HELMHOLTZ ASSOCIATION

HELMHOLTZ CENTRE POTSDAM
GFZ GERMAN RESEARCH CENTRE FOR GEOSCIENCES

GFZRNX 1.07 Users Guide

DokuWiki

Tue Jan 10 15:07:17 UTC 2017

Contents

Chapter 1

Before You Start

1.1 End User License Agreement

- The software **gfzrnx-RINEX GNSS Data Conversion and Manipulation Toolbox** can be used under the following license conditions:

1. With this license the copyright holder **GFZ** grants you permission to use the software **gfzrnx** free of charge in executable form and for non-commercial purposes only.
2. When using the software please cite it as:
   Nischan, Thomas (2016): GFZRNX - RINEX GNSS Data Conversion and Manipulation Toolbox. GFZ Data Services. http://dx.doi.org/10.5880/GFZ.1.1.2016.002
3. You may not copy, modify, sublicense, distribute or create any derivate works of the software except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License.
4. The software is provided "as is" without warranty of any kind, either expressed or implied, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. The entire risk as to the quality and performance of the program is with you. Should the program prove defective, you assume the cost of all necessary servicing, repair or correction.
5. In no event unless required by applicable law or agreed to in writing will the copyright holder be liable to you for damages, including any general, special, incidental or consequential damages arising out of the use or inability to use the program (including but not limited to loss of data or data being rendered inaccurate or losses sustained by you or third parties or a failure of the program to operate with any other programs), even if the holder has been advised of the possibility of such damages.
6. This terms shall be governed by and construed and enforced in accordance with the laws of the Federal Republic of Germany.
7. This license does not include the permission for commercial usage of the software. The right for commercial usage is subject to a different license agreement including license fee and further conditions.

1.2 Scope of Operation

The **gfzrnx** is a toolbox for RINEX file manipulation for the major versions 2 and 3.
The following RINEX data types are supported:

- **O**bservation data
- **N**avigation data
- **M**eteorological data

The following operations/tasks are supported:

- RINEX file check and repair,
- RINEX file format conversion (version 3 to 2 and vice versa),
- RINEX file splice,
- RINEX file split,
- RINEX file statistics generation,
- RINEX file manipulations like:
  - data sampling,
  - observation types selection,
  - satellite systems selection,
  - elimination of overall empty or sparse observation types.
- Automatic version dependent file naming on output file.
- RINEX file (re)naming support (version 2 to 3)
- RINEX header editing
- RINEX file meta data extraction
- RINEX file comparison

See also the **Rinex Standard Extensions/NonConformity** section for further information.

1.3 Examples

You can always find examples in boxes with light grey background like the one below.

```
Example Box
```

All given examples are valid for the UNIX based systems like Linux, SunOS or OSX.
You will find almost **gfzrnx** used in the example boxes which is always used as a synonym for the operating system dependent executable (gfzrnx_lx, gfzrnx_osx, ...).

1.4 Follow us

1.4.1 Join Mailing List

There is a mailing list **gfzrnx@gfz-potsdam.de** which will be used for information transfer (new features, versions, etc.). It can be also used for questions which are not covered by the documentation.

One can join the mailing list sending an empty e-mail to:
gfzrnx-on@gfz-potsdam.de .
After getting a **Confirmation Request** e-mail, please don't forget to **reply** to this Confirmation Request. This reply is mandatory to finish your list joining.

1.4.2 Drop Out of Mailing List

One can drop out of the mailing list sending an empty e-mail to:
gfzrnx-off@gfz-potsdam.de .

1.4.3 Twitter: @gfzrnx

Figure 1.1: Twitter: @gfzrnx

1.5 Bug Reports / Comments

For bug reports or comments please use the mailing address **gfzrnx_bug@gfz-potsdam.de** .
Please use the following procedure for bug reports:

- Make sure, that you are using the latest version.
- If you are using the latest version, please provide the complete command line you have used.
- attach your input file(s) to your e-mail or provide a link for the input data download. Shrink the input file(s) if possible.

Chapter 2

Basics

2.1 Software

2.1.1 Download

One can download the software via:
http://semisys.gfz-potsdam.de/semisys [Download → GFZ Software → gfzrnx]
You will find an **official** version with a version number and a **development** version (DEVEL) with ongoing bug fixing and may be new features. The **manual** (pdf) can be downloaded from there too.

2.1.2 Install

The software consists of a single executable (operating system dependent) to be used at the command prompt of a Terminal window or in batch scripts.

Linux (64)	gfzrnx_lx
Linux (32)	gfzrnx_lx32
SunOS (Sparc)	gfzrnx_sun
SunOS (i86)	gfzsun_suni86
MS Windows (64)	gfzrnx_win64.exe
MS Windows (32)	gfzrnx_win32.exe
Mac OSX	gfzrnx_osx

UNIX: Copy the executable into a directory covered by your system search PATH variable.
WINDOWS: Copy the executable into your **Windows** directory for ease of use.

2.1.3 Usage

gfzrnx is a **command line executable**. It can be used in a terminal window or batch scripts.
It has **no graphical interface !**

2.1.3.1 Unix

For Unix (Linux, MacOS, SunOS) users it can be run in any Terminal application or used in shell-scripts ...

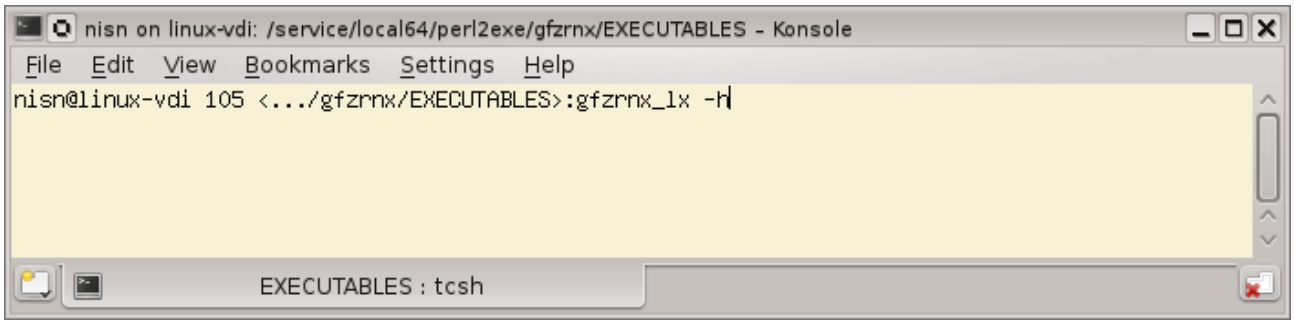

Figure 2.1: Unix Terminal - command line

2.1.3.2 Windows

For MS Windows you can use e.g. the **cmd.exe** or create and execute batch-scripts (whatever.bat).

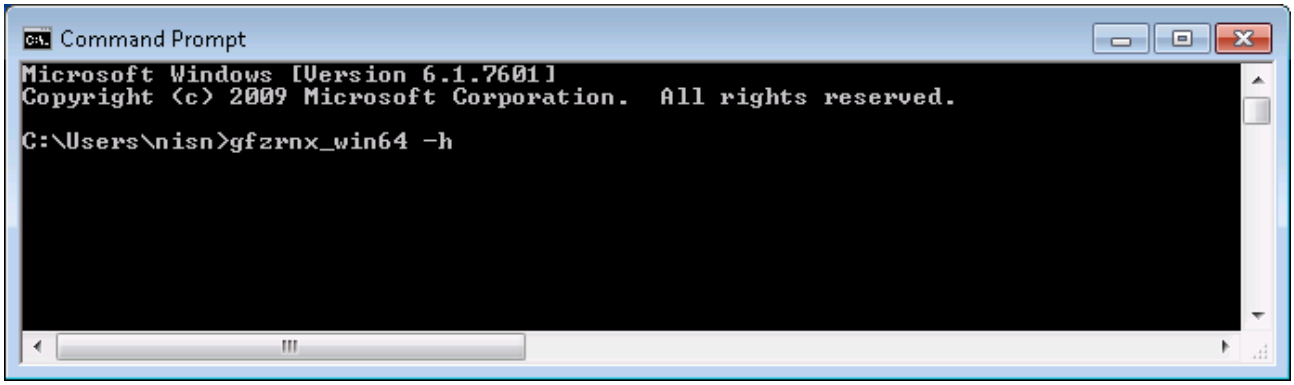

Figure 2.2: MS Windows command window - command line

Here a small batch file **example.bat** is shown. The input data are sampled to 30 s time interval.

```
gfzrnx_win64.exe -finp C:\data\XXXX0010.15o -fout C:\data_30\XXXX0010.15o -smp 30
gfzrnx_win64.exe -finp C:\data\XXXX0020.15o -fout C:\data_30\XXXX0020.15o -smp 30
...
gfzrnx_win64.exe -finp C:\data\XXXX3650.15o -fout C:\data_30\XXXX3650.15o -smp 30
```

2.1.4 Fast Help

A simple usage information you can get via command line parameter -**h** or -**help**.

```
gfzrnx -h

***** USAGE: ./gfzrnx

     file only or common options
     ------------------------------------------------------------------------------------
     [-h]                       - show this usage message
     [-help]

     [-finp <file list>]        - input  rinex file(s) (std. STDIN).
                                  STDIN is only valid for a single file input.

                                  the following file name types are supported to derive the
                                  nominal epoch/duration information.
```

```
                                RINEX-2 file naming

                                ssssDDD0.YYx       - daily      file
                                ssssDDD[a-x].YYx   - hourly     file
                                ssssDDD[a-x]mm.YYx - sub-hourly file

                                RINEX-3 file naming

                                SSSSMRCCC_S_YYYYDDDHHMM_NNN_FRQ_TT.FMT
                                SSSSMRCCC_S_YYYYDDDHHMM_NNN_TT.FMT

                                see Documentation for details

                                splice mode:
                                ------------
                                * list of input files

   [-fout <file>]             - output rinex or statistics file (std. STDOUT)
                                automatic file_name if filename given is "::RX2::" or "::RX3::".


   [-4to9 <file>]             - renaming information for rinex-3 type (re)naming
                                ( NNNN -> NNNNMRCCC / POTS -> POTS00DEU )

   [-f]                       - force overwrite of output file if it already exists
                                (std. no overwrite)

   [-sifl]                    - perform an operation on a single file if a file list is
   [-single_file]               provided via "-finp"

[-nomren23 <[s,][mr,][iso]>] - fast nominal output file name for RINEX-2 to RINEX-3 file rena  +
                                                                                        ming.
                                RINEX-3 output file name is written to STDOUT.

                                   s   - data source (S|R)        (default R)
                                   mr  - marker receiver number   (default 00)
                                   iso - 3 char. iso country code (default XXX)

                                the input parameters can be given in any order.
                                supported input file names nnnnddde.yyt[.cmp] or nnnnddddedd.yy  +
                                                                                      t[.cmp]

                                if providing a compressed file all information which is usuall  +
                                                                                        y taken
                                from file header (sat. system(s), data frequency) has to be gi  +
                                                                                   ven via the
                                command line parameter (see documion for details).

   [-vo <2|3>]                - output RINEX version (std. 3)
   [--version_out <2|3>]

   [-pr3rx2 <list>]           - komma separated list of list of signal priorities used for rin  +
                                                                              ex 3 -> 2 conversion
                                to overwrite the standard settings, see documentation for details.


                                S:n[n...]:STRING

                                S      - satellite System [CEGJRSI]
                                n      - frequency number(s)
                                STRING - prority STRING
```

```
                                G:12:PWCSLXYN,G:5:QXI,R:12:CP

80      [-errlog <file>]          - store (append) error logs to a file (std. print to STDERR)

        [-smp <num>]              - sampling rate in sec. (std. no sampling / resolution 1 ms)

        [-stk_obs]                - output data statistics information (std. STDOUT)
        [-stk_only]

        [-crux <file>]            - rinex header manipulations definitions for input files

        [-show_crux]              - show crux structure adopted and used by the program
90
        [-hded]                   - perform the header edit ONLY mode (with -crux)

        [-stk_epo <n[:list]>]     - ASCII timeplot of data availability (std. STDOUT)
                                    n    - time resolution in seconds
                                    list - comma separated list (prn,otp) (std. prn)

        [-ot <list>]              - obs. types list to be used (pattern matching). the list can be  +
                                                                                              given
        [--obs_types <list>]        globaly or sat. system dependent. the sat. system dependent record
100
                                    replaces fully a global one.

                                    list can be: [S:]OT1,OT2,...[+S:OT3,OT4,...][+...]

                                    S  - satellite system [CEGJRSI]
                                    OT - observation type identifier

                                    L1,L2,C1,C2,P1,P2
                                    L1,L2,C1,C2,P1,P2+C:L1,L7,C1,C7+G:L1C,L2W,C1,C2
110
        [-ots <string>]           - obs. types output sorting
      [--obs_types_sort <string>]   the string can consist of the 1st obs. types id. character ( e  +
                                                                                          .g. CPLDS )

        [-prn <prn-list>]         - komma separated list of PRNs to be used
                                    range notations are possible G1-32,C01-5,R01-10,E14,E18

        [-no_prn <prn-list>]      - komma separated list of PRNs to be skipped
                                    range notations are possible G1-32,C01-5,R01-10,E14,E18
120
        [-kaot]                   - keep all obs. types (including fully empty ones)

        [-rsot <n>]               - remove sparse obs. types.
    [--remove_sparse_obs_types <n>]  n - defines the % limit of the median number of observations
                                         per observation type used to delete an observation type fully.


        [-satsys <letters>]       - satellite system(s) to be used (CEGIJRS) (std. CEGIJRS)
                                    C - Beidou
130                                 E - Galileo
                                    G - GPS
                                    I - IRNSS
                                    J - QZSS
                                    R - Glonass
                                    S - SBAS

        [-ns <type>]              - output order of navigation records.   type = [time|prn] (std. prn)
```

```
      [--nav_sort <type>]         time - sort by time,prn
                                  prn  - sort by prn,time

      [-split n]                - split input file in <n seconds> pieces
                                  - valid only with -fout ::RX2:: or ::RX3::
                                  - valid if n is a multiple of 60 seconds.
                                  - only supported for single input file

      [-chk]                    - extended formal checks on input file (slower)

      [-meta <type[:format]>]   - extract file meta data. the type can be (basic|full).
                                  supported formats are json|xml|txt|dump

      [-fdiff]                  - compare two rinex files of the same format (major version id.)
                                  the two input files have to be given via -finp

      [-kv]                     - keep major output version number (2|3) same as in input

      [-q]                      - quiet mode

      [-d <sec>]                - file duration (seconds) (std. ignored on input
      [--duration <sec>]                                   std. 86400   on output )

      [-epo_beg <EPOCH>]        - first output epoch (<EPOCH> see below)

      [-sei <in|out>]
[--strict_epoch_interval <in|out>] - output epoch interval according to in/output file name
                                     (only valid in case of RINEX conform file names)

      [-enb <n>]                - extend the nav. epoch interval by -n and +n seconds
                                  (when using strict epoch interval)

      [-use_obs_map <file>]     - use modified obs. types mapping
      [-out_obs_map]            - output std.  obs. types mapping


      epoch <EPOCH> parameter
      ---------------------------------------------------------------------------------------
      mjd             56753   or        56753_123000
      wwwwd           17870   or        17870_12:30:00
      yyyyddd       2014096   or      2014096_123000
      yyyymmdd     20140406   or     20140406_12:30:00
      yyyy-mm-dd 2014-04-06   or   2014-04-06_123000

      all these date types can be combined via '_' with a time string of type:
      hhmmss
      hh:mm:ss


  ---------------------------------------------------------------------------------------
  © Helmholtz-Centre Potsdam - GFZ German Research Centre for Geosciences
    Section 1.1 Space Geodetic Techniques

    see http://semisys.gfz-potsdam.de/semisys  [Download -> GFZ Software -> gfzrnx]

    for the manual with license details

    Thomas Nischan, nisn@gfz-potsdam.de
  ---------------------------------------------------------------------------------------

    VERSION: gfzrnx-1.07-7060
```

2.2 Data Input/Output

2.2.1 Supported Format Versions

gfzrnx supports all versions 2.x and 3.x formats as input. The output format will be only the latest standard format of the major formats 2 and 3. for version 2 it is 2.11 and for version 3 this is currently 3.03 .

2.2.2 Input

The input of a single file can be done via the -**finp** command line parameter or via **STDIN**.

2.2.3 Output

The standard output channel is **STDOUT**. The output to a dedicated file can be also done via the -**fout** command line parameter.

2.2.4 Examples Input/Output

2.2.4.1 Input via STDIN

```
cat     pots007a.15o   | gfzrnx ...
crx2rnx pots007a.15d - | gfzrnx ...
```

2.2.4.2 Input via -finp

```
gfzrnx -finp pots007a.15o ...
```

2.2.4.3 Output via STDOUT

```
gfzrnx -finp pots007a.15o > pots007a.15o_rx3
gfzrnx -finp pots007a.15o | rnx2crx > pots007a.15d
gfzrnx -finp pots007a.15o | rnx2crx | gzip > pots007a.15d.gz
```

The program **rnx2crx** is here the Hatanaka RINEX compression and **gzip** a common file compression.

2.2.4.4 Output via -fout

```
gfzrnx -finp pots007a.15o -fout pots007a.15o_rx3
```

2.2.5 Log Messages

By default log messages (**N**otices, **E**rrors, **W**arnings) are sent to **STDERR**. One can store the log messages into a file using the -**errlog** command line parameter.

```
> rnxall -finp leid2000.13o -fout leid2000.13o_rx3

DATE/TIME     | C | EPOCH/FILE  | SITE | T | MESSAGE
--------------+---+-------------+------+---+-----------------------------------------------
2015-01-09 .. | N | .. 00:00:00 | LEID | O | file duration set to 86400 s
2015-01-09 .. | W | .. 00:00:00 | LEID | O | no MARKER NAME in header / taken from file name
2015-01-09 .. | W | .. 00:00:00 | LEID | O | HEADER -> missing receiver type ><
2015-01-09 .. | W | .. 23:59:30 | LEID | O | BEIDOU obs. types update: D2_ -> D1_ !
2015-01-09 .. | W | .. 23:59:30 | LEID | O | BEIDOU obs. types update: L2_ -> L1_ !
2015-01-09 .. | W | .. 23:59:30 | LEID | O | BEIDOU obs. types update: P2_ -> P1_ !
2015-01-09 .. | W | .. 23:59:30 | LEID | O | BEIDOU obs. types update: S2_ -> S1_ !
```

```
2015-01-09 .. | N | .. 23:59:30 | LEID | O | mandatory HEADER label >GLONASS COD/PHS/BIS< added
2015-01-09 .. | N | .. 23:59:30 | LEID | O | mandatory HEADER label >SYS / PHASE SHIFT< added
2015-01-09 .. | N | .. 23:59:30 | LEID | O | label >#  / TYPES OF OBSERV< skipped via output
```

The log table information consists of:

Label	Description
DATE/TIME	processing epoch
C(ode)	N(otice), W(arning), E(rror)
EPOCH / FILE	affected epoch in input file
SITE	4-char. station identifier
T(ype)	Data Type
MESSAGE	log meessage

Output of log information to a file via **-errlog** command line parameter.

```
gfzrnx -finp leid2000.13o -fout xxxx -errlog leid2000.13o_log
```

2.3 Supported File Names

The following input file names are supported and used to initialize the nominal data epoch interval.

2.3.1 RINEX-2 naming convention

File Name	Description	Example
SSSSDDD0.YYT	daily file	pots0070.15o
SSSSDDD[a-x].YYT	hourly file	pots007a.15o
SSSSDDD[a-x]MM.YYT	sub-hourly file	pots007r45.15o

Var.	Description	Example
SSSS	4-char. station identifier	pots
DDD	day of year	007
YY	2-digit year	15
MM	minute of data begin	45
T	data type (o,d,m,n,...)	o

2.3.1.1 Examples

- daily file

```
pots0070.15o
```

- hourly files

```
pots007a.15o pots007b.15o pots007c.15o ... pots007v.15o pots007w.15o pots007x.15o
```

- sub-hourly files (15 min)

```
pots007a00.15o pots007a15.15o pots007a30.15o pots007a45.15o
```

2.3.2 RINEX-3 naming convention

File Nmae	Example
SSSSMRCCC_S_YYYYDDDHHMM_NNN_FRQ_TT.FMT[.CMP]	POTS00DEU_R_20150070000_01H_30S MO.rnx.bz2
SSSSMRCCC_S_YYYYDDDHHMM_NNN_TT.FMT[.CMP]	POTS00DEU_R_20150070000_01H_MN.rnx.gz

Var.	Description	Example
SSSSMRCCC	station identifier	POTS00DEU
SSSS	4-char. identifier	POTS
M	Monument number	0
R	Receiver number	0
CCC	ISO country code	DEU
S	data source	R
YYYYDDDHHMM	start epoch	20150070000
YYYY	year	2015
DDD	day of year	007
HH	hour	00
MM	minute	00
NNN	nominal file period (nominal)	01H
FRQ	data frequency	30S
TT	data type	MO
FMT	format extension	rnx
CMP	compression method	gz, bz2, ...

For more details see RINEX-3 file format definitions.

2.3.3 Automatic Output File Naming

For an automatic output file naming one can use the **::RX2::** or **::RX3::** parameter for the **-fout** command line switch.

2.3.3.1 RINEX-2 Site Name

The 4 character site name is taken from the **"MARKER NAME"** header record. If the site name is not given in the file header it is taken from the input file name (if standard file name). In all other cases it has to be provided via the **-site** command line parameter

2.3.3.2 RINEX-3 Site Name

```
gfzrnx -finp pots0070.15o -fout      ::RX3::
gfzrnx -finp pots0070.15o -fout /tmp/::RX3::
```

This works fully if the header **MARKER NAME** fully matches the RINEX-3 "SSSSMRCCC" naming style. For a 4-char. **MARKER NAME** one has to provide at least the **marker-, receiver numbers** and the **ISO country code** on the command line. If no station information is found the full information has to be given on the command line.

```
gfzrnx -finp pots0070.15o -fout      ::RX3::pots,00,DEU
gfzrnx -finp pots0070.15o -fout /tmp/::RX3::pots,00,DEU
```

The following examples will give the same result if the header 4-char. **MARKER NAME** is set. The parameters order is not relevant.

```
gfzrnx -finp pots0070.15o -fout ::RX3::00,DEU
gfzrnx -finp pots0070.15o -fout ::RX3::DEU,00
```

The output file name will be: **POTS00DEU_R_20150070000_01H_30S_MO.rnx**.

The default **data source** identifier is **R** (Receiver). If one needs the **S** (Streaming), simply add it to the **::RX3::** sub-information.

```
gfzrnx -finp pots0070.15o -fout      ::RX3::00,DEU,S
gfzrnx -finp pots0070.15o -fout /tmp/::RX3::00,DEU,S
```

The output file name will be: **POTS00DEU_S_20150070000_01H_30S_MO.rnx**.

2.3.3.3 RINEX-3 Site Name (-4to9)

Beside the naming definitions on the command line (-fout ::RX3::00,DEU) multiple site identifier definitions can be provided via the **-4to9** command line parameter providing a simple file with the naming information.

```
gfzrnx -finp pots0070.15o -fout ::RX3:: -4to9 four2nine.conf
```

The **-4to9** input file (e.g.) must have the following structure:

```
#    name mr iso
0001 pots 00 DEU
0002 brux 00 BEL
0003 tash 00 UZB
...
```

A correct numbering can be ignored if it is out of interest to you. In this case you can use the same number for all stations.

```
#    name mr iso
1    pots 00 DEU
1    brux 00 BEL
1    tash 00 UZB
...
```

An up to date **4to9** configuration file for diverse networks like **IGS**, **MGEX**, **EUREF**, **TIGA** and others can be derived from **GFZ**'s **SE**nsor **M**eta **I**nformation **SYS**tem (SEMISYS) via a simple command line:

```
curl -G http://semisys.gfz-potsdam.de/semisys/api/ -d 'symname=1005' -d 'network=EPN'           +
                                                                          -o EPN_4to9.txt
curl -G http://semisys.gfz-potsdam.de/semisys/api/ -d 'symname=1005' -d 'network=IGS,MGEX'      +
                                                                       -o IGS_MGEX_4to9.txt
curl -G http://semisys.gfz-potsdam.de/semisys/api/ -d 'symname=1005' -d 'network=EPN,IGS,MGEX,T  +
                                                                       IGA' -o ALL_4to9.txt
```

```
wget 'http://semisys.gfz-potsdam.de/semisys/api/?symname=1005&network=EPN' -O EPN_4to9.txt
```

For more details see the SEMISYS **api** and **download** page http://semisys.gfz-potsdam.de/semisys/download .

2.3.3.4 RINEX-2 Start Epoch/Duration

By default the start epoch and file duration are used to create the epoch parts of the output name. To force the automatic file naming to a distinct type **::RX2::** can be extended by the letters **L**, **S** or **D** (**L**ong, **S**hort, **D**ay) to **::RX2L::**, **::RX2S::** or **::RX2D::**.
The following examples illustrate the standard behavior for a station **ABCD** with start epoch **2015-123 03:05** and different durations.

Duration	$<$ 1 hour	1 hour	$>$ 1 hour
::RX2::	abcd122d05.15o	abcd122d.15o	abcd1220.15o
::RX2L::	abcd122d05.15o	abcd122d05.15o	abcd122d05.15o
::RX2S::	abcd122d.15o	abcd122d.15o	abcd122d.15o
::RX2D::	abcd1220.15o	abcd1220.15o	abcd1220.15o

The cases ::RX2L::, ::RX2S:: allow to store not only hourly or sub-hourly files. For durations larger than 1 hour one can use it to store sub-daily files too. In this case the file epoch indicates the start time (hour, minute) only.

2.3.3.5 RINEX-3 Start Epoch/Duration (real)

For the RINEX-3 file renaming the following rules are valid for all observation types (O/N/M). The example obs. files in the table below with the following characteristics are used to illustrate the (re)naming process.

Characteristics	**pots0070.15o**	**pots007c.15o**	**pots007c30.15o**
Time Begin	01:12:30	02:13:30	02:33:13
Time End	23:59:30	02:55:30	02:44:50
Duration (implicit)	1 day	1 hour	unknown
Duration (nominal)	1 day	1 hour	15 min
Duration (real hh:mm:ss)	22:47:00	00:42:00	00:11:37
Sampling Rate	30s	30s	1s

Using the following basic command you will get file names containing the real values derived from the file content.

```
gfzrnx -finp <RINEX-2 Name> -fout ::RX3::01,DEU
```

By default the real begin epoch and duration information based on the file content are used:

RINEX-2	**RINEX-3**
pots0070.15o	POTS00DEU_R_20150070112_23H_30S_MO.rnx
pots007c.15o	POTS00DEU_R_20150070213_42M_30S_MO.rnx
pots007c30.15o	POTS00DEU_R_20150070233_12M_01S_MO.rnx

2.3.3.6 RINEX-3 Start Epoch/Duration (nominal)

To get, similar to the RINEX-2 file naming, **nominal** begin and duration information in the RINEX-3 file name additional command line parameters are needed.

The general method is to give the begin epoch and the duration information via the **-epo_beg** and **-d** command line parameters.

```
gfzrnx -finp file.rnx     -fout ::RX3::ABCD,05,DEU -epo_beg 20150812_020000 -d 3600
gfzrnx -finp pots0070.15o -fout ::RX3::00,DEU      -epo_beg 20150107_000000 -d 86400
```

Assuming 30 s sampling rate and GPS only data, the output file names will be:
ABCD05DEU_R_20152240200_01H_30S_GO.rnx,
POTS00DEU_R_20150070000_01D_30S_GO.rnx.

In the case of **nominal** standard RINEX input file names you can get nominal RINEX-3 output file names providing the **-sei in** command line parameter (strict epoch interval), which uses the epoch and implicit duration information from the **in**put file name. If no implicit duration information is given (RINEX-2 11.3 file names) it has to be provided in addition via the **-d** (duration) command line parameter (otherwise the real duration is used).
This can be useful in renaming scenarios.

RINEX-2	command line parameters	RINEX-3
pots0070.15o	-sei in	POTS00DEU_R_20150070000_01D_30S_MO.rnx
pots007c.15o	-sei in	POTS00DEU_R_20150070200_01H_30S_MO.rnx
pots007c30.15o	-sei in -d 900	POTS00DEU_R_20150070230_15M_01S_MO.rnx

2.3.3.7 Remark

In the file **split mode** the duration information will be nominal (split interval).
The **nominal** mode has to be used with caution, especially in renaming operations.

CAUTION !
Using the NOMINAL mode gfzrnx does not only (re)name the given output files. **It ensures, that the file content fits to the file name.** **This way extra observations are removed !**

For navigation files this nominal interval can be extended via the **-enb** command line parameter (extend navigation boundaries). See the **Operation/Tasks - Rinex File Epoch Interval** section.

Chapter 3

Operation / Tasks

To get the full available checks via data input one has to use the -**chk** option, to make sure that the output data are formally correct. If you are sure, that your files are correct and you want to do some data manipulation only you can omit this commandline parameter to speed up the work.
Please keep in mind, that compared to other tools, which work on a single epoch level, **gfzrnx** stores the whole RINEX data set in the computers memory before output. This leads to some performance degradation but offers complete data handling opportunities.

The standard output format of **gfzrnx** is **RINEX-3** !

3.1 RINEX File Check and Repair

If one gets data of unknown quality one should pass them at least once through a check procedure. If an output file is created it will be RINEX conform nevertheless the input was corrupt.

With **gfzrnx** this can be done via:

```
gfzrnx -finp pots0070.15o -fout pots0070.15o_chk -chk -kv
```

with -**chk** all formal checks are done on the input file.
The -**kv** (keep version) ensures the same output version as the input file (standard output format is RINEX-3).
The following modifications are done in the output file:

- Statistical information are added or updated in the file header.
  - PRN / ♯ OF OBS
  - ♯ OF SATELLITES
  - INTERVAL
  - TIME OF FIRST OBS
  - TIME OF LAST OBS
- overall empty observation types are removed

Here is an example of an updated RINEX header information:

```
C   10 C1I C6I C7I D1I L1I L6I L7I S1I S6I S7I              SYS / # / OBS TYPES
E   13 C1X C5X C7X C8X D1X L1X L5X L7X L8X S1X S5X S7X S8X  SYS / # / OBS TYPES
G   20 C1C C2W C2X C5X D1C D1P D1W D2W L1C L1P L1W L2W L2X  SYS / # / OBS TYPES
       L5X S1C S1W S2C S2W S2X S5X                          SYS / # / OBS TYPES
J   19 C1C C1X C1Z C2X C5X C6L D1C L1C L1X L1Z L2X L5X L6L  SYS / # / OBS TYPES
       S1C S1X S1Z S2X S5X S6L                              SYS / # / OBS TYPES
R   13 C1C C1P C2C C2P D1C L1C L1P L2C L2P S1C S1P S2C S2P  SYS / # / OBS TYPES
S    4 C1C D1C L1C S1C                                      SYS / # / OBS TYPES
    76                                                      # OF SATELLITES
```

```
    C01  2863  2863  2863  2863  2863  2863  2863  2863  2863PRN / # OF OBS
         2863                                                  PRN / # OF OBS
...
    C14  1365  1363  1363  1365  1365  1363  1363  1365  1363PRN / # OF OBS
         1363                                                  PRN / # OF OBS
    E11   900   895   893   899   900   900   895   893   899PRN / # OF OBS
          900   895   893   899                               PRN / # OF OBS
    E19  1605  1601  1601  1603  1605  1605  1601  1601  1603PRN / # OF OBS
         1605  1601  1601  1603                               PRN / # OF OBS
    G01  1189  1148  1181  1181  1189                        1189PRN / # OF OBS
                     1148  1181  1181  1189        1181  1148PRN / # OF OBS
         1181  1181                                           PRN / # OF OBS
...
    G32  1247  1241              1247                        1247PRN / # OF OBS
                     1241              1247                  1241PRN / # OF OBS
                                                              PRN / # OF OBS
    J01  2863  2863  2863  2863  2863  2863  2863  2863  2863PRN / # OF OBS
         2863  2863  2863  2863  2863  2863  2863  2863  2863PRN / # OF OBS
         2863                                                 PRN / # OF OBS
    R01   713   713   709   706   713   713   713   709   706PRN / # OF OBS
          713   713   709   706                               PRN / # OF OBS
...
    R24   695   695   695   695   695   695   695   695   695PRN / # OF OBS
          695   695   695   695                               PRN / # OF OBS
    S26  1973  1973  1973  1973                               PRN / # OF OBS
...
    S37  2863  2863  2863  2863                               PRN / # OF OBS
     30.000                                                   INTERVAL
   2014     8    17     0     0    0.0000000     GPS         TIME OF FIRST OBS
   2014     8    17    23    59   30.0000000     GPS         TIME OF LAST OBS
...
```

The repair of a file file is different concerning RINEX-2 and RINEX-3. Data values are not corrected ! Via the repair operation formally corrupt observation parts are omitted only.

- RINEX-2
  1. A complete epoch block is removed in case of corrupted data detection.
- RINEX-3
  1. A complete satellite block (line) is removed in case of corrupted data detection.

3.2 RINEX File Statistics / Informations

3.2.1 Observations Statistics

The **-stk_only** or **-stk_obs** outputs an observations statistics information to **STDOUT**. Only the nonzero (nonempty) data values are counted.

```
gfzrnx -finp pots0070.15o -stk_obs
```

you can store it into a file using the **-fout** command line parameter.

```
gfzrnx -finp pots0070.15o -stk_obs -fout pots0070.15o_stk
```

Here is an example for the observations file sin12290.14o:

```
gfzrnx -finp sin12290.14o -stk_only

 STP sin1 C TYP   C1I   C6I   C7I   D1I   L1I   L6I   L7I   S1I   S6I   S7I
```

```
 STO sin1 C C01  2863  2863  2863  2863  2863  2863  2863  2863  2863  2863
 STO sin1 C C02  2863  2863  2863  2863  2863  2863  2863  2863  2863  2863
...
 STO sin1 C C14  1365  1363  1363  1365  1365  1363  1363  1365  1363  1363

 STP sin1 E TYP   C1X   C5X   C7X   C8X   D1X   L1X   L5X   L7X   L8X   S1X   S5X   S7X   S8X
 STO sin1 E E11   900   895   893   899   900   900   895   893   899   900   895   893   899
 STO sin1 E E12  1230  1230  1230  1230  1230  1230  1230  1230  1230  1230  1230  1230  1230
 STO sin1 E E19  1605  1601  1601  1603  1605  1605  1601  1601  1603  1605  1601  1601  1603

 STP sin1 G TYP   C1C   C2W   C2X   C5X   D1C   ...   L1C   L1P   L1W   L2W   L2X   L5X   S1C ...

 STO sin1 G G01  1189  1148  1181  1181  1189   ...  1189     0     0  1148  1181  1181  1189 ...

...
 STO sin1 G G10   886   881     0     0   886   ...   886     9     9   881     0     0   886 ...

...
 STO sin1 G G32  1247  1241     0     0  1247   ...  1247     0     0  1241     0     0  1247 ...


 STP sin1 J TYP   C1C   C1X   C1Z   C2X   C5X   C6L   D1C   L1C   L1X   L1Z   L2X   L5X   L6L ...

 STO sin1 J J01  2863  2863  2863  2863  2863  2863  2863  2863  2863  2863  2863  2863  2863 ...


 STP sin1 R TYP   C1C   C1P   C2C   C2P   D1C   L1C   L1P   L2C   L2P   S1C   S1P   S2C   S2P
 STO sin1 R R01   713   713   709   706   713   713   713   709   706   713   713   709   706
 STO sin1 R R02  1143  1143  1141  1141  1143  1143  1143  1141  1141  1143  1143  1141  1141
...
 STO sin1 R R24   695   695   695   695   695   695   695   695   695   695   695   695   695

 STO sin1 S TYP   C1C   D1C   L1C   S1C
 STO sin1 S S26  1973  1973  1973  1973
 STO sin1 S S27  2863  2863  2863  2863
...
 STO sin1 S S37  2863  2863  2863  2863
```

3.2.2 ASCII Timeplot of Observables

The **-stk_epo** command line parameter can be used to create an ASCII timeplot to show the availability of observations per **PRN** (std.) and/or **observation type**.
In the simplest mode one has to provide the time bin to be used in seconds (here 1800).

3.2.2.1 Timeplot per PRN

```
rnxall -finp stas0400.15o -stk_epo 1800
rnxall -finp stas0400.15o -stk_epo 1800:prn
```

```
 STT 20150209 00:00   04:00   08:00   12:00   16:00   20:00   00:00
 STH            +---+---+---+---+---+---+---+---+---+---+---+---+
 STE stas C C05 ************************************************| C05
 STE stas C C06 **********  |   |   |   |   |   |   |   |   ****| C06
 STE stas C C07 |   |   |   **********  |   |   |   |   |   |   | C07
 STE stas C C08 |   |   |   |   |   |   |   ********** |   |   | C08
 STE stas C C09 *****************   |   |   |   |   |   |   | **| C09
 STE stas C C10 |   |   |   | ******************|   |   |   |   | C10
 STE stas C C11 ****|   |   |   | ******|   |   |   | ******| C11
 STE stas C C12 |   |   |   |   | ********  |   |   |  *********| C12
```

```
 STE stas C C14 **********  |   |   |   |   | ********* |   |   | C14
 STS            |---|---|---|---|---|---|---|---|---|---|---|---|
 STE stas E E11 ******* |   |   |   |   | ******|   |   |   |   | E11
 STE stas E E12 *** |   |   |   |   |  *******  |   |   |   |***| E12
 STE stas E E19 *****   |   |   |   |   | ************* |   |   | E19
 STE stas E E20 **  |   |   |   |   | ***************   |   |   | E20
 STS            |---|---|---|---|---|---|---|---|---|---|---|---|
 STE stas G G01 |   |  **************   |   |   | ***** |   |   | G01
 STE stas G G02 *   |   |   |   | ***********   |   |  *********| G02
 STE stas G G03 |   |   | ***************|   |   |   |  *****|   | G03
...
 STE stas G G30 |   ******  |   |   | ************  |   |   |   | G30
 STE stas G G31 *** |   |   | ********  |   |   |   |   ********| G31
 STE stas G G32 |   |   **************   |   |   |   |****** |   | G32
 STS            |---|---|---|---|---|---|---|---|---|---|---|---|
 STE stas J J01 **  |   |   |   |   |   |   |   |   |   |  *** *| J01
 STS            |---|---|---|---|---|---|---|---|---|---|---|---|
 STE stas R R01 *****   |   |   |   |************|   |   |   ****| R01
 STE stas R R02 ********|   |   |   |   *********** |   |   | **| R02
 STE stas R R03 | ********* |   |   |   |   **********  |   |   | R03
...
 STE stas R R22 |   |   |   **********  |   |   ********|   |   | R22
 STE stas R R23 |   |   |   |  *********|   |   | ********* |   | R23
 STE stas R R24 **  |   |   |   | ********  |   |   |*********  | R24
 STH            +---+---+---+---+---+---+---+---+---+---+---+---+
 STT 20150209 00:00   04:00   08:00   12:00   16:00   20:00   00:00
```

3.2.2.2 Timeplot per PRN and/or Observation Type

A timeplot per observation type is available providing the **[:[prn/otp]]** parameter list.
This can be combined with other parameters like -**smp**, -**satsys**, –**obs_types**, -**prn**, -**no_prn** etc.

```
rnxall -finp stas0400.15o -stk_epo 1800:prn,otp -satsys E -ot C,L
```

```
 STT 20150209     00:00   04:00   08:00   12:00   16:00   20:00   00:00
 STH               +---+---+---+---+---+---+---+---+---+---+---+---+
 STE stas E E11     ******* |   |   |   |   | ******|   |   |   |   |     E11
 SOT stas E E11 C1X xxxxxxx |   |   |   |   | xxxxxx|   |   |   |   | C1X E11
 SOT stas E E11 C7X xxxxxxx |   |   |   |   | xxxxxx|   |   |   |   | C7X E11
 SOT stas E E11 C8X xxxxxxx |   |   |   |   | xxxxxx|   |   |   |   | C8X E11
 SOT stas E E11 L1X xxxxxxx |   |   |   |   | xxxxxx|   |   |   |   | L1X E11
 SOT stas E E11 L7X xxxxxxx |   |   |   |   | xxxxxx|   |   |   |   | L7X E11
 SOT stas E E11 L8X xxxxxxx |   |   |   |   | xxxxxx|   |   |   |   | L8X E11
 STE stas E E12     *** |   |   |   |   |  *******  |   |   |   |***|     E12
 SOT stas E E12 C1X xxx |   |   |   |   |  xxxxxxx  |   |   |   |xxx| C1X E12
 SOT stas E E12 C7X xxx |   |   |   |   |  xxxxxxx  |   |   |   |xxx| C7X E12
 SOT stas E E12 C8X xxx |   |   |   |   |  xxxxxxx  |   |   |   |xxx| C8X E12
 SOT stas E E12 L1X xxx |   |   |   |   |  xxxxxxx  |   |   |   |xxx| L1X E12
 SOT stas E E12 L7X xxx |   |   |   |   |  xxxxxxx  |   |   |   |xxx| L7X E12
 SOT stas E E12 L8X xxx |   |   |   |   |  xxxxxxx  |   |   |   |xxx| L8X E12
 STE stas E E19     *****   |   |   |   |   | ************* |   |   |     E19
 SOT stas E E19 C1X xxxxx   |   |   |   |   | xxxxxxxxxxxxx |   |   | C1X E19
 SOT stas E E19 C7X xxxxx   |   |   |   |   | xxxxxxxxxxxxx |   |   | C7X E19
 SOT stas E E19 C8X xxxx    |   |   |   |   |  xxxxxxxxxxx  |   |   | C8X E19
 SOT stas E E19 L1X xxxxx   |   |   |   |   | xxxxxxxxxxxxx |   |   | L1X E19
 SOT stas E E19 L7X xxxxx   |   |   |   |   | xxxxxxxxxxxxx |   |   | L7X E19
 SOT stas E E19 L8X xxxxx   |   |   |   |   | xxxxxxxxxxxxx |   |   | L8X E19
 STE stas E E20     **  |   |   |   |   | ***************   |   |   |     E20
 SOT stas E E20 C1X xx  |   |   |   |   | xxxxxxxxxxxxx     |   |   | C1X E20
 SOT stas E E20 L1X xx  |   |   |   |   | xxxxxxxxxxxxxxx   |   |   | L1X E20
```

```
STH                +---+---+---+---+---+---+---+---+---+---+---+---+
STT 20150209      00:00   04:00   08:00   12:00   16:00   20:00   00:00
```

Using an Editor, which is able to scroll horizontally through a text file (**nedit** for Unix, or **Notepad++** for MS Windows) one can check visually data availability details down to the single observation in case of problems.
Here an example of an input file with 5 s sampling rate:

```
gfzrnx -finp stas0010.15o -stk_epo 5:prn,otp -fout xxxx
```

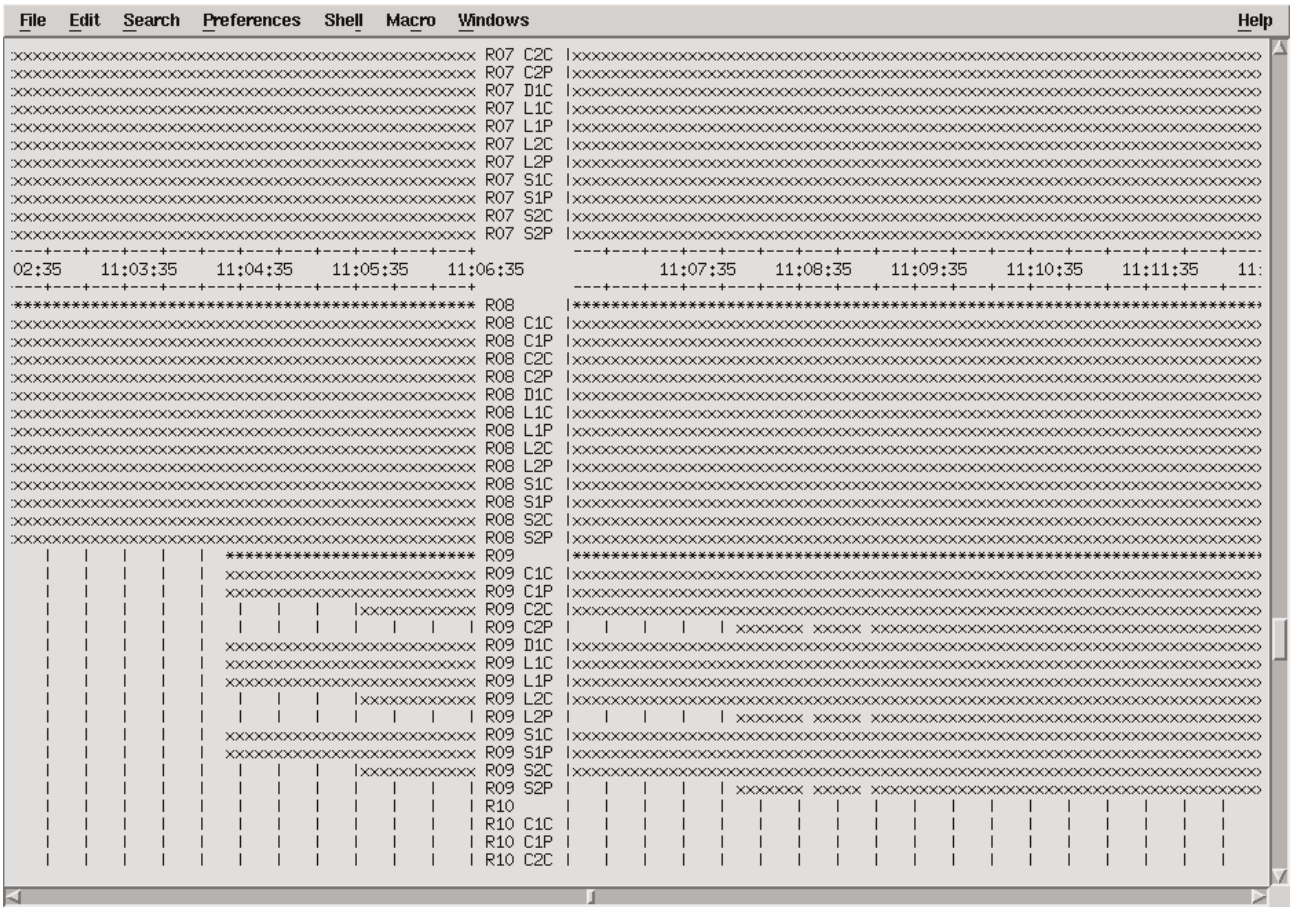

Figure 3.1: Editor Window - ASCII Timeplot per PRN and Observation Type

3.3 RINEX File Format Conversion (3/2, 2/3)

3.3.1 Observation Types Mapping

The used observation types mapping is hardcoded in **gfzrnx**. It can be shown up via the following command.

```
gfzrnx -out_obs_map
gfzrnx -out_obs_map -fout obs_types_map.txt
```

The information in the columns 2,3,4 are treated as comment only and are not used.

3.3.2 REMARK

During the conversion process the data values (observation, loss of lock indicator(LLI), signal strength) are left as they are. The LLI meaning differs between version 2 and 3 and the Interpretation of bit 1 and 2 has to be used with caution !

3.3.3 RINEX-2 to RINEX-3

Please use this conversion only if you are sure, that the output files are usable in the environment to which the data are supplied !
The output format for this conversion/transition is RINEX-3.01 to be standard conform.
The 2-char. observation types are kept as they are except the code observations for GPS and GLONASS (see below).
As **RINEX-3** is the standard output format of **gfzrnx** simply run:

```
gfzrnx -finp pots0070.15o -fout pots0070.15o_rx3
```

or

```
gfzrnx -finp pots0070.15o -fout ::RX3::00,DEU
gfzrnx -finp pots0070.15o -fout ::RX3::DEU,00
```

or

```
gfzrnx -finp pots0070.15o -fout ::RX3::00,DEU -sei in
gfzrnx -finp pots0070.15o -fout ::RX3::DEU,00 -sei in
```

to create a RINEX-3 conform output file name **POTS00DEU_R_201500700_01D_30S_MO.rnx** .
For naming details see the **Automatic Output File Naming** section.

A hard coded observation types mapping for the GPS and GLONASS **code observations** is implemented:

System	RINEX-2	RINEX-3
G	P1	C1W
G	C1	C1C
G	P2	C2W
G	C2	C2C

System	RINEX-2	RINEX-3
R	P1	C1P
R	C1	C1C
R	P2	C2P
R	C2	C2C

This is used, because both **Px** and **Cx** RINEX-2 code types are mapped to the single **Cx?** code type.

3.3.4 RINEX-3 to RINEX-2

The RINEX-2 output version is 2.11 .
Use the **–version_out** or **-vo** command line parameter to define RINEX format version of the output file.

```
gfzrnx -finp pots0070.15o                          -fout pots0070.15o_rx2 -vo 2
gfzrnx -finp POTS00DEU_R_201500700_01D_30S_MO.rnx -fout pots0070.15o     --version_out 2
```

3.3.4.1 Specific Observation Type Selection

In the RINEX-3 format one can have multiple observation types per data type and frequency (tracking mode or channel attribute). For a specific observation type selection for the format conversion you can use the observation

types selection feature in addition. Add the **-ot** command line parameter to the upper command like in the example below to select the RINEX-3 obs. types to be converted and to get a distinct conversion.

```
-ot G:C1W,L1W,D1W,S1W+C2W,L2W,D2W,S2W+R:C1P,L1P,S1P,D1P,C2P,L2P,S2P,D2P
```

3.3.4.2 Observation Type Selection via Signal Priorities

By default the following signal priorities per frequency and satellite system are used for the RINEX-3 to RINEX-2 conversion:

Sat. System	Freq. Num.	RINEX-3 Signal Priority
G - GPS	1	PWCSLXYMN
G - GPS	2	PWCSLXYMN
G - GPS	5	IQX
.		
R - GLO	1	PC
R - GLO	2	PC
R - GLO	3	IQX
.		
E - GAL	1	BCX
E - GAL	5	IQX
E - GAL	6	BCX
E - GAL	7	IQX
E - GAL	8	IQX
.		
J - QZS	1	SLXCZ
J - QZS	2	SLX
J - QZS	5	IQX
J - QZS	6	SLX
.		
C - BDS	1	IQX
C - BDS	6	IQX
C - BDS	7	IQX
.		
I - IRN	5	ABCX
I - IRN	9	ABCX
.		
S - SBS	1	C
S - SBS	5	IQX

The observation codes priority is **LCDS** (phase, code, doppler, signal strength). It defines the basis for the selection of the other obs. types of that frequency if existing. You can update the internal signal priority list providing update records via the **-pr3rx2** command line parameter. According to the upper table it should consist of a comma separated

list of a satellite system identifier , colon, frequency number, colon and the signal priority string. Observation types not covered by the priority string are simply ignored via conversion. See the following example.

```
-pr3rx2 G:5:QXI,I:59:CXAB
```

The same priority string per satellite system for different frequencies can be given combined.

3.3.4.3 Used Observation Types

The observation types per satellite system used for the format conversion can be found as **COMMENT**s in the RINEX file header.

```
*********************************************************  COMMENT
*             WARNING - FORMAT CONVERSION               *  COMMENT
*  -----------------------------------------------------  *  COMMENT
*  The data values (observation, loss of lock indicator  *  COMMENT
*      (LLI), signal strength) are left as they are.     *  COMMENT
*  The LLI meaning differs between versions 2 and 3 and  *  COMMENT
*    the Interpretation of bit 1 and 2 has to be used    *  COMMENT
*                  with caution !!!                     *  COMMENT
*********************************************************  COMMENT
                                                           COMMENT
RINEX 3 -> 2 TYPE CONVERSION DETAILS:                      COMMENT
-------------------------------------                      COMMENT
   C C1I -> C1                                             COMMENT
   C C6I -> C6                                             COMMENT
   C C7I -> C7                                             COMMENT
   C D1I -> D1                                             COMMENT
   C L1I -> L1                                             COMMENT
   C L6I -> L6                                             COMMENT
   C L7I -> L7                                             COMMENT
   C S1I -> S1                                             COMMENT
   C S6I -> S6                                             COMMENT
   C S7I -> S7                                             COMMENT
-------------------------------------                      COMMENT
   E C1X -> C1                                             COMMENT
   E C5X -> C5                                             COMMENT
   E C7X -> C7                                             COMMENT
   E C8X -> C8                                             COMMENT
   E D1X -> D1                                             COMMENT
   E L1X -> L1                                             COMMENT
   E L5X -> L5                                             COMMENT
   E L7X -> L7                                             COMMENT
   E L8X -> L8                                             COMMENT
   E S1X -> S1                                             COMMENT
   E S5X -> S5                                             COMMENT
   E S7X -> S7                                             COMMENT
   E S8X -> S8                                             COMMENT
-------------------------------------                      COMMENT
   G C1C -> C1                                             COMMENT
   G C2X -> C2                                             COMMENT
   G C5X -> C5                                             COMMENT
   G D1C -> D1                                             COMMENT
   G L1C -> L1                                             COMMENT
   G L2W -> L2                                             COMMENT
   G L5X -> L5                                             COMMENT
   G C2W -> P2                                             COMMENT
   G S1C -> S1                                             COMMENT
   G S2W -> S2                                             COMMENT
   G S5X -> S5                                             COMMENT
...
```

3.3.4.4 Remark

To avoid the selection of an obs. type with sparse observations using **Signal Priorities** mode it can be useful to add the **-rsot** command line parameter (remove sparse observations obs. types) in addition.

```
gfzrnx -finp pots0070.15o                          -fout pots0070.15o_rx2 -vo 2 -rsot 40
gfzrnx -finp POTS00DEU_R_201500700_01D_30S_MO.rnx -fout pots0070.15o     -vo 2 -rsot 40
```

3.4 RINEX File Nominal Renaming Support (2/3)

A fast file name conversion of RINEX-3 files with RINEX-2 style file names to RINEX-3 style file names is supported. It can be used without reading the input files using all necessary information from the RINEX-2 style file name and from information provided via command line parameters (useful for compressed files).
For uncompressed observation files, including hatanaka compressed files, some required information can also be derived from the file header.
The supported RINEX-2 style file names are:

Name	Example	Description
nnnnddd0.yyt	pots1230.15o	daily obs. file
	pots1230.15d	daily obs. file (hatanaka compressed)
nnnnddd[a-z].yyt.	pots123a.15n	hourly nav. file
nnnnddd[a-z]mm.yyt	pots123x15.15m	sub-hourly met. file

The renaming support can be invoked via the **-nomren23** (nominal rename) command line parameter. The output is the RINEX-3 file name (printed to STDOUT) which can be used for renaming operations. The input can be a full path, the output is the file name only.

```
gfzrnx -finp           pots1230.15n -nomren23
POTS00XXX_R_20151230000_01D_GN.rnx

gfzrnx -finp /tmp/data/pots1230.15n -nomren23
POTS00XXX_R_20151230000_01D_GN.rnx
```

Using **-nomren23** command line parameter the following additional information **s,mr,iso** has to be be provided via command line, because they are not available from the RINEX-2 style file name or RINEX file header.

	Information	Values	Default
s	data source	R or S	R
mr	marker/receiver number	mr	00
iso	iso country code	ISO	XXX

```
gfzrnx -finp pots1230.15n -nomren23 DEU,12
POTS12DEU_R_20151230000_01D_GN.rnx

gfzrnx -finp pots1230.15g -nomren23 S,DEU,12
POTS12DEU_S_20151230000_01D_RN.rnx

gfzrnx -finp pots1230.15m -nomren23 DEU
POTS00DEU_R_20151230000_01D_00U_MM.rnx
```

Via the **-4to9** command line parameter one can provide multiple site identifier information from a provided configuration file. See the **Automatic Output File Naming** section for details on **-4to9**.

```
gfzrnx -finp pots1230.15o -nomren23 -4to9 four2nine.conf
gfzrnx -finp tash1230.15o -nomren23 -4to9 four2nine.conf
```

Meteo- and Navigation files don't have additional information which can be derived from the file header.
For observation files the data frequency and satellite system can be derived from the **"INTERVAL"** and **"SYS / # / OBS TYPES"** RINEX header records. For compressed files this information can be provided via the command line parameters -**smp** and -**satsys**.
Here some examples, including hatanaka compressed files:

```
gfzrnx -finp pots1230.15o.gz -nomren23 DEU -smp 30 -satsys G
POTS00DEU_R_20151230000_01D_30S_GO.rnx.gz

gfzrnx -finp pots1230.15o.gz -nomren23 DEU -smp 30 -satsys GR
POTS00DEU_R_20151230000_01D_30S_MO.rnx.gz

gfzrnx -finp pots1230.15d.gz -nomren23 DEU -smp 30 -satsys GR
POTS00DEU_R_20151230000_01D_30S_MO.crx.gz

gfzrnx -finp pots1230.15d.gz -nomren23 DEU
POTS00DEU_R_20151230000_01D_00U_MO.crx.gz
```

Using the following RINEX-3 header information:

```
E    6 C1X C5X L1X L5X S1X S5X                           SYS / # / OBS TYPES
G    8 C1C C1P C2C C2P L1P L2P S1P S2P                   SYS / # / OBS TYPES
R    8 C1C C1P C2C C2P L1P L2P S1P S2P                   SYS / # / OBS TYPES
    10.000                                               INTERVAL
```

leads to the following file names:

```
gfzrnx -finp pots1230.15o -nomren23 DEU
POTS00DEU_R_20151230000_01D_10S_MO.rnx

gfzrnx -finp pots1230.15d -nomren23 DEU
POTS00DEU_R_20151230000_01D_10S_MO.crx
```

A single satellite system file with the following information:

```
E    6 C1X C5X L1X L5X S1X S5X                           SYS / # / OBS TYPES
     5.000                                               INTERVAL
```

leads to the file names:

```
gfzrnx -finp pots1230.15o -nomren23 DEU
POTS00DEU_R_20151230000_01D_05S_EO.rnx

gfzrnx -finp pots1230.15d -nomren23 DEU
POTS00DEU_R_20151230000_01D_05S_EO.crx
```

Sub-daily files need the additional duration information if it is not 15 minutes (std.). It can be given via the -**d**, -**duration** command line parameter.

```
gfzrnx -finp pots123b30.15o -nomren23 DEU
POTS00DEU_R_20151230130_15m_01S_MO.rnx
```

```
gfzrnx -finp pots1230c35.15o.gz -nomren23 DEU -d 300 -smp 5
POTS00DEU_R_20151230235_05M_05S_MO.rnx.gz
```

3.4.1 Remark

Information provided via command line has priority.

3.5 RINEX File Splice

For the RINEX file splicing one can give an unsorted list of input files of a single station. The observation types order can also differ from input file to input file and an observation type order change inside of a single file is also taken into account.
Simply provide a list of input files and the output file:

```
gfzrnx -finp pots007b.14o pots007a.14o ... pots007x.14o -fout pots0070.14o      -kv
```

For **bash** command shell it can be shortened using filename expansion options.

```
gfzrnx -finp      pots007{a..x}.14o -fout pots0070.14o      -kv
gfzrnx -finp /tmp/pots007{a..x}.14o -fout /tmp/pots0070.14o -kv
```

For **csh** command shell it is:

```
gfzrnx -finp      pots007[a-x].14o -fout pots0070.14o      -kv
gfzrnx -finp /tmp/pots007[a-x].14o -fout /tmp/pots0070.14o -kv
```

For windows-users in **cmd.exe** or **powershell.exe** it is:

```
gfzrnx -finp        pots007[a-x].14o -fout pots0070.14o        -kv
gfzrnx -finp c:\tmp\pots007[a-x].14o -fout c:\tmp\pots0070.14o -kv
```

This works similar for navigation and meteo files.

```
gfzrnx -finp      pots007[a-x].14m -fout /tmp/pots0070.14m --version_out 2
gfzrnx -finp /tmp/pots007[a-x]     -fout /tmp/brds0070.14n --version_out 3
```

3.5.1 Remark - Splice/Split

It is possible to combine the **splice** and **split** operation of **observation data** via a single command line call. Here an example splicing e.g. 15 min. input files and split to hourly files keeping the version in output.

```
gfzrnx -finp pots007[a-x]??.14o -fout /tmp/::RX2:: -kv -split 3600
```

This can be additionally combined with data sampling, satellite system- and observation type selection etc..

3.5.2 Remark - Filename Expansion

3.5.2.1 UNIX

On UNIX systems the file name expansion is usually done by the calling command shell. Please adopt the filename expansion options like "?", "*", "[]", etc. to your used command shell. The "[a-x]" can be also e.g. an a..x in another command shell.

3.5.2.2 MS Windows

MS Windows does not support the file name expansion in its command line interfaces. Therefore this is done within **gfzrnx**. Only "?", "*", "[]" are supported here.

3.6 RINEX File Split

The RINEX file split can be initiated providing a split interval in seconds via **-split** command line parameter. For the output file the automatic file naming **::RX2/3::** is mandatory.
The following command:

```
gfzrnx -finp pots0070.15o -fout /tmp/::RX2:: -split 3600 -kv
```

will split a daily file into hourly files keeping the input file RINEX version and using the RINEX-2 file naming.

```
pots007a.15o pots007b.15o pots007c.15o pots007d.15o pots007e.15o pots007f.15o
pots007g.15o pots007h.15o pots007i.15o pots007j.15o pots007k.15o pots007l.15o
pots007m.15o pots007n.15o pots007o.15o pots007p.15o pots007q.15o pots007r.15o
pots007s.15o pots007t.15o pots007u.15o pots007v.15o pots007w.15o pots007x.15o
```

The following command:

```
gfzrnx -finp pots0070.15o -fout /tmp/::RX3::00,DEU -split 3600
```

will split a daily file into RINEX-3 hourly files using the RINEX-3 file naming.

```
/tmp/POTS00DEU_R_20150070000_01H_30S_MO.rnx /tmp/POTS00DEU_R_20150070100_01H_30S_MO.rnx
/tmp/POTS00DEU_R_20150070200_01H_30S_MO.rnx /tmp/POTS00DEU_R_20150070300_01H_30S_MO.rnx
/tmp/POTS00DEU_R_20150070400_01H_30S_MO.rnx /tmp/POTS00DEU_R_20150070500_01H_30S_MO.rnx
/tmp/POTS00DEU_R_20150070600_01H_30S_MO.rnx /tmp/POTS00DEU_R_20150070700_01H_30S_MO.rnx
/tmp/POTS00DEU_R_20150070800_01H_30S_MO.rnx /tmp/POTS00DEU_R_20150070900_01H_30S_MO.rnx
/tmp/POTS00DEU_R_20150071000_01H_30S_MO.rnx /tmp/POTS00DEU_R_20150071100_01H_30S_MO.rnx
/tmp/POTS00DEU_R_20150071200_01H_30S_MO.rnx /tmp/POTS00DEU_R_20150071300_01H_30S_MO.rnx
/tmp/POTS00DEU_R_20150071400_01H_30S_MO.rnx /tmp/POTS00DEU_R_20150071500_01H_30S_MO.rnx
/tmp/POTS00DEU_R_20150071600_01H_30S_MO.rnx /tmp/POTS00DEU_R_20150071700_01H_30S_MO.rnx
/tmp/POTS00DEU_R_20150071800_01H_30S_MO.rnx /tmp/POTS00DEU_R_20150071900_01H_30S_MO.rnx
/tmp/POTS00DEU_R_20150072000_01H_30S_MO.rnx /tmp/POTS00DEU_R_20150072100_01H_30S_MO.rnx
/tmp/POTS00DEU_R_20150072200_01H_30S_MO.rnx /tmp/POTS00DEU_R_20150072300_01H_30S_MO.rnx
```

3.6.1 Remark - Split/Splice

It is possible to combine the **split** with a **splice** operation of observation data. See splice section for details.

3.7 RINEX File Output Epoch Interval

3.7.1 Supported Date/Time/Epoch Formats

3.7.1.1 Date

Date Type	Abbreviation	Example
MJD	MJD	56753
GPSweekWeekday	WWWWD	17870
YearDayofyear	YYYYDDD	2014096
YearMonthDay	YYYYMMDD	20140406
Year-Month-Day	YYYY-MM-DD	2014-04-06

3.7.1.2 Time

Time Type	Abbreviation	Example
HourMinuteSecond	HHMMSS	123000
Hour:Minute:Second	HH:MM:SS	12:30:00

3.7.1.3 Epoch

An Epoch string can be formed connecting any Date-string via '_' with a Time-string.

Date Type	Example
MJD	56753_123000
GPSweekWeekday	17870_12:30:00
YearDayofyear	2014096_123000
YearMonthDay	20140406_12:30:00
Year-Month-Day	2014-04-06_123000

3.7.2 Dedicated Output Epoch Interval

To extract a dedicated epoch interval from a RINEX-file you have to provide a Start-Epoch via **-epo_beg** and the Duration **-d** or **-duration** in seconds.
Here an example to extract the first hour of a daily input file.

```
gfzrnx -finp pots0070.15o -fout pots007a.15o -epo_beg 2015-01-07_000000   -d 3600
gfzrnx -finp pots0070.15o -fout pots007a.15o -epo_beg    2015007_00:00:00 -d 3600
gfzrnx -finp pots0070.15o -fout pots007a.15o -epo_beg   20150107_000000   -d 3600
```

3.7.3 Strict Epoch interval (-sei)

If you want, that your output epoch interval strictly follows a RINEX file naming, you can give the **-sei** command line parameter to omit all data, which don't fit to the implicitly given epoch interval of your input or output file name. You have to use the parameters **in,out** to the **-sei** switch to indicate if either the input- or the output filename has to be used for the strict epoch interval handling.

```
gfzrnx -finp pots0070.15o -fout pots007a.15o_chk -chk    -sei in
gfzrnx -finp pots0070.15o -fout pots007a.15o_smp -smp 30 -sei out
```

The last example extracts the first hour from the daily input file including a data sampling operation.

3.7.4 Extend Navigation File Boundaries (-enb)

Navigation information files contain often records which don't correspond to the nominal time interval given via the in/out file names. To avoid the elimination of data extending the nominal time interval one can extend the interval to be checked via the **-enb** command line parameter.The check time interval will be extended at both boundaries by the number of seconds given. Choose a reasonable value to ensure the quality of the output file.

```
gfzrnx -finp grac182n.15f -fout ::RX3::FRA -f -sei in -enb 86400
```

3.8 RINEX File Manipulation

The following manipulations are useful mainly to shrink an input file to a size and content really needed for the analysis purpose. All these manipulations can be combined with the other described operations.

3.8.1 Data Sampling (-smp)

Provide the sampling rate [sec] and the optional tolerance range [sec] to link an observation epoch to its nominal epoch via **-smp** command line parameter. This parameter can be given for any **gfzrnx** operation.

```
-smp num[:eps]
```

For observation data the default tolerance range (eps) is 0.5 times of the input sampling rate taken from the INTERVAL header element.
In case the INTERVAL header element is not available or not mandatory (e.g. meteorological data) the default tolerance range (eps) is 0.5 times of the via "-smp" specified sampling rate (num).

```
gfzrnx -finp pots0070.15o -fout pots0070.15o_rx3_5min -smp 300
gfzrnx -finp pots0070.15o -fout pots0070.15o_rx3_5min -smp 300:0.5
```

3.8.1.1 REMARK

If more than one observation epoch is found in the tolerance range only the nearest to the nominal epoch is used. Having several observation epochs within a tolerance range slows down the sampling process, especially for observation files. You can fasten the sampling process providing a reasonable tolerance range (eps) on the command line.
The default tolerance ranges are:

Sampling Rate	Default eps
$>= 1$ s	0.5 s
< 1 s	5 ms

3.8.2 Satellite System Selection (-satsys)

If you are interested in a subset of satellite systems only you can use the -**satsys** command line parameter to provide your wished satellite system. All other satellite systems are omitted in the output file.

```
-satsys <string>
```

The satellite systems string (string) consists of Satellite system letters (G-GPS, R-Glonass, E-Galileo, C-Beidou ...).

```
gfzrnx -finp pots0070.15o -fout pots0070.15o_rx3_GR  -satsys GR
gfzrnx -finp pots0070.15o -fout pots0070.15o_rx3_GRE -satsys GRE
gfzrnx -finp pots0070.15o -fout pots0070.15o_rx2_G   -satsys G   --version_out 2
```

3.8.3 PRN Selection (-prn, -no_prn)

For RINEX Observation files one can use a PRN selection/deselection via -**prn** and -**no_prn** command line parameters to include/exclude specific PRNs in the RINEX or statistics output. Both parameters can be mixed (-no_prn is prioritized). Simply provide a comma separated list of PRNs or PRN-ranges.

```
gfzrnx -finp pots0070.15o -fout pots0070.15o_rx3_small -prn G01,G05-20,R01-24,C05,C06 \
                                                       -no_prn G10,R05-7,R10
```

3.8.4 Observation Types Selection (-obs_types)

If you are interested in a subset of observation types only, you can use the -**obs_types** command line parameter to provide your wished observation types via a comma separated list of pattern.
The observation types selection works via a pattern matching mode. Here some examples:

3.8.4.1 RINEX-2

The input file contains the following observation types.

```
     8    C1    D1    L1    L2    P2    D2    S2    S1    P1# / TYPES OF OBSERV
```

Select code and phase observations only.

```
gfzrnx -finp pots0070.15o -fout pots0070.15o -obs_types P,C,L
```

The result will be a file containing the following observation types only.

```
     5    C1    L1    L2    P1    P2                          # / TYPES OF OBSERV
```

The following command line

```
gfzrnx -finp pots0070.15o -fout pots0070.15o -obs_types P2,C,L
```

will result in a file containing the following observation types, omitting the P1 observable too.

```
     4    C1    L1    L2    P2                                # / TYPES OF OBSERV
```

3.8.4.2 RINEX-3

In a simple case it works same way as for RINEX-2. For RINEX-3 it is possible to do the selection down to the satellite systems. One has to concatenate the global and the satellite system dependent definitions via the + character. For satellite system dependent selections you have to start with the satellite system character and colon.

```
list can be: [S:]OT1,OT2,...[+S:OT3,OT4,...][+...]

              S  - satellite system [CEGJRS]
              OT - observation type identifier
```

A satellite system dependent record replaces fully a global one.

Here is a global selection over all satellite systems (simple mode):

```
gfzrnx ... -obs_types L1,L2,C1,C2,P1,P2
```

Here is a global selection with special selections for **C** (Beidou) and **G** (GPS).

```
gfzrnx ... -obs_types L1,L2,C1,C2,P1,P2+C:L1,L7,C1,C7+G:L1C,L2W,C1,C2
```

3.8.5 Remove of Sparse Observation Types (-remove_sparse_obs_types)

One can give a limit in % which can be used to eliminate sparse observation types. The basis is the median of the number of observations per single observation type.

```
gfzrnx -finp pots0070.15o -fout pots0070.15o_ok --remove_sparse_obs_types 5
gfzrnx -finp pots0070.15o -fout pots0070.15o_ok -rsot 5
```

3.8.6 Keep all Observation Types (-kaot)

For GNSS observation files complete empty observation types are removed by default. Complete empty PRN data records are removed too. To keep all these data use the "-kaot" command line parameter.

3.8.7 Observation Types Sorting (-ots)

The default observation types output sorting order is alphanumeric. To controle the observation types output order (GNSS obs. files only) a string of the first observation types letters should be given.

```
-ots <CPLSD>
```

3.8.8 Navigation File Sorting

The output order of the navigation records can be controlled via **-nav_sort** or **-ns** command line parameter. Two options **prn**, **time** are possible.

- In the **time** mode the sorting order is by time and prn.
- In the **prn** mode the sorting order is by prn and time.

The standard mode is **prn**.

```
gfzrnx -finp pots0070.15n -fout pots0070.15o_srt -ns time
```

This can be used for any operation on navigation files (check, splice, split, ...).

```
gfzrnx -finp ????0070.15n -fout brds0070.15n -ns time
gfzrnx -finp ????0070.15n -fout ::RX3:: -split 3600 --nav_sort time
```

3.9 Handling a Group of Files with a Single Command (-single_file)

Usually a list of input files via **"-finp"** leads to a splice operation where the output is a single file. To initiate a file by file operation for a group of input files with a single command the command line parameter **"-single_file"** or **"-sifi"** has to be used.
For the output file naming the automatic file naming must be used **(::RX2::, ::RX3::)** or the "**::INP::**" variable, which means that the output file name is the same as the input file name.

Here an example for a data sampling operation on a group of input files:

```
gfzrnx.exe -finp c:\Rinex10sec\????3050.16o -fout e:\Rinex30sec\::INP:: -smp 30 --single_file
gfzrnx.exe -finp c:\Rinex10sec\????3050.16o -fout e:\Rinex30sec\::RX3:: -smp 30 -sifi
```

```
gfzrnx     -finp ????3050.16o -fout ./Rinex30sec/::INP:: -smp 30 --single_file
gfzrnx     -finp ????3050.16o -fout ./Rinex30sec/::RX2:: -smp 30 -sifi
```

3.10 Rinex File Header/Data Editing

RINEX file header editing can be invoked providing a configuration file for the header manipulations to be done. It has to be specified via the **-crux** command line parameter providing the configuration file name.

There are two modes available:

- Header editing as part of other operations on the input RINEX file.
- Header editing only. Only the header input, editing and check is performed but the data part is simply copied as it is.

In the following examples the configuration file **header_crux.txt** is used.

3.10.1 Header Editing (Standard)

```
rnxall -finp mizt1600.15o -fout mizt1600.15o_new -crux header_crux.txt
```

3.10.2 Header Editing (Only)

For the **editing only** mode one has to use the **-hded** option in addition.

```
rnxall -finp mizt1600.15o -fout mizt1600.15o_hded -crux header_crux.txt -hded
```

An additional epoch and station identifier has to be given if no standard RINEX file names are used. If no additional information is provided the **MARKER NAME** and the first data epoch is used if existing. This information is needed to extract the right header editing information from the overall configuration information.

```
gfzrnx -finp file.rnx -fout file.rnx_hded -crux header_crux.txt -hded -epo_beg 2015234_000000
                                                                                    -site POTS
```

3.10.3 Editing Operations

The following operations are supported:

- update single elements of an existing header line (label),
- insert single elements of a non existing header line (label),
- update(insert) a complete header line or multiple header lines per label.
- common string replacement in a string- or regular expression mode,
- renaming of PRN in the header and data part,
- renaming of OBS. types in the header part,
- station-, data type- and epoch interval dependent settings in a single configuration file are possible.

3.10.4 Show Config. File Interpretation (-show_crux)

Due to the variety of input options one can check how the configuration is interpreted in the program. This can be used as a kind of check via the **-show_crux** option before real use.

```
gfzrnx -crux header_crux.txt -show_crux
gfzrnx -crux header_crux.txt -show_crux -fout crux.log -f
```

The default header edit settings are shown via:

```
gfzrnx -show_crux
```

3.10.5 Configuration file

Formally there are 3 major modes: **update_insert**, **replace** or **rename** delimited by colon.
In case of **rename** a type (prn—obs) hast to be given additionally. The mode definition line has to be followed by an optional data type identifier string (OMN / **O**bs.,**M**et.,**N**av.) delimited with a hyphen, an optional epoch interval delimited by a hyphen and a valid station identifier (4 char.) or dot-separated list of station identifiers delimited by a colon. Now the editing definitions can follow.

```
update_insert :
#--------------
  [OMN-][YYYYMMDD:HHMMSS YYYYMMDD:HHMMSS-] ALL:
...
  [OMN-][YYYYDDD:SSSSS   YYYYDDD:SSSSS-]   STA1[.STA2[.STA3...] :
...

replace :
#--------
  [OMN-][YYYYMMDD:HHMMSS YYYYMMDD:HHMMSS-] ALL:
...
  [OMN-][YYYYDDD:SSSSS   YYYYDDD:SSSSS-]   STA1[.STA2[.STA3...]:
...
```

Every **rename** setting has to be done completely on a single line using the following syntax:

```
rename : prn
#--------
  [ON-][YYYYMMDD:HHMMSS YYYYMMDD:HHMMSS-] - <prn-from> - <prn-to> : ALL
  [ON-][YYYYDDD:SSSSS   YYYYDDD:SSSSS-]   - <prn-from> - <prn-to> : STA1[.STA2[.STA3...]


rename : obs
#--------
  [OM][YYYYMMDD:HHMMSS YYYYMMDD:HHMMSS-] <obs-from> - <obs-to> - <sat.sys> : ALL
  [OM][YYYYDDD:SSSSS   YYYYDDD:SSSSS-]   <obs-from> - <obs-to> - <sat.sys> : STA1[.STA2[.STA3...]
```

The following rules have to be taken into account:

- Comment lines have to begin with ♯.
- The file name station identifier has to be used for the station name.
  At the moment only the 4 char. station identifier is supported (RINEX-2 file naming).
- For non specific station definitions the **ALL** station identifier can be used.
- Omitting the data types identifier extends the validity to all supported data types (OMN).
- Omitting the epoch interval leads to an overall validity.
- Station dependent settings overwrite non specific **ALL** settings.
- Overlapping epoch intervals for the same header label and station lead to an error.
- The **date** of the epoch interval can be given either as **YYYYDDD** (year, day of year) or **YYYYMMDD** (year, month, day of month)
- The **time** of the epoch interval can be given as **SSSSS** (second of day 0-86399) or **HHMMSS** (hour, minute, second)
- An unlimited begin or end of an epoch interval can be given using zeros in the date and time values (e.g. 0000000:000000)

See also the examples below.

3.10.5.1 Update - Single Header Element

Single header element update/insert can be done providing the label in double quotes , "+"an optional time interval, ":" and the list of index-value pairs enclosed in curly brackets. Every definition should cover only one line !

```
"<label>" [+ YYYYMMDD:HHMMSS YYYYMMDD:HHMMSS ] : { k: "<value>", [ [ l: "<value> ], ... ] }
"<label>" [+  YYYYDDD:HHMMSS YYYYDDDD:HHMMSS ] : { k: "<value>", [ [ l: "<value> ], ... ] }
"<label>" [+  YYYYDDD:SSSSS  YYYYDDDD:SSSSS  ] : { k: "<value>", [ [ l: "<value> ], ... ] }
...

indexes k,l,... = 0,1,...
```

See some examples below:

```
update_insert :
#--------------
  0 - POTS.OUST.WINT:
    "REC # / TYPE / VERS"  : { 1 : "TRIMBLE NETR9" }

  0 - 2015209:00000 0000000:00000 - MIZT:
    "APPROX POSITION XYZ"  : { 0: -3857167.6484, 1: 3108694.9138, 2: 4004041.6876 }
    "ANTENNA: DELTA H/E/N" : { 0: 0.1209,        1: 0.0008,       2: 0.0007 }

  0 - POTS:
    "OBSERVER / AGENCY" + 0000000:00000 2013126:86399  : { 0:automatic, 1:"GFZ" }
    "OBSERVER / AGENCY" + 2013127:00000 0000000:00000  : { 0:gfz,       1:"GFZ/IHL" }
```

- Multi string elements in the index-value pairs have to be enclosed with double quotes.
  Please make sure, that the given values don't exceed the elements format length !
- The first header element is at index 0.

3.10.5.2 Update - Multi Header

Multiple header elements like the **"SENSOR MOD/TYPE/ACC"** or **"SENSOR POS XYZ/H"** for meteo data need an additional condition (here the sensor identifiers TD,PR,HR,...). An additional "+ column_number:value" pair has to be added to the label and optional epoch interval information. The column counter starts with 0. Here a **crux** example block.

```
"<label>" [+ YYYYMMDD:HHMMSS YYYYMMDD:HHMMSS ] i:"CC" : { k:"<value>", [ [ l:"<value> ],..] }
"<label>" [+  YYYYDDD:HHMMSS YYYYDDDD:HHMMSS ] i:"CC" : { k:"<value>", [ [ l:"<value> ],..] }
"<label>" [+  YYYYDDD:SSSSS  YYYYDDDD:SSSSS  ] i:"CC" : { k:"<value>", [ [ l:"<value> ],..] }

indexes i,k,l,... = 0,1,...
CC                = condition string
```

Here a **crux** example block.

```
update_insert :

#-----

M - 2015209:00000 0000000:00000 - ALL :

    "SENSOR MOD/TYPE/ACC" + 3:"TD" : { 0:"Vaisala", 1:"PTU 303/5.14", 2:"0.10" }
    "SENSOR MOD/TYPE/ACC" + 3:"PR" : { 0:"Vaisala", 1:"PTU 303/5.14", 2:"0.05" }
    "SENSOR MOD/TYPE/ACC" + 3:"HR" : { 0:"Vaisala", 1:"PTU 303/5.14", 2:"1.7"  }

    "SENSOR MOD/TYPE/ACC" + 3:"XX" : { 0:"XXXXXXX", 1:"XXX 125",      2:"1.0"  }

M - POTS :

    "SENSOR POS XYZ/H" + 4:"TD"    : { 0:3275753.9120, 1:321110.8651, 2:5445041.8829, 3:5 }
    "SENSOR POS XYZ/H" + 4:"PR"    : { 0:3275753.9120, 1:321110.8651, 2:5445041.8829, 3:5 }
    "SENSOR POS XYZ/H" + 4:"HR"    : { 0:3275753.9120, 1:321110.8651, 2:5445041.8829, 3:5 }

    "SENSOR POS XYZ/H" + 4:"XX"    : { 0:3275753.9120, 1:321110.8651, 2:5445041.8829, 3:5 }
```

If an element is not found it will be added (see the "XX" sensor).

See below a small example fore a header manipulation with the initial header and the manipulation result.

```
rnxall -finp pots3410.15m -f -fout pots3410.15m_new -crux crux.txt
```

pots3410.15m

```
     2.11           METEOROLOGICAL DATA                     RINEX VERSION / TYPE
TPP 3.1                                2015-12-07 00:01:03 PGM / RUN BY / DATE
pots                                                       MARKER NAME
     3    TD    HR    PR                                   # / TYPES OF OBSERV
Paroscientific      Model 760                    0.1    TD SENSOR MOD/TYPE/ACC
Paroscientific      Model 760                    2.0    HR SENSOR MOD/TYPE/ACC
Paroscientific      Model 760                    0.1    PR SENSOR MOD/TYPE/ACC
  3275756.3423   321111.4422  5445046.8829        0.0000 TD SENSOR POS XYZ/H
  3275756.3423   321111.4422  5445046.8829        0.0000 HR SENSOR POS XYZ/H
  3275756.3423   321111.4422  5445046.8829        0.0000 PR SENSOR POS XYZ/H
                                                           END OF HEADER
```

pots3410.15m_new

```
     3.03           METEOROLOGICAL DATA                     RINEX VERSION / TYPE
TPP 3.1                              2015-12-07 00:01:03 COMMENT
RINEX_DB.pm         GFZ FILE CONVERSION 20150807 14:32:19UTCPGM / RUN BY / DATE
pots                                                        MARKER NAME
Vaisala             PTU 303/5.14                0.1    TD SENSOR MOD/TYPE/ACC
Vaisala             PTU 303/5.14                1.7    HR SENSOR MOD/TYPE/ACC
Vaisala             PTU 303/5.14                0.1    PR SENSOR MOD/TYPE/ACC
  3275753.9120   321110.8651  5445041.8829        5.0000 TD SENSOR POS XYZ/H
  3275753.9120   321110.8651  5445041.8829        5.0000 HR SENSOR POS XYZ/H
  3275753.9120   321110.8651  5445041.8829        5.0000 PR SENSOR POS XYZ/H
XXXXXXX             XXX 125                     1.0    XX SENSOR MOD/TYPE/ACC
  3275753.9120   321110.8651  5445041.8829        5.0000 XX SENSOR POS XYZ/H
     3    HR    PR    TD                                  # / TYPES OF OBSERV
                                                          END OF HEADER
```

3.10.5.3 Proposed Use

There are several possibilities to organize the header editing configuration file. The most clear form would be to organize it per station.
Below you can find a configuration example for the single station POTS covering the whole station history information for **O**bservation and **M**eteo file header entries.

```
update_insert:

OM - POTS:

"APPROX POSITION XYZ" : { 0:"3800689.6341", 1:"882077.3857", 2:"5028791.3179" }
"MARKER NAME"         : { 0:"POTS" }
"MARKER NUMBER"       : { 0:"14106M003" }
"OBSERVER / AGENCY"   : { 0:"GFZ", 1:"GFZ" }

"REC # / TYPE / VERS" + 1994274:00000 1996015:86340 : { 0:"289",  1:"ROGUE SNR-8000",      ...}
"REC # / TYPE / VERS" + 1996016:49680 1996151:28380 : { 0:"279",  1:"ROGUE SNR-8000",      ...}
"REC # / TYPE / VERS" + 1996151:28860 1999231:00000 : { 0:"289",  1:"ROGUE SNR-8000",      ...}
"REC # / TYPE / VERS" + 1999232:00000 2000232:00000 : { 0:"281",  1:"AOA SNR-8000 ACT",    ...}
"REC # / TYPE / VERS" + 2000233:00000 2009089:00000 : { 0:"281-U",1:"AOA SNR-8000 ACT",    ...}
"REC # / TYPE / VERS" + 2009089:00000 2011046:61200 : { 0:"1358", 1:"SEPT POLARX2",        ...}
"REC # / TYPE / VERS" + 2011046:61200 2011307:52200 : { 0:"205",  1:"JAVAD TRE_G3TH DELTA",...}
"REC # / TYPE / VERS" + 2011307:52200 2011354:38280 : { 0:"205",  1:"JAVAD TRE_G3TH DELTA",...}
"REC # / TYPE / VERS" + 2011354:38280 2012164:32400 : { 0:"205",  1:"JAVAD TRE_G3TH DELTA",...}
"REC # / TYPE / VERS" + 2012164:32400 2013009:36720 : { 0:"205",  1:"JAVAD TRE_G3TH DELTA",...}
"REC # / TYPE / VERS" + 2013009:36780 2015258:50280 : { 0:"205",  1:"JAVAD TRE_G3TH DELTA",...}
"REC # / TYPE / VERS" + 2015258:50280 0000000:00000 : { 0:"205",  1:"JAVAD TRE_G3TH DELTA",...}

"ANT # / TYPE"        + 1994301:00000 1995276:28800 : { 0:"261", 1:"AOAD/M_T",       2:"NONE" }
"ANT # / TYPE"        + 1995276:28800 2009105:47700 : { 0:"235", 1:"AOAD/M_T",       2:"NONE" }
"ANT # / TYPE"        + 2009105:47700 2011046:61200 : { 0:"354-U",1:"AOAD/M_T",      2:"NONE" }
"ANT # / TYPE"        + 2011046:61200 0000000:00000 : { 0:"316", 1:"JAV_RINGANT_G3T",2:"NONE" }

"ANTENNA: DELTA H/E/N"+ 1994301:00000 1995276:28800 : { 0:"0.046", 1:"0", 2:"0" }
"ANTENNA: DELTA H/E/N"+ 1995276:28800 2009105:47700 : { 0:"0.046", 1:"0", 2:"0" }
"ANTENNA: DELTA H/E/N"+ 2009105:47700 2011046:61200 : { 0:"0.046", 1:"0", 2:"0" }
"ANTENNA: DELTA H/E/N"+ 2011046:61200 0000000:00000 : { 0:"0.121", 1:"0", 2:"0" }

"SENSOR MOD/TYPE/ACC" + 1996254:00000 2006011:00000 + 3:"PR" : { 0:"Vaisala", 1:"PTB100B",... }
"SENSOR MOD/TYPE/ACC" + 2006011:00000 0000000:00000 + 3:"PR" : { 0:"Vaisala", 1:"PTU200", ... }

"SENSOR MOD/TYPE/ACC" + 1996254:00000 2006011:00000 + 3:"HR" : { 0:"Timetech",1:"HC 500", ... }
"SENSOR MOD/TYPE/ACC" + 2006011:00000 0000000:00000 + 3:"HR" : { 0:"Vaisala", 1:"HMP45A-P",.. }
```

```
"SENSOR MOD/TYPE/ACC" + 1996254:00000 2006011:00000 + 3:"TD" : { 0:"Timetech",1:"PT100",  ... }
"SENSOR MOD/TYPE/ACC" + 2006011:00000 0000000:00000 + 3:"TD" : { 0:"Vaisala", 1:"HMP45A-P",.. }
```

Depending on the first data epoch the appropriate header entry is updated.

3.10.5.4 Remark

There is one exclusion concerning the RINEX header fields manipulation. According to IGS antenna definition (number, antenna + radome) the **"ANT # / TYPE"** record consists of 3 columns, which is a deviation from the RINEX standard.
This means, the standard (A20,A20) RINEX definition is in gfzrnx handled as (A20,A16,A4). A correction record should be of the following form:

```
update_insert :
# -------------
POTS:
     "ANT # / TYPE"  : { 0:"30336561", 1:"TRM55971.00", 2:"NONE" }
```

3.10.5.5 Complete Header Line(s) Update

For a single line definition one has to give the label name in double quotes followed by an "+" optional epoch interval string followed by a colon and the 60 char. string to be updated or inserted. The multi-line definition has to be enclosed in square brackets as a comma separated list of 60 char. strings with one string per line.
The square brackets have to be given on the first (**[**) and last (**]**) 60 char. string definition line.

```
"<label>" [+ YYYYMMDD:HHMMSS YYYYMMDD:HHMMSS ] : [ "<60-char. string>",
                                                   "<60-char. string>",
                                                   ...
                                                   "<60-char. string>" ]
```

```
update_insert :
#--------------
  0 - 2015010:00000 0000000:00000 - POTS:
     "OBSERVER / AGENCY"    :   "Automatic          Deutsches GeoForschungsZentrum (GFZ)    "

     "SYS / PHASE SHIFT"    : [ "G L1C  0.00000                                             ",
                                "J L1C  0.00000                                             ",
                                "J L1X  0.25000                                             ",
                                "E L1X  0.00000                                             ",
                                "C L7I  0.00000                                             ",
                                "R L1P  0.25000                                             ",
                                "R L2C  0.00000                                             ",
                                "R L2P  0.25000                                             ",
                                "G L2X -0.25000                                             ",
                                "G L5X  0.00000                                             "  ]
```

Please keep in mind, that an already existing header label content is completely removed. Only **COMMENT** header lines are appended.

3.10.5.6 Header Label Independent String Replacement

For the string replacement the major mode **replace** has to be used. One has to define the station identifier as before. Afterwards you can define from/to pairs of type **regexp** or **string**. The **regular expression** syntax follows **Perl** syntax. Each pair element (from/to) should be given on a separate line.
The example below shows how to correct an erroneous label name.

```
replace :
#--------
  ALL:
    regexp_from : "^(.{60})PGM\s*/\s*RUN\s*BY\s*/\s*DATE\s*$"
    regexp_to   : "$1PGM / RUN BY / DATE"
  ALL:
    string_from : "PGM/RUN BY/DATE"
    string_to   : "PGM / RUN BY / DATE"
```

For the remove of single header label lines on **input** use an empty **regexp_to** (""). To remove all COMMENT lines use:

```
replace :
#--------
  ALL:
    regexp_from : "^.{60}COMMENT\s*$"
    regexp_to   : ""
```

To remove lines containing the string "ABC DEF" use:

```
replace :
#--------
  ALL:
    regexp_from : "^.*ABC DEF.*$"
    regexp_to   : ""
```

3.10.5.7 Rename - PRNs

If raw data conversion programs don't assign the right PRN, this can be changed via the **"rename: PRN"** mode. Here the crux configuration syntax:
Here some examples:

```
rename: prn
#----------

ON - 20140105:000000 20150101:000000 - E51 - E01: ALL
ON - 20140105:000000 00000000:000000 - E52 - E02 : ABC1.ABC2.ABC3

E51 - E01 : ALL
E52 - E02 : ALL
```

3.10.5.8 Rename - OBS types

```
rename: obs
#----------

20140105:000000 20150101:000000 - L2X - L2L - G : ABCD
20140105:000000 20150101:000000 - L2L - L2X - G : ABCD

20140105:000000 20150101:000000 - *2* - *1* - C : ALL
20140105:000000 20150101:000000 - *2  - *1  - C : ALL

20140105:000000 20150101:000000 - **X - **L - C : ALL
20140105:000000 20150101:000000 - *2  - *1  - C : ALL

20140105:000000 20150101:000000 - **X - **L - G04.G08 : ALL
20140105:000000 20150101:000000 - *2  - *1  - G04.G08 : ALL

*2* - *1* - C : ALL
*2  - *1  - C : ALL
```

3.10.5.9 Remark

The **replace** mode is done directly on input, the **update_insert** and **rename** modes are done after the whole header has been read.

3.11 Rinex File Meta Data Extraction (-meta)

RINEX file meta informations can be extracted from header and data in different output formats.

```
-meta [mode:format]          mode=[basic:full], format=[txt,json,xml,dump]
```

- The **basic** mode extracts only the header information and the first, last epoch from the RINEX file without reading the whole file (fast).
- The **full** mode extends/updates the basic information with information derived from the complete data file like data statistics, the real data interval and so on.
- There are supported the following output **format**s: **txt**(default),**json**,**xml**,**dump** to be used for fast view or further applications.

Here some simple examples:

```
gfzrnx -finp pots0070.15o -meta basic
gfzrnx -finp pots0070.15o -meta basic:txt
gfzrnx -finp pots0070.15o -meta basic:json -fout pots0070.15o.json
gfzrnx -finp pots0070.15o -meta  full:xml  -fout pots0070.15o.xml
```

```
rnxall -finp pots0070.15o -meta basic:txt
```

```
antenna:
     height:
          x = 0
          y = 0
          z = 0
     name = JAV_RINGANT_G3T
     number = 316
     radome = NONE
data:
     epoch:
          first = 2015 01 07 00 00 00.0000000
          interval = 30.000
          last = 2015 01 07 23 59 30.0000000
file:
     md5 = e1202b7ef2bb19266356b2fd12c4f023
     system = M
     type = O
     version = 3.02
receiver:
     firmware = 3.4.7
     name = JAVAD TRE_G3TH DELTA
     number = 205
site:
     agency = GFZ
     name = POTS
     number = 14106M003
     observer = GFZ
     position:
          x = 3800689.6333
          y = 882077.3949
          z = 5028791.3131
```

```
rnxall -finp pots0070.15o -meta basic:json
```


```
{"receiver":{"number":"205","name":"JAVAD TRE_G3TH DELTA","firmware":"3.4.7"},
"site":{"number":"14106M003","position{"y":"882077.3949","x":"3800689.6333",
"z":"5028791.3131"},"name":"POTS","agency":"GFZ","observer":"GFZ"},"file"{
"system":"M","version":"3.02","type":"O","md5":"e1202b7ef2bb19266356b2fd12c4f023"},
"data":{"epoch":{"first":"2015 01 07 00 00 00.0000000","last":
"2015 01 07 23 59 30.0000000","interval":"30.000"}},"antenna":{"number":"316",
"name":"JAV_RINGANT_G3T","height":{"y":0,"x":0,"z":0},"radome":"NONE"}}
```


3.12 Rinex File Comparison (-fdiff)

The comparison of single site RINEX files of the same time interval and from different sources (e.g. real time data, data from different rinex-converters, ...) are often not possible in an easy way. gfzrnx offers a possibility to compare two input files of the same format (major version id.) via the **-fdiff** command line parameter. NOTE, different observation types orders in the input files are allowed !

```
gfzrnx -fdiff -finp <rinex_file_1> <rinex_file_2>
```

The output is RINEX-3 like, storing only the data epochs and data records where both files differ in the data records. Internal or data headers are ignored.

- If per epoch an observation type exists in both files its numerical difference (file1-file2) is shown.
- If per epoch an observation type is missing in one of the input files the original data value of the corresponding input file is shown (merged).
- For the LLI and SNR values always **absolute** differences are reported.

```
gfzrnx -fdiff -finp pots0140.16o_1 pots0140.16o_2 -fout pots0140.16o_diff
```

In the header you can find the observation types order and the PRN-statistics of detected differences.

```
     3.00           DATA COMPARISON                         RINEX VERSION / TYPE
------------------------------------------------------------COMMENT
pots0140.16o_1                                               FILE_1
pots0140.16o_2                                               FILE_2
------------------------------------------------------------COMMENT
...
G    4 C1C L1  L2  C2W                                       SYS / # / OBS TYPES
R    4 C1C L1  L2  C2P                                       SYS / # / OBS TYPES
    20                                                       # OF SATELLITES
   G02     2     1           1                               PRN / # OF OBS
   G03     2     1     1     1                               PRN / # OF OBS
   G06     2     1     1     1                               PRN / # OF OBS
...
```

The data or differences part will look like the following example:

```
> 2016 01 14 11 00 00.0000000  0  2
G02             1
G03             1
> 2016 01 14 11 00 01.0000000  0  2
G02         0.052           0.098                           0.012
G19  19699748.072   105380370.084    81962499.868    19699744.832
> 2016 01 14 11 00 02.0000000  0 19
G03  22232325.432   116831670.250    91037637.373    22232315.592
G06  23394480.604   122938818.380    95796470.667    23394477.044
...
G31  23924131.742   125722160.848    97965321.818    23924126.722
> 2016 01 14 11 00 02.0000000  0 19
...
```

- In the first epoch the data of two PRNs differ by "1" in the LLI (loss of lock indicator) value for the C1C observation type.
- In the second epoch the PRN G02 differs (file1-file2) by the given values for the observation types C1C, L1, C2W.
  The PRN G19 seems to be fully missing in one of the files or you see a merged record, where an observation type is missing either in the the first or the second file.
- The third epoch seems to be fully missing in one of the files or you see a merged record, where a full PRN or an observation type is missing either in the the first or the second file.

3.13 Rinex Standard Extensions/NonConformity

3.13.1 RINEX-2 BDS,QZSS,IRNSS support

As an extension to the RINEX-2.11 standard, the BEIDOU-, QZSS-, IRNSS- satellite systems are formally supported.

3.13.2 RINEX-2 to RINEX-3 conversion

The RINEX-3.03 standard does not allow an empty attribute identifier (tracking mode or channel) in observation type naming (**tna** - obs. type—band/frequency—attribute). Converting files from RINEX-2 to RINEX-3 show up the problem to safely map 2-char. to 3-char. obs. type names (e.g. **L2** to **L2?**). As it is not foreseen to have an "unknown" or "converted" attribute identifier the output version used is **3.01** to stay format conform.

3.13.3 Handling of unsupported observation types

gfzrnx is driven by a hardcoded observation types and mapping table conform to the RINEX standards. Running the program for unsupported or non standard observations types leads to an omitting of these data. To avoid this behaviour one has to extend the standard. This can be done with the following procedure:

- Extract the hardcoded table from **gfzrnx** executable.

```
gfzrnx -out_obs_map
gfzrnx -out_obs_map -fout obs_types_map.txt
```

- Add new obs. types records to the map.
  The information in the columns 2,3,4 are treated as comment only and are not used.
- Run any **gfzrnx** command call with the modified table.

```
gfzrnx -use_obs_map obs_types_map.txt -finp ...
```

3.13.4 Remark

Please use this feature with special caution !
Be aware that this undermines the given RINEX standard and can be an error source if not used properly. The generated files should be for internal use only !

List of Figures