BKG Ntrip Cient (BNC) Version 2.2

Manual

July 11, 2010

Federal Agency for Cartography and Geodesy, Frankfurt, Germany

BKG Ntrip Client (BNC) Version 2.2 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.

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

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 supports decoding the following GNSS data formats:

- 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 (GPS, SBAS) or 1004 (GPS), 1010 or 1012 (GLONASS), 1019 or 1020 (broadcast ephemeris), 1057-1068 (premature State Space Representation messages for GPS and GLONASS)
- 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 pupose 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.
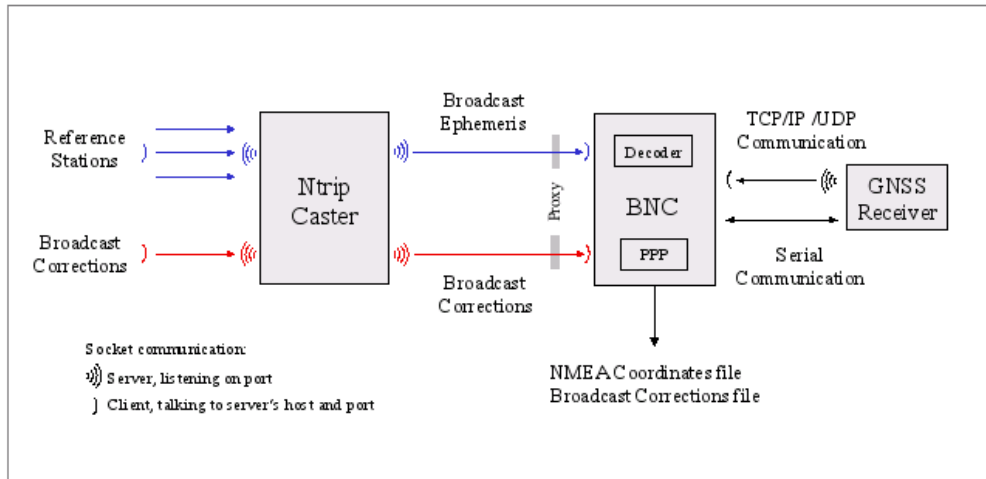
Figure: Flowchart, BNC connected to a GNSS receiver for Precise Point Positioning.
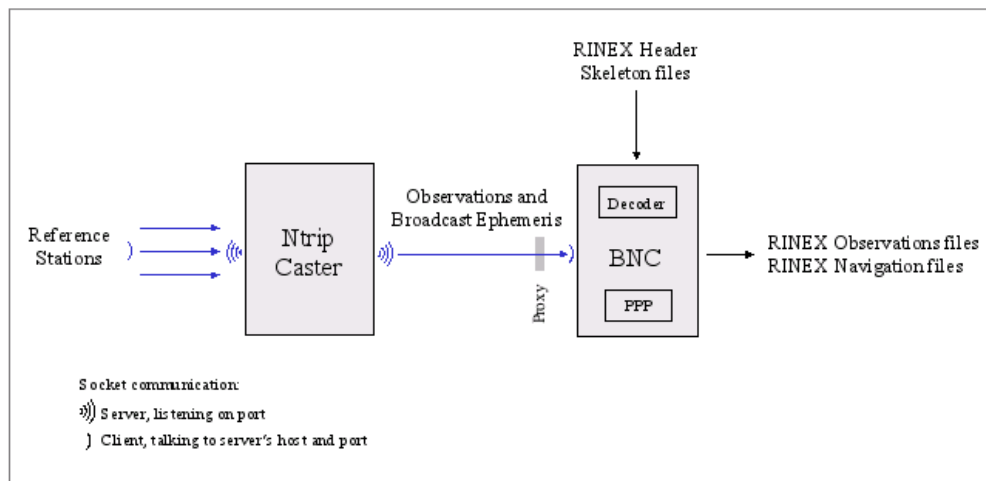

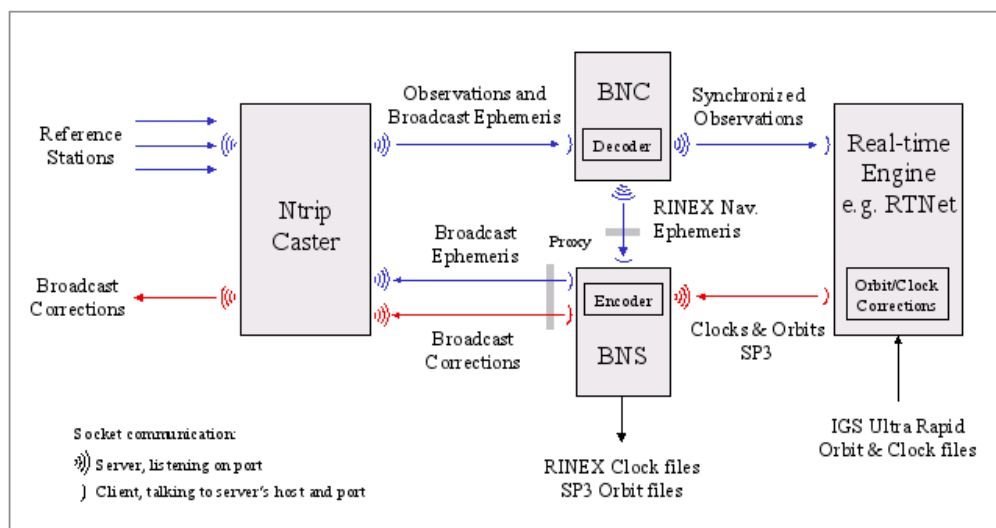Figure: Flowchart, BNC converting RTCM streams to RINEX batches.

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 availabe 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.
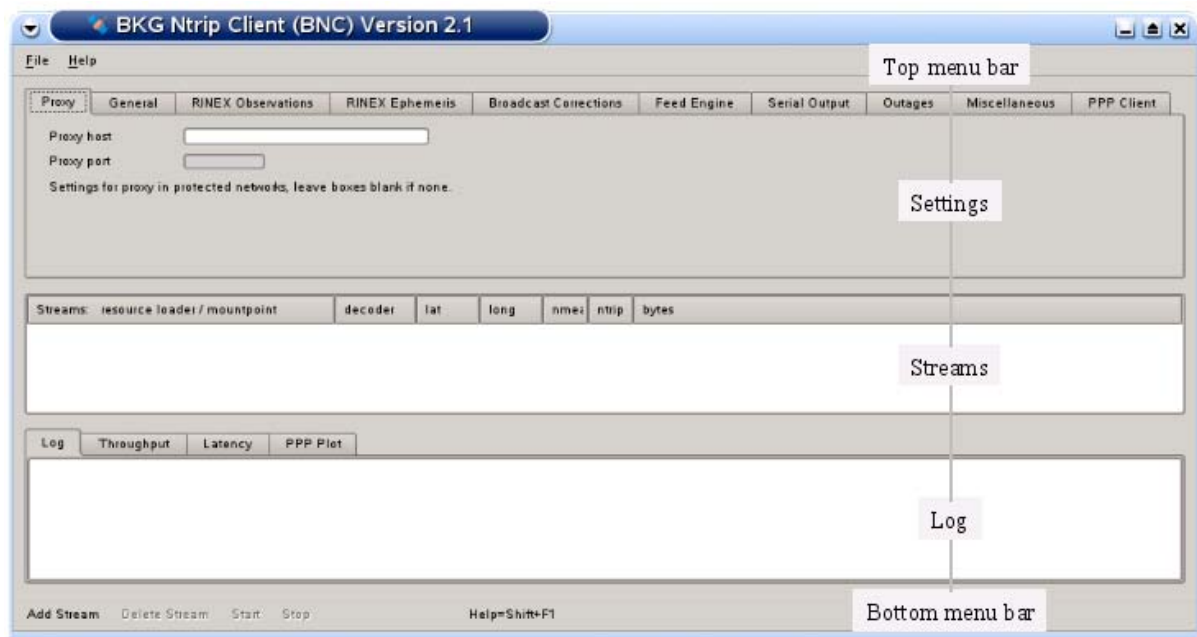

Figure: Sections on BNC's main window.

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. 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

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 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.

The default value for 'Raw output file (full path)' is an empty option field, meaning that BNC will not save raw data into a 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 Observation files generally contain C1, C2, P1, P2, L1, L2, S1, and S2 observations. 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 does not contain any GLONASS or SABAS data.

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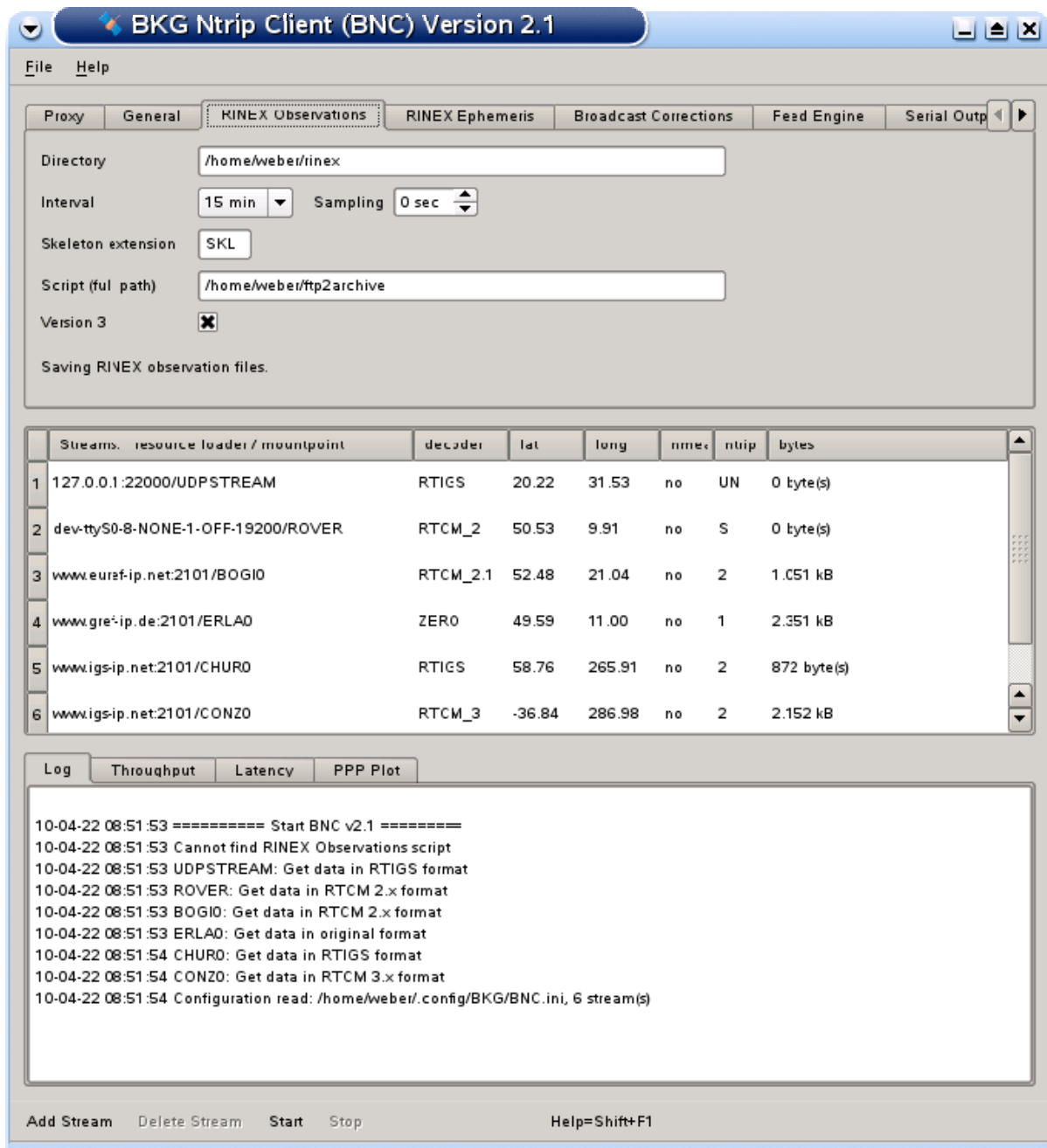
Figure: BNC translating incoming streams to 15 min RINEX Version 3 files.

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.skl
FRAN_KFURT.skl
FRAN_CE.skl
BRUS_0.skl
BRUS_1.skl

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

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 / DATE
  BNC 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 OBS
  BNC will include these lines in the final RINEX file header together with an additional
  - COMMENT
  line 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) and 1020 (GLONASS) 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 ephemeris saved together in one RINEX Version 3 Navigation file.

Note that streams dedicated to carry Broadacst 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_bnc_eph.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:

- 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 beimportant to understand that they are not corrected for the 2nd-order relativistic effect. The 2nd-order relativistic effect is a periodic time correction defined as $-2 (R * V) / c^2$ where R * V is the scalar product of the satellite position and velocity and c is the speed of light.

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_bnc_eph.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. The output can be produced in a binary format through an IP port and/or a plain ASCII format to save the observations in a local file. It comprises the following observations where available:

StatID, SVPRN, GPSWeek, GPSWeeks, C1, C2, P1, P2, L1, L2, slip_cnt_L1, slip_cnt_L2, lock_timei_L1, lock_timei_L2, S1, S2, SNR1, SNR2

Note that slip_cnt stands for the cumulative loss of continuity indicator, lock_timei for the lock time indicator, and SNR for the signal-to-noise ratio 'S' mapped to integer numbers 1 to 9. In case an observation is not available, its value is set to zero '0.000'. Loss of continuity indicator and lock time indicator are set to negative values if undefined.

The binary output is a continuous stream in the following order:

```
begEpoch
t_obsInternal
t_obsInternal
...
t_obsInternal
endEpoch
begEpoch
t_obsInternal
...
```

The corresponding structures are defined as follow:

```
  const char begEpoch[] = "BEGEPOCH";
  const char endEpoch[] = "ENDEPOCH";
...
...
class t_obsInternal {
 public:
  int    flags;
  char   StatID[20+1];  // Station ID
  char   satSys;        // Satellite System ('G' or 'R')
  int    satNum;        // Satellite Number (PRN for GPS NAVSTAR)
  int    slot;          // Slot Number (for Glonass)
  int    GPSWeek;       // Week of GPS-Time
  double GPSWeeks;      // Second of Week (GPS-Time)
  double C1;            // CA-code pseudorange (meters)
  double C2;            // CA-code pseudorange (meters)
  double P1;            // P1-code pseudorange (meters)
  double P2;            // P2-code pseudorange (meters)
  double L1;            // L1 carrier phase (cycles)
  double L2;            // L2 carrier phase (cycles)
```

```
  int     slip_cnt_L1;   // L1 cumulative loss of continuity indicator
(negative value = undefined)
  int     slip_cnt_L2;   // L2 cumulative loss of continuity indicator
(negative value = undefined)
  int     lock_timei_L1; // L1 last lock time indicator
(negative value = undefined)
  int     lock_timei_L2; // L2 last lock time indicator
(negative value = undefined)
  double S1;             // L1 signal-to noise ratio
  double S2;             // L2 signal-to noise ratio
  int    SNR1;           // L1 signal-to noise ratio (mapped to integer)
  int    SNR2;           // L2 signal-to noise ratio (mapped to integer)
};
```

The source code for BNC comes with an example program called 'test_bnc_qt.cpp' that allows you to read BNC's (synchronized or unsynchronized) binary observation output from the IP port and print the observations in a plain ASCII format 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 if streams is pulled from different NTRIP broadcasters to feed a GNSS engine via IP port output.
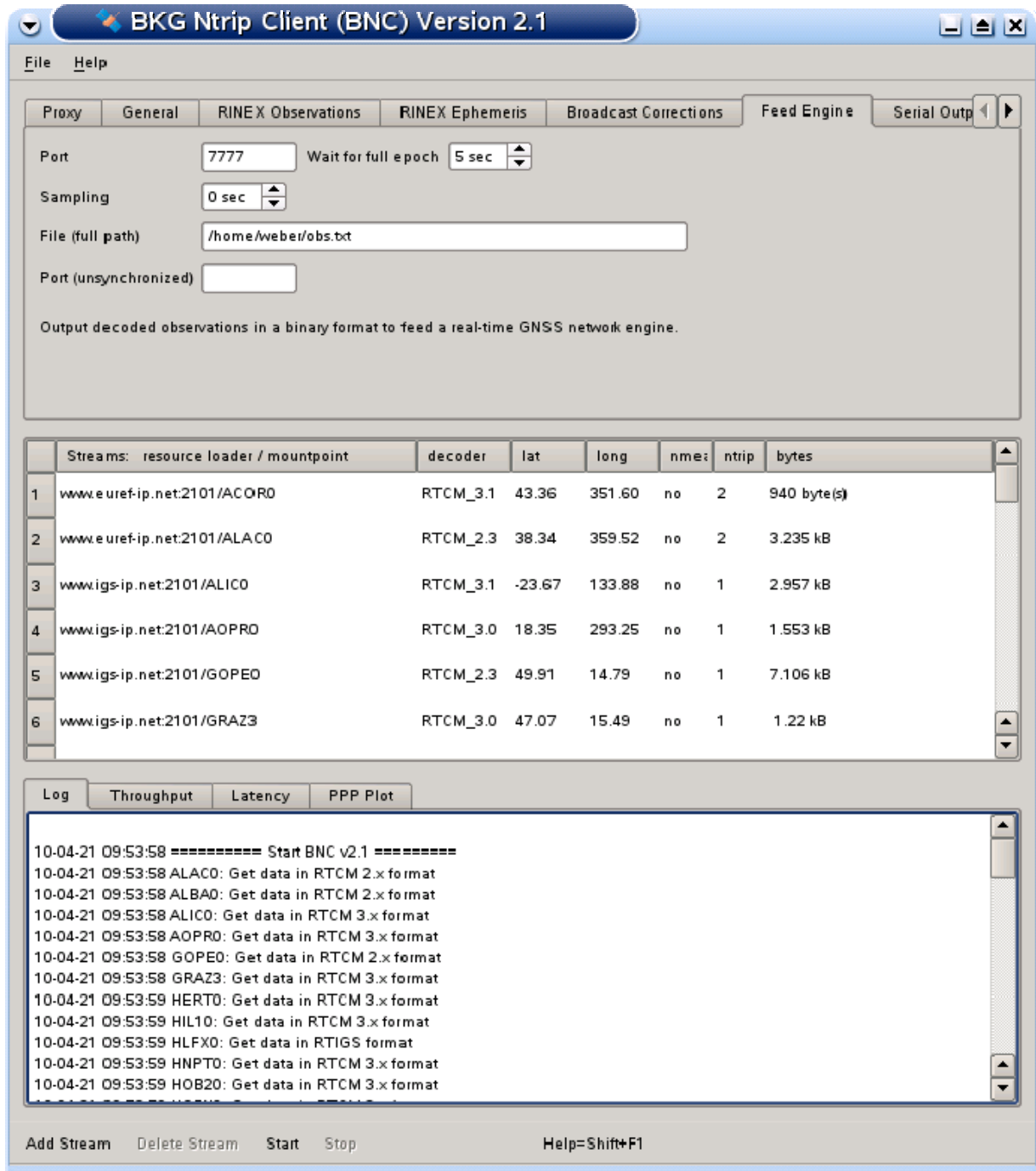
Figure: Synchronized BNC output via IP port to feed a GNSS real-time engine.

3.7.1 Port - optional

BNC can produce synchronized observations in binary 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 binary 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.
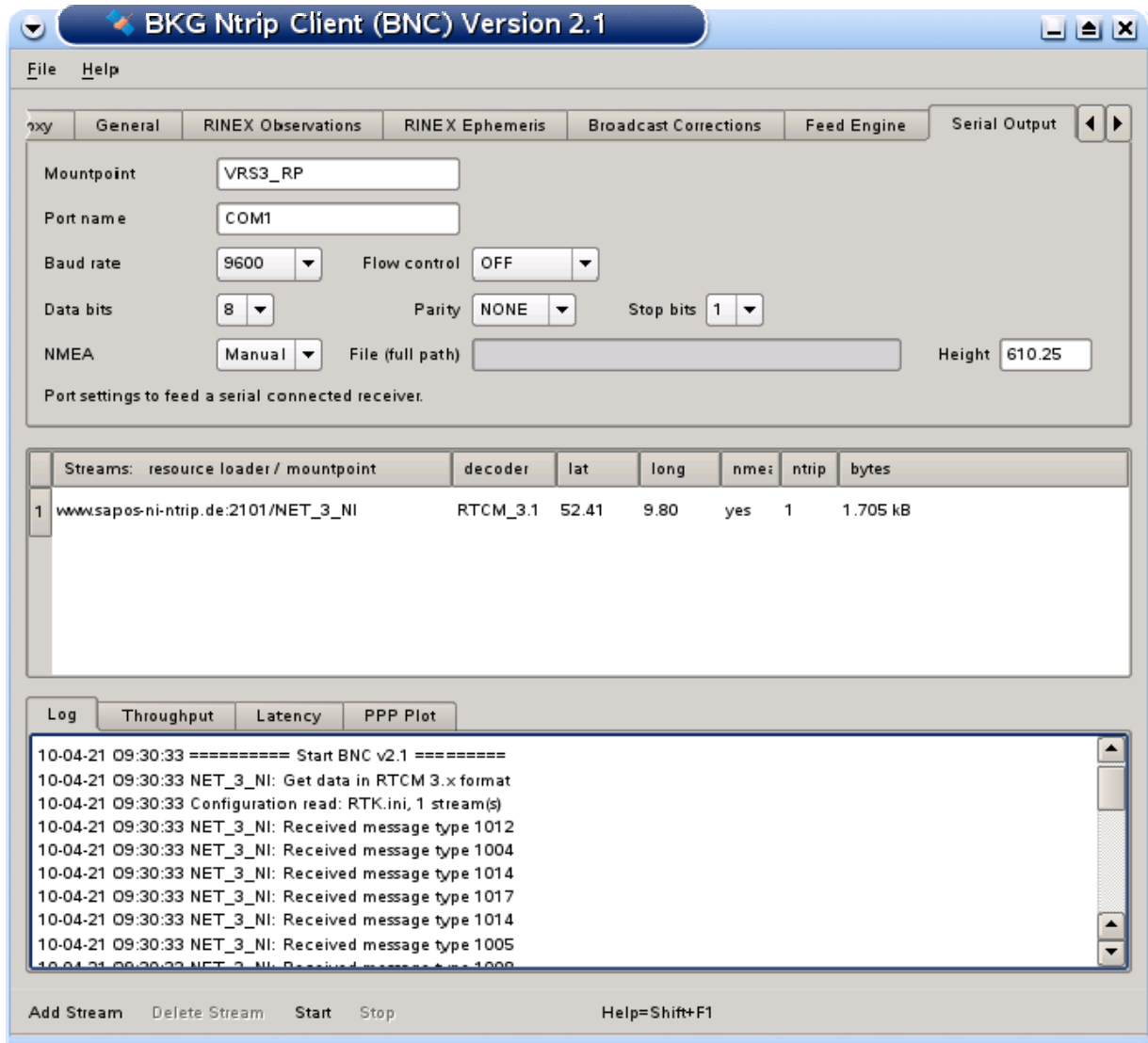
Figure: BNC pulling a VRS stream to feed a serial connected rover.

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/tty1p0, /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 priory 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 innundate 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 innundate 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 avoids 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.
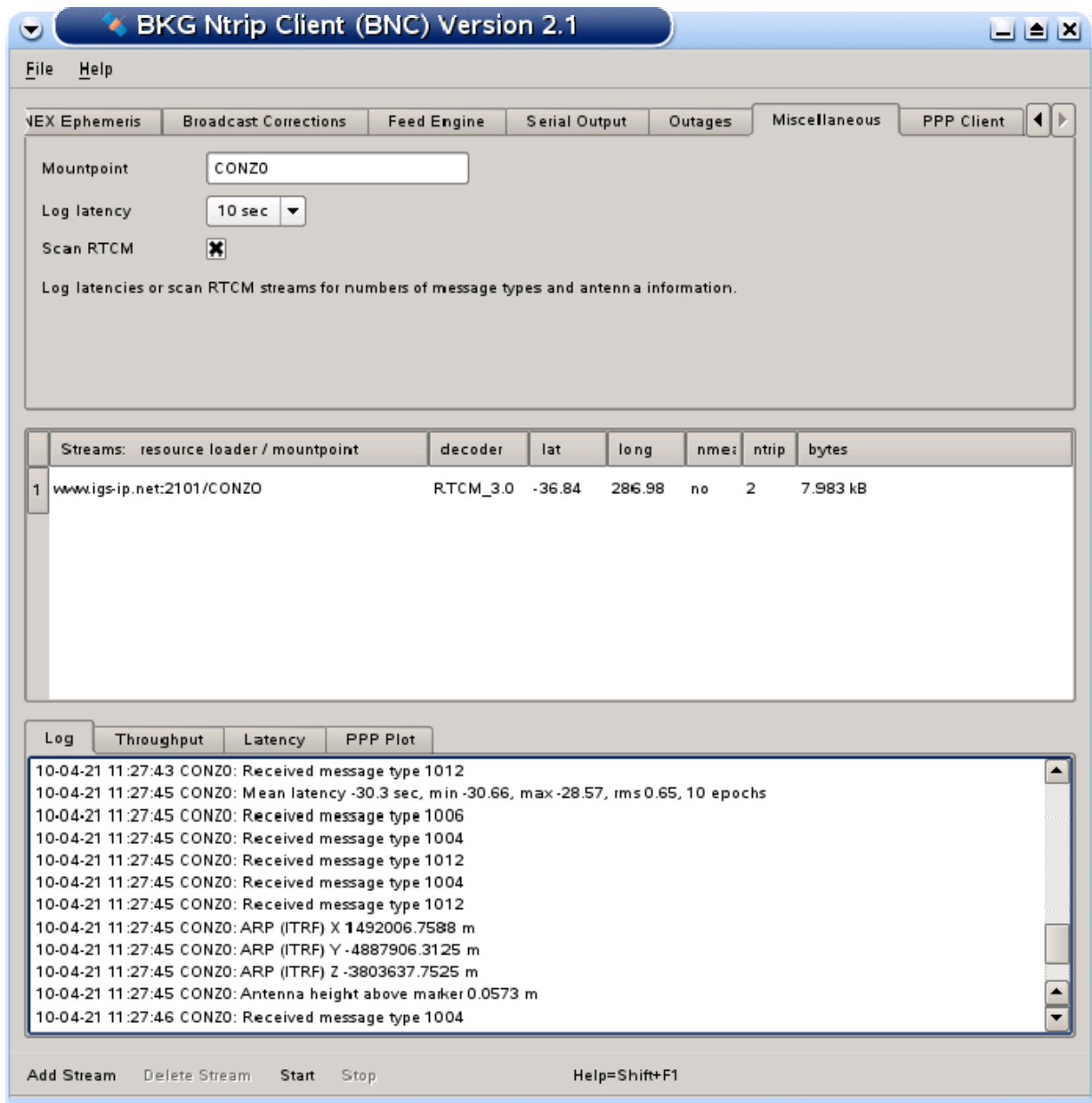
Figure: RTCM message numbers and latencies.

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:

```
  UTC time provided by BNC's host
- GPS time of currently processed epoch
+ Leap seconds between UTC and GPS time
  --------------
= Latency
```

Statistics: BNC counts the number of GPS seconds covered by at least one observation. It also estimates an observation rate (independent from the a priory 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 www.igs-ip.net:2101/CLK11 is an example.
- a stream carrying Broadcast Ephemeris available as RTCM Version 3 message types 1019 and 1020. This is a must when the stream coming from the receiver does not contain Broadcast Ephemeris or provides them only at low repetition rate. Streams providing such messages are listed on http://igs.bkg.bund.de/ntrip/ephemeris. Stream www.igs-ip.net:2101/RTCM3EPH is an example.

The following figure provides the screenshot of an example PPP session with BNC.
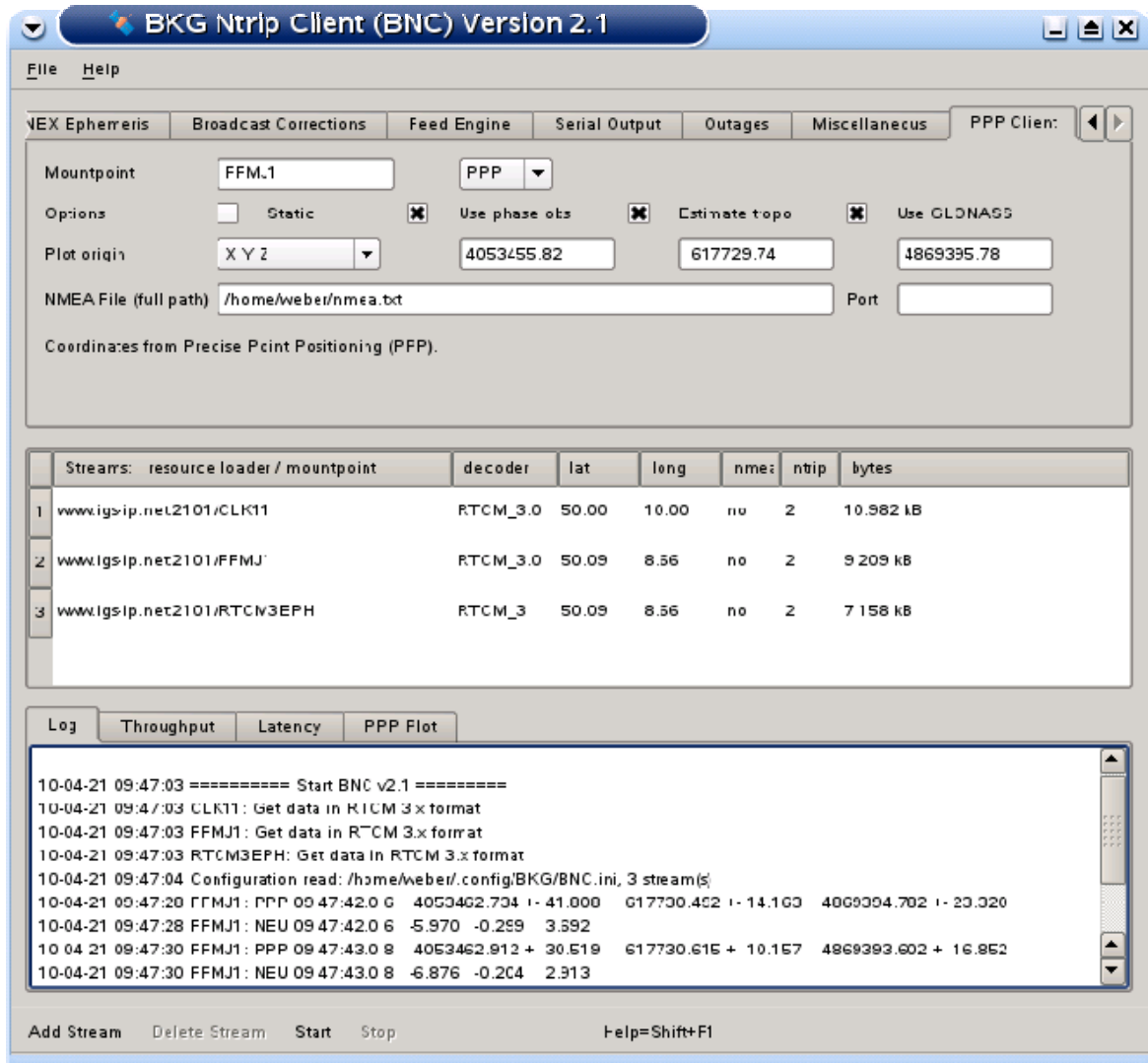
Figure: Precise Point Positioning (PPP) with BNC.

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):

```
09-12-15 22:30:39  PPP CAS10 22:30:44.0 9  -901775.887 +- 0.058
2409382.427 +- 0.046  -5816746.721 +- 0.087
```

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 in [m],
- receiver clock errors in [m],
- a-priori and correction values of tropospheric zenith delay in [m],
- L3 biases, also known as 'floated ambiguities', given per satellite.

31

These parameters are saved together with their standard deviation. Example extract from a log file:

```
10-01-07 15:18:02
residuals code  0.778 -0.608 -1.269 1.697
residuals phase 0.002 0.001 -0.002 -0.002
residuals glo   -0.000 0.000 0.001

   clk = 112578.741 +-  9.345
   trp =    2.339 +0.000 +-  0.010
   amb G11 = -1.762 +-  3.590
   amb G19 = -18.195 +-  5.386
   amb G23 =  1.316 +-  4.843
   amb G32 =  9.299 +-  3.920
   amb R01 = -146.297 +-  4.568
   amb R02 = -146.738 +- 15.037
   amb R17 = -156.881 +- 13.671
```

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 corretors, 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 'Mounpoint': 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 --date 2010-03-27 --time 06:58:00 --format RTCM_3

3.11.1 Mountpoint - optional

Specify a mountpoint 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).

Note that a plain SPP solution does not require to pull a stream carrying Broadcast Ephemeris Corrections. It only needs Broadcast Ephemeris.

3.11.2 Options

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

3.11.2.1 Static - optional

By default BNC considers the rover as mobile. It means that the rover coordinates are estimated as stochastic parameters with white noise 100 meters per epoch. Tick 'Static' in a static observation situation to adapt appropriate filter characteristics for that.

3.11.2.2 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.

3.11.2.3 Estimate Tropo - optional

BNC estimates the tropospheric delay according to equation

```
T(z) = T_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.

3.11.2.4 Use GLONASS - optional

By default BNC does not process GLONASS observations when in Point Positioning mode. Tick 'Use GLONASS' to use GPS and GLONASS observations for estimating coordinates in Point Positioning mode.

3.11.3 Plot Origin - optional

Select an origin for North/East/Up time series plots of derived coordinates in the 'PPP Plot' tab. Note that this makes only sense for a stationary receiver. Available options are

- 'No plot', meaning that BNC not plot a time series of estimated position.
- 'Start position', meaning that BNC will refer time series plots to the first estimated position.
- 'X Y Z', meaning the known reference coordinate components for the receiver's position.

Default is the selection of an empty option field meaning that no PPP time series will be plotted in the "PPP Plot' tab..

If option 'X Y Z' is selected, the following line (example) is recorded in BNC's logfile

```
10-03-12 09:25:25 FFMJ1: NEU 09:25:24.0 8   -0.042   -0.084    0.026
```

The 'NEU' string in that is followed by a PPP time stamp in GPS Time, the number of processed satellites, and the Nort, East and Up value of the current displacement in meter.

3.11.4 NMEA

BNC allows to output results from Point Positioning in NMEA format. The NMEA messages generated are

- one leading GPRMC message which carries only date and time information, plus

- about once per second a GPGGA message which mainly carries the estimated latitude, longitude, and height values.

3.11.4.1 File - optional

Specify the full path to a file where Point Positioning results are saved as NMEA messages. The generated NMEA file begins with a single GPRMC message which carries only date and time information. The default value for 'File (full path)' is an empty option field, meaning that BNC will not saved NMEA messages into a file.

3.11.4.2 Port - optional

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

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 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 consumtion, 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 comsumption 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.

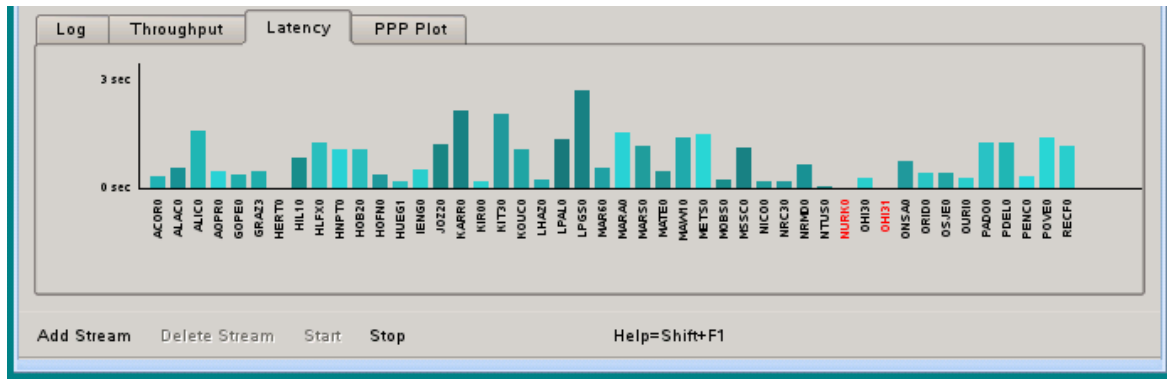Figure: Latency of incoming streams.

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 'Plot 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 coordiate components.
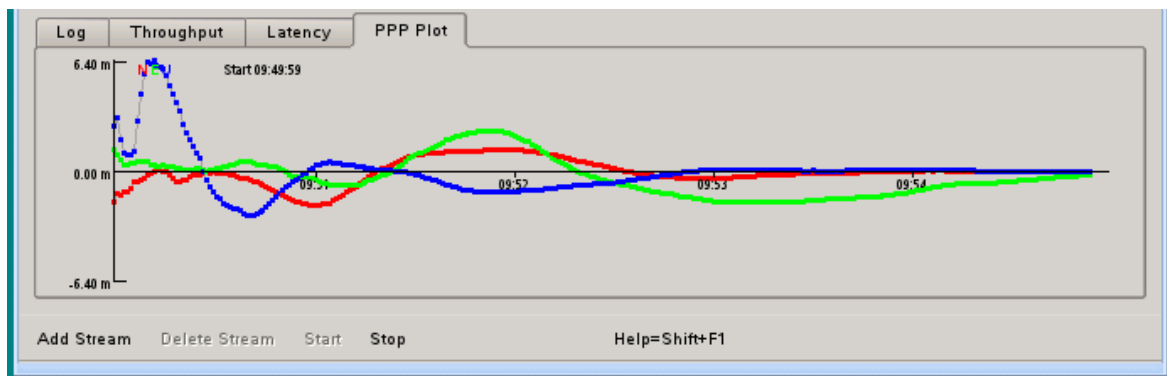

Figure: Time series plot of PPP session.

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.
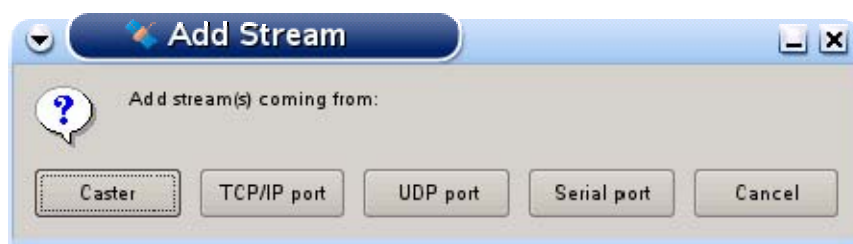

Figure: Steam input communication links.

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.

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.

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.rtcm-ntrip.org/home. A window opens which allows to select a broadcaster for stream retrieval, see figure below.

Figure: Casters table.

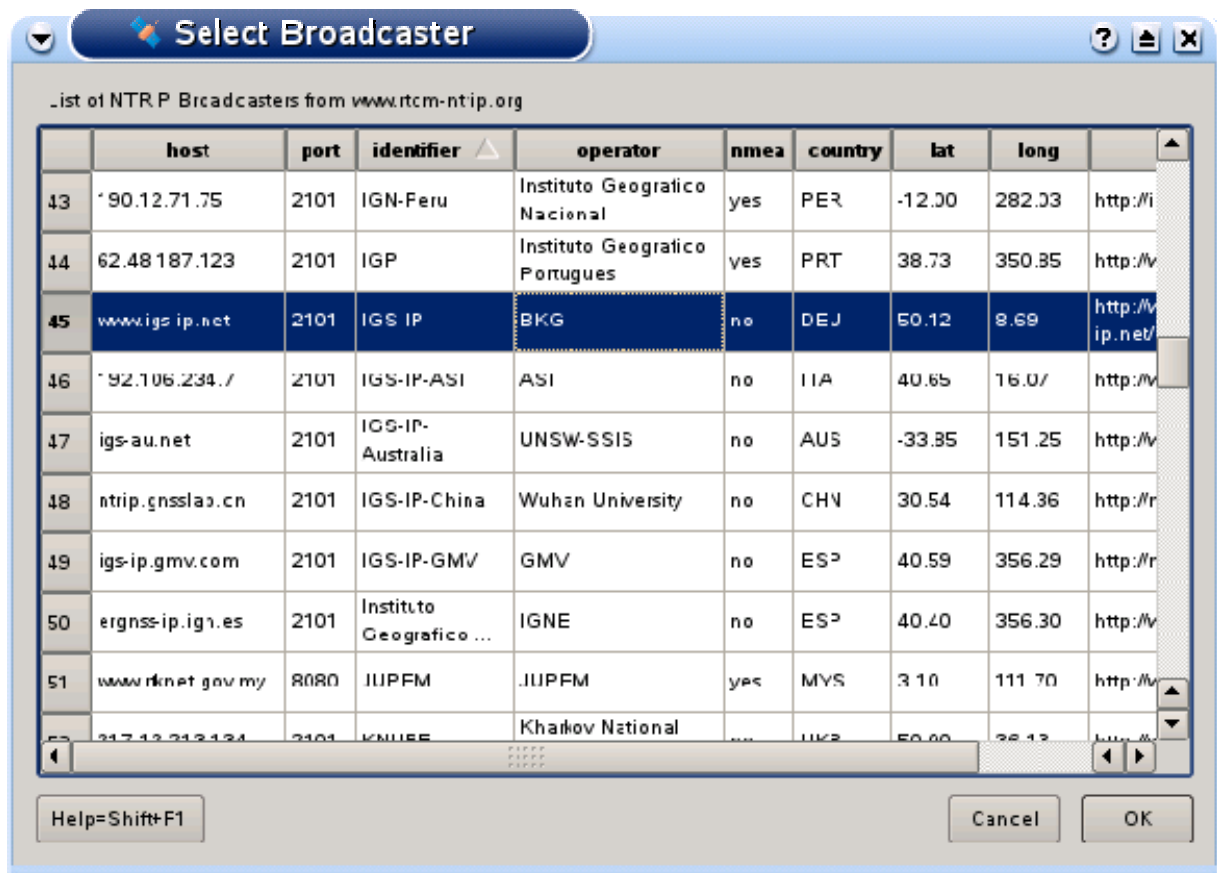### 3.14.1.3 User and Password - mandatory for protected streams

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 and www.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. RTCM Version 2.x streams must contain message types 18 and 19 or 20 and 21 while RTCM Version 3.x streams must contain GPS or SBAS message types 1002 or 1004 and may contain GLONASS message types 1010 or 1012, 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) 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.
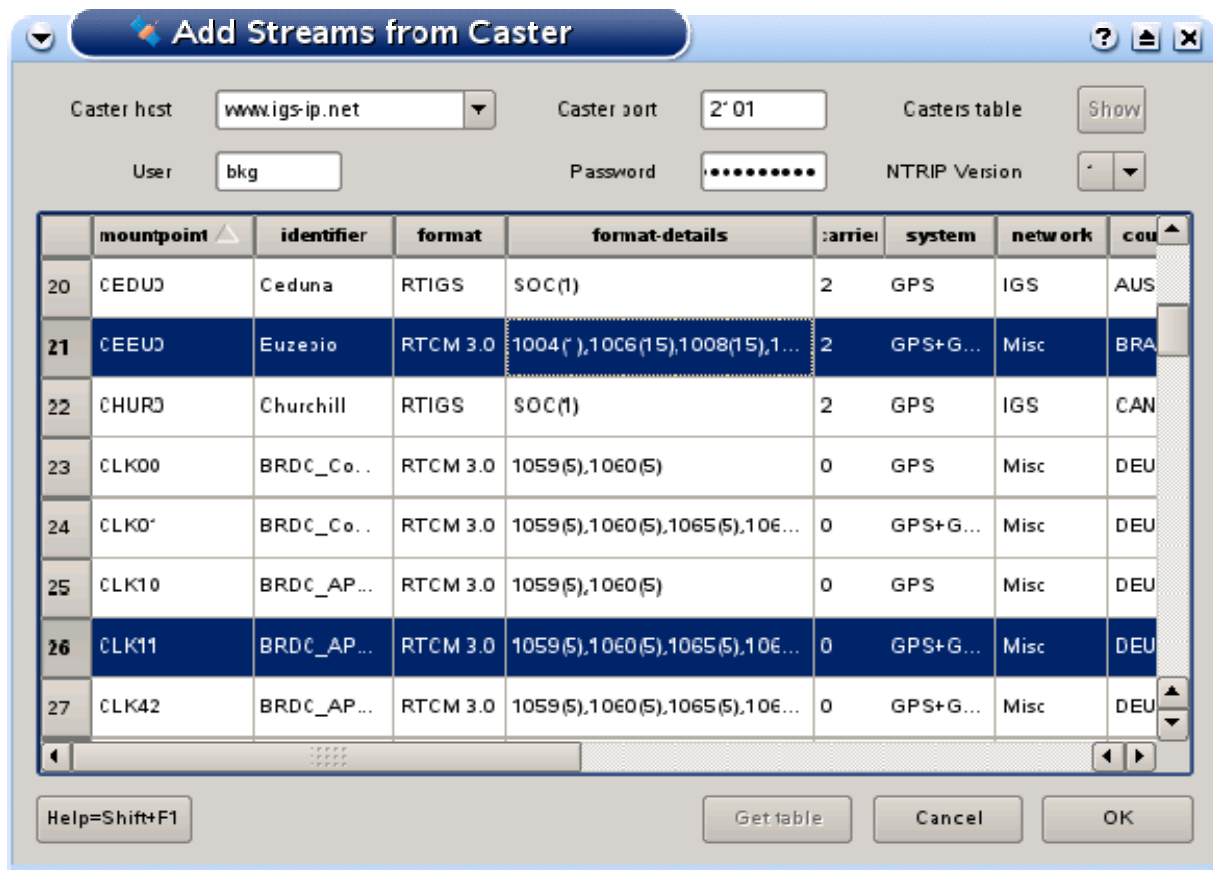

Figure: Broadcaster source-table.

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.

3.14.2 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.
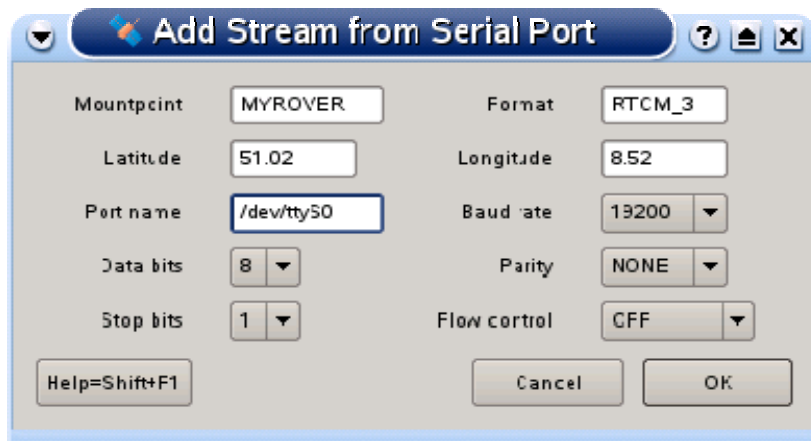
Figure: BNC setup for pulling a stream via serial port.

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 file for post-processing purposes. Enter the following four command line options for that:

- '--file <u>inputFileName</u>' to enter the full path to an input file containing data in RTCM Version 2.x, or Version 3.x, or RTIGS/SOC format.

- '--format <u><format></u>' to enter one of the file format describing strings 'RTCM_2', 'RTCM_3' or 'RTIGS'.
- '--date YYYY-MM-DD' to enter a date for the first epoch.
- '--time HH:MM:SS' to enter a time for the first epoch.

Example:

./bnc --file FFMJ.dat --format RTCM_3 --date 2008-10-27 --time 23:12:56

Note that it is necessary to define a date and time for the first epoch because RTCM streams do not contain complete time stamps as needed i.e. for RINEX files or for PPP in post-processing mode. Note further that when running BNC in offline mode, it will use options for file saving, interval, sampling etc. from its configuration file.

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 <u><confFileName></u>'. This allows to run several BNC jobs in parallel on the same host using different sets of configuration options. <u>confFileName</u> 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.

4. Limitations

- 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 and GLONASS data. Galileo is not yet supported.
- BNC currently will only handle C1, C2, P1, P2, L1, L2, S1, and S2 observations as well as the cumulative loss of continuity and lock time indicators. 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 while RTCM Version 3.x streams can only transport one code observable per frequency. Note that signal-to-noise ratios 'S' are also logged mapped to integer numbers 1 to 9.
- Using RTCM Version 3.x, BNC will properly handle message types 1002, 1004, 1010, and 1012. Note that 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 â€' Set 10'), a final decision is not yet made. Note the what's implemented in BNC is just a temporary solutions.
- 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 and www.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.

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]

6. Annex

6.1 Revision History

Date	Version	Changes
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
Apr 2007	Version 1.2b	[Bug] Output only through IP port [Bug] Method 'reconnecting' now thread-save [Add] ZERO decoder added [Mod] Download public RINEX skeletons once per day [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'
Apr 2008	Version 1.5	[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 [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 ouput 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
Nov 2009	Version 1.7	[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.rtcm-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 [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
May 2010	Version 2.1	[Bug] SSR GLONASS message generation [Add] PPP in post-processing mode [Mod] Update of SSR messages following draft dated 2010-04-12
Jul 2010	Version 2.2	[Bug] GLONASS ephemeris time

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.rtcm.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 XZY 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).

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://igscb.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 form 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	Add Stream: NTRIP Version
obsRate=	Outages: Observation rate
onTheFlyInterval=1 day	General: Reread configuration
outEphPort=	RINEX Ephemeris: Port
outFile=	Feed Engine: File (full path)
outPort=	Feed Engine: Port
outUPort=	Feed Engine: Port (unsynchronized)
perfIntr=	Miscellaneous: Log latency

proxyHost=	Proxy: Proxy host
proxyPort=	Proxy: Proxy port
rnxAppend=2	General: Append files
rnxIntr=15 min	RINEX Observations: Interval
rnxPath=/home/user/rinex	RINEX Observations: Directory
rnxSample=0	RINEX Observations: Sampling
rnxScript=/home/user/rinex/up2archive	RINEX Observations: Script (full path)
rnxSkel=	RINEX Observations: Skeleton extension
rnxV3=0	RINEX Observation: Version 3
scanRTCM=0	Miscellaneous: Scan RTCM
serialAutoNMEA=Auto	Serial Output: NMEA
serialBaudRate=9600	Serial Output: Baud rate
serialDataBits=8	Serial Output: Data bits
serialHeightNMEA=	Serial Output: Height
serialMountPoint=	Serial Output: Mountpoint
serialParity=NONE	Serial Output: Parity
serialPortName=	Serial Output: Port name
serialStopBits=1	Serial Output: Stop bits
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: Mountpoint
pppSPP=	PPP Client: PPP/SPP
pppStatic=0	PPP Client: Static mode
pppUsePhase=0	PPP Client: Use phase data
pppEstTropo=0	PPP Client: Estimate troposphere
pppGLONASS=0	PPP Client: Use GLONASS
pppOrigin=	PPP Client: Origin of time series plot
pppRefCrdX=	PPP Client: X coordinate of plot origin
pppRefCrdY=	PPP Client: Y coordinate of plot origin
pppRefCrdZ=	PPP Client: Z coordinate of plot origin
nmeaFile=	PPP Client: NMEA outputfile
nmeaPort=	PPP Client: NMEA IP output port

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
NTRIP broadcaster overview	http://www.rtcm-ntrip.org/home
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.rtcm.org |