GFZRNX 1.12 Users Guide

DokuWiki

Wed Nov 6 14:50:05 UTC 2019
# Contents

## 1 Before You Start
1.1 End User License Agreement ........................................ 4
1.2 Scope of Operation .................................................. 4
1.3 Examples ............................................................. 5
1.4 Follow us ............................................................ 5
1.4.1 Join Mailing List .................................................. 5
1.4.2 Drop Out of Mailing List ......................................... 5
1.4.3 Twitter: @gfzrnx .................................................. 5
1.5 Bug Reports / Comments ............................................. 6

## 2 Basics
2.1 Software ............................................................. 7
2.1.1 Download ........................................................ 7
2.1.2 Install ........................................................... 7
2.1.3 Usage ............................................................ 8
2.1.4 Fast Help ......................................................... 8
2.2 Data Input/Output ................................................... 12
2.2.1 Supported Format Versions ..................................... 12
2.2.2 Input ............................................................ 12
2.2.3 Output .......................................................... 12
2.2.4 Examples Input/Output ........................................ 12
2.2.5 Log Messages .................................................. 12
2.3 Supported File Names .............................................. 13
2.3.1 RINEX-2 naming convention .................................. 13
2.3.2 RINEX-3 naming convention .................................. 14
2.3.3 Automatic Output File Naming ................................ 16

## 3 Operation / Tasks
3.1 RINEX File Check and Repair ..................................... 20
3.1.1 Navigation Data Epoch Filter ................................ 21
3.2 RINEX File Statistics / Informations ............................... 22
3.2.1 Observations Statistics ....................................... 22
3.2.2 ASCII Timeplot of Observables ............................... 22
3.3 RINEX File Format Conversion (3/2, 2/3) ......................... 24
3.3.1 Observation Types Mapping .................................. 24
3.3.2 REMARK ....................................................... 25
3.3.3 RINEX-2 to RINEX-3 ......................................... 25
3.3.4 RINEX-3 to RINEX-2 ......................................... 26
3.4 RINEX File Nominal Renaming Support (2/3) .................... 28
3.4.1 Remark ........................................................ 30
3.5 RINEX File Splice .................................................. 31
3.5.1 Observation Data Splice Specials ............................. 31
3.5.2 Navigation Data Splice Specials .............................. 32
3.5.3 Remark - Splice/Split ....................................... 33
3.5.4 Remark - Filename Expansion ................................ 34
### 3.6 RINEX File Split
- 3.6.1 Remark - Split/Splice

### 3.7 RINEX File Output Epoch Interval
- 3.7.1 Supported Date/Time/Epoch Formats
- 3.7.2 Dedicated Output Epoch Interval
- 3.7.3 Strict Epoch interval (-sei)
- 3.7.4 Extend Navigation File Boundaries (-enb)

### 3.8 RINEX File Manipulation
- 3.8.1 Data Sampling (-smp)
- 3.8.2 Satellite System Selection (-satsys)
- 3.8.3 PRN Selection (-prn, -no_prn)
- 3.8.4 Observation Types Selection (-obs_types)
- 3.8.5 Remove of Sparse Observation Types (-remove_sparse_obs_types)
- 3.8.6 Keep all Observation Types (-kaot)
- 3.8.7 Observation Types Sorting (-ots)
- 3.8.8 Navigation File Sorting (-nav_sort)
- 3.8.9 GPSweek Rollover Correction (-shift_gpsw)
- 3.8.10 Antenna Rename (-ant_rename)
- 3.8.11 Antenna Rename Table output (-ant_rename_out)
- 3.8.12 Antenna Rename Table input (-ant_rename_inp)

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

### 3.10 Rinex File Header/Data Editing
- 3.10.1 Header Editing (Standard)
- 3.10.2 Header Editing (Only)
- 3.10.3 Editing Operations
- 3.10.4 Show Config. File Interpretation (-show_crux)
- 3.10.5 Configuration file
- 3.10.6 Header edit via command line (-cx_updins)
- 3.10.7 Internal/Data Headers via crux-file (-cx_addinthd)
- 3.10.8 Manipulate Header Version Number (-vnum)

### 3.11 Rinex File Meta Data Extraction (-meta)

### 3.12 Rinex File Comparison (-fdiff)

### 3.13 Rinex Hatanaka Compressed Files

### 3.14 Rinex to Tabular Observations Output
- 3.14.1 Standard Output
- 3.14.2 Date/Time Formats
- 3.14.3 Column Separator

### 3.15 Rinex Standard Extensions/NonConformity
- 3.15.1 RINEX-2 BDS, QZSS, IRNSS support
- 3.15.2 RINEX-2 to RINEX-3 conversion
- 3.15.3 Handling of unsupported observation types
- 3.15.4 Remark
Chapter 1

Before You Start

1.1 End User License Agreement

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

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

1.2 Scope of Operation

The gfzrnx is a toolbox for RINEX file check and manipulation for the major versions 2 and 3. 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 (version 3 to 2 and vice versa),
• RINEX file splice,
• RINEX file split,
• RINEX file statistics generation,
• RINEX file manipulations like:
  – data sampling,
  – observation types selection,
  – satellite systems selection,
  – elimination of overall empty or sparse observation types.
• Automatic version dependent file naming on output file.
• RINEX file (re)naming support (version 2 to 3)
• RINEX header editing
• RINEX file meta data extraction
• RINEX file comparison

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

1.3 Examples

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

Example Box

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

1.4 Follow us

1.4.1 Join Mailing List

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

One can join the mailing list sending an empty e-mail to: gfzrnx-on@gfz-potsdam.de.

After getting a Confirmation Request e-mail, please don’t forget to reply to this Confirmation Request. This reply is mandatory to finish your list joining.

1.4.2 Drop Out of Mailing List

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

1.4.3 Twitter: @gfzrnx

Figure 1.1: Twitter: @gfzrnx
1.5 Bug Reports / Comments

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

- Make sure, that you are using the latest version.
- If you are using the latest version, please provide the complete command line you have used.
- Attach your input file(s) to your e-mail or provide a link for the input data download. Shrink the input file(s) if possible.
Chapter 2

Basics

2.1 Software

2.1.1 Download

One can download the software via:

http://semisys.gfz-potsdam.de/semisys

You will find an official version with a version number and a development version (DEVEL) with ongoing bug fixing and new features. The manual (pdf) can be downloaded from there too.

2.1.2 Install

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

<table>
<thead>
<tr>
<th>OS</th>
<th>Default Temporary Directory</th>
</tr>
</thead>
<tbody>
<tr>
<td>UNIX</td>
<td>/tmp</td>
</tr>
<tr>
<td>Windows</td>
<td>$WINDIR (C:)</td>
</tr>
</tbody>
</table>

UNIX: Copy the executable into a directory covered by your system search PATH variable.

WINDOWS: Copy the executable into your Windows directory for ease of use.

2.1.2.1 Remark

gfzrnx will store and execute libraries in a temporary directory.

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.
2.1.3 Usage

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

2.1.3.1 Unix

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

![Figure 2.1: Unix Terminal - command line](image1)

2.1.3.2 Windows

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

![Figure 2.2: MS Windows command window - command line](image2)

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

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

2.1.4 Fast Help

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

```
./gfzrnx_lx -h

***** USAGE: /dsk/perl2exe/gfzrnx/www/1.12/gfzrnx_lx

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

[-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

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

RINEX-3 file naming

SSSMRRCC_S_YYYYDDHHMM_NNN_FRQ_TT.FMT
SSSMRRCC_S_YYYYDDHHMM_NNN_TT.FMT

see Documentation for details

splice mode:

* list of input files

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

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

[-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
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 documentation for details).

[-vo <2|3>] - output RINEX version (std. 3)
[---version_out <2|3>] - 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 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:PWCSLYX,G:5:QXI,R:12:CP
[-errlog <file>] - store (append) error logs to a file (std. print to STDOUT)
[-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
[-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_addithd] - 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
globally 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 [CEGIJRSI]
  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
  the "string" consists of the 1st obs. type id. characters (e.g. CPLDS ),
  the "attr" can be [fraqs:freqdsc|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,CD1-5,R01-10,E14,E18
[-no_prn <prn-list>] - komma separated list of PRNs to be skipped
  range notations are possible G1-32,CD1-5,R01-10,E14,E18
[-kaot] - keep all obs. types (including fully empty ones)
[-rsot <n>] - remove sparse obs. types.
  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 - Gionass
  S - SBAS
[-ns <type>] - output order of navigation records. type = [time|prn] (std. prn)
[-nav_sort <type>] - time - sort by time,prn
prn - sort by prn,time

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

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

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

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

[-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 (2|3) same as in input

[-q] - quiet mode

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

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

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

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

[-nav_epo_filter] - only standard epochs are passed to the output

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

[try_append] - try append mode to fasten the splice process

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

[-out_obs_map] - output std. obs. types mapping

[-tab_obs] - create a tabular observation output

[-tab_date] - use other date (pattern) for tabular observation output (yyyy-mm-dd|yyyymmdd|wwwd|yyyyddd|yyyyyddd|wwwd|mjd|ddd)

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

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

ePOCH <EPOCH> parameter

mjd  56753 or 56753_123000
wwwd 17870 or 17870_123000
yyyyddd 2014096 or 2014096_123000
yyyyymmdd 20140406 or 20140406_123000
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:
  hmmmss
  hh:mm:ss

© Helmholtz-Centre Potsdam - GFZ German Research Centre for Geosciences
Section 1.1 Space Geodetic Techniques
2.2 Data Input/Output

2.2.1 Supported Format Versions

gfzrnx supports all versions 2.x and 3.x formats as input. The output format will be only the latest standard format of the major formats 2 and 3. for version 2 it is 2.11 and for version 3 this is currently 3.04.

2.2.2 Input

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

2.2.3 Output

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

2.2.4 Examples Input/Output

2.2.4.1 Input via STDIN

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

2.2.4.2 Input via -finp

gfzrnx -finp pots007a.15o ...

2.2.4.3 Output via STDOUT

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

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

2.2.4.4 Output via -fout

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

2.2.5 Log Messages

By default log messages (Notices, Errors, Warnings) are sent to STDERR. One can store the log messages into a file using the -errlog command line parameter.
The log table information consists of:

<table>
<thead>
<tr>
<th>Label</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATE/TIME</td>
<td>processing epoch</td>
</tr>
<tr>
<td>C(ode)</td>
<td>N(otice), W(arning), E(rror)</td>
</tr>
<tr>
<td>EPOCH / FILE</td>
<td>affected epoch in input file</td>
</tr>
<tr>
<td>SITE</td>
<td>4-char. station identifier</td>
</tr>
<tr>
<td>T(ype)</td>
<td>Data Type</td>
</tr>
<tr>
<td>MESSAGE</td>
<td>log message</td>
</tr>
</tbody>
</table>

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

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

### 2.3 Supported File Names

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

#### 2.3.1 RINEX-2 naming convention

<table>
<thead>
<tr>
<th>File Name</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>SSSSSDDDD0.YYT</td>
<td>daily file</td>
<td>pots0070.15o</td>
</tr>
<tr>
<td>SSSSSDDDD[a-x].YYT</td>
<td>hourly file</td>
<td>pots007a.15o</td>
</tr>
<tr>
<td>SSSSSDDDD[a-x]MM.YYT</td>
<td>sub-hourly file</td>
<td>pots007r45.15o</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Var.</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>SSSS</td>
<td>4-char. station identifier</td>
<td>pots</td>
</tr>
<tr>
<td>DDD</td>
<td>day of year</td>
<td>007</td>
</tr>
<tr>
<td>YY</td>
<td>2-digit year</td>
<td>15</td>
</tr>
<tr>
<td>MM</td>
<td>minute of data begin</td>
<td>45</td>
</tr>
<tr>
<td>T</td>
<td>data type (o,d,m,n,...)</td>
<td>o</td>
</tr>
</tbody>
</table>
2.3.1.1 Examples

- daily file
  
  pots0070.15o

- hourly files
  
  pots007a.15o pots007b.15o pots007c.15o ... pots007v.15o pots007w.15o pots007x.15o

- sub-hourly files (15 min)
  
  pots007a00.15o pots007a15.15o pots007a30.15o pots007a45.15o

2.3.2 RINEX-3 naming convention

Table follows on next page ...
<table>
<thead>
<tr>
<th>File Name</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>SSSSMRCC_S_YYYYYDDHHMM_NNN_FRQ TT .FMT[CMP]</td>
<td>POTS00DEU_R_20150070000_01H_30S MO .mx .bz2</td>
</tr>
<tr>
<td>SSSSMRCC_S_YYYYYDDHHMM_NNN TT .FMT[CMP]</td>
<td>POTS00DEU_R_20150070000_01H_MN .mx .gz</td>
</tr>
</tbody>
</table>
2.3 Supported File Names

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

For more details see RINEX-3 file format definitions.

2.3.3 Automatic Output File Naming

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

2.3.3.1 RINEX-2 Site Name

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

2.3.3.2 RINEX-3 Site Name

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

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

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

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

```
gfzrnx -finp pots0070.15o -fout ::RX3::00,DEU
gfzrnx -finp pots0070.15o -fout ::RX3::DEU,00
```
The output file name will be: **POTS00DEU_R_20150070000_01H_30S_MO.rnx**.

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

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

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

### 2.3.3.3 RINEX-3 Site Name (-4to9)

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

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

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

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

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

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

An up to date **4to9** configuration file for diverse networks like **IGS**, **MGEX**, **EUREF**, **TIGA** and others can be derived from **GFZ**'s **SEnSor Meta Information SYStem** (**SEMSYS**) 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](http://semisys.gfz-potsdam.de/semisys/download).

### 2.3.3.4 RINEX-2 Start Epoch/Duration

By default the start epoch and file duration are used to create the epoch parts of the output name. To force the automatic file naming to a distinct type **::RX2::** can be extended by the letters **L**, **S** or **D** (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.

<table>
<thead>
<tr>
<th>Duration</th>
<th>&lt; 1 hour</th>
<th>1 hour</th>
<th>&gt; 1 hour</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>::RX2::</strong></td>
<td>abcd122d05.15o</td>
<td>abcd122d15o</td>
<td>abcd1220.15o</td>
</tr>
<tr>
<td><strong>::RX2L::</strong></td>
<td>abcd122d05.15o</td>
<td>abcd122d05.15o</td>
<td>abcd122d05.15o</td>
</tr>
<tr>
<td><strong>::RX2S::</strong></td>
<td>abcd122d15o</td>
<td>abcd122d15o</td>
<td>abcd122d15o</td>
</tr>
<tr>
<td><strong>::RX2D::</strong></td>
<td>abcd1220.15o</td>
<td>abcd1220.15o</td>
<td>abcd1220.15o</td>
</tr>
</tbody>
</table>
The cases ::RX2L::, ::RX2S:: allow to store not only hourly or sub-hourly files. For durations larger than 1 hour one can use it to store sub-daily files too. In this case the file epoch indicates the start time (hour, minute) only. In case of sub hourly file names with nominal begin epochs (-epo_beg / -sei in) and the nominal duration -d 900 is 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 pots126x15.13o -fout TMP/::RX2:: 
  TMP/pots126x15.13o

gfzrnx -kv -finp pots126x15.13o -fout TMP/::RX2L:: 
  TMP/pots126x15.13o

gfzrnx -kv -finp pots126x15.13o -fout TMP/::RX2L:: -d 120 
  TMP/pots126x16.13o

gfzrnx -kv -finp pots126x15.13o -fout TMP/::RX2L:: -sei in 
  TMP/pots126x15.13o

gfzrnx -kv -finp pots126x15.13o -fout TMP/::RX2L:: -epo_beg 2015125_230000 -d 1800 
  TMP/pots126x00.13o

gfzrnx -kv -finp pots126x15.13o -fout TMP/::RX2S:: 
  TMP/pots126x.13o
```

### 2.3.3.5 RINEX-3 Start Epoch/Duration (real)

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

<table>
<thead>
<tr>
<th>Characteristics</th>
<th>pots0070.15o</th>
<th>pots007c15o</th>
<th>pots007c30.15o</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time Begin</td>
<td>01:12:30</td>
<td>02:13:30</td>
<td>02:33:13</td>
</tr>
<tr>
<td>Time End</td>
<td>23:59:30</td>
<td>02:55:30</td>
<td>02:44:50</td>
</tr>
<tr>
<td>Duration (implicit)</td>
<td>1 day</td>
<td>1 hour</td>
<td>unknown</td>
</tr>
<tr>
<td>Duration (nominal)</td>
<td>1 day</td>
<td>1 hour</td>
<td>15 min</td>
</tr>
<tr>
<td>Duration (real hh:mm:ss)</td>
<td>22:47:00</td>
<td>00:42:00</td>
<td>00:11:37</td>
</tr>
<tr>
<td>Sampling Rate</td>
<td>30s</td>
<td>30s</td>
<td>1s</td>
</tr>
</tbody>
</table>

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

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

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

<table>
<thead>
<tr>
<th>RINEX-2</th>
<th>RINEX-3</th>
</tr>
</thead>
<tbody>
<tr>
<td>pots0070.15o</td>
<td>POTS00DEU_R_20150070112_23H_30S_MO.rnx</td>
</tr>
<tr>
<td>pots007c15o</td>
<td>POTS00DEU_R_20150070213_42M_30S_MO.rnx</td>
</tr>
<tr>
<td>pots007c30.15o</td>
<td>POTS00DEU_R_20150070233_12M_01S_MO.rnx</td>
</tr>
</tbody>
</table>

### 2.3.3.6 RINEX-3 Start Epoch/Duration (nominal)

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

The general method is to give the begin epoch and the duration information via the `-epo_beg` and `-d` command line
Basics

parameters.

\begin{verbatim}
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
\end{verbatim}

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.

\begin{verbatim}
<table>
<thead>
<tr>
<th>RINEX-2</th>
<th>command line parameters</th>
<th>RINEX-3</th>
</tr>
</thead>
<tbody>
<tr>
<td>pots0070.15o</td>
<td>-sei in</td>
<td>POTS00DEU_R_20150070000_01D_30S_MO.rnx</td>
</tr>
<tr>
<td>pots007c.15o</td>
<td>-sei in</td>
<td>POTS00DEU_R_20150070200_01H_30S_MO.rnx</td>
</tr>
<tr>
<td>pots007c30.15o</td>
<td>-sei in -d 900</td>
<td>POTS00DEU_R_20150070230_15M_01S_MO.rnx</td>
</tr>
</tbody>
</table>
\end{verbatim}

2.3.3.7 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 only a single satellite system is found in the given file(s).

2.3.3.8 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.

\begin{center}
\textbf{CAUTION !}

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

For navigation files this nominal interval can be extended via the -enb command line parameter (extend navigation boundaries). See the Operation/Tasks - Rinex File Epoch Interval section.
Chapter 3
Operation / Tasks

To get the full available checks via data input one has to use the -chk option, to make sure that the output data are formally correct. If you are sure, that your files are correct and you want to do some data manipulation only you can omit this commandline parameter to speed up the work.

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

The standard output format of gfzrnx is RINEX-3!

3.1 RINEX File Check and Repair

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

With gfzrnx this can be done via:

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

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

The following modifications are done in the output file:

- update of observation types to really existing ones, overall empty observation types are removed.
  - SYS / # / OBS TYPES
  - # / TYPES OF OBSERV
- Statistical information are 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
E 13 C1X C5X C7X C8X D1X L1X L5X L7X L8X S1X S5X S7X S8X
G 20 C1C C2W C2X C5X D1C D1P D1W D2W L1C L1P L1W L2W L2X
J 19 C1C C1Z C2X C5X C6L D1C L1X L1Z L2X L5X L6L
```

Helmholtz Centre Potsdam / GFZ German Research Centre for Geosciences Wed Nov 6 14:50:05 UTC 2019
The repair of a file file is different concerning RINEX-2 and RINEX-3. Data values are not corrected! Via the repair operation formally corrupt observation parts are omitted only.

- **RINEX-2**
  1. A complete epoch block is removed in case of corrupted data detection.

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

### 3.1.1 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, hours are checked at the moment. The following table shows valid hours, minutes per satellite system:

<table>
<thead>
<tr>
<th>Sat. System</th>
<th>Minutes</th>
<th>Hours (modulo)</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>E</td>
<td>0,10,20,30,40,50</td>
<td>1</td>
</tr>
<tr>
<td>G</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>R</td>
<td>15,45</td>
<td>1</td>
</tr>
<tr>
<td>J</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>
3.2 RINEX File Statistics / Informations

3.2.1 Observations Statistics

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

gfzrnx -finp pots0070.15o -stk_obs

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

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

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

gfzrnx -finp sin12290.14o -stk_obs

STP sin1 C TYP C1I C6I C7I D1I L1I L6I L7I S1I S6I S7I
STD sin1 C C01 2863 2863 2863 2863 2863 2863 2863 2863 2863 2863
STD sin1 C C02 2863 2863 2863 2863 2863 2863 2863 2863 2863 2863
...   
STD sin1 C C14 1365 1363 1363 1365 1363 1363 1363 1363 1363 1363

STP sin1 E TYP C1X C5X C7X C8X D1X L1X L6X L7X L8X S1X S5X S7X S8X
STD sin1 E E11 900 895 893 899 900 900 895 893 899 900 895 893 899
STD sin1 E E12 1230 1230 1230 1230 1230 1230 1230 1230 1230 1230 1230
STD sin1 E E19 1605 1601 1601 1603 1605 1601 1601 1601 1601 1601

STP sin1 G TYP C1C C2W C2X C5X D1C ... L1C L1P L1W L2W L2X L5X S1C ... 
STD sin1 G G01 1189 1148 1181 1181 1189 ... 1189 0 0 1148 1181 1181 1189 ...
...   
STD sin1 G G10 886 881 0 0 886 ... 886 9 9 881 0 0 886 ...

STP sin1 J TYP C1C C1X C1Z C2X C5X D6L D1C L1C L1X L1Z L2X L5X L6L ...
STD sin1 J J01 2863 2863 2863 2863 2863 2863 2863 2863 2863 2863 2863

3.2.2 ASCII Timeplot of Observables

The -stk_epo command line parameter can be used to create an ASCII timeplot to show the availability of observations per PRN (std.) and/or observation type.

In the simplest mode one has to provide the time bin to be used in seconds (here 1800).

3.2.2.1 Timeplot per PRN

rnxall -finp stas0400.15o -stk_epo 1800
rnxall -finp stas0400.15o -stk_epo 1800:prn
### 3.2.2.2 Timeplot per PRN and/or Observation Type

A timeplot per observation type is available providing the \([\text{[prn/otp]}]\) parameter list. This can be combined with other parameters like \([-\text{smp}, -\text{satsys}, -\text{obs_types}, -\text{prn}, -\text{no_prn} \text{ etc.}]

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

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>STT</th>
<th>20150209 00:00 04:00 08:00 12:00 16:00 20:00 00:00</th>
</tr>
</thead>
<tbody>
<tr>
<td>STH</td>
<td>1800:prn,otp -satsys E -ot C,L</td>
</tr>
<tr>
<td>STH</td>
<td>1800:prn,otp -satsys E -ot C,L</td>
</tr>
</tbody>
</table>

---

**Operation / Tasks**
3.3 RINEX File Format Conversion (3/2, 2/3)

3.3.1 Observation Types Mapping

The used observation types mapping is hardcoded in \texttt{gfzrnx}. It can be shown up via the following command.

\begin{verbatim}
gfzrnx -finp stas0010.15o -stk_epo 5:prn,otp -fout xxxx
\end{verbatim}
The information in the columns 2,3,4 are treated as comment only and are not used.

### 3.3.2 REMARK

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

### 3.3.3 RINEX-2 to RINEX-3

Please use this conversion only if you are sure, that the output files are usable in the environment to which the data are supplied!

The output format for this conversion/transition is RINEX-3.01 to be standard conform.

The 2-char. observation types are kept as they are except the code observations for GPS and GLONASS (see below).

As **RINEX-3** is the standard output format of **gfzrnx** simply run:

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

or

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

or

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

to create a RINEX-3 conform output file name **POTS00DEU_R_201500700_01D_30S_MO.rnx**.

For naming details see the **Automatic Output File Naming** section.

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

<table>
<thead>
<tr>
<th>System</th>
<th>RINEX-2</th>
<th>RINEX-3</th>
</tr>
</thead>
<tbody>
<tr>
<td>G</td>
<td>P1</td>
<td>C1W</td>
</tr>
<tr>
<td>G</td>
<td>C1</td>
<td>C1C</td>
</tr>
<tr>
<td>G</td>
<td>P2</td>
<td>C2W</td>
</tr>
<tr>
<td>G</td>
<td>C2</td>
<td>C2C</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>System</th>
<th>RINEX-2</th>
<th>RINEX-3</th>
</tr>
</thead>
<tbody>
<tr>
<td>R</td>
<td>P1</td>
<td>C1P</td>
</tr>
<tr>
<td>R</td>
<td>C1</td>
<td>C1C</td>
</tr>
<tr>
<td>R</td>
<td>P2</td>
<td>C2P</td>
</tr>
<tr>
<td>R</td>
<td>C2</td>
<td>C2C</td>
</tr>
</tbody>
</table>

This is used, because both **Px** and **Cx** RINEX-2 code types are mapped to the single **Cx** RINEX-3 code type.
3.3.4 RINEX-3 to RINEX-2

The RINEX-2 output version is 2.11.

Use the -version_out or -vo command line parameter to define RINEX format version of the output file.

\[
gfzrnx -finp pots0070.15o -fout pots0070.15o_rx2 -vo 2
\]

\[
gfzrnx -finp POTS00DEU_R_201500700_01D_30S_MD.rnx -fout pots0070.15o --version_out 2
\]

3.3.4.1 Specific Observation Type Selection

In the RINEX-3 format one can have multiple observation types per data type and frequency (tracking mode or channel attribute). For a specific observation type selection for the format conversion you can use the observation types selection feature in addition. Add the -ot command line parameter to the upper command like in the example below to select the RINEX-3 obs. types to be converted and to get a distinct conversion.

\[
\]

3.3.4.2 Observation Type Selection via Signal Priorities

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

<table>
<thead>
<tr>
<th>Sat. System</th>
<th>Freq. Num.</th>
<th>RINEX-3 Signal Priority</th>
</tr>
</thead>
<tbody>
<tr>
<td>G - GPS</td>
<td>1</td>
<td>PWCSLXYMN</td>
</tr>
<tr>
<td>G - GPS</td>
<td>2</td>
<td>PWCSLXYMN</td>
</tr>
<tr>
<td>G - GPS</td>
<td>5</td>
<td>IQX</td>
</tr>
<tr>
<td>R - GLO</td>
<td>1</td>
<td>PC</td>
</tr>
<tr>
<td>R - GLO</td>
<td>2</td>
<td>PC</td>
</tr>
<tr>
<td>R - GLO</td>
<td>3</td>
<td>IQX</td>
</tr>
<tr>
<td>R - GLO</td>
<td>4</td>
<td>ABX</td>
</tr>
<tr>
<td>R - GLO</td>
<td>6</td>
<td>ABX</td>
</tr>
<tr>
<td>E - GAL</td>
<td>1</td>
<td>BCX</td>
</tr>
<tr>
<td>E - GAL</td>
<td>5</td>
<td>IQX</td>
</tr>
<tr>
<td>E - GAL</td>
<td>6</td>
<td>BCX</td>
</tr>
<tr>
<td>E - GAL</td>
<td>7</td>
<td>IQX</td>
</tr>
<tr>
<td>E - GAL</td>
<td>8</td>
<td>IQX</td>
</tr>
<tr>
<td>J - QZS</td>
<td>1</td>
<td>SLX</td>
</tr>
<tr>
<td>J - QZS</td>
<td>2</td>
<td>SLX</td>
</tr>
<tr>
<td>J - QZS</td>
<td>5</td>
<td>IQX</td>
</tr>
<tr>
<td>J - QZS</td>
<td>6</td>
<td>SLX</td>
</tr>
<tr>
<td>C - BDS</td>
<td>1</td>
<td>IQX</td>
</tr>
<tr>
<td>C - BDS</td>
<td>2</td>
<td>IQX</td>
</tr>
</tbody>
</table>
### Operation / Tasks

<table>
<thead>
<tr>
<th>Sat. System</th>
<th>Freq. Num.</th>
<th>RINEX-3 Signal Priority</th>
</tr>
</thead>
<tbody>
<tr>
<td>C - BDS</td>
<td>5</td>
<td>DPX</td>
</tr>
<tr>
<td>C - BDS</td>
<td>6</td>
<td>IQX</td>
</tr>
<tr>
<td>C - BDS</td>
<td>7</td>
<td>IQX</td>
</tr>
<tr>
<td>C - BDS</td>
<td>8</td>
<td>DPX</td>
</tr>
<tr>
<td>I - IRN</td>
<td>5</td>
<td>ABCX</td>
</tr>
<tr>
<td>I - IRN</td>
<td>9</td>
<td>ABCX</td>
</tr>
<tr>
<td>S - SBS</td>
<td>1</td>
<td>C</td>
</tr>
<tr>
<td>S - SBS</td>
<td>5</td>
<td>IQX</td>
</tr>
</tbody>
</table>

The observation codes priority is **LCDS**: phase, code, doppler, 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 providing update records via the `-pr3rx2` command line parameter. According to the upper table it should consist of a comma separated list of a satellite system identifier, colon, frequency number, colon and the signal priority string. Observation types not covered by the priority string are simply ignored via conversion. See the following example.

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

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

#### 3.3.4.3 Used Observation Types

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

```
RINEX 3 -> 2 TYPE CONVERSION DETAILS:
-------------------------------------
C C1I -> C1
C C6I -> C6
C C7I -> C7
C D1I -> D1
C L1I -> L1
C L6I -> L6
C L7I -> L7
C S1I -> S1
C S6I -> S6
C S7I -> S7
-------------------------------------
E C1X -> C1
E C5X -> C5
E C7X -> C7
E C8X -> C8
```

---

**Operation / Tasks**

28

---

**Helmholtz Centre Potsdam / GFZ German Research Centre for Geosciences**

Wed Nov 6 14:50:05 UTC 2019
29 3.4 RINEX File Nominal Renaming Support (2/3)

A fast file name conversion of RINEX-3 files with RINEX-2 style file names to RINEX-3 style file names is supported. It can be used without reading the input files using all necessary information from the RINEX-2 style file name and from information provided via command line parameters (useful for compressed files).

For uncompressed observation files, including hatanaka compressed files, some required information can also be derived from the file header.

The supported RINEX-2 style file names are:

<table>
<thead>
<tr>
<th>Name</th>
<th>Example</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>nnnnddd0.yyt</td>
<td>pots1230.15o</td>
<td>daily obs. file</td>
</tr>
<tr>
<td></td>
<td>pots1230.15d</td>
<td>daily obs. file (hatanaka compressed)</td>
</tr>
<tr>
<td>nnnnddd[a-z].yyt</td>
<td>pots123a.15n</td>
<td>hourly nav. file</td>
</tr>
<tr>
<td>nnnnddd[a-z]mm.yt</td>
<td>pots123x15.15m</td>
<td>sub-hourly met. file</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Information</th>
<th>Values</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>s</td>
<td>data source</td>
<td>R or S</td>
</tr>
<tr>
<td>mr</td>
<td>marker/receiver number</td>
<td>mr</td>
</tr>
<tr>
<td>iso</td>
<td>iso country code</td>
<td>ISO</td>
</tr>
</tbody>
</table>

```
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 extension letter to the RINEX-3 data type identifier:

<table>
<thead>
<tr>
<th>Extension</th>
<th>Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>o</td>
<td>_MO.rnx</td>
</tr>
<tr>
<td>d</td>
<td>_MO.crx</td>
</tr>
<tr>
<td>n</td>
<td>_GN.rnx</td>
</tr>
<tr>
<td>g</td>
<td>_RN.rnx</td>
</tr>
<tr>
<td>l</td>
<td>_EN.rnx</td>
</tr>
<tr>
<td>c</td>
<td>_CN.rnx</td>
</tr>
<tr>
<td>q</td>
<td>_JN.rnx</td>
</tr>
<tr>
<td>j</td>
<td>_JN.rnx</td>
</tr>
<tr>
<td>h</td>
<td>_SN.rnx</td>
</tr>
<tr>
<td>p</td>
<td>_MN.rnx</td>
</tr>
<tr>
<td>m</td>
<td>_MM.rnx</td>
</tr>
</tbody>
</table>

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 this 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 some examples, including hatanaka compressed files:

```bash
gfzrnx -finp pots1230.15o.gz -nomren23 DEU -smp 30 -satsys G
POTS00DEU_R_20151230000_01D_30S_G0.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
```

Using the following RINEX-3 header information:

```
E  6 C1X C5X L1X S1X S5X
G  8 C1C C1P C2C L1P L2P S1P S2P
R  8 C1C C1P C2C L1P L2P S1P S2P
10.000 INTERVAL
```

leads to the following file names:

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

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

A single satellite system file with the following information:

```
E  6 C1X C5X L1X S1X S5X
5.000 INTERVAL
```

leads to the file names:

```
gfzrnx -finp pots1230.15o -nomren23 DEU
POTS00DEU_R_20151230000_01D_05S_MO.rnx

gfzrnx -finp pots1230.15d -nomren23 DEU
POTS00DEU_R_20151230000_01D_05S_MO.crx
```

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

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

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

### 3.4.1 Remark

Information provided via command line has priority.
3.5 RINEX File Splice

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

Simply provide a list of input files and the output file:

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

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

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

For **csh** command shell it is:

```
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
```

This works similar for navigation and meteo files.

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

3.5.1 Observation Data Splice Specials

There are two different splice modes available

3.5.1.1 Standard Mode (default)

The input file order is derived automatically. In case of overlapping input files the file with less 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 by input statistics. This allows to omit "empty" observation types.

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

3.5.1.2 Fast / RAM save Mode (-splice_memsave)

Via the `-splice_memsave` only 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 is used to derive the output observation types.

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

3.5.1.3 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
```
3.5.1.4 Try Append (-try_append)

The -try_append command line parameter initiates an initial check over all input files if an 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.

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

3.5.2 Navigation Data Splice Specials

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

```
# of NAV. EPOCHS based on # of FILE CONTRIBUTIONS
------------------------------------------------------
BRD_HD   PRN  T =1 <5 <10 <15 <25 <50 >=50 #EPO
BRD_LN   ----------------------------------------
BRD_STK  C01_ . . 1 . 2 . . 3                COMMENT
BRD_STK  C02_ 2 1 . . 1 1 1 6                COMMENT
BRD_STK  C03_ 1 . . 1 . 2 . . 4                COMMENT
...   COMMENT
BRD_STK  C39_ 2 . . . . . . 2                COMMENT
BRD_STK  C59_ 2 . . . . . . 2                COMMENT
BRD_LN   ----------------------------------------
BRD_SYS  C - 250                COMMENT
BRD_SYS  E - 447                COMMENT
BRD_SYS  G - 214                COMMENT
```
3.5.2.1 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.

3.5.3 Remark - Splice/Split

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

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

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

3.5.4.1 UNIX

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

3.5.4.2 MS Windows

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

3.6 RINEX File Split

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

The following command:

\begin{verbatim}
gfzrnx -finp pots0070.15o -fout /tmp/::RX2:: -split 3600 -kv
\end{verbatim}

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

\begin{verbatim}
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
\end{verbatim}

The following command:

\begin{verbatim}
gfzrnx -finp pots0070.15o -fout /tmp/::RX3::00,DEU -split 3600
\end{verbatim}

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

\begin{verbatim}
/tmp/POTS0DEU_R_20150070000_01H_30S_MO.rnx /tmp/POTS0DEU_R_20150071000_01H_30S_MO.rnx /tmp/POTS0DEU_R_20150072000_01H_30S_MO.rnx /tmp/POTS0DEU_R_20150073000_01H_30S_MO.rnx /tmp/POTS0DEU_R_20150074000_01H_30S_MO.rnx /tmp/POTS0DEU_R_20150075000_01H_30S_MO.rnx /tmp/POTS0DEU_R_20150076000_01H_30S_MO.rnx /tmp/POTS0DEU_R_20150077000_01H_30S_MO.rnx /tmp/POTS0DEU_R_20150078000_01H_30S_MO.rnx /tmp/POTS0DEU_R_20150079000_01H_30S_MO.rnx /tmp/POTS0DEU_R_20150071000_01H_30S_MO.rnx /tmp/POTS0DEU_R_20150072000_01H_30S_MO.rnx /tmp/POTS0DEU_R_20150073000_01H_30S_MO.rnx /tmp/POTS0DEU_R_20150074000_01H_30S_MO.rnx /tmp/POTS0DEU_R_20150075000_01H_30S_MO.rnx /tmp/POTS0DEU_R_20150076000_01H_30S_MO.rnx /tmp/POTS0DEU_R_20150077000_01H_30S_MO.rnx /tmp/POTS0DEU_R_20150078000_01H_30S_MO.rnx /tmp/POTS0DEU_R_20150079000_01H_30S_MO.rnx /tmp/POTS0DEU_R_20150071000_01H_30S_MO.rnx /tmp/POTS0DEU_R_20150072000_01H_30S_MO.rnx /tmp/POTS0DEU_R_20150073000_01H_30S_MO.rnx /tmp/POTS0DEU_R_20150074000_01H_30S_MO.rnx /tmp/POTS0DEU_R_20150075000_01H_30S_MO.rnx /tmp/POTS0DEU_R_20150076000_01H_30S_MO.rnx /tmp/POTS0DEU_R_20150077000_01H_30S_MO.rnx /tmp/POTS0DEU_R_20150078000_01H_30S_MO.rnx /tmp/POTS0DEU_R_20150079000_01H_30S_MO.rnx
\end{verbatim}

3.6.1 Remark - Split/Splice

It is possible to combine the \texttt{split} with a \texttt{splice} operation of observation data. See splice section for details.

3.7 RINEX File Output Epoch Interval

3.7.1 Supported Date/Time/Epoch Formats

3.7.1.1 Date
### 3.7.1.2 Time

<table>
<thead>
<tr>
<th>Time Type</th>
<th>Abbreviation</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>HourMinuteSecond</td>
<td>HHMMSS</td>
<td>123000</td>
</tr>
<tr>
<td>Hour:Minute:Second</td>
<td>HH:MM:SS</td>
<td>12:30:00</td>
</tr>
</tbody>
</table>

### 3.7.1.3 Epoch

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

<table>
<thead>
<tr>
<th>Date Type</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>MJD</td>
<td>56753_123000</td>
</tr>
<tr>
<td>GPSweekWeekday</td>
<td>17870_12:30:00</td>
</tr>
<tr>
<td>YearDayofyear</td>
<td>2014096_123000</td>
</tr>
<tr>
<td>YearMonthDay</td>
<td>20140406_12:30:00</td>
</tr>
<tr>
<td>Year-Month-Day</td>
<td>2014-04-06_123000</td>
</tr>
</tbody>
</table>

### 3.7.2 Dedicated Output Epoch Interval

To extract a dedicated epoch interval from a RINEX-file you have to provide a Start-Epoch via `-epo_beg` and the Duration `-d` or `-duration` in seconds.

Here an example to extract the first hour of a daily input file.

```
gfzrnx -finp pots0070.15o -fout pots007a.15o -epo_beg 2015-01-07_000000 -d 3600

gfzrnx -finp pots0070.15o -fout pots007a.15o -epo_beg 2015007_00:00:00 -d 3600

gfzrnx -finp pots0070.15o -fout pots007a.15o -epo_beg 20150107_000000 -d 3600
```

### 3.7.3 Strict Epoch interval (-sei)

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

```
gfzrnx -finp pots0070.15o -fout pots007a.15o_chk -chk -sei in

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

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

### 3.7.4 Extend Navigation File Boundaries (-enb)

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

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

3.8 RINEX File Manipulation

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

3.8.1 Data Sampling (-smp)

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

{smp num[:eps]}

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

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

3.8.1.1 REMARK

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

<table>
<thead>
<tr>
<th>Sampling Rate</th>
<th>Default eps</th>
</tr>
</thead>
<tbody>
<tr>
<td>&gt;= 1 s</td>
<td>0.5 s</td>
</tr>
<tr>
<td>&lt; 1 s</td>
<td>5 ms</td>
</tr>
</tbody>
</table>

3.8.2 Satellite System Selection (-satsys)

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

{satsys <string>}

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

gfzrnx -finp pots0070.15o -fout pots0070.15o_rx3_GR -satsys GR
gfzrnx -finp pots0070.15o -fout pots0070.15o_rx3_GRE -satsys GRE

3.8.3 PRN Selection (-prn, -no_prn)

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

If you are interested in a subset of observation types only, you can use the –obs types command line parameter to provide your wished observation types via a comma separated list of pattern.

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

Here some examples:

3.8.4.1 RINEX-2

The input file contains the following observation types.

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

Select code and phase observations only.

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

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

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

The following command line

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

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

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

3.8.4.2 RINEX-3

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

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

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

A satellite system dependent record replaces fully a global one.

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

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

Here a selection of frequencies only:

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

Here is a global selection with special selections for C (Beidou) and G (GPS).
3.8.5 Remove of Sparse Observation Types (--remove_sparse_obs_types)

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

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

3.8.6 Keep all Observation Types (-kaot)

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

3.8.7 Observation Types Sorting (-ots)

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

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

<table>
<thead>
<tr>
<th>attribute</th>
<th>order by</th>
</tr>
</thead>
<tbody>
<tr>
<td>frqasc</td>
<td>frequency &amp; obs. type (ascending)</td>
</tr>
<tr>
<td>frqdsc</td>
<td>frequency &amp; obs. type (descending)</td>
</tr>
<tr>
<td>frq&lt;frq-list&gt;</td>
<td>distinct given comma separated list of frequencies</td>
</tr>
<tr>
<td>froasc</td>
<td>obs. type &amp; frequency (ascending)</td>
</tr>
<tr>
<td>frodsc</td>
<td>obs. type &amp; frequency (descending)</td>
</tr>
<tr>
<td>fro&lt;frq-list&gt;</td>
<td>distinct given comma separated list of frequencies</td>
</tr>
</tbody>
</table>

Some examples:

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

```
-ots CPLDS:frqasc
```
3.8.8 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 are possible.

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

The standard mode is prn.

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

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

```
gfzrnx -finp ???0070.15n -fout brds0070.15n -ns time
```

```
gfzrnx -finp ???0070.15n -fout ::RX3:: -split 3600 --nav_sort time
```

3.8.9 GPSweek Rollover Correction (-shift_gpsw)

Due to firmware or Rinex converter problems we have seen files which 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 one example for the file MAR100DEU_R_20190440015_15M_01S_GO.rnx, where the gpsweek for 20190440015 (2019 02 13) is 2040.

```
1999 06 30 00 15 0.0000000 GPS TIME OF FIRST OBS
1999 06 30 00 29 59.0000000 GPS TIME OF LAST OBS
```

The gpsweek of 1999 06 30 is 1016 (2040-1016 = 1024). The shift by 1024 weeks leads to 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
```
### 3.8 RINEX File Manipulation

**3.8.10 Antenna Rename (-ant_rename)**

Historical files, especially GPS observations files before 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.

<table>
<thead>
<tr>
<th>FROM</th>
<th>TO</th>
</tr>
</thead>
<tbody>
<tr>
<td>DORNE MARGOLIN ASH</td>
<td>ASH700936A_M</td>
</tr>
<tr>
<td>GEODETIC III L1/L2</td>
<td>ASH700718A</td>
</tr>
<tr>
<td>GEODETIC L1/L2 L</td>
<td>ASH700228A</td>
</tr>
<tr>
<td>GEODETIC L1/L2 P</td>
<td>ASH700228D</td>
</tr>
<tr>
<td>MARINE/RANGE</td>
<td>ASHMAR/RANGE</td>
</tr>
<tr>
<td>A-C L1</td>
<td>ASHAC_L1</td>
</tr>
<tr>
<td>A-C L1/L2</td>
<td>ASHAC_L1/L2</td>
</tr>
<tr>
<td>ASH701945.02B</td>
<td>ASH701945B_M</td>
</tr>
<tr>
<td>ASH701946.012</td>
<td>ASH701946.2</td>
</tr>
<tr>
<td>ASH701946.022</td>
<td>ASH701946.2</td>
</tr>
<tr>
<td>ASH701975.01Agp</td>
<td>ASH701975.01AGP</td>
</tr>
<tr>
<td>TR GEOD L1/L2 GP</td>
<td>TRM22020.00+GP</td>
</tr>
<tr>
<td>TR GEOD L1/L2 W/O GP</td>
<td>TRM22020.00-GP</td>
</tr>
<tr>
<td>TRM10877.10+RGP</td>
<td>TRM12333.00+RGP</td>
</tr>
<tr>
<td>JPSMARANT_GGD</td>
<td>JNSMARANT_GGD</td>
</tr>
<tr>
<td>TRM10877.10+SGP</td>
<td>TRM11877.10+SGP</td>
</tr>
<tr>
<td>DORNE MARGOLIN LEICA</td>
<td>LEIAT504</td>
</tr>
<tr>
<td>LEICA AT201</td>
<td>LEIAT201</td>
</tr>
<tr>
<td>LEICA AT202</td>
<td>LEIAT202-GP</td>
</tr>
<tr>
<td>LEICA AT302</td>
<td>LEIAT302-GP</td>
</tr>
</tbody>
</table>

Please turn over
### 3.8.11 Antenna Rename Table output (-ant_rename_out)

The table for the antenna rename 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 of json format.

```plaintext
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:

```plaintext
gfzrnx -ant_rename_out -fout ant_rename.json
```
3.8.12 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 fully overwrites the internal table. The input file must be of json format.

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

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

Usually a list of input files via `-finp` leads to a splice operation where the output is a single file. To initiate a file by file operation for a group of input files with a single command the command line parameter `-single_file` or `-sifi` has to be used.

For the output file naming the automatic file naming must be used (::RX2::, ::RX3::) or the `::INP::` variable, which means that the output file name is the same as the input file name.

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

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

3.10 Rinex File Header/Data Editing

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

There are two modes available:

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

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

3.10.1 Header Editing (Standard)

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

3.10.2 Header Editing (Only)

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

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

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

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

```
gfzrnx -finp file.rnx -fout file.rnx_hded -crux header_crux.txt -hded -epo_beg 2015234_000000 -site POTS00DEU
```
3.10.3 Editing Operations

The following operations are supported:

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

3.10.4 Show Config. File Interpretation (-show_crux)

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

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

The default header edit settings are shown via:

```
$ gfzrnx -show_crux
```

3.10.5 Configuration file

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

In case of rename a type (prn—obs) 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
#--------
[OM-][YYYYMMDD:HHMMSS YYYYMMDD:HHMMSS-] - <prn-from> - <prn-to> : ALL 
[OM-][YYYYDDD:SSSSS YYYYDDD:SSSSS-] - <prn-from> - <prn-to> : STA1[.STA2[.STA3...]] 
[OM-][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. At the moment only the 4 char. station identifier is supported (RINEX-2 file naming).
• For non specific station definitions the ALL station identifier can be used.
• Omitting the data types identifier extends the validity to all supported data types (OMN).
• Omitting the epoch interval leads to an overall validity.
• Station dependent settings overwrite non specific ALL settings.
• Overlapping epoch intervals for the same header label and station lead to an error.
• The date of the epoch interval can be given either as YYYYDDD (year, day of year) or YYYYMMDD (year, month, day of month)
• The time of the epoch interval can be given as SSSSS (second of day 0-86399) or HHMMSS ( hour, minute, second )
• An unlimited begin or end of an epoch interval can be given using zeros in the date and time values (e.g. 0000000:000000)

See also the examples below.

3.10.5.1 Update - Single Header Element

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

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

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

See some examples below:

```
update_insert :

#--------------
O - POTSDAM.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" }
```
<table>
<thead>
<tr>
<th>Substitute String</th>
<th>Substitute/Example</th>
<th>Description</th>
<th>Environment variables</th>
</tr>
</thead>
<tbody>
<tr>
<td>uSeR</td>
<td>nisn</td>
<td>user name provided by os</td>
<td>USERNAME, USER</td>
</tr>
<tr>
<td>pRoGrAm</td>
<td>gfzrnx-1.08-8003</td>
<td>gfzrnx-version-revision</td>
<td></td>
</tr>
<tr>
<td>hOsTnAmE</td>
<td>serv01</td>
<td>simple hostname provided by os</td>
<td>HOSTNAME</td>
</tr>
<tr>
<td>hOsTdOmAiNnAmE</td>
<td>serv01.gfz-potsdam.de</td>
<td>fully qualified hostname provided by os</td>
<td>HOSTFQDN</td>
</tr>
<tr>
<td>dOmAiNnAmE</td>
<td>gfz-potsdam.de</td>
<td>domain name provided by os</td>
<td>USERDOMAIN</td>
</tr>
<tr>
<td>tImEsTaMp</td>
<td>20170712 113126 UTC</td>
<td>time stamp of current time</td>
<td></td>
</tr>
</tbody>
</table>
Remark: Please check in advance if you get the right results for your operating system!

```
update_insert :
#---
MNO - ALL:
"COMMENT" : "PG tiMeTaMp pRoGrAm uSeR@dOmAiNnAmE"
"PGM / RUN BY / DATE" : { 0: "pRoGrAm" , 1: "uSeR@dOmAiNnAmE", 2: "tiMeTaMp" }
```

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 cutted to 60!

### 3.10.5.3 Update - Multi Header

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

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

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

CC = condition string

Here a crux example block.

```
update_insert :
#---
M - 2015209:00000 0000000:00000 - ALL :
"SENSOR MOD/TYPE/ACC" + 3:"TD" : { 0:"Vaisala" , 1:"FTU 303/5.14", 2:"0.10" } 
"SENSOR MOD/TYPE/ACC" + 3:"PR" : { 0:"Vaisala" , 1:"FTU 303/5.14", 2:"0.05" } 
"SENSOR MOD/TYPE/ACC" + 3:"HR" : { 0:"Vaisala" , 1:"FTU 303/5.14", 2:"1.7" } 
"SENSOR MOD/TYPE/ACC" + 3:"XX" : { 0:"XXXXXXX" , 1:"XXX 125", 2:"1.0" } 
M - POTSODEU :
"SENSOR POS XYZ/H" + 4:"TD" : { 0:"3275753.9120" , 1:"321110.8661", 2:"5445041.8829", 3:"5" } 
"SENSOR POS XYZ/H" + 4:"PR" : { 0:"3275753.9120" , 1:"321110.8661", 2:"5445041.8829", 3:"5" } 
"SENSOR POS XYZ/H" + 4:"HR" : { 0:"3275753.9120" , 1:"321110.8661", 2:"5445041.8829", 3:"5" } 
"SENSOR POS XYZ/H" + 4:"XX" : { 0:"3275753.9120" , 1:"321110.8661", 2:"5445041.8829", 3:"5" } 
```

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

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

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

pots3410.15m
2.11 METEOROLOGICAL DATA

TPP 3.1

RINEX VERSION / TYPE
POM / RUN BY / DATE

pots

3 TD HR PR
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

3.03 METEOROLOGICAL DATA

TPP 3.1

RINEX VERSION / TYPE
POM / RUN BY / DATE

pots

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

END OF HEADER

3.10.5.4 Proposed Use

There are several possibilities to organize the header editing configuration file. The most clear form would be to organize it per station.

Below you can find a configuration example for the single station POTS covering the whole station history information for Observation and Meteo file header entries.

update_insert:

"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 1999231:00000 : { 0: "289", 1: "ROGUE SNR-8000", ...}
"REC # / TYPE / VERS" + 1999232:00000 2000232:00000 : { 0: "281", 1: "ADA SNR-8000 ACT", ...}
"REC # / TYPE / VERS" + 2000233:00000 2009089:00000 : { 0: "205", 1: "JAVAD TRE_G3TH DELTA", ...}
"REC # / TYPE / VERS" + 2009089:00000 2011046:61200 : { 0: "205", 1: "JAVAD TRE_G3TH DELTA", ...}
"REC # / TYPE / VERS" + 2011046:32400 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: "0.261", 1: "ADA/M_T", 2: "NONE" }
"ANTENNA: DELTA H/E/N" + 1995276:28800 2009105:47700 : { 0: "0.235", 1: "ADA/M_T", 2: "NONE" }
"ANTENNA: DELTA H/E/N" + 2009105:47700 2011046:61200 : { 0: "0.354", 1: "ADA/M_T", 2: "NONE" }
"ANTENNA: DELTA H/E/N" + 2011046:61200 0000000:00000 : { 0: "0.316", 1: "JAV_RINGANT_G3T", 2: "NONE" }

Wed Nov 6 14:50:05 UTC 2019
Helmholtz Centre Potsdam / GFZ German Research Centre for Geosciences
Depending on the first data epoch the appropriate header entry is updated.

### 3.10.5.5 Remark

There is one exclusion concerning the RINEX header fields manipulation. According to IGS antenna definition (number, antenna + radome) the "ANT♯ / TYPE" record consists of 3 columns, which is a deviation from the RINEX standard.

This means, the standard (A20,A16,A4) RINEX definition is in {\textit{gfzrnx}} handled as (A20,A16,A4). A correction record should be of the following form:

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

### 3.10.5.6 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>" ]
```

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

### 3.10.5.7 Header Label Independent String Replacement

For the string replacement the major mode \textit{replace} has to be used. One has to define the station identifier as before. Afterwards you can define from/to pairs of type \textit{regexp} or \textit{string}. The \textit{regular expression} syntax follows \textit{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 : "^(\d{60})PGM\s*/\s*RUN\s*/\s*BY\s*/\s*DATE\s*$"
  regexp_to : "$1PGM / RUN BY / DATE"
ALL:
  string_from : "PGM/RUN BY/DATE"
  string_to : "PGM / RUN BY / DATE"

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

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

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

replace :
#--------
ALL:
  regexp_from : "^\d{60}ABC DEF.*$"
  regexp_to : ""

3.10.5.8 Rename - PRNs

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

Here some examples:

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

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

E51 - E01 : ALL
E52 - E02 : ALL

3.10.5.9 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

*2* - *1* - C : ALL
*2 - *1 - C : ALL
3.10.5.10 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.

3.10.6 Header edit via command line (-cx_updins)
Single update_insert header edit options can be provided also 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:

```
```
gfzrnx -finp /datadir/VALD00CAN_R_20181001200_01H_30S_MO.rnx -fout /datadir/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"'
```
```

3.10.6.1 Remark
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"'
```
```

3.10.7 Internal/Data Headers via crux-file (-cx_addinthd)
Meta data changes following e.g. hardware changes can be introduced at the event epochs into the data part of a RINEX file if an 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 an an example:

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

The following crux-configuration
```
```
update_insert:
```
```
0 - 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")
0 - 20180417:061500 00000000:000000 - OBWT:
"REC # / TYPE / VERS" : { 0: "1705310", 1: "LEICA GR30", 2: "4.20.232")
"ANT # / TYPE" : { 0: "09440002", 1: "LEIA25.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
38.700
23773818.648 127129528.196 4 18 04 17 06 15 00.0000000 4 2
1705310 LEICA GR30 4.20.232 REC # / TYPE / VERS
09440002 LEIA25.R3 LEIT ANT # / TYPE
18 04 17 06 15 00.0000000 0 16G02G05G07G09G13G27G28G30R06R07R08R09
R10R16R23R24
3.10.8 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 to this version. If a special version number is needed (for what ever reason) one can use the -vnum command line parameter to manipulate the version number to a certain value.

<table>
<thead>
<tr>
<th>3.04 OBSERVATION DATA</th>
<th>M</th>
<th>RINEX VERSION / TYPE</th>
</tr>
</thead>
<tbody>
<tr>
<td>gfzrnx -finp ... -vnum 3.03</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>3.03 OBSERVATION DATA</th>
<th>M</th>
<th>RINEX VERSION / TYPE</th>
</tr>
</thead>
</table>

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

3.10.8.1 Remark

The -vnum version number change is only a formal exchange of the version number to fulfill e.g. external circumstances over which one has no influence. The file content will be still conform to highest supported version number!

3.11 Rinex File Meta Data Extraction (-meta)

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

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

- The basic mode extracts only the header information and the first, last epoch from the RINEX file without reading the whole file (fast).
- The 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.
- There are supported the following output formats: txt (default), json, jsonp (pretty json), xml, dump to be used for fast view or further applications.

The file-, site-, receiver-, antenna- sections information is derived from the RINEX header part only and the data-section holds information derived from the RINEX data part. Here some simple examples:

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

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

```
gfzrnx -finp pots0070.15o -meta basic.json -fout pots0070.15o.json
```

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

```
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 00.000000
    interval = 30.000
    last = 2018 03 05 23 59 30.000000
  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 00.000000
    interval = 30.000
    md5 = 9a49ad078b4bcbe1d1a2f4de440de1
    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
      R = 1 2
    sysobs:
      E = C1X C5X D1X L1X S1X S5X
      G = C1C C1W C2W C2X C5X D1C D1W D2W D2X D5X L1C L1W L2X L2W 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
    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

rnxall -finp pots0070.15o -meta basic:jsonp
rnxall -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" : "9a49ad078b4bcfbeb12a2fe4de400d1",
    "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"
      ]
    }
  }
}
3.12 Rinex File Comparison (-fdiff)

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

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

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

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

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

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

```
<table>
<thead>
<tr>
<th>3.00</th>
<th>DATA COMPARISON</th>
<th>RINEX VERSION / TYPE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>----------------</td>
<td>---------------------</td>
</tr>
<tr>
<td>pots0140.16o_1</td>
<td>FILE_1</td>
<td></td>
</tr>
<tr>
<td>pots0140.16o_2</td>
<td>FILE_2</td>
<td></td>
</tr>
</tbody>
</table>

...  

| 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 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:
In the first epoch the data of two PRNs differ by "1" in the LLI (loss of lock indicator) value for the C1C observation type.

In the second epoch the PRN G02 differs (file1-file2) by the given values for the observation types C1C, L1, C2W.

The PRN G19 seems to be fully missing in one of the files or you see a merged record, where an observation type is missing either in the the first or the second file.

The third epoch seems to be fully missing in one of the files or you see a merged record, where a full PRN or an observation type is missing either in the the first or the second file.

3.13 Rinex 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 for 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 - | 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
3.14 Rinex to Tabular Observations Output

3.14.1 Standard Output

The tabular observations output allows to convert a RINEX observations input file into a data table which can be used for simple visualization or for 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.

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

```
gefrnx -finp POTSO00DEU_R_20150070000_01D_30S_MO.rnx -tab_obs -fout POTSO00DEU_2015007_G03.tab
gefrnx -finp POTSO00DEU_R_20150070000_01D_30S_MO.rnx -tab_obs -fout POTSO00DEU_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):

<table>
<thead>
<tr>
<th>Line type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>#HD</td>
<td>header line with column description</td>
</tr>
<tr>
<td>OBS</td>
<td>observation line</td>
</tr>
</tbody>
</table>

The first columns are fix, showing the: + Line Type + Satellite System, + Date, + Time, + PRN, followed by the list of wished or given observation types as provided in the satellite system specific header line order.

3.14.2 Date/Time Formats

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

<table>
<thead>
<tr>
<th>Date Pattern</th>
<th>Example</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>mjd</td>
<td>57029</td>
<td>Modified Julian Date (MJD)</td>
</tr>
<tr>
<td>ddd</td>
<td>007</td>
<td>day of year</td>
</tr>
<tr>
<td>wwwwd</td>
<td>18263</td>
<td>gps-week,weekday</td>
</tr>
<tr>
<td>wwwww-d</td>
<td>1826-3</td>
<td>gps-week,weekday</td>
</tr>
<tr>
<td>yyyyddd</td>
<td>2015007</td>
<td>year, day of year</td>
</tr>
<tr>
<td>yyyy-ddd</td>
<td>2015-007</td>
<td>year, day of year</td>
</tr>
<tr>
<td>yyyyymmdd</td>
<td>20150107</td>
<td>year, month, day of month</td>
</tr>
<tr>
<td>yyyy-mm-dd</td>
<td>2015-01-07</td>
<td>year, month, day of month</td>
</tr>
<tr>
<td>yymmdd</td>
<td>150107</td>
<td>2-digit year, month, day of month</td>
</tr>
<tr>
<td>yy-mm-dd</td>
<td>15-01-07</td>
<td>2-digit year, month, day of month</td>
</tr>
</tbody>
</table>
### Operation / Tasks

#### Time Pattern

<table>
<thead>
<tr>
<th>Time Pattern</th>
<th>Example</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>hhmmss</td>
<td>01:35:16.000000</td>
<td>hour, minutes, seconds</td>
</tr>
<tr>
<td>hh:mm:ss</td>
<td>01:35:16.000000</td>
<td>hour, minutes, seconds</td>
</tr>
<tr>
<td>sod</td>
<td>5716.000000</td>
<td>seconds of day</td>
</tr>
<tr>
<td>fod</td>
<td>0.0661574074074</td>
<td>fractions of day</td>
</tr>
</tbody>
</table>

The above used Date/Time pattern **ddd** and **sod** lead to the output below.

---

#### 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 simply empty.

---

3.15 Rinex Standard Extensions/NonConformity

3.15.1 RINEX-2 BDS,QZSS,IRNSS support

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

3.15.1.1 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`:

<table>
<thead>
<tr>
<th>System</th>
<th>Letter</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>BDS</td>
<td>c</td>
<td>pots0750.17c</td>
</tr>
<tr>
<td>QZSS</td>
<td>j</td>
<td>pots0750.17j</td>
</tr>
<tr>
<td>IRNSS</td>
<td>i</td>
<td>pots0750.17i</td>
</tr>
</tbody>
</table>

3.15.2 RINEX-2 to RINEX-3 conversion

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

3.15.3 Handling of unsupported observation types

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

- Extract the hardcoded table from gfzrnx executable.

```
gfzrnx -out_obs_map
```

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

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

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

3.15.4 Remark

Please use this feature with special caution!

Be aware that this undermines the given RINEX standard and can be an error source if not used properly.

The generated files should be for internal use only!
List of Figures

1.1 Twitter: @gfzrnx .......................................................... 5
2.1 Unix Terminal - command line ............................................. 8
2.2 MS Windows command window - command line ...................... 8
3.1 Editor Window - ASCII Timeplot per PRN and Observation Type ...................................................... 24