

The BKG Ntrip Client (BNC)

by

Georg Weber, Federal Agency for Cartography and Geodesy (BKG), Frankfurt, Germany
Leos Mervart, Czech Technical University Prague, Department of Geodesy

Introduction

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 using option -m32), Solaris, and Mac systems from http://igs.bkg.bund.de/index_ntrip_down.htm. We used the MinGW Version 5.3.1 compiler to create the Windows binary. It is likely that BNC can be compiled on other systems where a GNU compiler and Qt Version 4.3.2 are installed.

BNC's 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. It 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.

The following is a complete description of BNC comprising its purpose, user resources, options and limitations. Background information concerning NTRIP, RTCM, RTIGS and configuration files is provided as an annex. The paper is written in a way that it can serve as a BNC user manual.

1. Purpose

The purpose of BNC is to

- retrieve real-time GNSS data streams available through NTRIP transport protocol,
- generate high-rate RINEX Observation and Navigation files to support near real-time GNSS post-processing applications, and/or
- generate ephemeris and synchronized observations epoch by epoch through an IP port to support real-time GNSS engines, and/or
- monitor the performance of a network of real-time GNSS reference stations to generate advisory notes.

BNC supports the following GNSS data formats:

- RTCM Version 2.x containing message types 18 and 19 (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)
- 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.

2. Resources

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.

Note that running BNC requires the clock of the host computer to be properly synchronized.

BNC 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.

3. Options

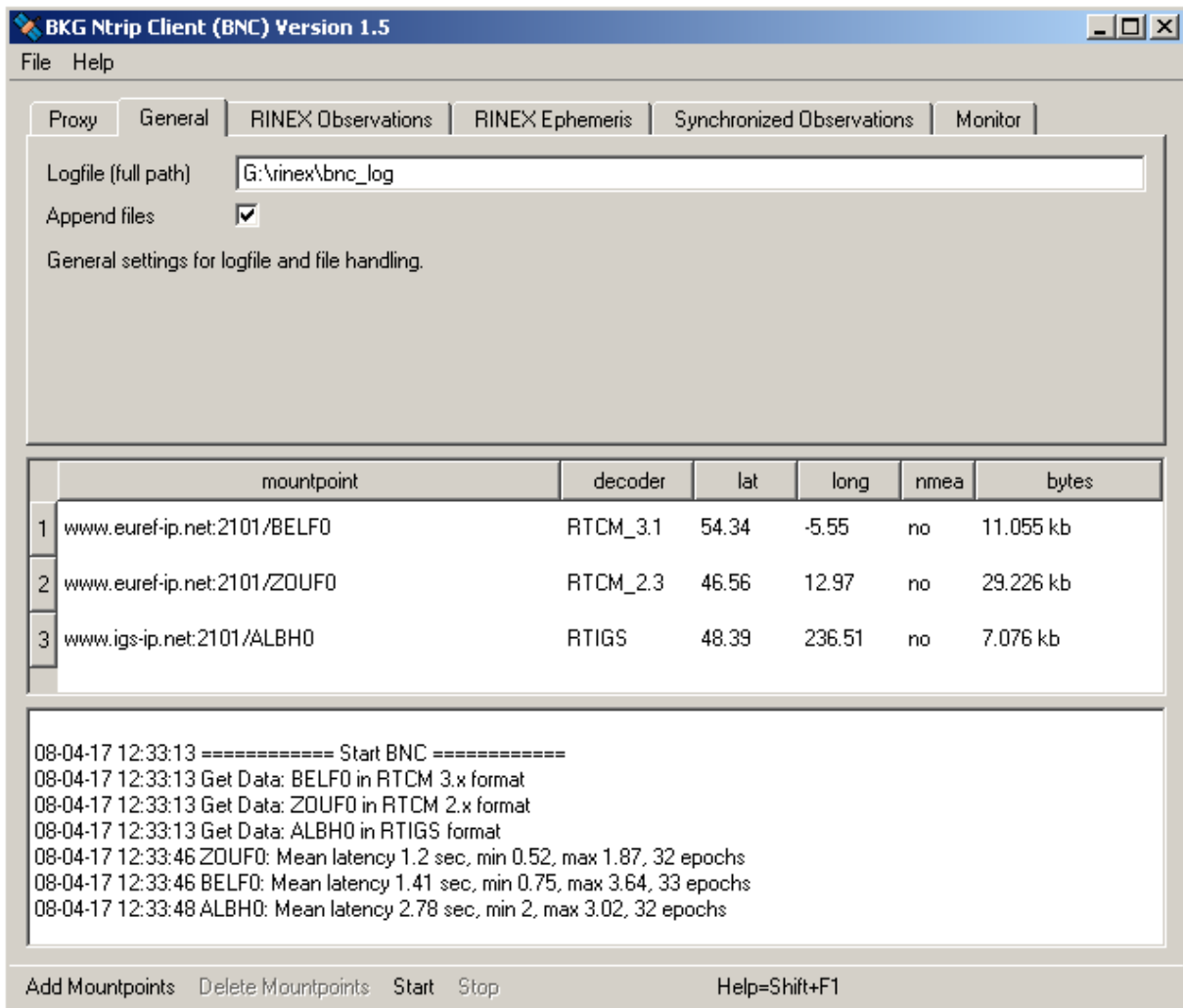


Fig 1: BNC v1.5 options

3.1. File

The 'File' button (see Fig. 1) lets you

- select an appropriate font.
Use smaller font size if the BNC main window exceeds the size of your screen.
- save selected options.
Note that on Windows systems options are saved in register BKG_NTRIP_Client. On Unix/Linux systems options are saved in file
\${HOME}/.config/BKG/BKG_NTRIP_Client.conf.
- quit the BNC program.

3.2. Help

The 'Help' button (see Fig. 2) provides access to

- help contents.
You may keep the 'Help Contents' window open while configuring BNC.
- 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.

3.3. Proxy - for usage in a protected LAN

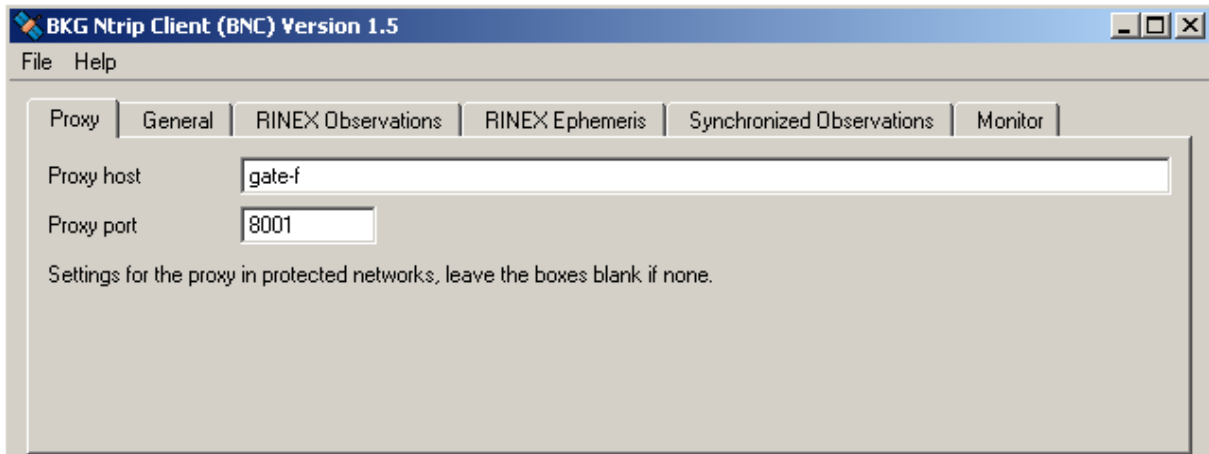


Fig 2: Proxy options

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 (see Fig. 2) 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.

3.4. General

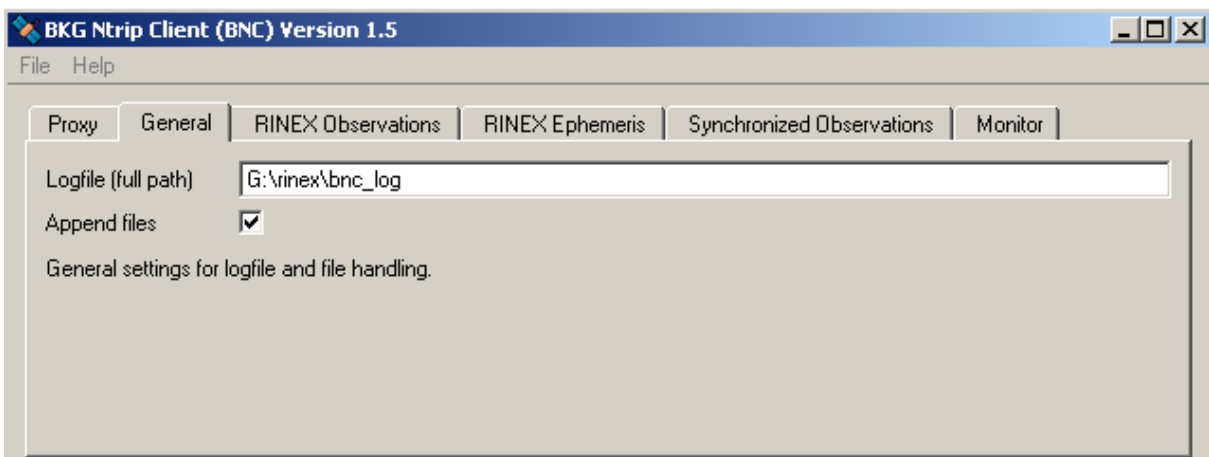


Fig 3: General options

3.4.1 Logfile - optional

Records of BNC's activities are shown in the Log section on the bottom of the main window. These logs can be saved into a file when a valid path is specified in the 'Log (full path)' field (see Fig. 3). The message log covers 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 'Log (full path)' is an empty option field, meaning that BNC logs will not be saved into a file.

3.4.2 Append Files

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.

3.5. RINEX - Observations

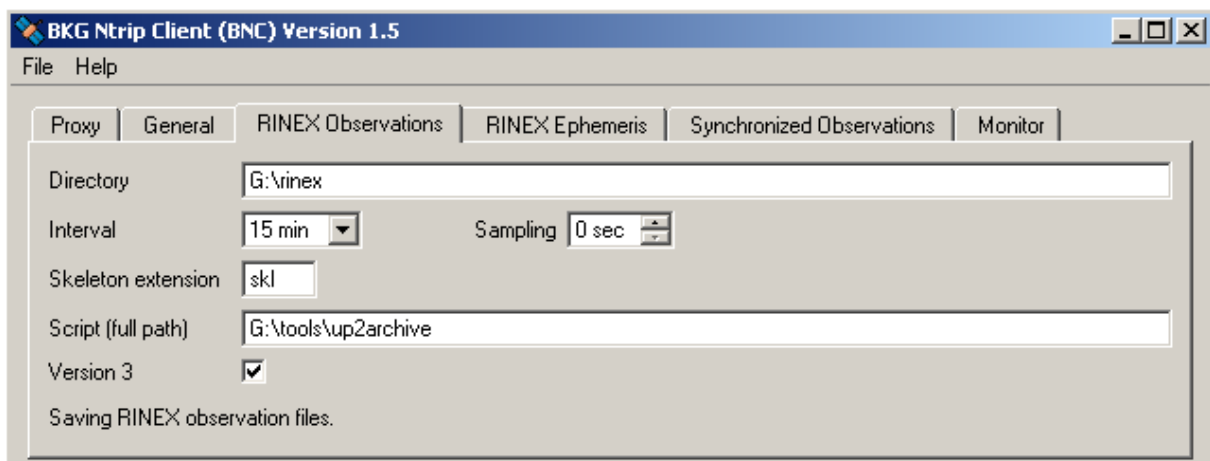


Fig. 4: RINEX Observations options

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.

3.5.1 RINEX File Names

RINEX file names are derived by BNC from the first 4 characters of the corresponding 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.

3.5.2 Directory - optional

Here you can specify the path to where the RINEX Observation files will be stored (see Fig. 4). 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.

3.5.3 Interval - mandatory if 'Directory' is set

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

3.5.4 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.

3.5.5 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 'Directory' with a 'Skeleton extension' 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 RINEX '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 / DATEBNC will complete this line and include it in the actual RINEX file header.
- They should further contain complete header records of type
 - MARKER NAME
 - OBSERVER / AGENCY
 - REC # / TYPE / VERS
 - ANT # / TYPE
 - APPROX POSITION XYZ
 - ANTENNA: DELTA H/E/N
 - WAVELENGTH FACT L1/2
- They may contain any other optional complete header record as defined in the RINEX documentation.
- They should then contain empty header records of type
 - # / TYPES OF OBSERV
 - TIME OF FIRST OBSBNC will include these lines in the final RINEX file header together with an additional
 - COMMENTline describing the source of the stream.
- They should finally contain an empty header record of type
 - END OF HEADER (last record)

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

3.5.6 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 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'.

3.5.7 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.

3.6. Ephemeris

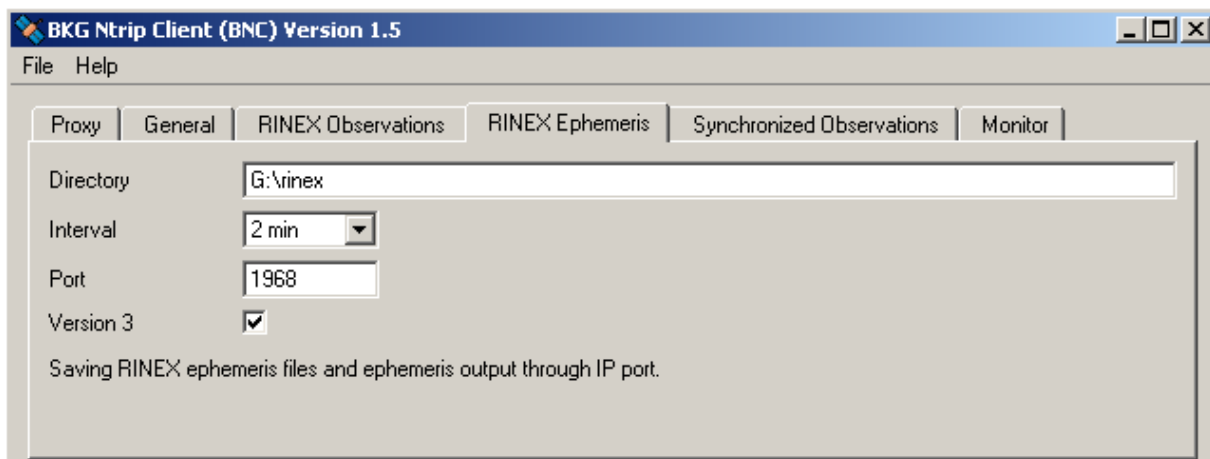


Fig. 5: RINEX Ephemeris options

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.

3.6.1 Directory - optional

Specify the path for saving broadcast ephemeris data as RINEX Navigation files (see Fig. 5). 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.

3.6.2 Interval - mandatory if Ephemeris 'Directory' is set

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

3.6.3 Port - optional

BNC can output broadcast ephemeris in RINEX 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 ASCII ephemeris output via IP port is generated.

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

3.6.4 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.

3.7. Synchronized Observations

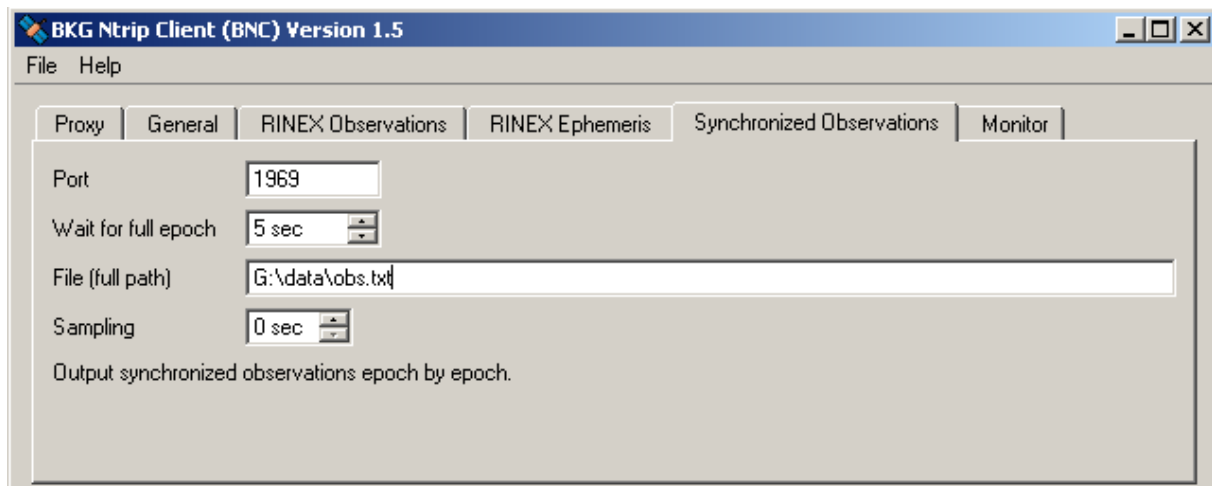


Fig 6: Synchronized Observations options

BNC can generate synchronized epoch by epoch observations from all stations and satellites. The output can be in either a plain ASCII format and/or a binary format. It comprises the following observations where available:

StatID, SVPRN, GPSWeek, GPSWeeks, C1, C2, P1, P2, L1, L2, S1, S2, SNR1, SNR2

Note that SNR stands 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'.

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'. Specify an IP port number here (see Fig. 6) to activate this function. The default is an empty option field, meaning that no binary output is generated.

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

```
begEpoch
begObs
Observation
begObs
Observation
begObs
Observation
...
endEpoch
begEpoch
...
```

with the corresponding structures defined as follow:

```
const char begEpoch = 'A';
const char begObs    = 'B';
const char endEpoch = 'C';
struct Observation {
    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)
    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 binary observation output from the IP port (hard-coded to 1968) and save the observations in file 'obs.txt'.

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

When feeding a real-time GNSS engine waiting for 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 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.

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

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

3.8. Monitor

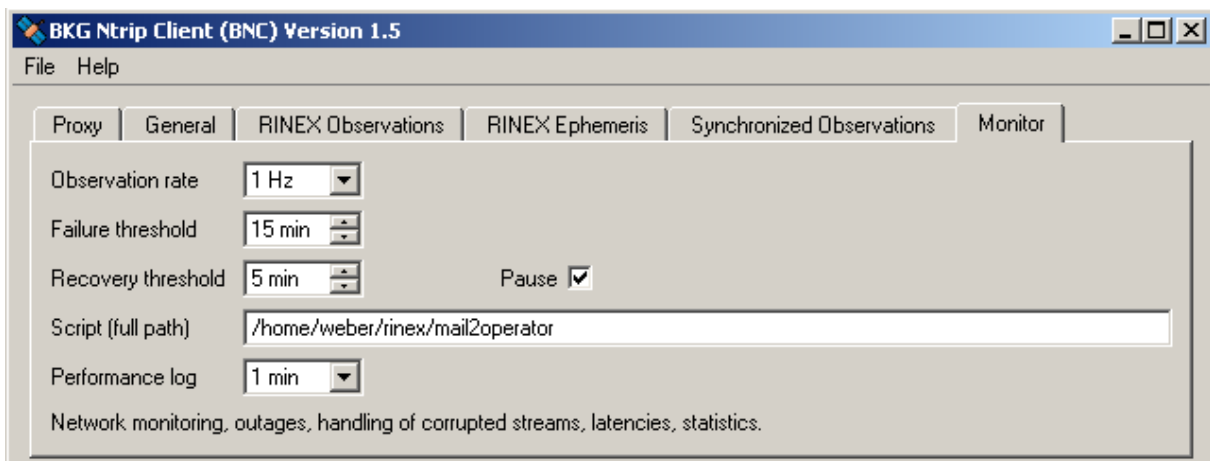


Fig. 7: Monitor options

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 file/section. 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 utilise BNC as a real-time performance monitor and alarm system for a network of GNSS reference stations.

3.8.1 Observation Rate - mandatory if 'Failure threshold', 'Recovery threshold', 'Pause' and 'Advisory Script' is set

BNC can collect all returns (success or failure) coming from a decoder within a certain short time span to then decide whether a stream has an outage or its content is corrupted. This procedure needs a rough a priori estimate of the expected observation rate of the incoming streams (see Fig. 7).

An empty option field (default) means that you don't want an explicit information from BNC about stream outages and incoming streams that can not be decoded and that the special procedure for handling of corrupted streams is by-passed (decoding attempt will never get paused).

3.8.2 Failure Threshold - optional

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

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

3.8.3 Recovery Threshold - optional

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

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

3.8.4 Pause - optional

In case of a corrupted stream, the decoding process can be paused and decodings are then attempted again at decreasing rate. BNC will first attempt to decode again after a 30 second lag and if unsuccessful, make another attempt within 60 seconds after the previous attempt. If it is still unsuccessful, it will make another attempt to decode within 120 seconds after the

previous attempt and so on. Each decoding attempt doubles the wait time since the previous attempt. The maximum wait time between attempts is limited to 960 seconds. Tick 'Pause' to activate this function. Note that it is only effective if an 'Observation rate' is specified.

Do not tick 'Pause' if you want to prevent BNC from making any decoding pause. Be aware that this may incur an unnecessary workload.

3.8.5 Advisory 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 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 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 systems:

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

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

3.8.6 Performance Log - optional

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, hard-coded to 14
-----
= Latency
```

BNC can average the latencies per stream over a certain period of GPS time, the 'Performance log' interval. Mean latencies are calculated from the individual latencies of at most one (first incoming) observation per second. Note that computing correct latencies requires the clock of the host computer to be properly synchronized.

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

Latencies and statistical information can be recorded in the Log file/section at the end of each 'Performance log' interval. A typical output from a 1 hour 'Performance log' interval would be:

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

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

3.9. Mountpoints

Each stream on an NTRIP broadcaster 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 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 'Mountpoints' section on BNC's main window (see Fig. 1). The list provides the following information extracted from source-table(s) produced by the NTRIP broadcasters:

mountpoint	NTRIP broadcaster URL, port, and mountpoint
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'
bytes	Number of bytes retrieved

3.9.1 Add Mountpoints

Button 'Add Mountpoints' (see Fig. 1) opens a window that allows user to select data streams from an NTRIP broadcaster according to their mountpoints.

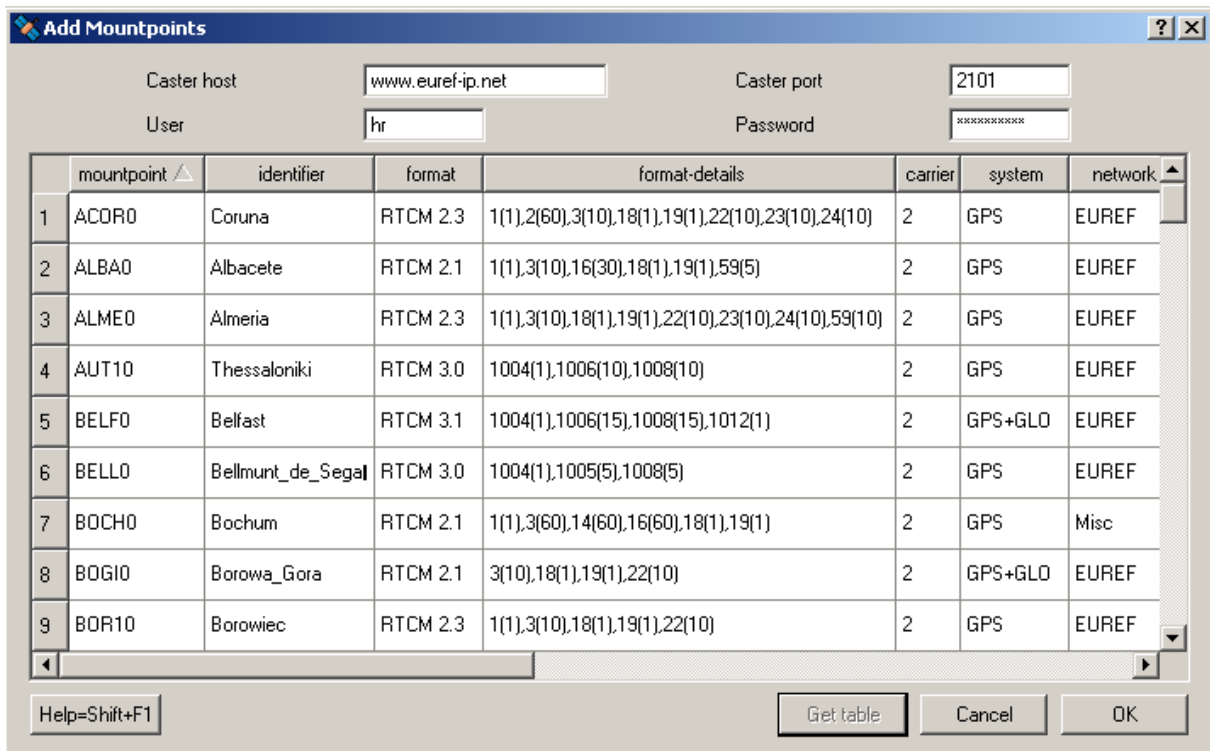


Fig. 8: Add Mountpoints options

3.9.2 Broadcaster Host and Port - required

Enter the NTRIP broadcaster host IP and port number (see Fig. 8). <http://www.rtcntrip.org/home> provides information about known NTRIP broadcaster installations. Note that EUREF and IGS operate NTRIP broadcasters at <http://www.euref-ip.net/home> and <http://www.igs-ip.net/home>.

3.9.3 Broadcaster User and Password - required 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/index_ntrip_reg.htm for access to protected streams on www.euref-ip.net and www.igs-ip.net.

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

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

3.9.5 Delete Mountpoints

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

3.9.6 Edit Mountpoints

- 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 (see Fig. 1), 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 (see Fig. 1) as 'ZERO'. The generated file names are created from the characters of the 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. This stream is customized to the exact latitude and longitude as shown in the 'lat' and 'long' columns under 'Mountpoints'. These VRS streams are indicated by a 'yes' in the 'nmea' column under 'Mountpoints' as well as in the source-table.

The default 'lat' and 'long' values are taken from the source-table however in most cases you would probably want to change this according to your requirement. Double-click on 'lat' and 'long' fields (see Fig. 1), 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 mountpoints with a 'yes' in its 'nmea' column can be edited. The position must preferably be a point within the 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.

3.10. 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.

3.11. Stop

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

3.12. No Window - optional

On all systems BNC can be started in batch mode with the command line option '-nw'. BNC will then run in 'no window' mode, using options from the configuration file `${HOME}/.config/BKG/BNC_NTRIP_Client.conf` (Unix/Linux, see Config File example in the Annex) or from the register `BKG_NTRIP_Client` (Windows).

Note that the self-explaining contents of the configuration file or the Windows register can easily be edited. Terminate BNC using Windows Task Manager when running it in 'no window' mode on Windows systems.

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. Which observables 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 2.x, BNC will only handle message types 18 and 19. 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).
- 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, its configuration can not be changed as long as it is stopped. A reconfiguration on-the-fly is not implemented.

Acknowledgements

BNC's Help Contents has been proofread by Thomas Yan, University of New South Wales, Australia. Scott Glazier, OmniSTAR Australia, included the decoding of broadcast ephemeris from RTIGS streams and has been helpful in finding BNC's bugs. James Perlt, BKG, helped fixing bugs and redesigned BNC's main window. Andre Hauschild, German Space Operations Center, DLR, revised the RTCMv2 decoder.

Annex

A.1. NTRIP

'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.

A.1.1 Source-table

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: 'identifiery', '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'.

A.2. RTCM

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

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

A.2.1 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.

A.2.2 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).

A.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://igs.cb.jpl.nasa.gov/mail/igs-rtwg/2004/msg00001.html> for further details.

A.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.

A.4. Configuration File

The following is an example for the contents of a Unix/Linux configuration file `/${HOME}/.config/BKG/BKG_NTRIP_Client.conf`. It enables the retrieval of stream ACOR0 from www.euref-ip.net and FFMJ3 from www.igs-ip.net for the generation of 15 min RINEX files. RINEX files are uploaded to an archive using script 'up2archive' :

```
[General]
adviseFail=15
adviseReco=5
adviseScript=
binSampl=0
casterHost=www.euref-ip.net
casterPassword=pass
casterPort=80
casterUser=user
ephIntr=1 day
ephPath=/home/user/rinex
ephV3=2
logFile=/home/user/log.txt
makePause=0
mountPoints=//user:pass@www.euref-ip.net:2101/ACOR0 RTCM_2.3 43.36 351.60
no, //user:pass@www.igs-ip.net:2101/FFMJ3 RTCM_3.0 41.58 1.40 no
obsRate=
outEphPort=2102
outFile=/home/user/ascii
```

outPort=2101
perfIntr=
proxyHost=proxyhost
proxyPort=8001
rnxApend=2
rnxIntr=15 min
rnXPath=/home/weber/rinex
rnxSampl=0
rnxScript=/home/weber/up2archive
rnxSkel=SKL
rnxV3=2
waitTime=5

A.5 Links

NTRIP	http://igs.bkg.bund.de/index_ntrip.htm
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.rtcn-ntrip.org/home
EUREF-IP Project	http://www.epncb.oma.be/dataproducts/data_access/real_time/
Real-time IGS Pilot Project	http://www.rtigs.net/pilot
Radio Technical Commission for Maritime Services	http://www.rtcn.org