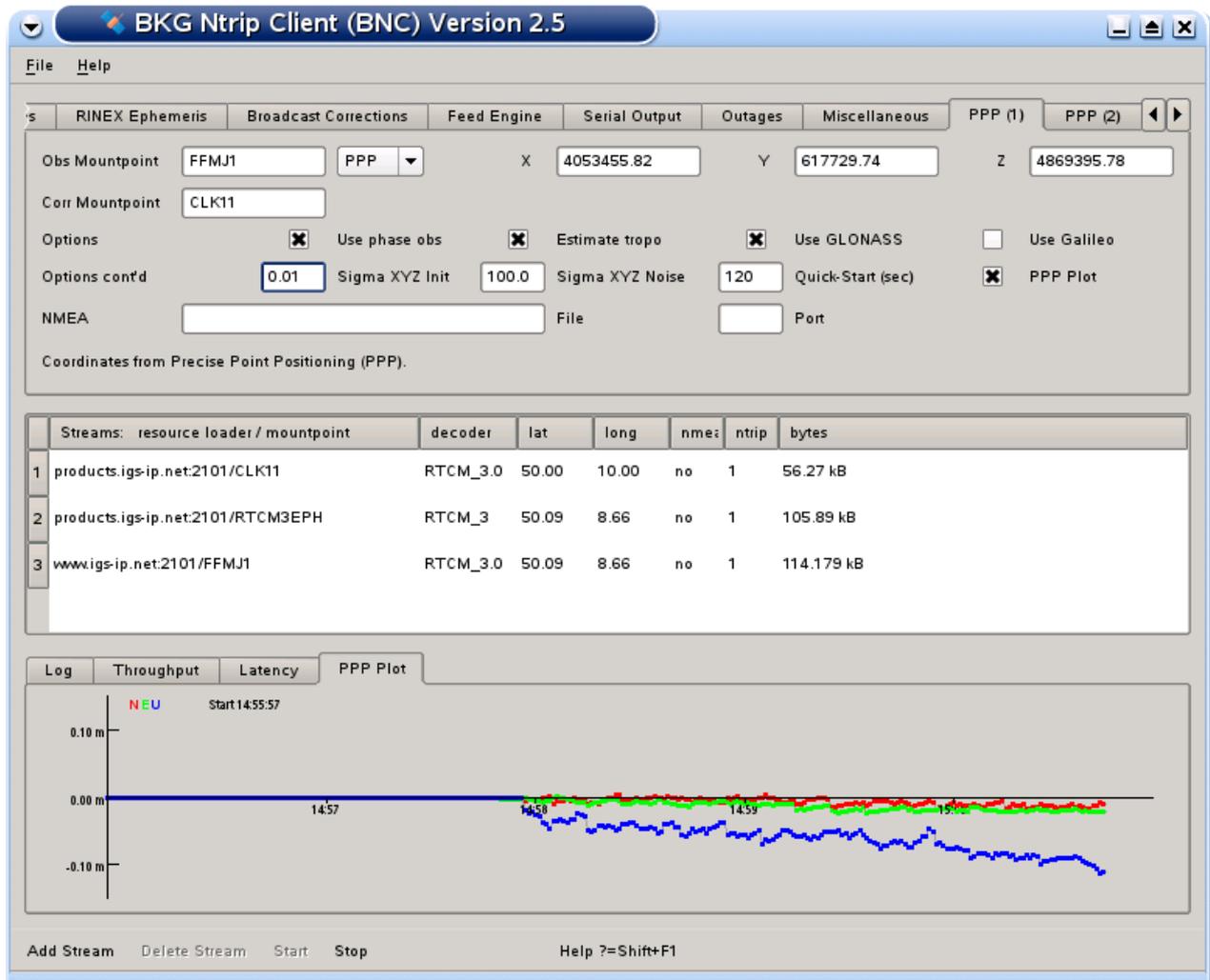


Figure: BNC in 'Quick-Start' mode

3.11.4.4 PPP Plot - optional

PPP time series of North (red), East (green) and Up (blue) coordinate components will be plotted in the 'PPP Plot' tab when this option is ticked. Values will be either referred to an XYZ reference coordinate (if specified) or referred to the first estimated XYZ coordinate. The sliding PPP time series window will cover the period of the latest 5 minutes.

Note that a PPP time series makes only sense for a stationary operated receiver.

3.11.5 NMEA - optional

BNC allows to output results from Precise Point Positioning in NMEA format.

3.11.5.1 NMEA File - optional

The NMEA sentences generated about once per second are pairs of

- GPGGA sentences which mainly carry the estimated latitude, longitude, and height values, plus
- GPRMC sentences which mainly carry date and time information.

Specify the full path to a file where Point Positioning results are saved as NMEA messages. The default value for 'NMEA file' is an empty option field, meaning that BNC will not save NMEA messages into a file.

Note that Tomoji Takasu has written a Windows program called RTKPlot for visualizing NMEA strings. It is available from <http://gpspp.sakura.ne.jp/rtklib/rtklib.htm> and compatible with the NMEA output of BNC's 'PPP Client' option.

3.11.5.2 NMEA Port - optional

Specify the IP port number of a local port where Point Positioning results become available as NMEA messages. The default value for 'NMEA Port' is an empty option field, meaning that BNC does not provide NMEA messages via IP port. Note that the NMEA file output and the NMEA IP port output are the same.

NASA's 'World Wind' software (see http://worldwindcentral.com/wiki/NASA_World_Wind_Download) can be used for real-time visualization of positions provided through BNC's NMEA IP output port. You need the 'GPS Tracker' plug-in available from http://worldwindcentral.com/wiki/GPS_Tracker for that. The 'World Wind' is not meant for showing centimeter level details.

3.11.6 Receiver Antenna - optional

BNC allows to correct observations for receiver antenna phase center offsets and variations.

3.11.6.1 ANTEX File - optional

IGS provides a file containing absolute phase center corrections for GNSS satellite and receiver antennas in ANTEX format. Entering the full path to such an ANTEX file is required for correcting observations for antenna phase center offsets and variations. It allows you to specify the name of your receiver's antenna (as contained in the ANTEX file) to apply such corrections.

Default is an empty option field meaning that you don't want to correct observations for antenna phase center offsets and variations.

3.11.6.2 Antenna Name - optional if 'ANTEX File' is set

Specify the receiver's antenna name as defined in your ANTEX file. Observations will be corrected for the antenna phase center's offset which may result in a reduction of a few centimeters at max. Corrections for phase center variations are not yet applied by BNC. The specified name must consist of 20 characters. Add trailing blanks if the antenna name has less than 20 characters. Examples:

```
'JPSREGANT_SD_E      ' (no radome)
'LEIAT504             NONE' (no radome)
'LEIAR25.R3          LEIT' (radome)
```

Default is an empty option field meaning that you don't want to correct observations for antenna phase center offsets.

3.11.7 Satellite Antenna - optional

BNC allows to correct observations for satellite antenna phase center offsets.

3.11.7.1 Apply Offsets - optional if 'ANTEX File' is set

Satellite orbit and clock corrections refer to the satellite's antenna phase centers and hence observations are not to be corrected for satellite antenna phase center offsets. Tick 'Ignore Offsets' to force BNC to not correct observations for satellite antenna phase center offsets. So far satellite antenna phase center variations remain unconsidered in BNC.

Default is to not correct observations for satellite antenna phase center offsets.

3.11.8 Parameter Sigmas

You may like to introduce specific sigmas for code and phase observations and for the estimation of troposphere parameters.

3.11.8.1 Code - mandatory if 'Use Phase Obs' is set

When 'Use phase obs' is set in BNC, the PPP solution will be carried out using both, code and phase observations. A sigma of 5.0 m for code observations and a sigma of 0.02 m for phase observations (defaults) is used to combine both types of observations. As the convergence characteristic of a PPP solution can be influenced by the ratio of the sigmas for code and phase, you may like to introduce you own sigmas for code and phase observations which differ from the default values.

- Introducing a smaller sigma (higher accuracy) for code observations or a larger sigma for phase observations leads to better results shortly after program start. However, it may take more time till you finally get the best possible solutions.
- Introducing a larger sigma (lower accuracy) for code observations or a smaller sigma for phase observations may lead to less accurate results shortly after program start and thus a prolonged period of convergence but could provide better positions in the long run.

Specify a sigma for code observations. Default is 5.0 m.

3.11.8.2 Phase - mandatory if 'Use Phase Obs' is set

Specify a sigma for phase observations. Default is 0.02 m.

3.11.8.3 Tropo Init - mandatory if 'Estimate tropo' is set

Enter a sigma in meters for the a-priory model based tropospheric delay estimation. A value of 0.1 (default) may be an appropriate choice.

3.11.8.4 Tropo White Noise - mandatory if 'Estimate tropo' is set

Enter a sigma in meters per second to describe the expected variation of the tropospheric effect. Supposing 1Hz observation data, a value of 1e-6 (default) would mean that the tropospheric effect may vary for $3600 * 1e-6 = 0.0036$ meters per hour.

3.11.9 Options cont'd - optional

You may like to introduce sigmas for code and phase observations and the estimation of troposphere parameters.

3.11.9.1 Sync Corr - optional

Zero value (or empty field) means that BNC processes each epoch of data immediately after its arrival using satellite clock corrections available at that time. Non-zero value 'Sync Corr' means that the epochs of data are buffered and the processing of each epoch is postponed till the satellite clock corrections not older than 'Sync Corr' are available. Specifying a value of half the update rate of the clock corrections as 'Sync Corr' (i.e. 5 sec) may be appropriate. Note that this causes an additional delay of the PPP solutions in the amount of the update rate.

Using observations in sync with the corrections can avoid a possible high frequency noise of PPP solutions. Such noise could result from processing observations regardless of how late after a clock correction they were received. Note that applying the 'Sync Corr' option significantly reduces the PPP computation effort for BNC.

Default is an empty option field, meaning that you want BNC to process observations immediately after their arrival through applying the latest received clock correction.

3.11.9.2 Averaging - optional if XYZ is set

Enter the length of a sliding time window in minutes. BNC will continuously output moving average values and their RMS as computed from those individual values obtained most recently throughout this period. RMS values presented for XYZ coordinates and tropospheric zenith path delays are bias reduced while RMS values for North/East/Up (NEU) displacements are not. Averaged values for XYZ coordinates and their RMS are marked with string "AVE-XYZ" in BNC's log file and 'Log' section while averaged values for NEU displacements and their RMS are marked with string "AVE-NEU" and averaged values for the tropospheric delays and their RMS are marked with string "AVE-TRP". Example:

```
10-09-08 09:13:05 FFMJ1 AVE-XYZ 09:13:04.0 4053455.948 +- 0.284
617730.422 +- 0.504 4869397.692 +- 0.089
10-09-08 09:13:05 FFMJ1 AVE-NEU 09:13:04.0 1.043 +- 0.179 0.640 +-
0.456 1.624 +- 0.331
10-09-08 09:13:05 FFMJ1 AVE-TRP 09:13:04.0 2.336 +- 0.002
```

Entering any positive value up to 1440 (24h mean value) is allowed. An empty option field (default) means that you don't want BNC to output moving average positions into the log file and the 'Log' section. Note that averaging positions makes only sense for a stationary receiver.

3.12. Streams

Each stream on an NTRIP broadcaster (and consequently on BNC) is defined using a unique source ID called mountpoint. An NTRIP client like BNC access the desired data stream by referring to its mountpoint. Information about streams and their mountpoints is available through the source-table maintained by the NTRIP broadcaster. Note that mountpoints could show up in BNC more than once when retrieving streams from several NTRIP broadcasters.

Streams selected for retrieval are listed under the 'Streams' canvas section on BNC's main window. The list provides the following information either extracted from source-table(s) produced by the NTRIP broadcasters or introduced by BNC's user:

'resource loader'	NTRIP broadcaster URL and port, or TCP/IP host and port, or UDP port, or Serial input port specification.
'mountpoint'	Mountpoint introduced by NTRIP broadcaster, or Mountpoint introduced by BNC's user.
'decoder'	Type of decoder used to handle the incoming stream content according to its format; editable.
'lat'	Approximate latitude of reference station, in degrees, north; editable if 'nmea' = 'yes'.
'long'	Approximate longitude of reference station, in degrees, east; editable if 'nmea' = 'yes'.
'nmea'	Indicates whether or not streaming needs to be initiated by BNC through sending NMEA-GGA message carrying position coordinates in 'lat' and 'long'.
'ntrip'	Selected NTRIP transport protocol version (1, 2, R, or U), or 'N' for TCP/IP streams without NTRIP, or 'UN' for UDP streams without NTRIP, or 'S' for serial input streams without NTRIP.
'bytes'	Number of bytes received.

3.12.1 Edit Streams

- BNC automatically allocates one of its internal decoders to a stream based on the stream's 'format' and 'format-details' as given in the source-table. However, there might be cases where you need to override the automatic selection due to incorrect source-table for example. BNC allows users to manually select the required decoder by editing the decoder string. Double click on the 'decoder' field, enter your preferred decoder and then hit Enter. The accepted decoder strings are 'RTCM_2.x', 'RTCM_3.x', and 'RTIGS'.
- In case you need to log the raw data as is, BNC allows users to by-pass its decoders and directly save the input in daily log files. To do this specify the decoder string as 'ZERO'. The generated file names are created from the characters of the streams mountpoints plus two-digit numbers each for year, month, and day. Example: Setting the 'decoder' string for mountpoint WTZZ0 to 'ZERO' and running BNC on March 29, 2007 would save the raw data in a file named WTZZ0_070329.
- BNC can also retrieve streams from virtual reference stations (VRS). To initiate these streams, an approximate rover position needs to be sent in NMEA format to the NTRIP broadcaster. In return, a user-specific data stream is generated, typically by a Network-RTK software. VRS streams are indicated by a 'yes' in the source-table as well as in the 'nmea' column on the 'Streams' canvas in BNC's main window. They are customized exactly to the latitude and longitude transmitted to the NTRIP broadcaster via NMEA-GGA messages.

If NMEA-GGA messages are not coming from a serial connected GNSS rover, BNC simulates them from the default latitude and longitude of the source-table as shown in the 'lat' and 'long' columns on the 'Streams' canvas. However, in most cases you would

probably want to change these defaults according to your requirement. Double-click on 'lat' and 'long' fields, enter the values you wish to send and then hit Enter. The format is in positive north latitude degrees (e.g. for northern hemisphere: 52.436, for southern hemisphere: -24.567) and eastern longitude degrees (example: 358.872 or -1.128). Only streams with a 'yes' in their 'nmea' column can be edited. The position must preferably be a point within the VRS service area of the network. RINEX files generated from these streams will contain an additional COMMENT line in the header beginning with 'NMEA' showing the 'lat' and 'long' used.

Note that when running BNC in a Local Area Network (LAN), NMEA strings may be blocked by a proxy server, firewall or virus scanner.

3.12.2 Delete Stream

To remove a stream from the 'Streams' canvas in the main window, highlight it by clicking on it and hit the 'Delete Stream' button. You can also remove multiple streams simultaneously by highlighting them using +Shift and +Ctrl.

3.12.3 Reconfigure Streams On-the-fly

The streams selection can be changed on-the-fly without interrupting uninvolved threads in the running BNC process.

Window mode: Hit 'Save & Reread Configuration' while BNC is in window mode and already processing data to let changes of your streams selection immediately become effective.

No window mode: When operating BNC online in 'no window' mode (command line option -nw), you force BNC to reread its 'mountPoints' configuration option from disk at pre-defined intervals. Select '1 min', '1 hour', or '1 day' as 'Reread configuration' option to reread the 'mountPoints' option every full minute, hour, or day. This lets a 'mountPoints' option edited in between in the configuration file become effective without terminating uninvolved threads. See annexed section 'Configuration Example' for a configuration file example and a list of other on-the-fly changeable options.

3.13. Logging

A tabs section on the bottom of the main window provides online control of BNC's activities. Tabs are available to show the records saved in a logfile, for a plot to control the bandwidth consumption, for a plot showing stream latencies, and for time series plots of PPP results.

3.13.1 Log

Records of BNC's activities are shown in the 'Log' tab. They can be saved into a file when a valid path is specified in the 'Logfile (full path)' field.

3.13.2 Throughput

The bandwidth consumption per stream is shown in the 'Throughput' tab in bits per second (bps) or kilo bits per second (kbps). The following figure shows the bandwidth consumption of incoming streams.

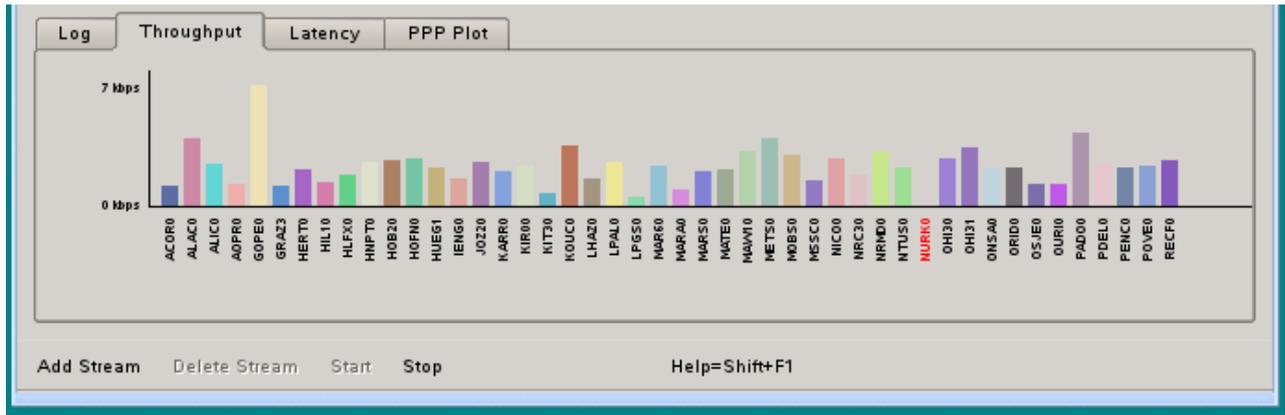


Figure: Bandwidth consumption of incoming streams.

3.13.3 Latency

The latency of observations in each incoming stream is shown in the 'Latency' tab in milliseconds or seconds. Streams not carrying observations (i.e. those providing only broadcast ephemeris messages) or having an outage are not considered here and shown in red color. Note that the calculation of correct latencies requires the clock of the host computer to be properly synchronized. The next figure shows the latency of incoming streams.

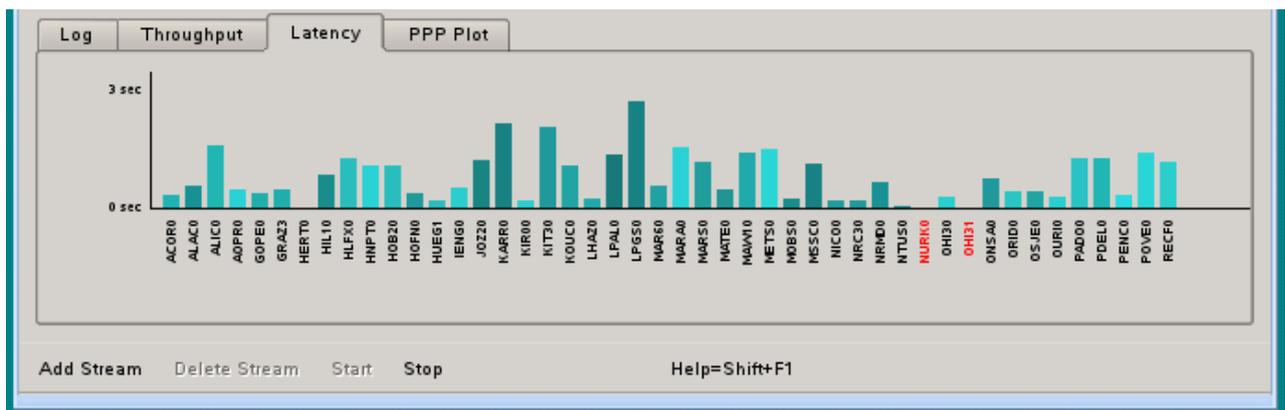


Figure: Latency of incoming streams.

3.13.4 PPP Plot

Precise Point Positioning time series of North (red), East (green) and Up (blue) coordinate components are shown in the 'PPP Plot' tab when a 'Origin' option is defined. Values are either referred to reference coordinates (if specified) or referred to the first estimated set of coordinate components. The time as given in format [hh:mm] refers to GPS Time. The sliding PPP time series window covers a period of 5 minutes. Note that it may take up to 30 seconds or more till the first PPP solutions becomes available. The following figure shows the screenshot of a PPP time series plot of North, East and Up coordinate components.

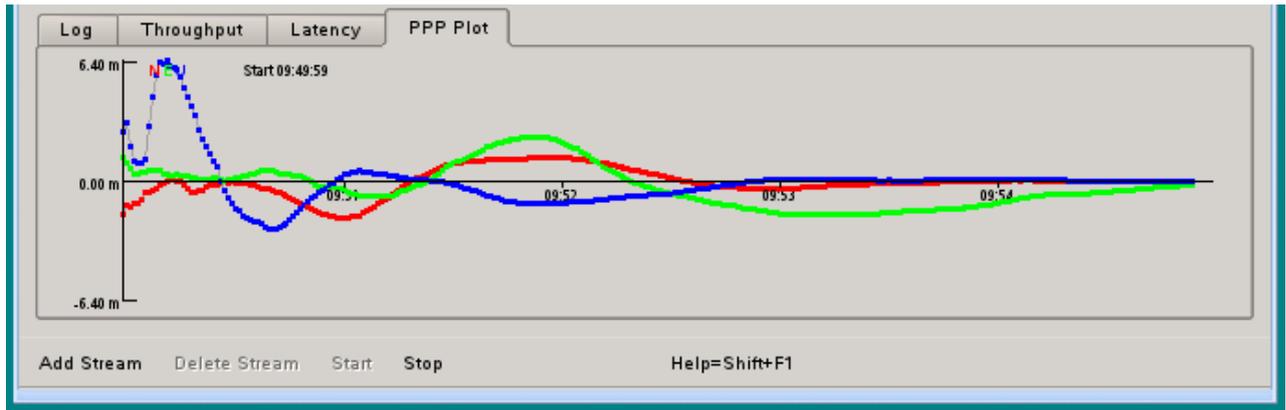


Figure: Time series plot of PPP session.

3.14. Bottom Menu Bar

The bottom menu bar allows to add or delete streams to BNC's configuration and to start or stop it. It also provides access to BNC's online help function. The 'Add Stream' button opens a window that allows user to select one of several input communication links, see figure below.

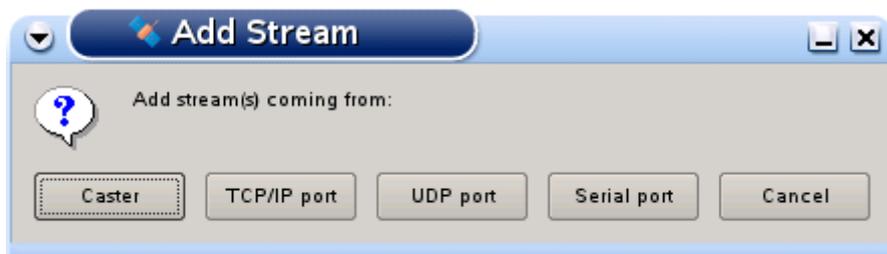


Figure: Steam input communication links.

3.14.1 Add Stream - Coming from Caster

Button 'Add Stream' > 'Coming from Caster' then opens a window that allows user to select data streams from an NTRIP broadcaster according to their mountpoints and show a distribution map of offered streams.

3.14.1.1 Caster Host and Port - mandatory

Enter the NTRIP broadcaster host IP and port number. Note that EUREF and IGS operate NTRIP broadcasters at <http://www.euref-ip.net/home> and <http://www.igs-ip.net/home> and <http://www.products.igs-ip.net/home>.

3.14.1.2 Casters Table - optional

It may be that your are not sure about your NTRIP broadcasters host and port number or you are interested in other broadcaster installations operated elsewhere. Hit 'Show' for a table of known broadcasters maintained at www.rtcn-ntrip.org/home. A window opens which allows to select a broadcaster for stream retrieval, see figure below.

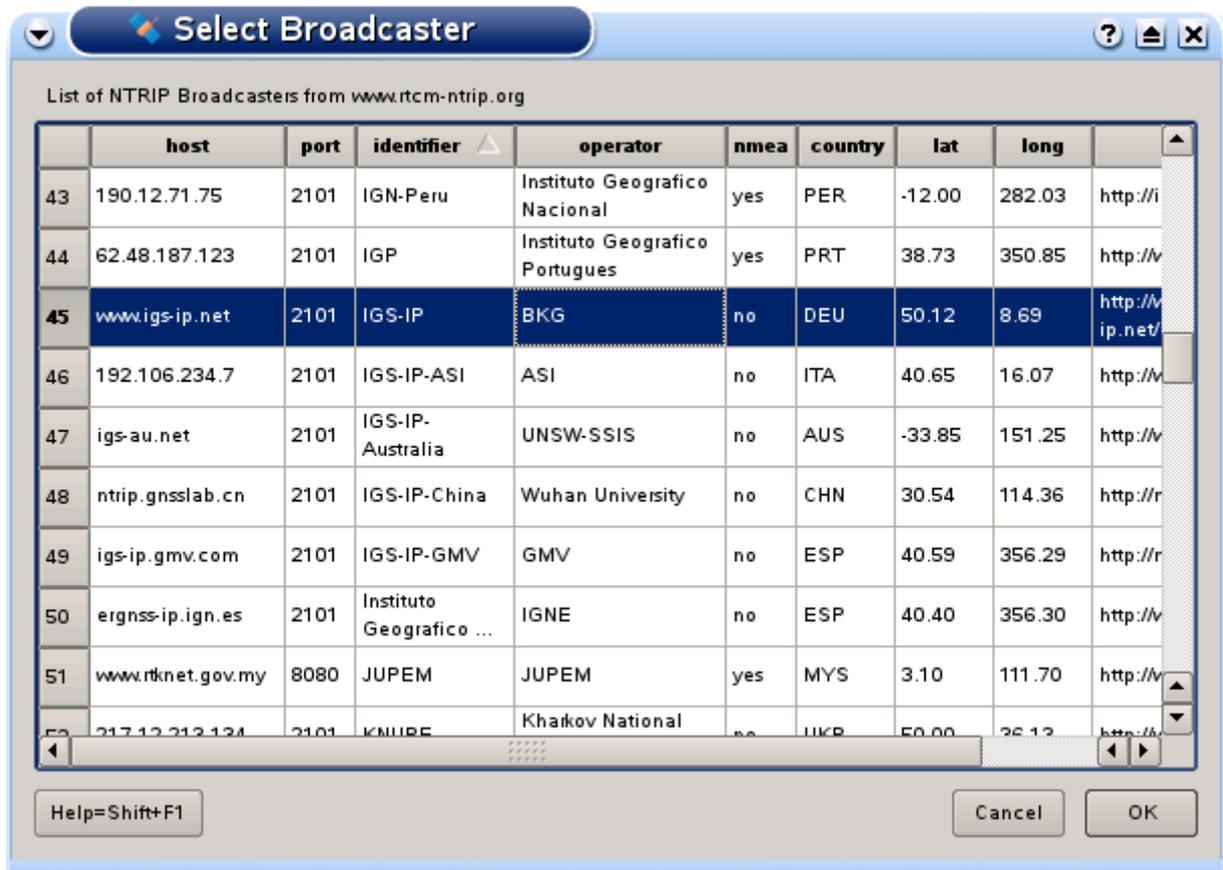


Figure: Casters table.

3.14.1.3 User and Password - mandatory for protected streams

Some streams on NTRIP broadcasters may be restricted. Enter a valid 'User' ID and 'Password' for access to protected streams. Accounts are usually provided per NTRIP broadcaster through a registration procedure. Register through <http://igs.bkg.bund.de/ntrip/registeruser> for access to protected streams on www.euref-ip.net or www.igs-ip.net or products.igs-ip.net.

3.14.1.4 Get Table

Use the 'Get Table' button to download the source-table from the NTRIP broadcaster. Pay attention to data fields 'format' and 'format-details'. Keep in mind that BNC can only decode and convert streams that come in RTCM Version 2.x, RTCM Version 3.x, or RTIGS format. For access to observations, ephemeris or ephemeris correctors, an RTCM Version 2.x streams must contain message types 18 and 19 or 20 and 21 while an RTCM Version 3.x streams must contain

- GPS or SBAS message types 1002 or 1004, or
- GLONASS message types 1010 or 1012, or
- proposed State Space Representation messages for GPS and GLONASS, types 1057-1068, or
- proposed 'Multiple Signal Messages' (MSM) for GPS, GLONASS, or Galileo, types 1071-1077, 1081-1087, or 1091-1097.

see data field 'format-details' for available message types and their repetition rates in brackets. Note that in order to produce RINEX Navigation files RTCM Version 3.x streams containing message types 1019 (GPS) and 1020 (GLONASS) and 1045 (Galileo) are required. Select your streams line by line, use +Shift and +Ctrl when necessary. The figure below provides an example source-table.

The contents of data field 'nmea' tells you whether a stream retrieval needs to be initiated by BNC through sending an NMEA-GGA message carrying approximate position coordinates (virtual reference station).

Hit 'OK' to return to the main window. If you wish you can click on 'Add Stream' and repeat the process again to retrieve streams from different casters.

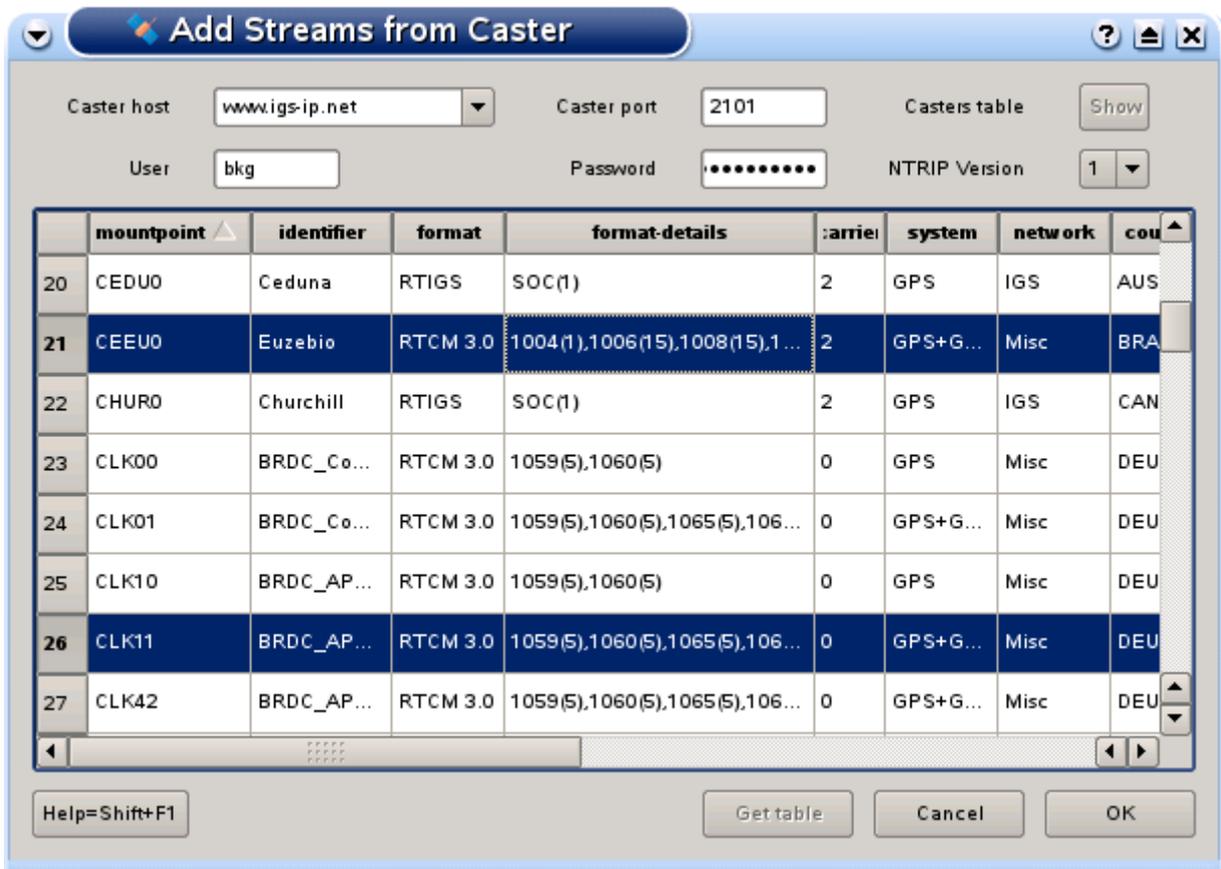


Figure: Broadcaster source-table.

3.14.1.5 NTRIP Version - mandatory

Some limitations and deficiencies of the NTRIP version 1 stream transport protocol are solved in NTRIP version 2. Improvements mainly concern a full HTTP compatibility in view of requirements coming from proxy servers. Version 2 is backwards compatible to Version 1. Options implemented in BNC are:

- 1: NTRIP version 1, TCP/IP.
- 2: NTRIP version 2 in TCP/IP mode.
- R: NTRIP version 2 in RTSP/RTP mode.
- U: NTRIP version 2 in UDP mode.

If NTRIP version 2 is supported by the broadcaster:

- Try using option '2' if your streams are otherwise blocked by a proxy server operated in front of BNC.
- Option 'R' or 'U' may be selected if latency is more important than completeness for your application. Note that the latency reduction is likely to be in the order of 0.5 sec or less. Note further that options 'R' (RTSP/RTP mode) and 'U' (UDP mode) are not accepted by proxy servers and a mobile Internet Service Provider may not support it.

Select option '1' if you are not sure whether the broadcaster supports NTRIP version 2.

3.14.1.6 Map - optional

Button 'Map' opens a window to show a distribution map of the casters's streams. You may like to zoom in or out using option 'Zoom +' or 'Zoom -'. You may also like to 'Clean' or 'Reset' a map or let it 'Fit' exactly to the current size of the window. Option 'Close' shuts the window.

3.14.2 Add Stream - Coming from TCP/IP Port

Button 'Add Stream' > 'Coming from TCP/IP Port' allows to retrieve streams via TCP directly from an IP address without using the NTRIP transport protocol. For that you:

- Enter the IP address of the stream providing host.
- Enter the IP port number of the stream providing host.
- Specify a mountpoint. Recommended is a 4-character station ID. Example: FFMJ
- Specify the stream format. Available options are 'RTCM_2', 'RTCM_3', 'RTIGS', and 'ZERO'.
- Enter the approximate latitude of the stream providing rover in degrees. Example: 45.32.
- Enter the approximate longitude of the stream providing rover in degrees. Example: -15.20.

Streams directly received from a TCP/IP port show up with an 'N' for 'No NTRIP' in the 'Streams' canvas section on BNC's main window . Latitude and longitude are to be entered just for informal reasons.

Note that this option works only if no proxy server is involved in the communication link.

3.14.3 Add Stream - Coming from UDP Port

Button 'Add Stream' > 'Coming from UDP Port' allows to pick up streams arriving directly at one of the local host's UDP ports without using the NTRIP transport protocol. For that you:

- Enter the local port number where the UDP stream arrives.
- Specify a mountpoint. Recommended is a 4-character station ID. Example: FFMJ
- Specify the stream format. Available options are 'RTCM_2', 'RTCM_3', 'RTIGS', and 'ZERO'.
- Enter the approximate latitude of the stream providing rover in degrees. Example: 45.32.

- Enter the approximate longitude of the stream providing rover in degrees. Example: -15.20.

Streams directly received at a UDP port show up with a 'UN' for 'UDP, No NTRIP' in the 'Streams' canvas section on BNC's main window . Latitude and longitude are to be entered just for informal reasons.

3.14.4 Add Stream - Coming from Serial Port

Button 'Add Stream' > 'Coming from Serial Port' allows to retrieve streams from a GNSS receiver via serial port without using the NTRIP transport protocol. For that you:

- Specify a mountpoint. Recommended is a 4-character station ID. Example: FFMJ
- Specify the stream format. Available options are 'RTCM_2', 'RTCM_3', 'RTIGS', and 'ZERO'.
- Enter the approximate latitude of the stream providing receiver in degrees. Example: 45.32.
- Enter the approximate longitude of the stream providing receiver in degrees. Example: -15.20.
- Enter the serial 'Port name' selected on your host for communication with the receiver. Valid port names are
 - Windows: COM1, COM2
 - Linux: /dev/ttyS0, /dev/ttyS1
 - FreeBSD: /dev/ttyd0, /dev/ttyd1
 - Digital Unix: /dev/tty01, /dev/tty02
 - HP-UX: /dev/tty1p0, /dev/tty2p0
 - SGI/IRIX: /dev/ttyf1, /dev/ttyf2
 - SunOS/Solaris: /dev/ttya, /dev/ttyb
- Select a 'Baud rate' for the serial input. Note that using a high baud rate is recommended.
- Select the number of 'Data bits' for the serial input. Note that often '8' data bits are used.
- Select the 'Parity' for the serial input. Note that parity is often set to 'NONE'.
- Select the number of 'Stop bits' for the serial input. Note that often '1' stop bit is used.
- Select a 'Flow control' for the serial link. Select 'OFF' if you don't know better.

When selecting the serial communication options listed above, make sure that you pick those configured to the serial connected GNSS receiver.

Streams received from a serial connected GNSS receiver show up with an 'S' (for Serial Port, no NTRIP) in the 'Streams' canvas section on BNC's main window . Latitude and longitude are to be entered just for informal reasons.

The following figure shows a BNC example setup for pulling a stream via serial port on a Linux operating system.

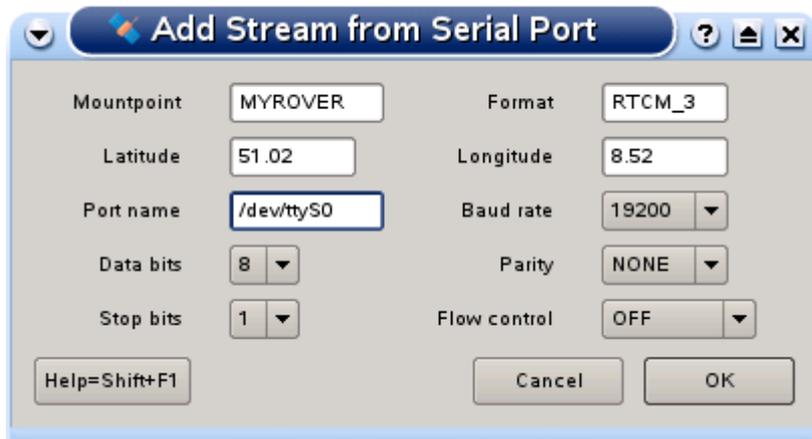


Figure: BNC setup for pulling a stream via serial port.

3.14.5 Start

Hit 'Start' to start retrieving, decoding, and converting GNSS data streams in real-time. Note that 'Start' generally forces BNC to begin with fresh RINEX which might overwrite existing files when necessary unless the option 'Append files' is ticked.

3.14.6 Stop

Hit the 'Stop' button in order to stop BNC.

3.15. Command Line Options

Command line options are available to run BNC in 'no window' mode or let it read data from a file in offline mode. BNC will then use processing options from the configuration file. Note that the self-explaining contents of the configuration file can easily be edited. It is possible to introduce a specific configuration file name instead of using the default name 'BNC.ini'.

3.15.1 No Window Mode - optional

Apart from its regular windows mode, BNC can be started on all systems as a background/batch job with command line option '-nw'. BNC will then run in 'no window' mode, using processing options from its configuration file on disk. Terminate BNC using Windows Task Manager when running it in 'no window' mode on Windows systems.

Example:

```
bnc.exe -nw
```

3.15.2 Offline Mode - optional

Although BNC is primarily a real-time online tool, it can be run in offline mode to read data from a previously saved file for post-processing purposes. Enter the following command line options for that:

- '--file <inputFileName>' to enter the full path to an input file containing data previously saved by BNC.

- '--format <format>' to enter one of the file format describing strings 'RTCM_2', 'RTCM_3' or 'RTIGS'.

Example:

```
./bnc --file FFMJ.dat --format RTCM_3
```

Note that when running BNC in offline mode, it will use options for file saving, interval, sampling, PPP etc. from its configuration file. Note further that only those data in the file will be processed offline which are encoded as specified with the --format option.

3.15.3 Configuration File - optional

The default configuration file name is 'BNC.ini'. You may change this name at startup time using the command line option '--conf <confFileName>'. This allows to run several BNC jobs in parallel on the same host using different sets of configuration options. confFileName stands either for the full path to a configuration file or just for a file name. If you introduce only a filename, the corresponding file will be saved in the current working directory from where BNC is started.

Example:

```
./bnc --conf MyConfig.ini
```

This leads to a BNC job using configuration file 'MyConfig.ini'. The configuration file will be saved in the current working directory.

On a Mac-OS X v10.6 (or higher) system the command line would be

```
open -a /Applications/bnc.app --args -conf /Users/tsyan/MyConfig.ini
```

if the program is in /Applications and the configuration file 'MyConfig.ini' in /Users/tsyan.

4. Limitations & Known Bugs

- In Qt-based desktop environments (like KDE) on Unix/Linux platforms it may happen that you experience a crash of BNC at startup even when running the program in the background using the '-nw' option. This is a known bug most likely resulting from an incompatibility of Qt libraries in the environment and in BNC. Entering the command 'unset SESSION_MANAGER' before running BNC may help as a work-around.
- Currently BNC only handles GPS, SBAS, GLONASS and Galileo data.
- BNC currently will only handle the following observation types:
 For GPS satellites, 'G': C1C L1C D1C S1C C1W L1W D1W S1W C2P L2P D2P S2P C2X L2X D2X S2X C5 L5 D5 S5
 For GLONASS satellites, 'R': C1C L1C D1C S1C C1P L1P D1P S1P C2P L2P D2P S2P C2C L2C D2C S2C
 For Geostationary signal payloads, 'S': C1C L1C D1C S1C C1W L1W D1W S1W
 For Galileo satellites, 'E': C1 L1 D1 S1 C5 L5 D5 S5
 Which observables and indicators are available on a particular stream will depend on the setup of source receiver and the data format used. RTCM Version 2.x streams do not carry signal-to-noise ratio 'S' values.
- Using RTCM Version 3.x to produce RINEX files, BNC will properly handle message types 1002, 1004, 1010, 1012, 1071-1077, 1081-1087, or 1091-1097. However, when handling message types 1001, 1003, 1009 and 1011 where the ambiguity field is not set, the output will be no valid RINEX. All values will be stored modulo 299792.458 (speed of light).
- Concerning the RTCM Version 3.x premature message types 1057-1068 (see RTCM document 091-2009-SC104-542 'Version 3 Proposed Messages - Set 10'), a final decision is not yet made. Note the what's implemented in BNC is just a temporary solution.
- Concerning the RTCM Version 3.x premature message types 1071-1077, 1081-1087, 1091-1097 (see RTCM document 086-2010-SC104-587 'New RTCM-3 Multiple Signal Message Proposal for GPS, GLONASS and Galileo'), a final decision is not yet made. Note that what is implemented in BNC is just a temporary solution.
- Using RTCM Version 2.x, BNC will only handle message types 18 and 19 or 20 and 21 together with position and the antenna offset information carried in types 3 and 22. Note that processing carrier phase corrections and pseudo-range corrections contained in message types 20 and 21 needs access to broadcast ephemeris. Hence, whenever dealing with message types 20 and 21, make sure that broadcast ephemeris become available for BNC through also retrieving at least one RTCM Version 3.x stream carrying message types 1019 (GPS ephemeris) and 1020 (GLONASS ephemeris).
- Streams coming in RTIGS format carry only GPS data.
- BNC's 'Get Table' function only shows the STR records of a source-table. You can use an Internet browser to download the full source-table contents of any NTRIP broadcaster by simply entering its URL in the form of <http://host:port>. Data field number 8 in the NET records may provide information about where to register for an NTRIP broadcaster account.
- EUREF as well as IGS adhere to an open data policy. Streams are made available through NTRIP broadcasters at www.euref-ip.net, www.igs-ip.net and products.igs-ip.net free of charge to anyone for any purpose. There is no indication up until now how many users will need to be supported simultaneously. The given situation may develop in such a way that it might become difficult to serve all registered users at the same times. In cases where limited resources on the NTRIP broadcaster side (software restrictions, bandwidth limitation etc.) dictates, first priority in stream provision will

be given to stream providers followed by re-broadcasting activities and real-time analysis centers while access to others might be temporarily denied.

- We experienced a limitation of the Standard Version of Microsoft Windows related to socket communication where sockets are not always handled properly. Since BNC makes intensive use of communication through sockets, we recommend to use the Server Version of Microsoft Windows when running BNC continuously for extended on a Windows platform.
- The source code provided by NRCan for decoding RTIGS streams is 32-bit dependent. Hence the BNC executable generated for 64-bit Linux systems would only run when compiled using the -m32 compiler option.
- Once BNC has been started, many of its configuration options cannot be changed as long as it is stopped. See chapter 'Reread Configuration' for on-the-fly configuration exceptions.
- On Mac systems you may experience a problem in passing a specific configuration file to the BNC application. On a Mac OS X 10.6 (or higher) system, you may then like to try the following command line to launch BNC:
open -a /Applications/bnc.app --args -conf /Users/tsyan/MyConfig.ini
(if the program is in /Applications and the configuration file 'MyConfig.ini' in /Users/tsyan)

5. Authors

The BKG Ntrip Client (BNC) Qt Graphic User Interface (GUI) has been developed for the Federal Agency for Cartography and Geodesy (BKG) by Leos Mervart, Czech Technical University Prague, Department of Geodesy. BNC includes the following GNU GPL software components:

- RTCM 2.x decoder, written by Oliver Montenbruck, German Space Operations Center, DLR, Oberpfaffenhofen
- RTCM 3.x decoder, written for BKG by Dirk Stoecker, Alberding GmbH, Schoenefeld
- RTIGS decoder, written by Ken MacLeod, Natural Resources, Canada.

Georg Weber
Federal Agency for Cartography and Geodesy (BKG)
Frankfurt, Germany
[euref-ip@bkg.bund.de] or [igs-ip@bkg.bund.de]

Acknowledgements

BNC's Help Contents has been proofread by Thomas Yan, University of New South Wales, Australia.

Scott Glazier, OmniSTAR Australia, included the decoding of broadcast ephemeris from RTIGS streams and has been helpful in finding BNC's bugs.

James Perlt, BKG, helped fixing bugs and redesigned BNC's main window.

Andre Hauschild, German Space Operations Center, DLR, revised the RTCMv2 decoder.

Zdenek Lukes, Czech Technical University Prague, Department of Geodesy, extended the RTCMv2 decoder to handle message types 3, 20, 21, and 22 and added loss of lock indicator.

Jan Dousa, Geodetic Observatory Pecny, Czech Republic, provided a tool for drawing stream distribution maps and also helped with fixing bugs.

Denis Laurichesse, Centre National d'Etudes Spatiales (CNES), suggested to synchronize observations and clock corrections to reduce high frequency noise in PPP solutions.

6. Annex

- 6.1. [Revision History](#)
- 6.2. [RTCM](#)
 - 6.2.1 NTRIP [Version 1](#)
 - 6.2.2 NTRIP [Version 2](#)
 - 6.2.3 RTCM [Version 2.x](#)
 - 6.2.4 RTCM [Version 3.x](#)
- 6.3. [RTIGS](#)
 - 6.3.1 [SOC](#)
- 6.4. [Configuration Example](#)
- 6.5. [Links](#)

6.1 Revision History

- | | | |
|----------|--------------|---|
| Dec 2006 | Version 1.0b | [Add] First Beta Binaries published based on Qt 4.2.3. |
| Jan 2007 | Version 1.1b | [Add] Observables C2, S1, and S2
[Add] Virtual reference station access
[Bug] RTCM2 decoder time tag fixed
[Mod] Small letters for public RINEX skeleton files
[Add] Online help through Shift+F1
[Bug] Output only through IP port
[Bug] Method 'reconnecting' now thread-save
[Add] ZERO decoder added
[Mod] Download public RINEX skeletons once per day |
| Apr 2007 | Version 1.2b | [Mod] Upgrade to Qt Version 4.2.3
[Mod] Replace 'system' call for RINEX script by 'QProcess'
[Add] HTTP Host directive for skeleton file download
[Add] Percent encoding for user IDs and passwords
[Bug] Exit execution of calling thread for RTCM3 streams
[Bug] Signal-slot mechanism for threads |
| May 2007 | Version 1.3 | [Add] Source code published. |
| Jul 2007 | Version 1.4 | [Bug] Skip messages from proxy server
[Bug] Call RINEX script through 'nohup'
[Add] Handle ephemeris from RTCM Version 3.x streams
[Add] Upgrade to Qt Version 4.3.2
[Add] Optional RINEX v3 output
[Add] SBAS support |
| Apr 2008 | Version 1.5 | [Bug] RINEX skeleton download following stream outage
[Add] Handle ephemeris from RTIGS streams
[Add] Monitor stream failure/recovery and latency
[Mod] Redesign of main window
[Bug] Freezing of About window on Mac systems
[Bug] Fixed problem with PRN 32 in RTCMv2 decoder
[Bug] Fix for Trimble 4000SSI receivers in RTCMv2 decoder
[Mod] Major revision of input buffer in RTCMv2 decoder |
| Dec 2008 | Version 1.6 | [Mod] Fill blanc columns in RINEXv3 with 0.000 |

- [Add] RTCMv3 decoder for clock and orbit corrections
- [Add] Check RTCMv3 streams for incoming message types
- [Add] Decode RTCMv2 message types 3, 20, 21, and 22
- [Add] Loss of lock and lock time indicator
- [Bug] Rounding error in RTCMv3 decoder concerning GLONASS height
- [Mod] Accept GLONASS in RTCMv3 when transmitted first
- [Add] Leap second 1 January 2009
- [Add] Offline mode, read data from file
- [Add] Output antenna descriptor, coordinates and excentricities from RTCMv3
- [Add] Reconfiguration on-the-fly
- [Mod] Binary output of synchronized observations
- [Add] Binary output of unsynchronized observations
- [Bug] Fixed problem with joined RTCMv3 blocks

- Dec 2008 Version 1.6.1 [Mod] HTTP GET when no proxy in front

- [Bug] RINEX navigation file format
- [Add] Upgrade to Qt Version 4.5.2
- [Add] Support of NTRIP v2
- [Add] Rover support via serial port
- [Add] Show broadcaster table from www.rtcn-ntrip.org
- [Add] Enable/disable tab widgets
- [Add] User defined configuration file name
- [Mod] Switch to configuration files in ini-Format
- [Add] Daily logfile rotation
- Nov 2009 Version 1.7 [Add] Read from TCP/IP port, by-pass NTRIP transport protocol
- [Add] Save NMEA messages coming from rover
- [Add] Auto start
- [Add] Drag and drop ini files
- [Add] Read from serial port, by-pass NTRIP transport protocol
- [Mod] Update of SSR messages following RTCM 091-2009-SC104-542
- [Add] Read from UPD port, by-pass NTRIP transport protocol
- [Mod] Output format of Broadcast Corrections
- [Add] Throughput plot
- [Add] Latency plot

- Nov 2009 Version 1.8 [Mod] On-the-fly reconfiguration of latency and throughput plots

- Feb 2010 Version 2.0 [Mod] Change sign of Broadcast Ephemeris correctors
- [Add] Real-time PPP option
- [Bug] SSR GLONASS message generation
- [Add] PPP in post-processing mode
- Jun 2010 Version 2.1 [Mod] Update of SSR messages following draft dated 2010-04-12
- [Mod] Generating error message when observation epoch is wrong

- Jul 2010 Version 2.2 [Bug] GLONASS ephemeris time

- Aug 2010 Version 2.3 [Mod] Internal format for saving raw streams
- [Bug] Outlier detection in GLONASS ambiguity resolution
- [Mod] Format of PPP logs in logfile

		[Bug] Complete acceleration terms for GLONASS ephemeris
		[Bug] Handling ephemeris IOD's in PPP mode
		[Add] Output of averaged positions when in PPP mode
		[Mod] Use always the latest received set of broadcast ephemeris
		[Add] QuickStart PPP option
Dec 2010	Version 2.4	[Mod] Improvement of data sharing efficiency among different threads
		[Mod] Design of PPP tab section
		[Add] Sigmas for observations and parameters
		[Add] Stream distribution map
		[Bug] GPS Ephemeris in RINEX v3 format
		[Add] PPP option for sync of clock observations and corrections
		[Add] Drafted RTCMv3 Galileo ephemeris messages 1045
		[Add] Drafted RTCMv3 Multiple Signal Messages
		[Add] Optional specification of sigmas for coordinates and troposphere in PPP
Feb 2011	Version 2.5	[Add] Include Galileo in SPP
		[Add] Include Galileo observations in output via IP port
		[Add] Include Galileo observations in output via RINEXv3 files
		[Mod] Interface format for feeding a real-time engine with observations
		[Add] Correct observations for antenna phase center offsets
		[Add] Combine orbit/clock correctors streams
		[Add] Specify corrections mountpoint in PPP tab

6.2. RTCM

The Radio Technical Commission for Maritime Services (RTCM) is an international non-profit scientific, professional and educational organization. Special Committees provide a forum in which governmental and non-governmental members work together to develop technical standards and consensus recommendations in regard to issues of particular concern. RTCM is engaged in the development of international standards for maritime radionavigation and radiocommunication systems. The output documents and reports prepared by RTCM Committees are published as RTCM Recommended Standards. Topics concerning Differential Global Navigation Satellite Systems (DGNSS) are handled by the Special Committee SC 104.

Personal copies of RTCM Recommended Standards can be ordered through <http://www.rtcn.org/orderinfo.php>.

6.2.1 NTRIP Version 1

'Networked Transport of RTCM via Internet Protocol' Version 1.0 (NTRIP) stands for an application-level protocol streaming Global Navigation Satellite System (GNSS) data over the Internet. NTRIP is a generic, stateless protocol based on the Hypertext Transfer Protocol HTTP/1.1. The HTTP objects are enhanced to GNSS data streams.

NTRIP Version 1.0 is an RTCM standard designed for disseminating differential correction data (e.g. in the RTCM-104 format) or other kinds of GNSS streaming data to stationary or mobile users over the Internet, allowing simultaneous PC, Laptop, PDA, or receiver

connections to a broadcasting host. NTRIP supports wireless Internet access through Mobile IP Networks like GSM, GPRS, EDGE, or UMTS.

NTRIP is implemented in three system software components: NTRIP clients, NTRIP servers and NTRIP broadcasters. The NTRIP broadcaster is the actual HTTP server program whereas NTRIP client and NTRIP server are acting as HTTP clients.

NTRIP is an open none-proprietary protocol. Major characteristics of NTRIP's dissemination technique are:

- Based on the popular HTTP streaming standard; comparatively easy to implement when having limited client and server platform resources available.
- Application not limited to one particular plain or coded stream content; ability to distribute any kind of GNSS data.
- Potential to support mass usage; disseminating hundreds of streams simultaneously for thousands of users possible when applying modified Internet Radio broadcasting software.
- Considering security needs; stream providers and users don't necessarily get into contact, streams often not blocked by firewalls or proxy servers protecting Local Area Networks.
- Enables streaming over mobile IP networks because of using TCP/IP.

The NTRIP broadcaster maintains a source-table containing information on available NTRIP streams, networks of NTRIP streams and NTRIP broadcasters. The source-table is sent to an NTRIP client on request. Source-table records are dedicated to one of the following: Data Streams (record type STR), Casters (record type CAS), or Networks of streams (record type NET).

Source-table records of type STR contain the following data fields: 'mountpoint', 'identifier', 'format', 'format-details', 'carrier', 'nav-system', 'network', 'country', 'latitude', 'longitude', 'nmea', 'solution', 'generator', 'compr-encryp', 'authentication', 'fee', 'bitrate', 'misc'.

Source-table records of type NET contain the following data fields: 'identifiey', 'operator', 'authentication', 'fee', 'web-net', 'web-str', 'web-reg', 'misc'.

Source-table records of type CAS contain the following data fields: 'host', 'port', 'identifier', 'operator', 'nmea', 'country', 'latitude', 'longitude', 'misc'.

6.2.1 NTRIP Version 2

The major changes of NTRIP version 2.0 compared to version 1.0 are:

- cleared and fixed design problems and HTTP protocol violations;
- replaced non standard directives;
- chunked transfer encoding;
- improvements in header records;
- source-table filtering; and
- RTSP communication.

NTRIP version 2 allows to either communicate in TCP/IP mode or in RTSP/RTP mode or in UDP mode whereas version 1 is limited to TCP/IP only.

6.2.3 RTCM Version 2.x

Transmitting GNSS carrier phase data can be done through RTCM Version 2.x messages. Please note that only RTCM Version 2.2 and 2.3 streams may include GLONASS data. Messages that may be of some interest here are:

- Type 1 message is the range correction message and is the primary message in code-phase differential positioning (DGPS). It is computed in the base receiver by computing the error in the range measurement for each tracked SV.
- Type 2 message is automatically generated when a new set of satellite ephemeris is downloaded to the base receiver. It is the computed difference between the old ephemeris and the new ephemeris. Type 2 messages are used when the base station is transmitting Type 1 messages.
- Type 3 and 22 messages are the base station position and the antenna offset. Type 3 and 22 are used in RTK processing to perform antenna reduction.
- Type 6 message is a null frame filler message that is provided for data links that require continuous transmission of data, even if there are no corrections to send. As many Type 6 messages are sent as required to fill in the gap between two correction messages (type 1). Message 6 is not sent in burst mode.
- Type 9 message serves the same purpose as Type 1, but does not require a complete satellite set. As a result, Type 9 messages require a more stable clock than a station transmitting Type 1 's, because the satellite corrections have different time references.
- Type 16 message is simply a text message entered by the user that is transmitted from the base station to the rover. It is used with code-phase differential.
- Type 18 and 20 messages are RTK uncorrected carrier phase data and carrier phase corrections.
- Type 19 and 21 messages are the uncorrected pseudo-range measurements and pseudo-range corrections used in RTK.
- Type 23 message provides the information on the antenna type used on the reference station.
- Type 24 message carries the coordinates of the installed antenna's ARP in the GNSS coordinate system coordinates.

6.2.4 RTCM Version 3.x

RTCM Version 3.x has been developed as a more efficient alternative to RTCM Version 2.x. Service providers and vendors have asked for a standard that would be more efficient, easy to use, and more easily adaptable to new situations. The main complaint was that the Version 2 parity scheme was wasteful of bandwidth. Another complaint was that the parity is not independent from word to word. Still another was that even with so many bits devoted to parity, the actual integrity of the message was not as high as it should be. Plus, 30-bit words are awkward to handle. The Version 3.x standard is intended to correct these weaknesses.

RTCM Version 3.x defines a number of message types. Messages that may be of interest here are:

- Type 1001, GPS L1 code and phase.
- Type 1002, GPS L1 code and phase and ambiguities and carrier to noise ratio.
- Type 1003, GPS L1 and L2 code and phase.
- Type 1004, GPS L1 and L2 code and phase and ambiguities and carrier to noise ratio.
- Type 1005, Station coordinates XYZ for antenna reference point.

- Type 1006, Station coordinates XYZ for antenna reference point and antenna height.
- Type 1007, Antenna descriptor and ID.
- Type 1008, Antenna serial number.
- Type 1009, GLONASS L1 code and phase.
- Type 1010, GLONASS L1 code and phase and ambiguities and carrier to noise ratio.
- Type 1011, GLONASS L1 and L2 code and phase.
- Type 1012, GLONASS L1 and L2 code and phase and ambiguities and carrier to noise ratio.
- Type 1013, Modified julian date, leap second, configured message types and interval.
- Type 1014 and 1017, Network RTK (MAK) messages (under development).
- Type 1019, GPS ephemeris.
- Type 1020, GLONASS ephemeris.
- Type 4088 and 4095, Proprietary messages (under development).

The following are proposed 'Multiple Signal Messages' (MSM) under discussion for standardization:

- Type 1045, Galileo ephemeris.
- Type 1071, Compact GPS pseudo-ranges
- Type 1072, Compact GPS carrier phases
- Type 1073, Compact GPS pseudo-ranges and carrier phases
- Type 1074, Full GPS pseudo-ranges and carrier phases plus signal strength
- Type 1075, Full GPS pseudo-ranges, carrier phases, Doppler and signal strength
- Type 1076, Full GPS pseudo-ranges and carrier phases plus signal strength (high resolution)
- Type 1077, Full GPS pseudo-ranges, carrier phases, Doppler and signal strength (high resolution)
- Type 1081, Compact GLONASS pseudo-ranges
- Type 1082, Compact GLONASS carrier phases
- Type 1083, Compact GLONASS pseudo-ranges and carrier phases
- Type 1084, Full GLONASS pseudo-ranges and carrier phases plus signal strength
- Type 1085, Full GLONASS pseudo-ranges, carrier phases, Doppler and signal strength
- Type 1086, Full GLONASS pseudo-ranges and carrier phases plus signal strength (high resolution)
- Type 1087, Full GLONASS pseudo-ranges, carrier phases, Doppler and signal strength (high resolution)
- Type 1091, Compact Galileo pseudo-ranges
- Type 1092, Compact Galileo carrier phases
- Type 1093, Compact Galileo pseudo-ranges and carrier phases
- Type 1094, Full Galileo pseudo-ranges and carrier phases plus signal strength
- Type 1095, Full Galileo pseudo-ranges, carrier phases, Doppler and signal strength
- Type 1096, Full Galileo pseudo-ranges and carrier phases plus signal strength (high resolution)
- Type 1097, Full Galileo pseudo-ranges, carrier phases, Doppler and signal strength (high resolution)

The following are proposed 'State Space Representation' (SSR) messages under discussion for standardization:

- Type 1057, GPS orbit corrections to Broadcast Ephemeris
- Type 1058, GPS clock corrections to Broadcast Ephemeris
- Type 1059, GPS code biases
- Type 1060, Combined orbit and clock corrections to GPS Broadcast Ephemeris
- Type 1061, GPS User Range Accuracy (URA)
- Type 1062, High-rate GPS clock corrections to Broadcast Ephemeris
- Type 1063, GLONASS orbit corrections to Broadcast Ephemeris
- Type 1064, GLONASS clock corrections to Broadcast Ephemeris
- Type 1065, GLONASS code biases
- Type 1066, Combined orbit and clock corrections to GLONASS Broadcast Ephemeris
- Type 1067, GLONASS User Range Accuracy (URA)
- Type 1068, High-rate GLONASS clock corrections to Broadcast Ephemeris

6.3. RTIGS

RTIGS stands for a data format and transport protocol for GPS observations. It was defined by the Real-Time IGS Working Group (RTIGS WG). Its definition is based on the SOC format. Every RTIGS record has one of the following numbers:

Station record number 100
Observation record (O_T) number 200
Ephemeris record (E_T) number 300
Meteorological record (M_T) number 400

Every station has one of the following unique numbers:

1-99 reserved for JPL
100-199 reserved for NRCAN
200-299 reserved for NGS
300-399 reserved for ESOC
400-499 reserved for GFZ
500-599 reserved for BKG
600-699 reserved for GEOSCIENCE AUS
700-799 others
etc

The number of bytes in each real time message includes the header as well as the data content, but NOT the pointer.

For example:

- A station message is output once per hour and is 20 bytes.
- An observation message is output once per second. The header is 12 bytes long and the SOC data is 21 bytes per PRN. So a typical RTIGSO_T message will be 390 bytes if 8 sats are being tracked.
- An ephemeris message is output when the ephemeris is decoded by the GPS receiver. The time in the ephemeris header is the collected time. Only one ephemeris can be bundled in a RTIGSE_T message.

A RTIGSE_T message contains one eph. The message consists of 12 header bytes and 72 ephemeris bytes, for a total of 84 bytes.

- The RTIGSM_T (met) message should be issued once every 15 minutes. A basic met message consists of a 12 byte header and 3 longs (temp, press and relative humidity) for a total of 24 bytes.

All records are related to a station configuration indicated by the Issue of Data Station (IODS). The IODS will enable the user to identify the equipment and software that was used to derive the observation data.

Each record header contains the GPS Time in seconds which flows continuously from 6 Jan-1980 onwards.

The data payload of each record consists of observations. The structures indicate a pointer to data but in fact the broadcast messages do not contain the pointer, only the data. Users will have to manage the data and the pointer is shown in order to illustrate where the data is located in the message and one possible data management option.

All record data are in network byte order (Big Endian), i.e. IA32 users have to swap bytes.

Visit <http://igsceb.jpl.nasa.gov/mail/igs-rtwg/2004/msg00001.html> for further details.

6.3.1 SOC

The SOC format has been designed in July 1999 by the Jet Propulsion Laboratory (JPL) and the California Institute of Technology (CalTech) to transport 1Hz GPS data with minimal bandwidth over the open Internet. SOC follows the 'little-endian' byte order meaning that the low-order byte of a number is stored in memory at the lowest address, and the high-order byte at the highest address. Because the transport layer is UDP, the format does not include sync bits, a checksum, or cyclic redundancy checksum (CRC). SOC allows to transport the GPS observable CA, P1, P2, L1, and L2, efficiently compressed down to 14 bytes with 1 mm range resolution and 0.02 mm phase resolution. SOC contains epochs for cycle slips, a stand-alone time-tag per epoch, a minimum representation of the receiver's clock solution, 3 SNR numbers, a unique site id, a modulo 12 hour sequence number and flags for receiver type and GPS health. SOC's simple structure comprises an 8 byte header, a 9 byte overhead for timetag, number of gps, etc., plus 21 data bytes per gps.

Visit http://gipsy.jpl.nasa.gov/igdg/papers/SOC_FORMAT.ppt for further details.

6.4. Configuration Example

The following table's left column is an example for the contents of a configuration file 'BNC.ini'. It enables the retrieval of stream ACOR0 from www.euref-ip.net for the generation of 15 min RINEX files. RINEX files are uploaded to an archive using script 'up2archive' :

Option	Affiliation
[General]	Settings: Group
adviseFail=15	Outages: Failure threshold
adviseReco=5	Outages: Recovery threshold

adviseScript=	Outages: Script (full path)
autoStart=0	General: Auto start
binSample=0	Feed Engine: Sampling
casterUrlList=http://user:pass@euref-ip:2101	Internal memory: Visited URLs
corrIntr=1 day	Broadcast Corrections: Interval
corrPath=	Broadcast Corrections: Directory
corrPort=	Broadcast Corrections: Port
corrTime=5	Broadcast Corrections: Wait for full epoch
ephIntr=15 min	RINEX Ephemeris: Interval
ephPath=	RINEX Ephemeris: Directory
ephV3=0	RINEX Ephemeris: Version 3
font=	Internal memory: Used font
logFile=/home/weber/bnc.log	General: Logfile (full path)
rawOutFile=	General: Raw output file (full path)
miscMount=	Miscellaneous: Mountpoint
mountPoints=//user:pass@www.euref-ip.net:2101	
/ACOR0 RTCM_2.3 43.36 351.60 no 1	Streams: broadcaster:port/mountpoint
ntripVersion=1	
obsRate=	Add Stream: NTRIP Version
onTheFlyInterval=1 day	Outages: Observation rate
outEphPort=	General: Reread configuration
outFile=	RINEX Ephemeris: Port
outPort=	Feed Engine: File (full path)
outUPort=	Feed Engine: Port
perfIntr=	Feed Engine: Port (unsynchronized)
proxyHost=	Miscellaneous: Log latency
proxyPort=	Proxy: Proxy host
rnxApend=2	Proxy: Proxy port
rnxIntr=15 min	General: Append files
rnXPath=/home/user/rinex	RINEX Observations: Interval
rnxSample=0	RINEX Observations: Directory
rnxScript=/home/user/rinex/up2archive	RINEX Observations: Sampling
rnxSkel=	RINEX Observations: Script (full path)
rnxV3=0	RINEX Observations: Skeleton extension
scanRTCM=0	RINEX Observation: Version 3
serialAutoNMEA=Auto	Miscellaneous: Scan RTCM
serialBaudRate=9600	Serial Output: NMEA
serialDataBits=8	Serial Output: Baud rate
serialFileNMEA=	Serial Output: Data bits
serialHeightNMEA=	Serial Output: NMEA file name
serialMountPoint=	Serial Output: Height
	Serial Output: Mountpoint

serialParity=NONE	Serial Output: Parity
serialPortName=	Serial Output: Port name
serialStopBits=1	Serial Output: Stop bits
serialFlowControl=	Serial Output: Flow control
startTab=0	Internal memory: Top tab index
statusTab=0	Internal memory: Bottom tab index
waitTime=5	Feed Engine: Wait for full epoch
pppMount=	PPP Client: Observations Mountpoint
pppCorrMount=	PPP Client: Corrections Mountpoint
pppSPP=PPP	PPP Client: PPP/SPP
pppSigmaCode=5.0	PPP Client: Sigma for Code observations
pppSigmaPhase=0.02	PPP Client: Sigma for Phase observations
pppQuickStart=200	PPP Client: Quick-Start period
pppSigmaCrd0=100.0	PPP Client: Sigma for initial XYZ coordinate
pppSigmaCrdP=100.0	PPP Client: White noise for XYZ
pppSigmaTrp0=0.1	PPP Client: Sigma for initial tropospheric delay
pppSigmaTrpP=1e-6	PPP Client: White noise for tropospheric delay
pppAverage=	PPP Client: Length of time window for moving average
pppUsePhase=0	PPP Client: Use phase data
pppEstTropo=0	PPP Client: Estimate troposphere
pppGLONASS=0	PPP Client: Use GLONASS
pppGalileo=0	PPP Client: Use Galileo
pppPlotCoordinates=0	PPP Client: Plot NEU time series
pppRefCrdX=	PPP Client: X coordinate of plot origin
pppRefCrdY=	PPP Client: Y coordinate of plot origin
pppRefCrdZ=	PPP Client: Z coordinate of plot origin
pppAntenna=	PPP Client: Antenna name
pppAntex=	PPP Client: Path to ANTEX file
pppApplySatAnt=	PPP Client: Apply sat antenna phase center Offset
pppSync=	PPP Client: Sync observations and corrections
nmeaFile=	PPP Client: NMEA outputfile
nmeaPort=	PPP Client: NMEA IP output port
combineStreams=	Combination: List of correctors streams
cmbOutHost=	Combination: Ntrip caster for stream upload
cmbOutPort=	Combination: Port of Ntrip caster
cmbMountpoint=	Combination: Mountpoint on Ntrip caster
cmbPassword=	Combination: Stream upload password
cmbOutFile=	Combination: Raw output file
cmbSP3File=	Combination: SP3 output file

Note that the following configuration options saved on disk can be changed/edited on-the-fly while BNC is already processing data:

- 'mountPoints' to change the selection of streams to be processed, see section 'Streams',
- 'waitTime' to change the 'Wait for full epoch' option, see section 'Feed Engine', and
- 'binSampl' to change the 'Sampling' option, see section 'Feed Engine'.

6.5 Links

NTRIP	http://igs.bkg.bund.de/ntrip/about
EUREF-IP NTRIP broadcaster	http://www.euref-ip.net/home
IGS-IP NTRIP broadcaster	http://www.igs-ip.net/home
IGS products NTRIP broadcaster	http://products.igs-ip.net/home
Distribution of IGS-IP streams	http://www.igs.oma.be/real_time/
Completeness and latency of IGS-IP data	http://www.igs.oma.be/highrate/
NTRIP broadcaster overview	http://www.rtcn-ntrip.org/home
NTRIP Open Source software code	http://software.rtcn-ntrip.org
EUREF-IP Project	http://www.epncb.oma.be/euref_IP
Real-time IGS Pilot Project	http://www.rtigs.net/pilot
Radio Technical Commission for Maritime Services	http://www.rtcn.org