HELMHOLTZ
RESEARCH FOR GRAND CHALLENGES

HELMHOLTZ CENTRE POTSDAM
GFZ GERMAN RESEARCH CENTRE FOR GEOSCIENCES

GFZRNX - Users Guide

Version 2.1.10

Thomas Nischan

Content - GFZRNX - Users Guide

1. Overview

1.1 Welcome

Welcome to the online documentation of gfzrnx. The latest version of this documentation can be found via the following links.

1.1.1 Documentation (HTML)

The web based documentation is available via:

https://gnss.git-pages.gfz-potsdam.de/gfzrnx

1.1.2 Documentation (PDF)

The PDF version of this documentation is available via:

https://gnss.git-pages.gfz-potsdam.de/gfzrnx/pdf/GFZRNX_Users_Guide.pdf

1.2 Scope of Operation

The software **gfzrnx** is a toolbox for the check and manipulation of RINEX files for the major versions 2, 3, and 4.

The following RINEX data types are supported:

- Observation data
- Navigation data
- Meteorological data

The following operations/tasks are supported:

- RINEX file check and repair
- RINEX file format conversion (versions 2, 3, 4 into each other)
- 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
- PRN renaming
- Automatic version dependent file naming on output file.
- RINEX file (re)naming support (version 2 to 3 or 4)
- RINEX header editing
- RINEX file metadata extraction
- RINEX file comparison
- RINEX file tabular representation
- Multi RINEX file operations with a single command

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

1.3 End User License Agreement

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

- License for **Scientific Partners** (free)
- License for **Commercial Users** (chargeable).

1.4 Examples

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

```
Example Box
```

All given examples are valid for UNIX-based systems like Linux, SunOS or OSX. In the example boxes you will find `gfzrnx` used as a synonym for the operating system-dependent executable (gfzrnx_lx64, gfzrnx_osx64, ...).

1.5 Follow us

1.5.1 Documentation (HTML)

https://gnss.git-pages.gfz-potsdam.de/gfzrnx

1.5.2 Documentation (PDF)

https://gnss.git-pages.gfz-potsdam.de/gfzrnx/pdf/GFZRNX_Users_Guide.pdf

1.5.3 Join Mailing List

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

One can join the mailing list by sending an empty email to:

gfzrnx-on@gfz-potsdam.de

After getting a **Confirmation Request** email, please don't forget to reply to it. This reply is mandatory to finish your list joining.

1.5.4 Drop Out of Mailing List

One can drop out of the mailing list sending an empty email to:

gfzrnx-off@gfz-potsdam.de

1.5.5 Twitter: @gfzrnx

1.5.6 Bug Reports and 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.

2. Basics

2.1 Download

One can download the software via
https://gnss.gfz-potsdam.de/services/gfzrnx

You will find an **official version** with a version number and a **development version** (DEVEL) with ongoing bug fixing and new features.

2.2 Installation

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.

Operating Systen	Executable
Linux (64)	**gfzrnx_lx64**-x.x.x
Linux (32)	**gfzrnx_lx32**-x.x.x
SunOS (Sparc)	**gfzrnx_sun**-x.x.x
SunOS (i86)	**gfzsun_suni86**-x.x.x
MS Windows 11 (64)	**gfzrnx_win11_64**-x.x.x.**exe**
MS Windows 10 (64)	**gfzrnx_win10_64**-x.x.x.**exe**
MS Windows 7,8 (64)	**gfzrnx_win64**-x.x.x.**exe**
MS Windows 7,8 (32)	**gfzrnx_win32**-x.x.x.**exe**
Mac OSX (64)	**gfzrnx_osx64**-x.x.x
Mac OSX (arm64, Mx)	**gfzrnx_osxarm64**-x.x.x
Linux-ARM (64)	**gfzrnx_armlx64**-x.x.x
Linux-ARM (32)	**gfzrnx_armlx32**-x.x.x

2.2.1 UNIX

Put the executable to a directory of the search path **$PATH**. Simply run the following command to show the elements of your search path **$PATH**. This search path can differ depending on your overall setup.

```
> echo $PATH
/opt/local/bin:/opt/local/sbin:/usr/local/bin:/usr/bin:/bin:/usr/sbin:/sbin:/opt/X11/bin
```

Copy the executable into a directory covered by your system search **$PATH** variable and create a symbolic link **gfzrnx** for the ease of use. Here is an example using e.g. `/usr/local/bin` and the downloaded executable `gfzrnx_osxarm64-1.16-8154` :

```
> sudo copy gfzrnx_osxarm64-1.16-8154 /usr/local/bin
> sudo chmod a+x gfzrnx_osxarm64-1.16-8154
> cd /usr/local/bin
> sudo ln -s gfzrnx_osxarm64-1.16-8154 gfzrnx
```

The **which** command should show up with the search fullpath of the **gfzrnx**-command.

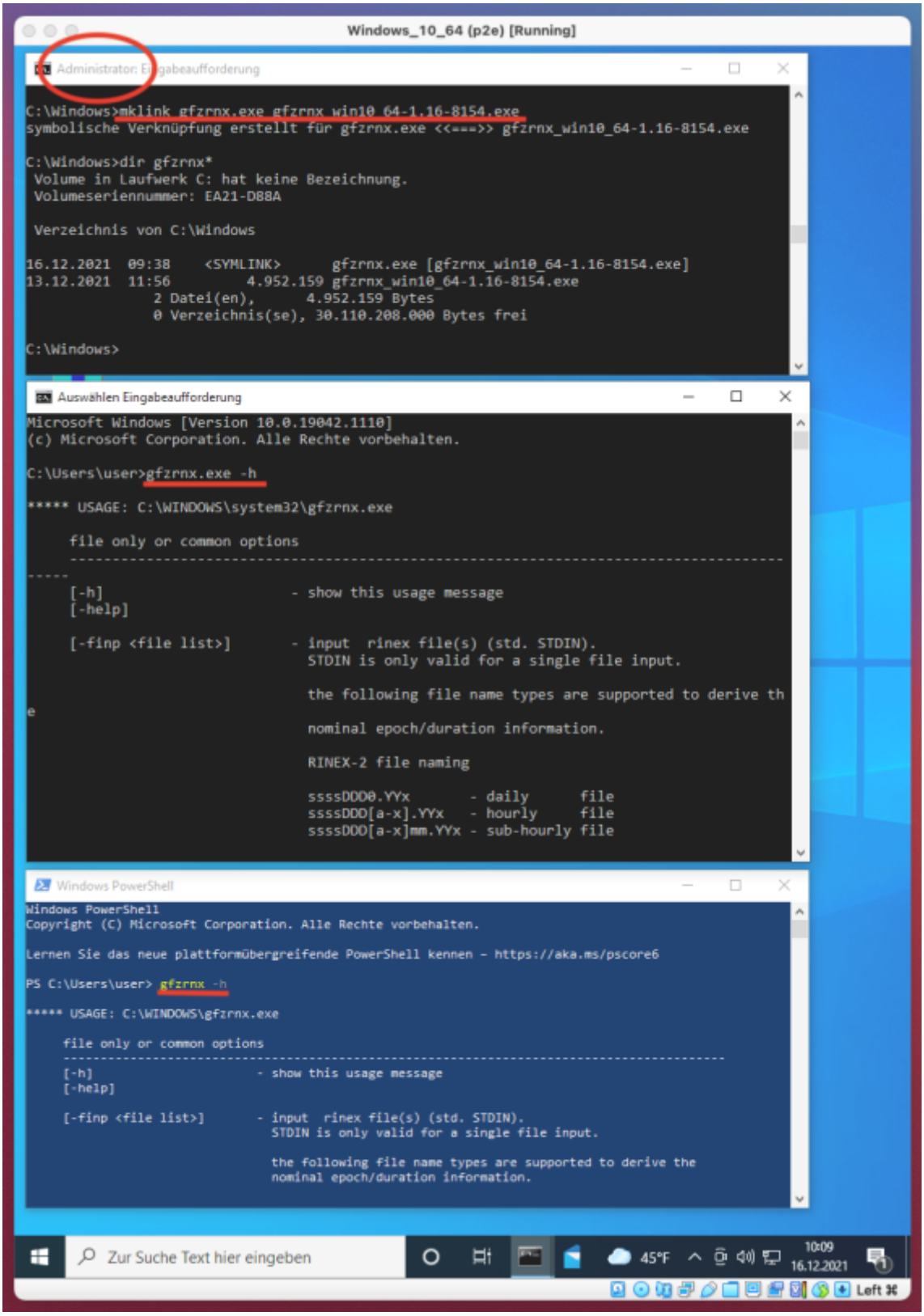
```
> which gfzrnx
/usr/local/bin/gfzrnx
> gfzrnx -h
```

2.2.2 MS Windows

Create a symbolic link **gfzrnx.exe** in your **C:\Windows** to the real executable location (e.g. c:\Users\user\gfzrnx_1.16-8154_win10.exe). This allows you now to use the short command name `gfzrnx.exe` or simply `gfzrnx` .

Open `cmd.exe` as **Administrator** (right mouse click on cmd.exe).

```
C:\Users\user > cd c:\Windows
C:\Windows    > mklink /J gfzrnx.exe c:\Users\user\gfzrnx_1.16-8154_win10.exe
```



2.2.3 Temporary Directory

The `gfzrnx` will store and execute libraries in a temporary directory.

OS	Default Temporary Directory
UNIX	**/tmp**
Windows	**$WINDIR** (C:\Windows)

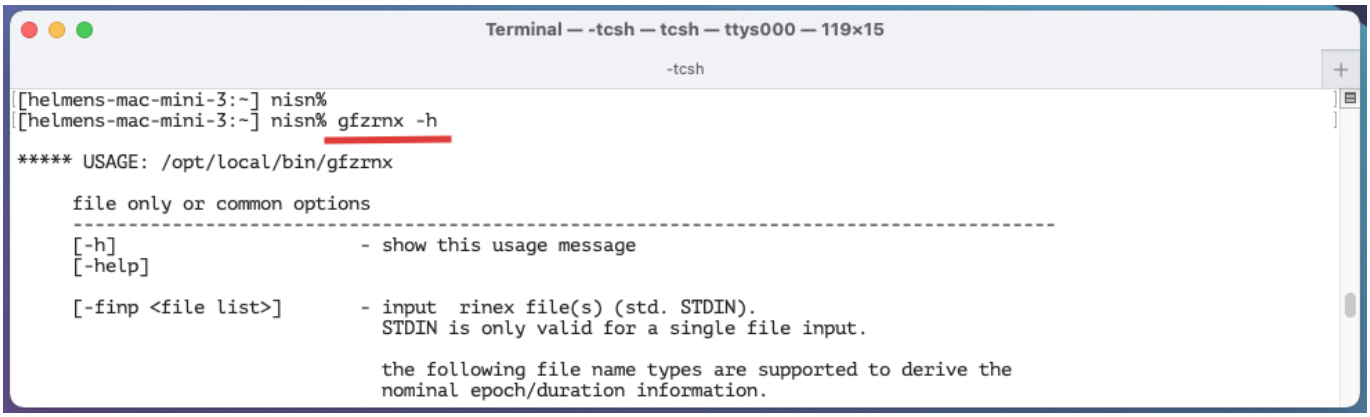
If this is not an option for you, you can specify an alternative temporary directory via the environment variables **$TEMP** or **$TMP** on all platforms.

2.3 Usage

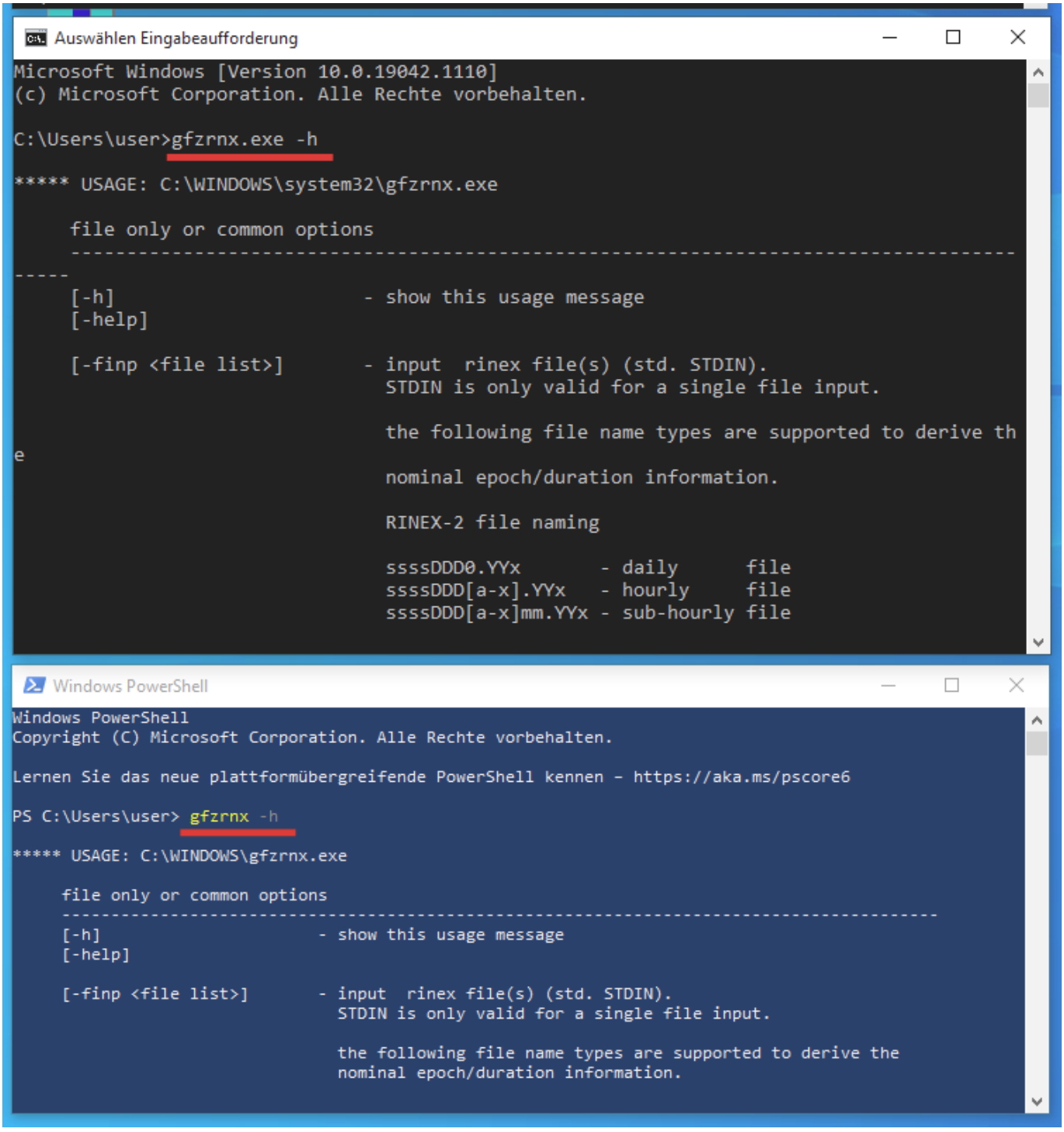
gfzrnx is a command line executable. It can be used in a terminal window or batch scripts. It has **NO graphical interface**!

2.3.1 Unix

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



2.3.2 Windows



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

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

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

2.3.3 Fast Help

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-3/4 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 output file name if filename given is "::RX2::", "::RX3::" or "::RX4::"

     [-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"

     [-ant_rename]             - rename historical antenna names to be IGS conform

   [-nomren23 <[s,][mr,][iso]>] - fast nominal output file name for RINEX-2 to RINEX-3 file renaming.
                                 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 nnnndddedd.yyt[.cmp]

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

     [-vo <2|3|4>]             - output RINEX version (std. latest)
     [--version_out <2|3|4>]
     [-vosc <2|3|4>]           - output RINEX version (fully standard conform)

     [-vnum m.nn]              - change header VERSION number and set output RINEX version
                                 (only the version number is changed / output RINEX version is the highest supported
```

```
one)

     [-pr3rx2 <list>]          - komma separated list of list of signal priorities used for rinex 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

     [-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)

     [-smp_nom <num>]          - sampling rate (num) in sec to be used for automatic file naming

     [-smp_lli_shift]          - perform LLI shifts via data sampling to sampling epoch

     [-nav_mixed]              - create a mixed nav. filename

     [-no_nav_stk]             - no nav. splice header statistic tables

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

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

     [-cx_updins <string(s)>]  - rinex header manipulation(s) definition for input files
                                 given via command line

     [-cx_addinthd]            - if using using a crux-file (-crux) internal/data headers are created
                                 at crux-settings starting epochs.

     [-show_crux]              - show crux structure adopted and used by the program

     [-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
                                 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

     [-ots <string>[:<attr>]]  - obs. types output sorting
[--obs_types_sort <string>[:<attr>]]
                                 the "string" consists of the 1st obs. type id. characters ( e.g. CPLDS ),
                                 the "attr" can be [frqasc|frqdsc|frqi,j,...] (frequ. numbers (i,j,...) = 1,...,n),
                                 which means a preferred sorting by frequency (ascending,descending or
                                 a list of distinct frequency numbers)

     [-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

     [-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
                                 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

     [-nt       <type-list>]   - '+' separated list of nav. selection records (version >= 4).
     [-nav_type <type-list>]     record = [<sat.system(s)>::]<nav.type(s)>:[<message.type(s)]
                                 type(s) are separated via '.'

     [-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

     [-met_nwm]                - edit a rinex meteo file(1) by the means of a reference NWM file(2).
                                 the two input files have to be given via -finp.
                                 the second file contains reference NWM data and check limits
                                 (can be used in conjunction with -obs_types, -ot)

     [-site <sitename>]        - use the 4- or 9-char sitename for output filename via automatic file naming
                                 or for header editing settings extractions (crux)
                                 or for "MARKER NAME" in case it is missing.

     [-kv]                     - keep major output version number 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 seconds
                                 (when using strict epoch interval)

     [-nav_epo_filter]         - only standard epochs are passed to the output
     [-nav_epo_strict]         - only nominal  epochs are passed to the output
     [-nav_latest]             - only latest nav. record per PRN are passed to the output

     [-splice_direct]          - use no RAM to store observations via splice operations
                                 (no header data statistics)

     [-try_append <sec>]       - try append mode to fasten the splice process with
                                 smallest nominal file duration (seconds) of part files

     [-direct]                 - direct input/output of single rinex OBS. files (!!! no header statistics !!!)

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

```
     [-tab]                    - create a tabular data representation output

     [-tab_date]               - use other date (pattern) for tabular observation output
                                 (yyyy-mm-dd|yy-mm-dd|yyyy-ddd|wwww-d|yyyymmdd|yymmdd|yyyyddd|wwwwd|mjd|ddd)

     [-tab_time]               - use other time pattern for tabular observation output
                                 (hh:mm:ss|hhmmss|sod|fod)

     [-tab_sep <string>]       - column separator string (default: BLANK)

     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 https://gnss.gfz-potsdam.de/services/gfzrnx

  for license details and manual

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

  VERSION: gfzrnx-2.1.8
```

2.4 Data Input/Output

2.4.1 Supported Format Versions

gfzrnx supports all versions 2.x, 3.x, 4.x formats as input. The output format will be only the latest standard format of the major formats 2, 3 or 4.

Standard out put version for major RINEX version:

Major	Standard Output
2	2.11
3	3.05
4	4.01

2.4.2 Input

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

2.4.3 Output

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

2.4.4 Examples Input/Output

Input via `-finp`

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

Input via STDIN

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

Output via `-fout`

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

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 program.

2.4.5 Log Messages

By default, log messages (Notices, Errors, Warnings) are sent to **STDERR**. One can store the log messages into a file using the `-errlog` command line parameter.

```
> gfzrnx -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 message

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

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

2.4.6 Direct Mode (-direct)

For a single **OBS.** rinex file it is possible to use a **direct** input/output mode. The I/O is done epoch by epoch and you miss information stored in the header like statistics and the automatic removal of empty obs. types. The advantage is the economic utilisation of RAM.

```
> gfzrnx -finp leid2000.13o -fout leid2000.13o_rx3 -direct
```

2.5 Supported File Names

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

2.5.1 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

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
```

2.5.2 RINEX-3 naming convention

File Name	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.

2.5.3 Automatic Output File Naming

For an automatic output file naming, one can use the **::RX2::**, **::RX3::** or **RX4::** parameter for the `-fout` command line switch.

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.

RINEX-3 Site Name

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

This works if the header **MARKER NAME** fully matches the RINEX-3 "SSSSMRCCC" naming style. For a 4-character **MARKER NAME** one has to provide at least the **marker-**, **receiver numbers**, and the **ISO country code** in 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 for a 4-char header. **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**.

RINEX-3 Site Name (-4to9)

Besides 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

SEnsor Meta Information SYStem (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,TIGA' -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.

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** (Long, Short, Day) 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 storing 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. In the case of sub-hourly, file names with nominal begin epochs (`-epo_beg` / `-sei in`) and the nominal duration `-d 900` are used by default. For other time intervals, the duration (`-d`) has to be given.

If the data start minute is 17 and the duration e.g. 300 s the following commands give different output file names:

```
gfzrnx -kv -finp pots125x15.13o -fout TMP/::RX2::
TMP/pots125x15.13o

gfzrnx -kv -finp pots125x15.13o -fout TMP/::RX2L::
TMP/pots125x15.13o

gfzrnx -kv -finp pots125x15.13o -fout TMP/::RX2L:: -d 120
TMP/pots125x16.13o

gfzrnx -kv -finp pots125x15.13o -fout TMP/::RX2L:: -sei in
TMP/pots125x15.13o

gfzrnx -kv -finp pots125x15.13o -fout TMP/::RX2L:: -epo_beg 2015125_230000 -d 1800
TMP/pots125x00.13o

gfzrnx -kv -finp pots125x15.13o -fout TMP/::RX2S::
TMP/pots125x.13o
```

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 observation 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 beginning 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

RINEX-3 Start Epoch/Duration (nominal)

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

The general method is to give the beginning 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 input 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

RINEX-3 Mixed Broadcast Splice File Naming -nav_mixed

If generating a mixed broadcast navigation file with automatic file naming (::RX3::) in an ongoing accumulation mode one should use the `-nav_mixed` command line parameter to ensure that a **_MN** file name is generated, nevertheless a single satellite system is found in the given file(s).

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.

> **Warning**
>
> 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.

3. Tasks

3.1 Operation / Tasks

The following operations/tasks are supported:

- RINEX file check and repair
- RINEX file format conversion (versions 2, 3, 4 into each other)
- RINEX file splice
- RINEX file split
- RINEX file statistics generation
- RINEX file manipulations like:
- data sampling
- observation types selection
- satellite systems selection
- elimination of empty or sparse observation types on a file level
- elimination of empty or sparse observation types on a PRN level
- PRN renaming
- Automatic version dependent file naming on output file.
- RINEX file (re)naming support (version 2 to 3 or 4)
- RINEX header editing
- RINEX file metadata extraction
- RINEX file comparison
- RINEX file tabular representation
- Multi RINEX file operations with a single command

Complete data check

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 input files are correct, and you want to do some data manipulation only, you can omit this command line parameter to speed up the work.

RAM utilisation

Please keep in mind that compared to other tools, working on a single epoch level, **gfzrnx** stores the whole RINEX data set in the computer's memory before output. This leads to some performance degradation but offers complete data handling opportunities.

Standard RINEX output version

The standard output format of **gfzrnx** is at the moment the latest supported version of major version **RINEX-3** (e.g. **3.05**).

RINEX-4 support

The major version **RINEX-4** is supported. Use the `-vo 4` command line parameter to force the RINEX-4 file output.

3.2 RINEX File Check and Repair

IIf 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 the latest supported format).

The following modifications are done in the output file:

- Update of observation types to really existing ones, overall empty observation types are removed.
- SYS / # / OBS TYPES
- # / TYPES OF OBSERV
- Statistical information is added or updated in the file header.
- PRN / # OF OBS
- # OF SATELLITES
- INTERVAL
- TIME OF FIRST OBS
- TIME OF LAST OBS

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 is different concerning RINEX-2 and RINEX-3(4,...). Data values are not corrected! Via the repair operation, formally corrupt observation parts are omitted only.

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

3.2.1 Navigation Data Epoch Filter

Use the `-nav_epo_filter` command line parameter to filter the navigation data input via epoch record checks. In this case, only **nominal** epochs are passed to the output file. Excluded records are given in the log table. Only epoch minutes and hours are checked at the moment. The following table shows valid hours and minutes per satellite system:

Sat. System	Minutes	Hours (modulo)
C	0	1
E	0,10,20,30,40,50	1
G	0	2
R	15,45	1
J	0	1

3.2.2 Meteo Data check/edit against NWM-Data

RINEX-meteo site data can be checked against a reference RINEX-file created from e.g. Numerical Weather Model (**NWM**) data (predicted or reprocessed). Two input files have to be given via the `-finp` command line parameter. The first file is the one to be checked, the second file is the reference file to be checked against.

The check limits for the different observation types are taken from the reference file "**SENSOR MOD/TYPE/ACC**" header records. Here is an example:

```
DNSXGFZ             ERA5/ECMWF                  10.0    PR SENSOR MOD/TYPE/ACC
DNSXGFZ             ERA5/ECMWF                  10.0    TD SENSOR MOD/TYPE/ACC
DNSXGFZ             ERA5/ECMWF                 100.0    HR SENSOR MOD/TYPE/ACC
```

Observations that exceed the difference limits are eliminated and don't go to the output file.

```
gfzrnx -met_nwm -finp POTS00DEU_R_20200010000_01D_05M_MM.rnx POTS00DEU_R_20200010000_01D_05M_MM.rnx_nwm -fout
POTS00DEU_R_20200010000_01D_05M_MM.rnx_chk
```

By default, all observation types found in the reference file are used for the differences checks. To limit the tests to single observation types, they can be given via the `-obs_types` or `-ot` command line parameters.

```
gfzrnx -met_nwm -finp ... -fout POTS00DEU_R_20200010000_01D_05M_MM.rnx_chk -ot TD,HR
```

The removed observations are documented in the log.

```
... | MESSAGE
...-+-------------------------------------------------------...
... | MET_NWM: ignoring checks for obs. type -> PR
    | MET_NWM: DELETED 2020-01-01 00:05:00.0000000 HR -   -81.4(   97.4) exceeds limit of 100.0(178.8)
    | MET_NWM: DELETED 2020-01-01 00:15:00.0000000 TD -    29.0(    2.6) exceeds limit of 10.0(26.4)
...
```

3.3 RINEX File Statistics / Information

3.3.1 Observations Statistics

The `-stk_only` or `-stk_obs` outputs 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_obs

 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
```

3.3.2 ASCII Time plot of Observables

The `-stk_epo` command line parameter can be used to create an ASCII time plot 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).

Timeplot per PRN

```
gfzrnx -finp stas0400.15o -stk_epo 1800
gfzrnx -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
```

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.

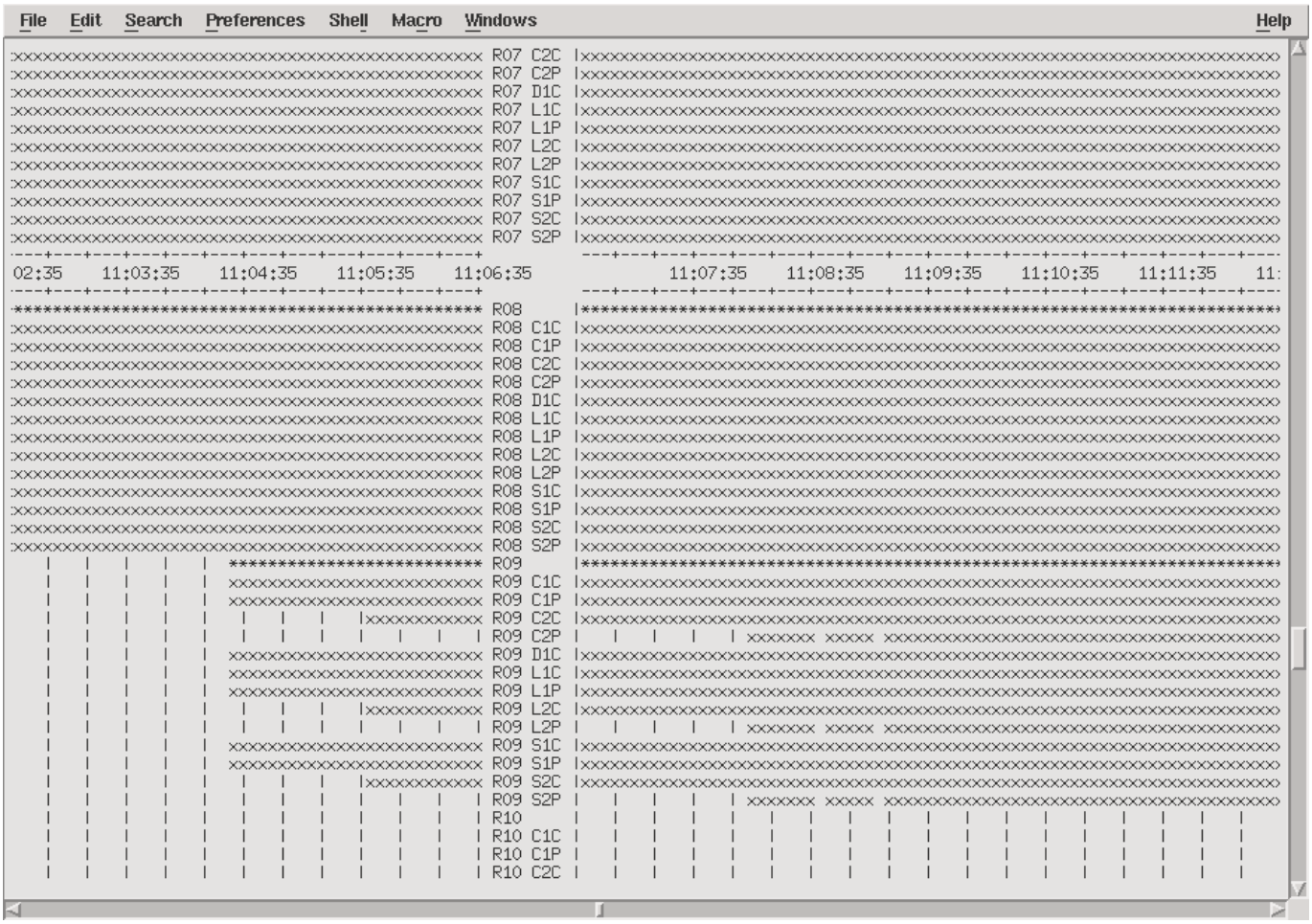
```
gfzrnx -finp stas0400.15o -stk_epo 1800:prn,otp -satsys E -ot C,L
```

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

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



3.4 RINEX File Format Conversion

3.4.1 RINEX OBS File Format Conversion (4, 3, 2 to 2, 3, 4)

Observation Types Mapping

The used observation types mapping is hard-coded 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 columns 2, 3 and 4 is treated as a comment only and is not used.

> **Remark**
>
> During the conversion process, the data values - observation, loss of lock indicator (LLI), signal strength indicator (SSI) - are left as they are. The LLI meaning differs between versions 2 and 3 or 4 and the Interpretation of bits 1 and 2 has to be used with caution!

RINEX-2 to RINEX-3 or 4

Please use this conversion only if you are sure that the output files are usable in the environment the data are supplied to! The output format for this conversion/transition is RINEX-3.01 to be standard-compliant. The 2-characters 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 compliant 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/ 4
G	P1	C1W
G	C1	C1C
G	P2	C2W
G	C2	C2C

System	RINEX-2	RINEX-3/ 4
R	P1	C1P
R	C1	C1C
R	P2	C2P
R	C2	C2C

This is used because both **Px** and **Cx** code types are mapped to the single **Cx?** RINEX-3 or 4 code type.

RINEX-3 or 4 to RINEX-2

The RINEX-2 output version is 2.11.

Use the **--version_out** or **-vo** command line parameter to define the 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
```

SPECIFIC OBSERVATION TYPE SELECTION

In the RINEX-3 or 4 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 or 4 observation 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
```

OBSERVATION TYPE SELECTION VIA SIGNAL PRIORITIES

By default, the following signal priorities per frequency and satellite system are used for the RINEX-3 or 4 to RINEX-2 conversion:

Sat. System	Freq. Num.	RINEX-3 Signal Priority
G - GPS	1	PRWCSLXYMN
G - GPS	2	PRWCDSLXYMN
G - GPS	5	IQX
.		
R - GLO	1	PC
R - GLO	2	PC
R - GLO	3	IQX
R - GLO	4	ABX
R - GLO	6	ABX
.		
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	2	IQX
C - BDS	5	DPX
C - BDS	6	IQX
C - BDS	7	IQX
C - BDS	8	DPX
.		
I - IRN	5	ABCX
I - IRN	9	ABCX
.		
S - SBS	1	C
S - SBS	5	IQX

The observation code's priority is **LCDS**: phase, code, doppler and signal strength. It defines the basis for the selection of the other observation types of that frequency, if existing. You can update the internal signal priority list by 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.

USED OBSERVATION TYPES

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

```
************************************************************ COMMENT
*                 WARNING - FORMAT CONVERSION              * COMMENT
* -------------------------------------------------------- * COMMENT
*  The data values: observation, loss of lock (LLI) and    * COMMENT
*  signal strength (SSI) indicators are left as they are. * COMMENT
*    The LLI meaning differs between versions 2 and 3      * COMMENT
*    and the Interpretation of bit 1 and 2 has to be       * COMMENT
*                   used 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
...
```

REMARK

To avoid the selection of an observation type with sparse observations using **Signal Priorities** mode, it can be useful to add the `-rsot` command line parameter (remove sparse observations 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
```

3.4.2 RINEX NAV File Format Conversion (3.04/3.05)

There is a significant change in the GLONASS broadcast records in the RINEX version 3.05 (one additional record).

The GFZRNX output version is always the highest supported one. For the RINEX-3 standard output, a "dummy" record is added to be 3.05 standard compliant.

If you can't use this latest 3.05 version because e.g. your software does not support this, you can create a 3.04-formatted file via the `-vo`, `-version_out` command line parameters.

```
gfzrnx -finp enao080a00.21G -fout enao080a00.21G -vo 3.04
```

This works for splice operations too.

```
gfzrnx -finp enao080a00.21* -fout ENAO00XXX_R_20210800015_15M_MN.rnx -vo 3.04
```

3.5 Rinex File Nominal Renaming Support

3.5.1 RINEX File Nominal Renaming Support (2, 3 or 4)

A fast file name conversion of RINEX-3 or 4 files with RINEX-2 style file names to RINEX-3 or 4 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 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
```

There are default mappings from the extension letter to the RINEX-3 data type identifier:

Extension	Data Type
o	_MO.rnx
d	_MO.crx
n	_GN.rnx
g	_RN.rnx
l	_EN.rnx
c	_CN.rnx
q	_JN.rnx
j	_JN.rnx
h	_SN.rnx
p	_MN.rnx
m	_MM.rnx

All other extension letters end up with **_XX.rnx**.

```
gfzrnx -finp pots1230.15b -nomren23 DEU,12
POTS12DEU_R_20151230000_01D_XX.rnx
```

To support additional extensions, these default mappings can be overwritten or extended via the `-extsysdt23` command line parameter, providing a comma-separated list of extension letter-colon-data type pairs.

```
gfzrnx -finp pots1230.15b -nomren23 DEU,12 -extsysdt23 b:SA,j:JN
POTS12DEU_R_20151230000_01D_SA.rnx
```

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 are 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 or 4 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
```

results in 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
```

Remark

Information provided via the command line has priority.

3.6 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 similarly for navigation and meteo files.

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

3.6.1 Observation Data Splice Specials

There are two different splice modes available

Standard Mode (default)

The input file order is derived automatically. In case of overlapping input files, the file with fewer epochs is preferred. This will allow the splice of resubmitted files into an existing "big" file. All output data records are stored in RAM to allow a full data statistics output in the header while reading any input file only once. The output data types are derived from input statistics. This allows omitting "empty" observation types.

Fast / RAM save Mode (`-splice_memsave`)

Via the `-splice_memsave` just the pure line-by-line output data block is stored in RAM for a fast output after the RINEX output header is written. Empty observation types are left in the output files because the observation types from the input header information are used to derive the output observation types.

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

Direct Mode (`-splice_direct`)

Via the `-splice_direct` command line parameter, an epoch by epoch output of the observations data can be reached, which leads to a small RAM utilization. Using this mode, a full data statistics header output is impossible.

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

Try Append (`-try_append`)

The `-try_append n` command line parameter initiates an initial check over all input files if append to the first file is possible. This can be useful in environments where e.g. a daily file is accumulating e.g. hourly files with time. In case of the append mode, the process will be significantly faster. The parameter of **-try_append** is the shortest nominal file duration (s)of the part files to be appended (e.g. **3600** for hourly files or **900** for 15-min files).

```
gfzrnx -finp pots007[a-x].14o -fout pots0070.14o -kv -try_append 3600
gfzrnx -finp pots007[a-x].14o -fout pots0070.14o -kv -try_append 3600 -splice_direct
```

3.6.2 Navigation Data Splice Specials

The navigation data splice is based on a majority filter for redundant navigation data records. There is a statistics table in the file header giving information about how many files contributed to the outputs per PRN. This can be useful in the case of creating navigation summary files for e.g. one day.

The header statistics table can be avoided via the `-no_nav_stk` command line parameter.

Here is shown an example header statistics table for BDS only:

```
B_TOP                                                  COMMENT
B_TOP   ---------------------------------------------      COMMENT
B_TOP    # of NAV. EPOCHS based on # of FILE CONTRIB.       COMMENT
B_TOP   ---------------------------------------------      COMMENT
B_TOP                                                  COMMENT
B_BEG C                                                COMMENT
B_HD  BTP C   MTP    =1   `<5  <10  <15  <25  <50 >`=50  #EPO COMMENT
B_LN  --- C ------------------------------------------------ COMMENT
B_STK EOP C19 CNVX    .    .    1    .    .    .    .    94 COMMENT
B_STK EOP C20 CNVX    .    .    1    .    .    .    .    72 COMMENT
...
B_STK EOP C45 CNVX    .    1    .    .    .    .    .    75 COMMENT
B_STK EOP C46 CNVX    1    .    1    .    .    .    .   106 COMMENT
B_STK EPH C01 D2      1   13   11    .    .    .    .    25 COMMENT
B_STK EPH C02 D2      1    .   12   12    .    .    .    25 COMMENT
...
B_STK EPH C56 D1      .    1    .    .    .    .    .     1 COMMENT
B_STK EPH C59 D2     13   12    .    .    .    .    .    25 COMMENT
B_STK EPH C60 D2      1   24    .    .    .    .    .    25 COMMENT
B_STK ION C01 D1D2    1    .    .    .    .    .    .     2 COMMENT
B_STK ION C02 D1D2    7    8    .    .    .    .    .    16 COMMENT
...
B_STK ION C59 D1D2    8    .    .    .    .    .    .    10 COMMENT
B_STK ION C60 D1D2    4    .    .    .    .    .    .     4 COMMENT
B_STK STO C01 D1D2    1    .    .    .    .    .    .     2 COMMENT
B_STK STO C02 D1D2    1    .    .    .    .    .    .    16 COMMENT
...
B_STK STO C46 D1D2    1    2    .    .    .    .    .    15 COMMENT
B_STK STO C59 D1D2    2    .    .    .    .    .    .    10 COMMENT
B_LN  --- C ------------------------------------------------ COMMENT
B_SUM EOP C   CNVX                                   2174 COMMENT
B_LN  --- C ------------------------------------------------ COMMENT
B_SUM EOP C                                          2174 COMMENT
B_LN  --- C ------------------------------------------------ COMMENT
B_SUM EPH C   CNV1                                    638 COMMENT
B_SUM EPH C   CNV2                                    641 COMMENT
B_SUM EPH C   CNV3                                    537 COMMENT
B_SUM EPH C   D1                                      932 COMMENT
B_SUM EPH C   D2                                      175 COMMENT
B_LN  --- C ------------------------------------------------ COMMENT
B_SUM EPH C                                          2923 COMMENT
B_LN  --- C ------------------------------------------------ COMMENT
B_SUM ION C   CNVX                                   2174 COMMENT
B_SUM ION C   D1D2                                    544 COMMENT
B_LN  --- C ------------------------------------------------ COMMENT
B_SUM ION C                                          2718 COMMENT
```

```
B_LN  --- C ------------------------------------------------ COMMENT
B_SUM STO C   CNVX                                    2174 COMMENT
B_SUM STO C   D1D2                                     482 COMMENT
B_LN  --- C ------------------------------------------------ COMMENT
B_SUM STO C                                           2656 COMMENT
B_LN  --- C ------------------------------------------------ COMMENT

...

B_ALL EOP -   -                                       3414 COMMENT
B_ALL EPH -   -                                      17109 COMMENT
B_ALL ION -   -                                       5455 COMMENT
B_ALL STO -   -                                       5207 COMMENT
                                                           COMMENT
        33                                                 MERGED_FILE
```

Navigation Data Epoch Filter

Use the `-nav_epo_filter` command line parameter to filter the navigation records. Only records with **standard** epochs are left in the output file.

Use the `-nav_epo_strict` command line parameter to filter the navigation records. Only records with **nominal** epochs are left in the output file.

Use the `-nav_latest` command line parameter to filter the navigation records. Only the latest record per PRN is left in the output file. In this case, the header statistics tables are omitted and the default **prn** output sorting is used.

Navigation Data output for GLONASS

With Rinex version 3.05 an additional record was introduced to the GLONASS navigation data block.

Use the `-vo 3.04` command line parameter to output version 3.04 omitting the additional GLONASS record to be compatible with your existing navigation data environment.

> **Remark - Splice/Split**
>
> It is possible to combine the **splice** and **split** operation of **observation data** via a single command line call.

Here is an example of splicing e.g. 15 min input files and splitting 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..

> **Remark - Filename Expansion - 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]` or `{a...x}` can be used too, depending on the used command shell.

> **Remark - Filename Expansion - Microsoft 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.

3.7 RINEX File Split

The RINEX file split can be initiated by providing a split interval in seconds via the -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
```

Remark

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

3.8 RINEX File Output Epoch Interval

3.8.1 Supported Date/Time/Epoch Formats

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

Time

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

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

3.8.2 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 is 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
```

3.8.3 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 that do not fit 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.

3.8.4 Extend Navigation File Boundaries (`-enb`)

Navigation information files often contain records that 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
```

3.9 RINEX File Manipulation

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

3.9.1 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
```

LLI shift

The LLIs (Loss of Lock Indicator) of the unused data epochs between two sample epochs are shifted to the sample epoch if you provide the `-smp_lli_shift` command line parameter. Otherwise, the LLIs of the sample epoch data are left as they are, and the information is lost. The use of this option slows down the sampling operation.

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

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

3.9.2 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 desired 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
```

3.9.3 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
```

3.9.4 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 desired observation types via a comma-separated list of patterns.

The observation types selection works via a pattern matching mode. The pattern matching is done left aligned (e.g. L,L2,L2C or 1,1C).

Here are some examples:

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
```

RINEX-3\/4

In a simple case, it works the 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 fully replaces a global one.

Here is a global selection overall satellite systems (simple mode) selecting phase and code observations only:

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

Here is a selection of frequencies only:

```
gfzrnx ... -obs_types 1,2
```

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

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

3.9.5 Remove of Sparse Observation Types (`-remove_sparse_obs_types`)

This option works on a satellite system or file level. 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 of a satellite system

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

3.9.6 Remove of Sparse Observation Types per PRN (`-rsot_prn`)

This option works on the PRN level. One can give a limit in % which can be used to eliminate sparse observation types on the PRN level. The basis is the median of the number of observations per single observation type of a satellite system.

```
gfzrnx -f -finp SVTL00RUS_R_20240840000_01D_30S_MO.rnx -fout SVTL00RUS_R_20240840000_01D_30S_MO.rnx_filtered -rsot_prn
20
```

The data statistics show here the sparse obs. types.

```
   G12   854   854   854   854   837   837   837   837   853PRN / # OF OBS
         853   853   853   837   837   837   837     1     1PRN / # OF OBS
           1     1                                         PRN / # OF OBS
...
   R10   945   945   945   945   945   945   945   945    12PRN / # OF OBS
          27    27    12     0     0     0     0     0     0PRN / # OF OBS
           0     0                                         PRN / # OF OBS
```

After using the option `-rsot_prn` sparse data are eliminated and your statistics should look like the following:

```
   G12   854   854   854   854   837   837   837   837   853PRN / # OF OBS
         853   853   853   837   837   837   837           PRN / # OF OBS
                                                           PRN / # OF OBS
...
   R10   945   945   945   945   945   945   945   945     PRN / # OF OBS
                                                           PRN / # OF OBS
                                                           PRN / # OF OBS
```

3.9.7 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 this data, use the `-kaot` command line parameter.

3.9.8 Observation Types Sorting (`-ots`)

```
-ots <CPLSD>[:<attribute>]
```

The default observation types output sorting order is alphanumeric. To control the observation types output order (GNSS observation files only) a string of the first observation types letters should be given. To order by frequency first, the following attributes are possible:

attribute	order by
frqasc	frequency & observation type (ascending)
frqdsc	frequency & observation type (descending)
frq `<frq-list>`	comma-separated list of frequencies given in a certain order
froasc	observation type & frequency (ascending)
frodsc	observation type & frequency (descending)
fro `<frq-list>`	comma-separated list of frequencies given in a certain order

Some examples:

```
-ots PCLDS
-ots CL
-ots PCLDS:frqasc
-ots PCLSD:frq1,5,7
-ots PCLDS:frodsc
-ots PCLDS:fro1,5,7
```

The following obs type order on input:

```
G   21 C1C L1C D1C S1C L1P D1P L1W D1W S1W D2C S2C C2W L2W  SYS / # / OBS TYPES
       D2W S2W C2X L2X S2X C5X L5X S5X
```

creates the following output order using different `-ots` parameters:

-ots CPLDS

```
G   21 C1C C2W C2X C5X L1C L1P L1W L2W L2X L5X D1C D2C D1P  SYS / # / OBS TYPES
       D1W D2W S1C S2C S1W S2W S2X S5X                     SYS / # / OBS TYPES
```

-ots CPLDS:frqasc

```
G   21 C1C L1C L1P L1W D1C D1P D1W S1C S1W C2W C2X L2W L2X  SYS / # / OBS TYPES
       D2C D2W S2C S2W S2X C5X L5X S5X                     SYS / # / OBS TYPES
```

-ots CPLDS:froasc

```
G   21 C1C C2W C2X C5X L1C L1P L1W L2W L2X L5X D1C D1P D1W  SYS / # / OBS TYPES
       D2C D2W S1C S1W S2C S2W S2X S5X                     SYS / # / OBS TYPES
```

3.9.9 Navigation File Sorting (`-nav_sort`)

The output order of the navigation records can be controlled via `-nav_sort` or `-ns` command line parameter. Two options **prn**, **time**, **prnmtype** 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.
- In the **prnmtype** mode, the sorting order is by prn message_type 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
```

```
gfzrnx -finp *.rnx -fout splice.rnx -vo 4 -ns prnmtype
```

3.9.10 Navigation Types Selection (`-nav_types`)

With RINEX version 4 navigation record types (EPH, STO, EOP, ION) have been introduced.

Type	Description
EPH	Satellite Orbit Ephemmeries
STO	System Time Offset
EOP	Earth Orientation Parameters
ION	Ionospheric Model Parameters

For every navigation record type message types (e.g. CNAV, LNAV, CNV3, ...) have been introduced additionally. If you are interested in a subset of navigation record types and selected message types only, you can use the `--nav_types` or `-nt` command line parameter to provide your desired selection.

The selection definition consists of a concatenation via '+' of global or satellite-dependent settings. One complete selection element consists of:

```
<satellite_system(s)>::<list_of_nav_types>:<list_of_message_types>
```

It is possible to omit the **satellite_system(s)** or the **list_of_message_types**. In this case, all valid supported elements are used. Lists are built via **dot**. Here are some examples:

```
--nav_types GC::EPH
--nav_types C::ION.STO
--nav_types C::ION.STO:CNVX
--nav_types EPH+C::ION.STO:CNAV
--nav_types C::ION.STO:CNVX+C::EPH:D1.D2.CNV1+E::EPH:INAV
--nav_types EPH.STO
```

Countless selection variations are possible to extract needed information only.

```
gfzrnx -finp *N.rnx -fout BRDC00GFZ_S_20210760000_01D_MN.rnx_EPH_STO -nav_types EPH,STO
gfzrnx -finp *N.rnx -fout BRDC00GFZ_S_20210760000_01D_MN.rnx_ION     -nav_types ION
gfzrnx -finp *N.rnx -fout BRDC00GFZ_S_20210760000_01D_MN.rnx_BDS     -nav_types C::EPH:D1D2+E::EPH:INAV
```

Remark

Additional command line selection options like `-satsys`, `-prn` can be used too.

Supported output versions (`-vo`) for NAV-files are **2**, **3.04**, **3.05** and **4**.

3.9.11 GPSweek Rollover Correction (`-shift_gpsw`)

Due to firmware or Rinex converter problems, we have seen files that show up with data epochs affected by 1024-week rollovers, which leads to data epoch shifts by a multiple of 1024. The week shift to be added must be provided via the `-shift_gpsw` command line parameter. The file name epoch needs to be corrected first before using the `-shift_gpsw` command line parameter. **gfzrnx** checks if the gpsweek difference between the first data epoch and the filename epoch is a multiple of 1024. Only in this case, the epoch shift will be applied.

Here is one example for the file **MAR100DEU_R_20190440015_15M_01S_GO.rnx**, where the gpsweek for **20190440015 (2019 02 13)** is **2040**.

```
     3.03           OBSERVATION DATA    I (IRNSS)           RINEX VERSION / TYPE
Convert 2.4         NovAtel             20190214 093312 UTC PGM / RUN BY / DATE
MAR100DEU                                                   MARKER NAME
MAR1                                                        MARKER NUMBER
gnss@gfz-potsdam.de GFZ                                     OBSERVER / AGENCY
DCH09470100         NOV OEMV1           3.01-TT             REC # / TYPE / VERS
DCH09470100         NOVSMART-V1     NONE                    ANT # / TYPE
G    4 C1C D1C L1C S1C                                      SYS / # / OBS TYPES
     1.000                                                  INTERVAL
...
  1999     6    30     0    15    0.0000000     GPS         TIME OF FIRST OBS
  1999     6    30     0    29   59.0000000     GPS         TIME OF LAST OBS
                                                            END OF HEADER
> 1999 06 30 00 15  0.0000000  0 12      -0.000000000000
G01  24177867.102 6      3413.676   127055545.211 6        41.000
G08  20596455.180 8       791.348   108235118.641 8        49.000
...
> 1999 06 30 00 15  1.0000000  0 13      -0.000000000000
G01  24177217.656 7      3412.410   127052132.391 7        42.000
G08  20596304.750 8       789.719   108234328.086 8        49.000
...
```

The gps-week of **1999 06 30** is **1016 (2040-1016=1024)**. The shift by 1024 weeks results in the correct data epochs.

```
gfzrnx -shift_gpsw 1024 -finp MAR100DEU_R_20190440015_15M_01S_GO.rnx -fout MAR100DEU_R_20190440015_15M_01S_GO.rnx_OK
```

```
     3.04           OBSERVATION DATA    G                   RINEX VERSION / TYPE
Convert 2.4         GFZ ODC             20190214 093312 UTC PGM / RUN BY / DATE
gfzrnx-1.12-2370    FILE CONVERSION     20190214 142041 UTC COMMENT
MAR100DEU                                                   MARKER NAME
MAR1                                                        MARKER NUMBER
gnss@gfz-potsdam.de GFZ                                     OBSERVER / AGENCY
DCH09470100         NOV OEMV1           3.01-TT             REC # / TYPE / VERS
DCH09470100         NOVSMART-V1     NONE                    ANT # / TYPE
G    4 C1C D1C L1C S1C                                      SYS / # / OBS TYPES
     1.000                                                  INTERVAL
...
  2019     2    13     0    15    0.0000000     GPS         TIME OF FIRST OBS
  2019     2    13     0    29   59.0000000     GPS         TIME OF LAST OBS
                                                            END OF HEADER
> 2019 02 13 00 15 00.0000000  0 12      -0.000000000000
G01  24177867.102 6      3413.676   127055545.211 6        41.000
```

```
G08  20596455.180 8       791.348   108235118.641 8        49.000
...
> 2019 02 13 00 15 01.0000000  0 13      -0.000000000000
G01  24177217.656 7      3412.410   127052132.391 7        42.000
G08  20596304.750 8       789.719   108234328.086 8        49.000
...
```

3.9.12 Antenna Rename (`-ant_rename`)

Historical files, especially GPS observation files before the year 2000, use outdated non-IGS-conform antenna names.

With the `-ant_rename` command line parameter, the antenna names can be updated using the fix implemented table below to have IGS-standard conform antenna names in the header. The renaming is documented in the RINEX header via a COMMENT record which is added.

FROM	TO
DORNE MARGOLIN ASH	ASH700936A_M
GEODETIC III L1/L2	ASH700718A
GEODETIC L1/L2 L	ASH700228A
GEODETIC L1/L2 P	ASH700228D
MARINE/RANGE	ASHMAR/RANGE
A-C L1	ASHAC_L1
A-C L1/L2	ASHAC_L1/L2
ASH701945.02B	ASH701945B_M
ASH701946.012	ASH701946.2
ASH701946.022	ASH701946.2
ASH701975.01Agp	ASH701975.01AGP
TR GEOD L1/L2 GP	TRM22020.00+GP
TR GEOD L1/L2 W/O GP	TRM22020.00-GP
TRM10877.10+RGP	TRM12333.00+RGP
JPSMARANT_GGD	JNSMARANT_GGD
TRM10877.10+SGP	TRM11877.10+SGP
DORNE MARGOLIN LEICA	LEIAT504
LEICA AT201	LEIAT201
LEICA AT202	LEIAT202-GP
LEICA AT302	LEIAT302-GP
LEICA AT202 GP	LEIAT202+GP
LEICA AT302 GP	LEIAT302+GP
LEICA AT303	LEIAT303
LEICA AT501	LEIAT501
LEICA AT502	LEIAT502
LEICA AT503	LEIAT503
MAGELLAN PM-500	MAGPM-500
M-PULSE L1/L2 SURVEY	MPLL1/L2_SURV
MACROMETER X-DIPOLE	MAC4647942
MINIMAC PATCH	MACPATCH
DORNE MARGOLIN B	AOAD/M_B
DORNE MARGOLIN R	JPLD/M_R
DORNE MARGOLIN T	AOAD/M_T

FROM	TO
TOPCR3_GGD	TPSCR3_GGD
4000SE INTERNAL	TRM17200.00
4000SL MICRO	TRM12333.00+RGP
4000SLD L1/L2	TRM12562.00+SGP
4000ST INTERNAL	TRM4000ST_INT
4000ST KINEMATIC	TRM14156.00-GP
4000ST L1 GEODETIC	TRM14177.00
4000ST L1/L2 GEOD	TRM14532.00
4000SX MICRO	TRM11877.10+SGP
DORNE MARGOLIN TRIM	TRM29659.00
STXS9+X001A	STXS9PX001A

3.9.13 Antenna Rename Table output (`-ant_rename_out`)

The table for the antenna renaming can be extended or corrected. Via the command line parameter `-ant_rename_out` one can get the currently used table for extension or correction. The output file is in **json** format.

```
gfzrnx -ant_rename_out
{
    "4000ST L1 GEODETIC"    : "TRM14177.00",
    "MINIMAC PATCH"         : "MACPATCH",
    ....
    "MAGELLAN PM-500"       : "MAGPM-500",
    "TR GEOD L1/L2 W/O GP" : "TRM22020.00-GP"
}
```

For a direct file output use:

```
gfzrnx -ant_rename_out -fout ant_rename.json
```

3.9.14 Antenna Rename Table input (`-ant_rename_inp`)

If you want to use an own or extended renaming table, you can provide it via the `-ant_rename_inp` command line parameter. It overwrites completely the internal table. The input file must be in **json** format.

```
gfzrnx -kv -finp pots0030.95o -fout pots0030.95o_new -ant_rename_inp ant_rename.json
```

3.10 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::` , `::RX4::`) or the `::INP::` variable. The `::INP::` means, that the output file name is the same as the input file name.

Here is 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 -sifl


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

3.11 RINEX File Header/Data Editing

RINEX file header editing can be invoked by 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 are performed, but the data part is simply copied as it is.

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

3.11.1 Header Editing (Standard)

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

3.11.2 Header Editing (Only)

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

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

An additional epoch and station identifier have to be given if no standard RINEX file names are used. If no additional information provided, the **MARKER NAME** and the first data epoch is used if it exists. 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
gfzrnx -finp file.rnx -fout file.rnx_hded -crux header_crux.txt -hded -epo_beg 2015234_000000 \
                                                                        -site POTS00DEU
```

3.11.3 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 par * Station-, data type- and epoch interval-dependent settings in a single configuration file are possible

3.11.4 Show Config. File Interpretation (`-show_crux`)

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

3.11.5 Configuration file

Formally there are 3 major modes: **update_insert**, **replace** or **rename** delimited by colon.

In case of **rename** a type (prn|obs) has to be given additionally. The mode definition line has to be followed by an optional data type identifier string (OMN Obs., Met., Nav.) delimited with a hyphen, an optional epoch interval delimited by a hyphen, and a valid station identifier (4- or 9-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...] :
    [OMN-][YYYYDDD:SSSSS   YYYYDDD:SSSSS-]   STA1MRCCC[.STA2MRCCC[.STA3MRCCC...] :
...

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...]
    [ON-][YYYYDDD:SSSSS   YYYYDDD:SSSSS-]   - <prn-from> - <prn-to> : STA1MRCCC[.STA2MRCCC[.STA3MRCCC...]


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...]
    [OM][YYYYDDD:SSSSS   YYYYDDD:SSSSS-]   <obs-from> - <obs-to> - <sat.sys> : STA1MRCCC[.STA2MRCCC[.STA3MRCCC...]
```

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. Currently, only the 4 characters 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)
- 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.

Update - Single Header Element

Single header element update/insert can be done by 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 :
#--------------
    O - POTS.OUST.WINT:
    "REC # / TYPE / VERS"   : { 1 : "TRIMBLE NETR9" }

    O - 2015209:00000 0000000:00000 - MIZT00JPN:
    "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" }

    O - POTS00DEU:
    "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 element's format length!
- The first header element is at index 0
- The **site** name used to **search** for site-dependent settings in the loaded **crux** information is extracted from the standard RINEX-2 or 3 input file name. In case of wrong or non-standard input file names or in pipe environments, the site name has to be provided via the `-site` command line parameter. The `-site` parameter overwrites any otherwise derived site name in general.

```
gfzrnx -finp xxxx282a.19o -crux crux.txt -fout ::RX3:: -kv -f -site MET300FIN
cat      MET300FIN_R_20192820000_01H_30S_MO.rnx   |gfzrnx -crux crux.txt -fout ::RX2:: -kv -f \
                                                                               -site MET300FIN
gzip -dc MET300FIN_R_20192820000_01H_30S_MO.rnx.gz|gfzrnx -crux crux.txt -fout ::RX3:: -kv -f \
                                                                               -site MET3
```

Supported String Substitutes

The following variable string substitutes are supported to be used via **crux** single header elements updates and **added** COMMENT lines. To be more independent of OS-derived values, the following environment variables are used with a higher preference if existing.

Substitute String	Substitute/Example	Description	Environment variables
uSeR	nisn	user name provided by os	USERNAME, USER
pRoGrAm	gfzrnx-1.08-8003	gfzrnx-version-revision	
hOsTnAmE	serv01	simple hostname provided by os	HOSTNAME
hOsTdOmAiNnAmE	serv01.gfz-potsdam.de	fully qualified hostname provided by os	HOSTFQDN
dOmAiNnAmE	gfz-potsdam.de	domain name provided by os	USERDOMAIN
tImEsTaMp	20170712 113126 UTC	time stamp of current time	

Warning

Please check in advance if you get the expected results for your operating system!

```
update_insert :
#---
MNO - ALL:
```

```
          "COMMENT"             : "PG tImEsTaMp pRoGrAm uSeR@dOmAiNnAmE"
          "PGM / RUN BY / DATE" : { 0: "pRoGrAm" , 1: "uSeR@dOmAiNnAmE", 2: "tImEsTaMp" }
```

For the upper configuration, the "PGM / RUN BY / DATE" record will be updated and the "COMMENT" record below will be added:

```
PG 20170712 120203 UTC gfzrnx-1.08-7179 nisn@gfz-potsdam.de COMMENT
gfzrnx-1.08-7179    nisn@gfz-potsdam.de 20170713 065255 UTC PGM / RUN BY / DATE
```

If the "COMMENT" string gets longer than 60 characters, it will be cut to 60!

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 is 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 is 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 - POTS00DEU :

    "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 of a header manipulation with the initial header and the manipulation result.

```
gfzrnx -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
```

Proposed Use

There are several possibilities to organize the header editing configuration file. The clearest 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 Observation and Meteo 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.

> **Remark**
>
> There is one exception 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" }
```

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 - POTS00DEU:
    "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.

> **Remark**
>
> **COMMENT** lines are **inserted only**

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. Afterward, 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"
```

To remove 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   : ""
```

Rename - PRNs

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

Here are 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
```

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
```

Remark

You can use 9-char. station names in crux-config-file for the handling of 4-char. station names too! The **replace** mode is done directly on input, the **update_insert** and **rename** modes are done after the whole header has been read.

3.11.6 Header edit via command line (`-cx_updins`)

Single **update_insert** header edit options can also be provided via command line using the `-cx_updins` command line parameter, providing a list of edit options. The site definition has to be given before the header label change option.

See an example below **Unix**:

```
gfzrnx -finp /data1/VALD00CAN_R_20181001200_01H_30S_MO.rnx \
        -fout /data1/VALD00CAN_R_20181001200_01H_30S_MO.rnx.hded -hded -cx_updins \
'O - VALD: "APPROX POSITION XYZ" : { 0:"3800689.6341", 1:"882077.3857", 2:"5028791.3179" }' \
'O - VALD: "REC # / TYPE / VERS" : { 0 : "", 1 : "JAVAD TRE_G3TH DELTA", 2 : "3.6.3 Jul,01,2017" }'
```

See an example below **Windows** (swapped single and double quotes):

```
gfzrnx -finp /data1/VALD00CAN_R_20181001200_01H_30S_MO.rnx \
       -fout /data1/VALD00CAN_R_20181001200_01H_30S_MO.rnx.hded -hded -cx_updins \
"O - VALD: "APPROX POSITION XYZ" : { 0:'3800689.6341', 1:'882077.3857', 2:'5028791.3179' }" \
"O - VALD: "REC # / TYPE / VERS" : { 0 : '', 1 : 'JAVAD TRE_G3TH DELTA', 2 : '3.6.3 Jul,01,2017' }"
```

Remark

Please pay attention to the different **single/double quote** usage on **Windows**- and **Unix**-based operating systems. Please check in advance with `-show_crux` the acceptance of your header edit options due to the mixture of different quotation marks after `-cx_updins`.

```
gfzrnx -show_crux -cx_updins \
'O - VALD: "APPROX POSITION XYZ" : { 0:"3800689.6341", 1:"882077.3857", 2:"5028791.3179" }' \
'O - VALD: "REC # / TYPE / VERS" : { 0 : "", 1 : "JAVAD TRE_G3TH DELTA", 2 : "3.6.3 Jul,01,2017" }'
```

Remark

COMMENT lines are **inserted only**

3.11.7 Internal/Data Headers via crux-file (`-cx_addinthd`)

Metadata changes following e.g. hardware changes can be introduced at the event epochs into the data part of a RINEX file if information is found in the crux-file. This mechanism can be activated additionally to the normal header edit operations via the `-cx_addinthd` command line parameter for **update_insert** crux-settings.

Here is an example:

```
gfzrnx -cx_addinthd -crux obwt_crux.txt -finp obwt107g.18o -fout obwt107g.18o_crx
```

The following crux-configuration

```
update_insert:
O - 20141105:071700 20180417:060500 - OBWT:
    "REC # / TYPE / VERS"    : { 0: "4831K57521", 1: "TRIMBLE NETR5", 2: "Nav 4.87 / Boot 4.18"}
    "ANT # / TYPE"           : { 0: "30767802",   1: "TRM55971.00",   2: "TZGD"}
O - 20180417:061500 00000000:000000 - OBWT:
    "REC # / TYPE / VERS"    : { 0: "1705310",    1: "LEICA GR30",    2: "4.20.232"}
    "ANT # / TYPE"           : { 0: "09440002",   1: "LEIAR25.R3",    2: "LEIT" }
```

will lead to file header records of e.g.:

```
4831K57521            TRIMBLE NETR5       Nav 4.87 / Boot 4.18REC # / TYPE / VERS
30767802              TRM55971.00     TZGD                        ANT # / TYPE
```

and a header block in the data part of a RINEX-2 file of:

```
    23913577.070   127921488.413 6  99494529.138 8  23913582.523          42.100
        33.300
    23773818.648   127129528.196 4                                       38.700

    18 04 17 06 15 00.0000000  4  2
1705310                 LEICA GR30          4.20.232            REC # / TYPE / VERS
09440002                LEIAR25.R3      LEIT                    ANT # / TYPE
    18 04 17 06 15 00.0000000  0 16G02G05G07G09G13G27G28G30R06R07R08R09
                              R10R16R23R24
    24247477.484   127421298.588 6  99289349.307 6  24247479.359          42.200
        25.500
    21028794.141   110507030.196 7  86109402.765 9  21028797.266          49.300
```

Remark

Windows users should **swap single and double quotes** using `-cx_addinthd` similar to `-cx_updins` command line options.

3.11.8 Manipulate Header Version Number (`-vnum`)

By default, the latest supported version number is used for the "**RINEX VERSION / TYPE**" header element, and there are made manipulations to fit this version. If a special version number is needed (for whatever reason) one can use the `-vnum` command line parameter to manipulate the version number to a certain value.

```
     3.04          OBSERVATION DATA    M                 RINEX VERSION / TYPE
```

```
gfzrnx -finp ... -vnum 3.03
```

```
     3.03          OBSERVATION DATA    M                 RINEX VERSION / TYPE
```

This will change the default output header value e.g. **3.04** to the wished value of **3.03**.

Remark

The `-vnum` version number change is only a formal exchange of the version number to meet any conditions of external software. The file content will be still conform to the highest supported version number!

3.12 RINEX File Metadata Extraction (`-meta`)

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

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

- The **basic** mode extracts only the header information, the first and last epoch from the RINEX file without reading the whole file (fast).
- The **medium** extends the basic information by real data interval, first/last epochs, and number of epochs.
- 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.
- The following output formats are supported: **txt**(default), **json**, **jsonp**(pretty json), **xml**, **dump**. They can be used for fast view or further applications.

The **file-**, **site-**, **receiver-**, **antenna-** sections information is derived from the RINEX header part only. The data-section holds information derived from the RINEX data part.

Here are 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
```

```
gfzrnx -finp POTS00DEU_00001024_FRO_RX3_MO_20180305_000000_01D_30S_GFZ.rnx -meta basic:txt
antenna:
        height:
            e = 0.0000
            h = 0.1206
            n = 0.0000
        name = JAV_RINGANT_G3T
        number = 316
        radome = NONE
data:
        epoch:
            first = 2018 03 05 00 00 00.0000000
            interval = 30.000
            last = 2018 03 05 23 59 30.0000000
exec:
        date = 2018-03-06 15:35:05 UTC
        meta = basic
        name = gfzrnx
        version = 1.10-7323
file:
        epo_first = 2018 03 05 00 00 00.0000000
        interval = 30.000
        md5 = 9a49ad078b4bcfbe1d1a2fe4de440de1
        name = POTS00DEU_00001024_FRO_RX3_MO_20180305_000000_01D_30S_GFZ.rnx
        pgm = JPS2RIN v.2.0.134
        pgm_date = 20180305 011547 UTC
        pgm_runby = GFZ ODC
        satsys = EGR
        site = POTS00DEU
        source = R
        sysfrq:
            E = 1 5
            G = 1 2 5
            R = 1 2
        sysobs:
            E = C1X C5X D1X D5X L1X L5X S1X S5X
            G = C1C C1W C2W C2X C5X D1C D1W D2W D2X D5X L1C L1W L2W L2X L5X S1C S1W S2W S2X S5X
            R = C1C C1P C2C C2P D1C D1P D2C D2P L1C L1P L2C L2P S1C S1P S2C S2P
```

```
        system = M
        systyp:
            E = C D L S
            G = C D L S
            R = C D L S
        type = O
        version = 3.03
receiver:
        firmware = 3.6.7
        name = JAVAD TRE_G3TH DELTA
        number = 205
site:
        agency = GFZ
        name = POTS
        number = 14106M003
        observer = GFZ
        position:
            x = 3800689.6341
            y = 882077.3857
            z = 5028791.3179
```

```
gfzrnx -finp pots0070.15o -meta basic:jsonp

    {"antenna":{"height":{"e":"0.0000","h":"0.1206","n":"0.0000"},"name":"JAV_RINGANT_G3T",
    "number":"316","radome":"NONE"},"data":{"epoch":{"first":"2018 03 05 00 00 00.0000000",
    "interval":"30.000","last":"2018 03 05 23 59 30.0000000"}},"exec":{"date":"2018-03-06 16:56:40 UTC",
    "meta":"basic","name":"gfzrnx","version":"1.10-7323"},"file":{"epo_first":
    "2018 03 05 00 00 00.0000000","interval":"30.000","md5":"9a49ad078b4bcfbe1d1a2fe4de440de1",
    "name":"POTS00DEU_00001024_FRO_RX3_MO_20180305_000000_01D_30S_GFZ.rnx","pgm":"JPS2RIN v.2.0.134",
    "pgm_date":"20180305 011547 UTC","pgm_runby":"GFZ ODC","satsys":"EGR","site":"POTS00DEU",
    "source":"R","sysfrq":{"E":["1","5"],"G":["1","2","5"],"R":["1","2"]},
    "sysobs":{"E":["C1X","C5X","D1X","D5X","L1X","L5X","S1X","S5X"],"G":["C1C","C1W","C2W","C2X",
    "C5X","D1C","D1W","D2W","D2X","D5X","L1C","L1W","L2W","L2X","L5X","S1C","S1W","S2W","S2X",
    "S5X"],"R":["C1C","C1P","C2C","C2P","D1C","D1P","D2C","D2P","L1C",
    "L1P","L2C","L2P","S1C","S1P","S2C","S2P"]},"system":"M","systyp":{"E":["C","D","L","S"],
    "G":["C","D","L","S"],"R":["C","D","L","S"]},"type":"O","version":"3.03"},
    "receiver":{"firmware":"3.6.7","name":"JAVAD TRE_G3TH DELTA","number":"205"},
    "site":{"agency":"GFZ","name":"POTS","number":"14106M003","observer":"GFZ",
    "position":{"x":"3800689.6341","y":"882077.3857","z":"5028791.3179"}}}
```

```
gfzrnx -finp pots0070.15o -meta basic:jsonp

{
    "antenna" : {
        "height" : {
            "e" : "0.0000",
            "h" : "0.1206",
            "n" : "0.0000"
        },
        "name" : "JAV_RINGANT_G3T",
        "number" : "316",
        "radome" : "NONE"
    },
    "data" : {
        "epoch" : {
            "first" : "2018 03 05 00 00 00.0000000",
            "interval" : "30.000",
            "last" : "2018 03 05 23 59 30.0000000"
        }
    },
    "exec" : {
        "date" : "2018-03-06 16:55:57 UTC",
        "meta" : "basic",
        "name" : "gfzrnx",
        "version" : "1.10-7323"
    },
    "file" : {
        "epo_first" : "2018 03 05 00 00 00.0000000",
        "interval" : "30.000",
        "md5" : "9a49ad078b4bcfbe1d1a2fe4de440de1",
```

```
        "name" : "POTS00DEU_00001024_FRO_RX3_MO_20180305_000000_01D_30S_GFZ.rnx",
        "pgm" : "JPS2RIN v.2.0.134",
        "pgm_date" : "20180305 011547 UTC",
        "pgm_runby" : "GFZ ODC",
        "satsys" : "EGR",
        "site" : "POTS00DEU",
        "source" : "R",
        "sysfrq" : {
            "E" : [
            "1",
            "5"
            ],
            "G" : [
            "1",
            "2",
            "5"
            ],
            "R" : [
            "1",
            "2"
            ]
        },
        "sysobs" : {
            "E" : [
            "C1X",
            "C5X",
            "D1X",
            "D5X",
            "L1X",
            "L5X",
            "S1X",
            "S5X"
            ],
            "G" : [
            "C1C",
            "C1W",
            "C2W",
            "C2X",
            "C5X",
            "D1C",
            "D1W",
            "D2W",
            "D2X",
            "D5X",
            "L1C",
            "L1W",
            "L2W",
            "L2X",
            "L5X",
            "S1C",
            "S1W",
            "S2W",
            "S2X",
            "S5X"
            ],
            "R" : [
            "C1C",
            "C1P",
            "C2C",
            "C2P",
            "D1C",
            "D1P",
            "D2C",
            "D2P",
            "L1C",
            "L1P",
            "L2C",
            "L2P",
            "S1C",
            "S1P",
            "S2C",
            "S2P"
```

```
            ]
        },
        "system" : "M",
        "systyp" : {
            "E" : [
            "C",
            "D",
            "L",
            "S"
            ],
            "G" : [
            "C",
            "D",
            "L",
            "S"
            ],
            "R" : [
            "C",
            "D",
            "L",
            "S"
            ]
        },
        "type" : "O",
        "version" : "3.03"
    },
    "receiver" : {
        "firmware" : "3.6.7",
        "name" : "JAVAD TRE_G3TH DELTA",
        "number" : "205"
    },
    "site" : {
        "agency" : "GFZ",
        "name" : "POTS",
        "number" : "14106M003",
        "observer" : "GFZ",
        "position" : {
            "x" : "3800689.6341",
            "y" : "882077.3857",
            "z" : "5028791.3179"
        }
    }
}
```

3.13 RINEX File Comparison (`-fdiff`)

The comparison of single-site RINEX files of the same time interval and files from different sources (e.g. real-time data, data from different rinex-converters, ...) are often not possible easily. **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, contains 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 SSI values, **absolute** differences are always 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 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 first or the second file.

3.14 RINEX Hatanaka Compressed Files

Hatanaka RINEX compressed files are **not** directly supported, but the Hatanaka RINEX compression or decompression can be combined with **gfzrnx** using the standard in/output (via pipes).

The Hatanaka RINEX compression/decompression utilities **RNXCMP** are free software and can be downloaded from http://terras.gsi.go.jp/ja/crx2rnx.html.

On the following page, you can find some examples of the **RNXCMP** decompression/compression in combination with **gfzrnx** and **gzip** compression.

```
Decompression:
--------------

gunzip -c pots0700.17d.Z | crx2rnx - | gfzrnx -kv          -fout pots0700.17o
gunzip -c pots0700.17d.Z | crx2rnx - | gfzrnx -kv  -smp 30 -fout pots0700.17o

gunzip -c POTS01DEU_R_20170700000_01D_30S_MO.crx.gz | crx2rnx - | gfzrnx -kv         -fout
POTS01DEU_R_20170700000_01D_30S_MO.rnx
gunzip -c POTS01DEU_R_20170700000_01D_01S_MO.crx.gz | crx2rnx - | gfzrnx -kv -smp 30 -fout
POTS01DEU_R_20170700000_01D_30S_MO.rnx


Compression:
------------

gfzrnx -finp pots0700.17o         -kv | rnx2crx - | gzip -c > pots0700.17d.gz
gfzrnx -finp pots0700.17o -smp 30 -kv | rnx2crx - | gzip -c > pots0700.17d.gz

gfzrnx -finp POTS01DEU_R_20170700000_01D_30S_MO.rnx | rnx2crx -           > POTS01DEU_R_20170700000_01D_30S_MO.crx
gfzrnx -finp POTS01DEU_R_20170700000_01D_30S_MO.rnx | rnx2crx - | gzip -c > POTS01DEU_R_20170700000_01D_30S_MO.crx.gz

cat POTS01DEU_R_20170700000_01D_30S_MO.rnx | gfzrnx         | rnx2crx -           >
POTS01DEU_R_20170700000_01D_30S_MO.crx
cat POTS01DEU_R_20170700000_01D_30S_MO.rnx | gfzrnx         | rnx2crx - | gzip -c >
POTS01DEU_R_20170700000_01D_30S_MO.crx.gz

cat POTS01DEU_R_20170700000_01D_01S_MO.rnx | gfzrnx -smp 30 | rnx2crx -           >
POTS01DEU_R_20170700000_01D_30S_MO.crx
cat POTS01DEU_R_20170700000_01D_01S_MO.rnx | gfzrnx -smp 30 | rnx2crx - | gzip -c >
POTS01DEU_R_20170700000_01D_30S_MO.crx.gz
```

3.15 RINEX to Tabular Output

3.15.1 Standard Output

The tabular observations output allows to output a RINEX observations input file into a data table that can be used for simple visualization or for an easier introduction into third-party applications like EXCEL, Matlab, etc... All main options like satellite system selection (`-satsys`) and/or satellites selection (`-prn`) and/or observation types selection (`-obs_types`) and others are supported. It can be used for all RINEX data types (OBS, MET, NAV).

The tabular observation output can be initiated via the `-tab` command line parameter. Here is an example for a single satellite and selected observation types:

Tabular OBS data

```
gfzrnx -finp POTS00DEU_R_20150070000_01D_30S_MO.rnx -tab -fout POTS00DEU_2015007_G03.tab
gfzrnx -finp POTS00DEU_R_20150070000_01D_30S_MO.rnx -tab -fout POTS00DEU_2015007_G03.tab -prn G03 -obs_types L1,L2
```

The last command leads to the following default tabular output, extracting phase observations for the PRN G03:

```
#HD G DATE          TIME          PRN        L1C            L1W            L2W            L2X
OBS G 2015-01-07 07:25:00.0000000 G03  134798128.476  134798125.823  105037501.328  105037506.181
OBS G 2015-01-07 07:25:30.0000000 G03  134629777.213  134629774.487  104906318.473  104906323.263
OBS G 2015-01-07 07:26:00.0000000 G03  134461452.299  134461449.545  104775156.193  104775160.914
OBS G 2015-01-07 07:26:30.0000000 G03  134293160.630  134293157.877  104644019.757  104644024.465
...
```

Every line begins with a line descriptor (**#HD,OBS**):

Line type	Description
#HD	header line with column description
OBS	observation line
NAV	navigation line
MET	meteo line

The first columns are fixed, showing the: + Line Type, + Satellite System, + Date, + Time, + PRN.

This is followed by the **list** of wished or given **observation types** as provided in the satellite system-specific header line order.

Tabular NAV Data

```
gfzrnx -finp *.rnx -fout RNX_C_CNV1.tab -f -vo 4 -tab -nt C::EPH:CNV1
```

The last command leads to the following tabular output, extracting only **BDS EPH**-records of message type **CNV1**

```
#HD S DATE          TIME          NAV PRN MTYP DATA ---------------
NAV C 2021-03-17    00:00:00      EPH C19 CNV1  7.642341079190e-04  1.389732773305e-11  0.000000000000e+00
2.871513366699e-03 ...
NAV C 2021-03-17    01:00:00      EPH C19 CNV1  7.642844575457e-04  1.394262483245e-11  0.000000000000e+00
2.720832824707e-03 ...
NAV C 2021-03-17    02:00:00      EPH C19 CNV1  7.643348653801e-04  1.396838200662e-11  0.000000000000e+00
2.270221710205e-03 ...
...
```

Here the same with a different column separator ';'.

```
gfzrnx -finp *.rnx -fout RNX_C_CNV1.tab -f -vo 4 -tab -tab_sep \; -nt C::EPH:CNV1
```

```
#HD;S;DATE;TIME;NAV;PRN;MTYP;DATA ---------------
NAV;C;2021-03-17;00:00:00;EPH;C19;CNV1;
7.642341079190e-04;1.389732773305e-11;0.000000000000e+00;2.871513366699e-03; ...
NAV;C;2021-03-17;01:00:00;EPH;C19;CNV1;
7.642844575457e-04;1.394262483245e-11;0.000000000000e+00;2.720832824707e-03; ...
NAV;C;2021-03-17;02:00:00;EPH;C19;CNV1;
7.643348653801e-04;1.396838200662e-11;0.000000000000e+00;2.270221710205e-03; ...
...
```

Tabular MET Data

This works similarly to OBS data.

3.15.2 Date/Time Formats

The Date/Time format can be controlled via the `-tab_date` , `-tab_time` command line parameters. The following pattern describes selected Date/Time formats:

Date Pattern	Example	Description
mjd	57029	Modified Julian Date (MJD)
ddd	007	day of year
wwwwd	18263	gps-week,weekday
wwww-d	1826-3	gps-week,weekday
yyyyddd	2015007	year, day of year
yyyy-ddd	2015-007	year, day of year
yyyymmdd	20150107	year, month, day of month
yyyy-mm-dd	2015-01-07	year, month, day of month
yymmdd	150107	2-digit year, month, day of month
yy-mm-dd	15-01-07	2-digit year, month, day of month

Time Pattern	Example	Description
hhmmss	013516.0000000	hour, minutes, seconds
hh:mm:ss	01:35:16.0000000	hour, minutes, seconds
sod	5716.0000000	seconds of day
fod	0.066157407407407	fractions of day

```
gfzrnx ... -tab_out -tab_date ddd -tab_time sod
```

The Date/Time patterns **ddd** and **sod** used above, result in the output below.

```
#HD G DATE    TIME      PRN        L1C           L1W           L2W           L2X
OBS G 007 26700.0000000 G03  134798128.476  134798125.823  105037501.328 9999999999.999
```

```
OBS G 007 26730.0000000 G03  134629777.213  134629774.487  104906318.473  104906323.263
OBS G 007 26760.0000000 G03  134461452.299  134461449.545  104775156.193  104775160.914
OBS G 007 26790.0000000 G03  134293160.630  134293157.877  104644019.757  104644024.465
...
```

3.15.3 Column Separator

By default the column **separator** is the **blank** character. Using the `-tab_sep` command line parameter you can choose any character or even string for column separation. In case of the **blank** column separator all missing/empty data values are replaced by **9999999999.999**, otherwise, they are just empty.

```
gfzrnx ... -tab_out -tab_date ddd -tab_time sod -tab_sep ','
```

The above command gives you a simple CSV output:

```
#HD,G,DATE,TIME,PRN,L1C,L1W,L2W,L2X
OBS,G,007,26700.0000000,G03,134798128.476,134798125.823,105037501.328,105037506.181
OBS,G,007,26730.0000000,G03,134629777.213,134629774.487,104906318.473,104906323.263
OBS,G,007,26760.0000000,G03,134461452.299,134461449.545,104775156.193,104775160.914
OBS,G,007,26790.0000000,G03,134293160.630,134293157.877,104644019.757,104644024.465
OBS,G,007,26820.0000000,G03,134124902.769,134124900.043,104512909.644,104512914.387
```

3.16 RINEX-2 BDS, QZSS, IRNSS support

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

3.16.1 Navigation file extensions

In the RINEX-2 standard, there are no extension letters defined for single system BEIDOU-, QZSS-, IRNSS- single system navigation files. The following characters are used by gfzrnx:

System	Letter	Example
BDS	c	pots0750.17c
QZSS	j	pots0750.17j
IRNSS	i	pots0750.17i

3.16.2 RINEX-2 to RINEX-3 or 4 conversion

The RINEX-3.03 standard (and higher) does not allow an empty attribute identifier (tracking mode or channel) in observation type naming (**tna** - observation type|band/frequency|attribute). Converting files from RINEX-2 to RINEX-3 shows the problem of safely map 2-characters to 3-characters observation 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.

3.16.3 Handling of unsupported observation types

gfzrnx s driven by hard-coded observation types, and the mapping table is compliant with RINEX standards. Running the program for unsupported or non-standard observation types results in the omitting of these data. To avoid this behavior, one has to extend the standard. This can be done with the following procedure:

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

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

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

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

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 are **for internal use only**!