GRUAN Technical Note 11

Brief Description of GruanToolRs92 (gt92)

Michael Sommer
Abstract

The tool gt92 is a powerful Java-based command-line utility. It was developed at the GRUAN Lead Centre for the daily work and is actively maintained and further developed by the author. It provides possibilities for converting and extracting data-tables and meta-data from several file-types like DC3DB, GNC-RAW, GNC-DATA, MWX, GSFZ, STRATO-ZIP and IGNC-RAW. In addition to these base functionality the tool gt92 has included some more features, e.g. handling and decoding of XData frames, gridding and merging of data tables, test and check files, identify supported file types. An internal batch processing mode is implemented for all main functions of the tool gt92. That means, the tool could e.g. convert hundreds of files with one run. Current version of this brief description is related to version 0.5.x of gt92.

Note: Please contact the author or the GRUAN Lead Centre (gruan.lc@dwd.de) if required file formats are not supported yet, if any functions are missed, or if a bug is found.

Note: Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not constitute or imply its endorsement by the GRUAN.

Revision history

<table>
<thead>
<tr>
<th>Version</th>
<th>Author / Editor</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0 (2020-10-01)</td>
<td>Michael Sommer</td>
<td>First published version as GRUAN-TN-11</td>
</tr>
<tr>
<td>0.8.0 (2019-04-04)</td>
<td>Michael Sommer</td>
<td>First draft version as GRUAN-TN using L\TeX{}</td>
</tr>
<tr>
<td>0.7.4 (2016-02-18)</td>
<td>Michael Sommer</td>
<td>Draft version as GRUAN-IN-4</td>
</tr>
</tbody>
</table>
# Table of contents

1 Installation .................................................. 6
   1.1 System requirements for running GruanToolRS92 (gt92) .......... 6
   1.2 Download and install Java ................................... 6
   1.3 Download tool GruanToolRs92 (gt92) ......................... 6
   1.4 Install tool GruanToolRs92 (gt92) ......................... 6

2 Usage ......................................................... 6
   2.1 Main purposes ............................................. 7
   2.2 File list .................................................. 7
   2.3 Supported file formats .................................... 7
      2.3.1 DC3DB (input) ......................................... 8
      2.3.2 MWX (input) ........................................... 8
      2.3.3 GSFZ (input) ........................................... 8
      2.3.4 STRATO-ZIP (input) .................................. 9
      2.3.5 GNC-DATA (input) .................................... 9
      2.3.6 IGN1-RAW (input) .................................... 9
      2.3.7 GNC-RAW (input, output) ............................ 10
      2.3.8 LDF (output) .......................................... 10
      2.3.9 CSV (output) .......................................... 11
      2.3.10 MD-TXT (output) .................................... 12
      2.3.11 TSV (output) .......................................... 14
      2.3.12 DC3-DUMP (output) .................................. 14
   2.4 Supported XData decoding .................................. 15

3 Options ....................................................... 15
   3.1 Option convert (-c, --convert) ............................. 15
   3.2 Option extract (-x, --extract) ............................. 16
   3.3 Option checksum (-k, --checksum) ........................ 18
   3.4 Option test (-t, --test) .................................. 18
   3.5 Option distribute (-d, --distribute) ....................... 19
   3.6 Option identify (-y, --identify) .......................... 19
   3.7 Option itemize (-z, --itemize) ............................ 19
   3.8 Option info (-i, --info) .................................. 19
   3.9 General options ........................................... 20

4 Examples .................................................... 21
   4.1 Example 1 – Convert GNC-DATA files to LDF format ........... 21
   4.2 Example 2 – Extract all data tables from DC3DB as TSV files ... 21
   4.3 Example 3 – Extract meta-data only from GNC-RAW ............. 21
   4.4 Example 4 – Extract a filtered part of meta-data from DC3DB .... 21
   4.5 Example 5 – Extract a filtered part of data tables from DC3DB .... 21
   4.6 Example 6 – Convert DC3DB files to GNC-RAW files ........... 21
   4.7 Example 7 – Convert MWX files to GNC-RAW files ............ 22
   4.8 Example 8 – Extract all data tables and meta-data from MWX files .... 22
4.9 Example 9 – Extract and collect a filtered part of meta-data to store in one general file

Appendix

A Full list of options .................................................. 24

B History of development of gt92 .................................. 26
   B.1 History of Version 0.5.x ...................................... 26
   B.2 History of Version 0.4.x ...................................... 26
   B.3 History of Version 0.3.x ...................................... 31

C Collection of tables describing XData columns ............... 32

D Example of file description ........................................ 41

Acronyms ............................................................... 43
1 Installation

1.1 System requirements for running GruanToolRS92 (gt92)

For using the gt92 following two requirements should be satisfied:

- a computer with any operating system (like Windows, Linux, Mac OS, ...)
- installed Java 8 (newer versions are not tested yet)

1.2 Download and install Java

The latest JRE should be installed on the preferred operating system. If a JRE is not installed, the latest version is available for download from:

- Oracle: www.java.com/en/download (last free update from January 2019), or
- AdoptOpenJDK: adoptopenjdk.net (security updates until at least September 2023).

1.3 Download tool GruanToolRs92 (gt92)

The latest version of the gt92 tool is available on the related page at the GRUAN website (www.gruan.org/data/software/gt92). The download option is visible after login only. Please register yourself at www.gruan.org/user/registration, if not yet registered.

1.4 Install tool GruanToolRs92 (gt92)

There are two types of files included in the packed zip-file. The main program file is a Java archive file (".jar"). All other files ("gt92.*") are start scripts for the different operating systems, like "gt92.bat" for Windows.

Unpack all files to a folder of your choice. Then this program (or the folder) should be announced to the system. There are two options:

- Edit the PATH system variable and add the directory of installation.
- Edit the dedicated script and correct the (absolute) path to the jar file, than copy this script in a folder which is known for executable programs to the system (e.g. a bin folder).

2 Usage

Common usage is easy if the user is familiar with command-line tools. The tool can be specifically instructed with the available options.

```
gt92 [OPTION]... [FILE]...
```
2.1 Main purposes

The tool can be used in the following modes (OPTION):

- **Convert** (-c) – Convert the file to another file format (see section 3.1)
- **Extract** (-x) – Extract meta-data and/or data-tables as CSV (see section 3.2)
- **Checksum** (-k) – Compute the checksum of the file (see section 3.3)
- **Test** (-t) – Test the file for correctness (see section 3.4)
- **Distribute** (-d) – Prepare a MWX file for external distribution (see section 3.5)
- **Identify** (-y) – Identify file formats for all given files (see section 3.6)
- **Itemize** (-z) – Itemize all tables of given files (see section 3.7)
- **Info** (-i) – Print a specific part of information (see section 3.8)
- **Version** (-v) – Print a general information about this program version (see section 3.9)
- **Help** (-h) – Print full list of options and all other description texts (see appendix A)

2.2 File list

The tool command *(FILE)* is designed to be able to handle multiple files and directories. All of these files (or if a directory is given all included files) are processed step be step. That allows the conversion of a mass of data files with one command. *(FILE)* can be called with one of following options:

- one file,
- a list of files,
- one directory with all included files,
- a list of directories with all included files,
- a combination of the above.

2.3 Supported file formats

The tool *gr92* supports handling of several file formats used for storing radiosounding data. These file formats include: original raw data files from manufacturer (often proprietary), specific GRUAN formats (often NetCDF) and text files.

File formats which can be read by this tool (input):

- **DC3DB** – Vaisala sounding file of MW31 or MW21 (see 2.3.1)
- **MWX** – Vaisala sounding file of MW41 (see 2.3.2)
- **GSFZ** – Packed Graw sounding files of GrawMet v5 (see 2.3.3)
- **STRATO-ZIP** – Zipped Strato Sounding File (see 2.3.4)
- **GNC-DATA** – GRUAN product data file (see 2.3.5)
- **IGNC-RAW** – IPSL GRUAN NetCDF Radiosonde Raw Data File (see 2.3.6)
- **GNC-RAW** – GRUAN raw data file (see 2.3.7)
File formats which can be written by this tool (output):

- **GNC-RAW** – GRUAN raw data file (see 2.3.7)
- **LDF** – Lindenberg data file (see 2.3.8)
- **CSV** – Comma separated values file (see 2.3.9)
- **MD-TXT** – Meta-data text file (see 2.3.10)
- **TSV** – Special Vaisala text file (see 2.3.11)
- **DC3-DUMP** – Special Vaisala DUMP file (see 2.3.12)

### 2.3.1 DC3DB (input)

The Vaisala DigiCORA® 3 DataBase file format ("*.dc3db" or "*.dc3") is referred to as DC3DB. These files could only be created by the outdated Vaisala DigiCORA® Sounding Systems MW31 and MW21. The DC3DB files are proprietary Microsoft DataBase files (MDB – "*.mdb") of Jet version 4.0. DC3DB has a well defined internal structure, which is not openly documented by Vaisala. The format is partly described in the Sommer (2020b).

The DC3DB files contain data tables (raw data, intermediate results, product data) and a complex metadata tree. The format is used for the following Vaisala radiosondes: RS80, RS90, RS92. It may also contain data from additional sensors, such as ozone probes.

Some details of the tables and metadata included can be found in the following document Vaisala (2015).

### 2.3.2 MWX (input)

The data file format MWX is the default output format of the current Vaisala DigiCORA® Sounding System MW41. A MWX file is a ZIP file, containing single table XML files. Each table contains either metadata, or data, including raw data, intermediate results or product data. Data from both RS41 and RS92 radiosondes can be stored in the MWX format. In addition it can contain data from auxiliary sensors connected via XData, including ECC Ozone sonde, CFH, and others.

Some details of the tables and metadata included can be found in the following document Vaisala (2019).

The gt92 tool can decode XData from some auxiliary sensors (see 2.4). In addition, a gridded multi-column table, including decoded XData, can be generated.

### 2.3.3 GSFZ (input)

The GSFZ data file format is a packed version of default output format of the Graw Sounding Software GRAWMET. Normally a series of Graw Sounding Files (GSF) is created during sounding ("*.gsf", "*.gsf1", "*.gsf2", ...). All these files are XML files and packed together in a GSFZ. A detailed documentation of internal structure of these XML files is not freely available. The content of these XML files is self-explanatory. The files contain both raw data and meta-data.
2.3.4 STRATO-ZIP (input)

The data file format STRATO-ZIP is a ZIP file which contains all sounding output files of software Strato. Strato is employed in cases of CFH flights where the InterMet iMet-1 is used as telemetry sonde.

A detailed description of possible output files of Strato can be found in the “Cryogenic Frost-point Hygrometer - Operations Manual” (Vömel, 2014a) and in the “Description of the Strato and Balloon - Data output” (Vömel, 2014b).

2.3.5 GNC-DATA (input)

The GNC-DATA data format is a specific NetCDF file. It is used for various GRUAN data products for radiosounding. The original variant uses NetCDF version 3, newer variants use NetCDF version 4. The GNC-DATA data format is not standardised and varies between different data products and their versions. Nevertheless, all data products follow the following rules:

- All variables together form exactly one table (all have the same dimension).
- The CF Convention is supported, of course, different versions over the years.
- Additional metadata is integrated with special GRUAN prefixes: g. and g. as global and local attributes, respectively.

The document “Brief Description of the RS92 GRUAN Data Product (RS92-GDP)” (Sommer et al., 2016) can be used as one example for a detailed description.

2.3.6 IGNC-RAW (input)

The gt92 tool recognises the “IPSL GRUAN Radiosonde Raw Data NetCDF File” data format under the IGNC-RAW name. The original name of this format is “Format of L1a GRUAN netCDF file for Meteomodem M10 radiosonde”. The format is described in an internal SIRTA/IPSL document and is available at GRUAN Lead Centre.

The IGNC-RAW format uses the grouping variables functionality of NetCDF Version 4. Currently the following groups are incorporated:

- station – Meteorological station/site
- rs_station – Radiosounding launching station
- balloon – Balloon type
- chamber_100_rh – Chamber at 100% of relative humidity
- gc_shelter – Ground calibration shelter
- std_met_station – Standard meteorological ground measurements
- ground_station – Meteomodem ground check
- launch – Launch
- burst – Burst
- rs – Radiosounding
2.3.7 GNC-RAW (input, output)

The GNC-RAW data format is based upon NetCDF Version 4 files with a specially defined structure. Its full name is “GRUAN NetCDF Raw Data Format for Radiosounding” and is developed to store raw data files from radiosounding systems of different manufacturers, standardised for the GRUAN network. This data format is intended to simplify the work with data from multiple manufacturers and sondes within the GRUAN network. Currently the tool gt92 supports the following original file formats: DC3DB, MWX, GSFZ, STRATO-ZIP.

Note: It is possible to integrate support for additional file formats. However, this requires an exact description of these formats. Please contact the author or the GRUAN Lead Centre (gruan.lc@dwd.de), if another original data format should be integrated.

This file format is the only one that can be used as input, e.g. using option extract (-x) (see section 3.2) and can be created as output using option convert (-c) (see section 3.1). The tool provides one special option to choose between the current and an old version of GNC-RAW:

• --nc-old – Convert a file in a NetCDF3 file without compression. Only possible with DC3DB files. This converts to the old version of GNC-RAW.

Note: This option is only available for compatibility reasons. Please use this option only, if you exactly know what the difference to newer GNC-RAW version 2 is. For example, such files are required as input files in case of RS92-GDP.2 processing.

The data format GNC-RAW is described in detail in the Sommer (2020a).

2.3.8 LDF (output)

LDF is an ASCII text file with a header which includes meta-data and a body which contains a data table.

All header lines are marked as comment lines and are starting always with a hash sign #. Each header line can contain a key/value pair which is separated by an equal sign =. A long list of sounding-related meta-data are included. In addition all columns of the data table are defined using the following attributes:

• Name – short free name/title
• Unit – unit
• StandardName – standard name of CF convention
• LongName – free long name or comment
• FormatType – data type (e.g. FLT, DBL, BOL, INT, LNG, STR, ...)
• FormatWidth – width of formatted column
• FormatNaN – used not a number text
• FormatFormat – used Fortran format for formatting of column

The data table is similar to a CSV file with a semicolon ; as default delimiter. All columns are formatted with a fixed width. The following preview shows a file content example:
2.3.9 CSV (output)

CSV is a comma separated values file in ASCII text.

The tool provides a couple of options to justify or permit the export:

- `--no-csv` – Extract no data tables as default csv files.
- `--no-csv-nan` – Do not print NaN values in csv files.
• **--csv-delimiter <DELIMITER>** – Use given delimiter within csv files.

The following example illustrates the CSV export options:

```plaintext
# original without any additional option
1 time [sec];P [hPa];T [K];U1 [%];U2 [%]
2 0.5625522136688232;996.571533203125;296.3365783691406;NaN;-0.01440700888633728
3 1.5625898838043213;996.5634155273438;296.38214111328125;NaN;-0.027024805545806885
4 2.5626277925583984;996.4644775390625;296.34832763671875;NaN;-0.026787370443344116
5 3.5626678466796875;996.5783081054688;296.3655090332031;NaN;-0.04264324903488159
6 4.562672138214111;996.5721435546875;296.36578369140625;NaN;-0.03854069113731384
7 5.562672138214111;996.5721435546875;296.36578369140625;NaN;-0.04685834050178528
8

# option --csv-delimiter ","
9 time [sec],P [hPa],T [K],U1 [%],U2 [%]
10 0.5625522136688232,996.571533203125,296.3365783691406,NaN,-0.01440700888633728
11 1.5625898838043213,996.5634155273438,296.38214111328125,NaN,-0.027024805545806885
12

# option --no-csv-nan
13 time [sec];P [hPa];T [K];U1 [%];U2 [%]
14 0.5625522136688232;996.571533203125;296.3365783691406;-32768.0;-0.01440700888633728
15 1.5625898838043213;996.5634155273438;296.38214111328125;-32768.0;-0.027024805545806885
16

2.3.10 MD-TXT (output)

Most original data files include both meta-data and data tables. During extraction meta-data is exported as meta-data text file **MD-TXT**. This meta-data text file has a very simple format. Each meta-data item is exported on separate line with an unique key and a value separated by “=” . The meta-data key is often like a path in a directory tree using a specific separator. This separator char depends on the original file format (e.g. “.” or “\” or “/”).

The tool provides a lot of options to justify, filter or permit the meta-data export:

• **--no-meta** – Extract no meta-data.
• **--meta-regex <REG-EX>** – Extract meta-data only if it matches the regEx.
• **--meta-sep <SEPARATOR>** – Extract meta-data using this separator for names (not default or original one).
• **--meta-osep** – Extract meta-data items using original separator in names (not default one).
• **--no-meta-group** – Extract no group items of meta-data.
• **--no-meta-name** – Extract only values of meta-data (not the names).
• **--no-meta-root** – Extract meta-data items without root.
• **--no-meta-valid** – Extract meta-data items using original names (not valid one).

Following example is a part of an extracted meta-data file of a **DC3DB**:
The following example shows the influence of different options:

```plaintext
# original without any additional option
E4211940!00\OperatorName=ozon
E4211940!00\RsGroundCheck=2010-05-26T16:50:16.593 (00|03)
E4211940!00\RsGroundCheck\Corrections=2010-05-26T16:50:16.609 (00|03)
E4211940!00\RsGroundCheck\Corrections\Temperature=-0.272957
E4211940!00\RsNumber=E4211940

# option --no-meta-root
E4211940!00\OperatorName=ozon
E4211940!00\RsGroundCheck=2010-05-26T16:50:16.593 (00|03)
E4211940!00\RsGroundCheck\Corrections=2010-05-26T16:50:16.609 (00|03)
E4211940!00\RsGroundCheck\Corrections\Temperature=-0.272957
E4211940!00\RsNumber=E4211940

# option --no-meta-group
E4211940!00\OperatorName=ozon
E4211940!00\RsGroundCheck\Corrections\Temperature=-0.272957
E4211940!00\RsNumber=E4211940

# option --no-meta-name
2010-05-26T16:33:27.859 (00|03)
ozon
2010-05-26T16:50:16.593 (00|03)
2010-05-26T16:50:16.609 (00|03)
-0.272957
E4211940

# option --meta-sep "," 
E4211940!00.RsGroundCheck.Corrections.Temperature=-0.272957

# option --meta-regex ".*(RsNumber).*"
E4211940!00\RsNumber=E4211940
E4211940!00\SOUNDING_DATA\RsNumber=E4211940!00
```
2.3.11 TSV (output)

A *TSV* file is a converted version of a binary data table which is included in *DC3DB* files (see section 2.3.1). The binary data table itself can also be extracted as a dump (see next section 2.3.12).

During extraction, *TSV* can be activated and fraction of numbers in the data table can be optimised:

- `--tsv` – Extract data tables in addition as tsv files. Only possible with *DC3DB* files.
- `--tsv-frac <FRACTION>` – Formatted numbers in tsv files use this fraction. Only expedient with option tsv.

Following example of a *TSV* file includes the header lines and some lines of data table:

<table>
<thead>
<tr>
<th>Record name:</th>
<th>Unit:</th>
<th>Data type:</th>
<th>Divisor:</th>
<th>Offset:</th>
</tr>
</thead>
<tbody>
<tr>
<td>time</td>
<td>sec</td>
<td>float (4)</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>P</td>
<td>hPa</td>
<td>float (4)</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>T</td>
<td>K</td>
<td>float (4)</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>U1</td>
<td>%</td>
<td>float (4)</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>U2</td>
<td>%</td>
<td>float (4)</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

| 503.95       | 996.87 | 293.17   | 32768.00 | 35.46   |
| 504.95       | 997.05 | 293.11   | 32768.00 | 35.53   |
| 505.95       | 996.83 | 292.92   | 32768.00 | 36.28   |
| 506.95       | 996.86 | 292.88   | 32.94    | -32768.00 |
| 507.95       | 996.86 | 292.87   | 33.60    | -32768.00 |
| 508.95       | 996.95 | 292.87   | 34.10    | -32768.00 |

2.3.12 DC3-DUMP (output)

The data format *DC3-DUMP* is a proprietary binary file of one data table which is included in the *DC3DB* file (see section 2.3.1). Such files are rarely used, because the tool decodes the content anyway. The format is further explained in the document *Sommer* (2020b).
2.4 Supported XData decoding

The tool gt92 supports decoding of XData frames which might be part of raw data files. XData frames of following instruments can be decode:

- **V7** – Innovative System Designs V7 Ozone interface board (id=01)
- **OIF411** – Vaisala Ozone interface board (id=05)
- **CFH** – Cryogen Frostpoint Hygrometer (id=08)
- **FPH** – Frost Point Hygrometer (id=10)
- **COBALD** – Compact Optical Backscatter and Aerosol Detector (id=18)
- **PCFH** – Peltier Cooled Frost point Hygrometer (id=3C)
- **FLASH-B** – Fluorescence Lyman-Alpha Stratospheric Hygrometer for Balloon (id=3D)
- **SKYDEW** – Peltier-based chilled-mirror hygrometer “SKYDEW” (id=3F)

**Note:** Please see tables included in appendix C.

3 Options

3.1 Option convert (-c, --convert)

Option `-c` is a very powerful option to convert a list of FILEs to another file format. The result format is currently pre-defined:

- Original raw data file
  - `DC3DB` (see 2.3.1) ⇒ to `GNC-RAW` (see 2.3.7)
  - `MWX` (see 2.3.2) ⇒ to `GNC-RAW` (see 2.3.7)
  - `GSFZ` (see 2.3.3) ⇒ to `GNC-RAW` (see 2.3.7)
  - `STRATO-ZIP` (see 2.3.4) ⇒ to `GNC-RAW` (see 2.3.7)
- NetCDF files
  - `GNC-DATA` (see 2.3.5) ⇒ to `LDF` (see 2.3.8)
  - `GNC-RAW` (see 2.3.7) ⇒ *not possible at moment*
  - `IGNC-RAW` (see 2.3.6) ⇒ *not possible at moment*

**Note:** To get more information about supported file formats see section 2.3.

Relevant options to parametrise convert mode:

- Directories
  - `-w, --work-dir <DIR>` – The working directory for all temporary files. If not given the current working directory where the tool is installed is used as default.
  - `-o, --out-dir <DIR>` – The output directory for all converted files. The working directory is used as default, if not specified.
3.2 Option extract (-x, --extract)

Option -x is a very powerful option to extract data tables and/or meta-data from a list of FILE and save as ASCII text files, e.g. CSV or TSV:

- **MWX, GSFZ, STRATO-ZIP, GNC-RAW, IGNC-RAW** ⇒ possible result files are: 
  - "*.csv",
  - "*_metadata.txt",
  - "<source-file-name>" (if included)
- **DC3DB** ⇒ possible result files are: 
  - "*.csv",
  - "*_metadata.txt",
  - "*.tsv",
  - "*.dump"
- **GNC-DATA** ⇒ possible result files are: 
  - "*.csv",
  - "*_metadata.txt"
Relevant options to parametrise extract mode:

- **Directories**
  - `--work-dir <DIR>` – The working directory for all temporary files. If not given the current working directory of program is used as default.
  - `--out-dir <DIR>` – The output directory for all extracted files. If not given the working directory is used as default.
  - `--separate` – Separate all extracted output files for every input file.

- **Export file formats**
  - `--no-csv` – Extract no data tables as default csv files.
  - `--no-csv-nan` – Do not print NaN values in csv files.
  - `--csv-delimiter <DELIMITER>` – Use given delimiter within csv files.
  - `--dump` – Extract data tables in addition as binary dump files. Only possible with DC3DB files.
  - `--tsv` – Extract data tables in addition as tsv files. Only possible with DC3DB files.
  - `--tsv-frac <FRACTION>` – Formatted numbers in tsv files use this fraction. Only expedient with option tsv.
  - `--no-source` – Do not extract an included source file during extracting a file. Only possible with GNC-RAW files.

- **Meta-data**
  - `--no-meta` – Extract no meta-data.
  - `--meta-regex <REG-EX>` – Extract meta-data only if it matches the regEx.
  - `--meta-sep <SEPARATOR>` – Extract meta-data using this separator for names (not default or original one).
  - `--meta-osep` – Extract meta-data items using original separator in names (not default one).
  - `--no-meta-group` – Extract no group items of meta-data.
  - `--no-meta-name` – Extract only values of meta-data (not the names).
  - `--no-meta-root` – Extract meta-data items without root.
  - `--no-meta-valid` – Extract meta-data items using original names (not valid one).
  - `--meta-out-delimiter <DELIMITER>` – Extract meta-data using this delimiter (file name, meta-data name, meta-data value) for general output file.
  - `--meta-out-file <FILE>` – Extract all meta-data of all files to this general meta-data output file.
  - `--no-meta-out-units` – Extract meta-data items without units (if available) to general output file.

- **Data tables**
  - `--no-tables` – Extract no data tables.
– `--import-table-filter <REG-EX>` – Import (read) tables only if they matches the regEx filter.
– `--export-table-filter <REG-EX>` – Export (write) tables only if they matches the regEx filter.
– `--no-xdata` – Do not decode xdata frames.

- Columns in data tables
  – `--export-col-filter <REG-EX>` – Export (write) columns only if they matches the regEx filter.

**Note:** Please find several examples which are related to option `--` in following sections 4.2, 4.3, 4.4, 4.5, 4.8, 4.9.

### 3.3 Option checksum (-k, --checksum)

Compute checksums for all given file(s). CRC-32 is used if option `--checksum-type` is not specified.

- `--checksum-type <TYPE>` – The type of checksum, digest or hash to calculate. Possible types are CRC-32, MD2, MD5, SHA-1, SHA-224, SHA-256, SHA-384, SHA-512.

The result can be adjusted using following option:

- `--json-result` – Get result on standard output (STDOUT) as JSON. (see section 3.9)

### 3.4 Option test (-t, --test)

Test all given file(s) for correctness. Under this option, the file (or files) are opened and read according to the file type. If the test does not detect any problems, then the result is **OK**, otherwise **CORRUPT**.

There is also the possibility to test the internal structure in more detail. This is currently only possible for the file type **GNC-DATA** (see 2.3.5). This requires an additional file description, which is tested against. Such a file description is a special XML file (see an example in appendix D).

- `--test-description <FILE>` – The file description to use during test.

Currently the following file descriptions are included in the tool:

- "Rs11Gv1GdpFileDescription.xml" – to check a RS-11G-GDP.1 file
- "Rs41v1EdtFileDescription.xml" – to check a RS41-EDT.1 file
- "Rs92v1EdtFileDescription.xml" – to check a RS92-EDT.1 file
- "Rs92v2GdpFileDescription.xml" – to check a RS92-GDP.2 file

The result can be adjusted using the following option:
• **--json-result** – Get result on standard output (STDOUT) as JSON. (see section 3.9)

### 3.5 Option distribute (-d, --distribute)

Prepare all given file(s) for distribution. Currently only an **MWX** can be subject to this option. The preparation of a MWX file for external distribution consists of the removal of some internal XML files.

### 3.6 Option identify (-y, --identify)

The tool **gt92** can identify supported file formats.

```
1  gt92 -y
2  # result lines, e.g.
3  d:\Temp\LIN-Test\Lindenberg_20170321_104624.dc3db = DC3DB
4  d:\Temp\LIN-Test\Lindenberg_20170321_104624.nc = GNC-RAW_0.3
```

The result can be adjusted using following option:

• **--json-result** – Get result on standard output (STDOUT) as JSON. (see section 3.9)

### 3.7 Option itemize (-z, --itemize)

The tool **gt92** can itemize all tables of given file(s).

```
1  gt92 -z -f DC3DB
2  # result line, e.g.
3  EDT,STD,FLEDT,FLSTD,FRAWPTU,GPS_ORB,GPSCCLOC,GPSCCREM,GPSDCC_RESULT,RAWPTUPOS,
   RS92SONDEID,RSSTATUS,SENS9FRQ
```

The result can be adjusted using following option:

• **--json-result** – Get result on standard output (STDOUT) as JSON. (see section 3.9)

### 3.8 Option info (-i, --info)

The tool provides an interface to get information about the tool itself, e.g. version and change history. Using option **--info <KEY>**, specific information can be acquired.

Print a specific information (default is **VERSION**) and exit. The possibilities of **<KEY>** are:

- **VERSION** – full version text
- **VERSION_NUMBER** – version number
- **VERSION_DATE** – version date
- **NAME** – name of tool
- **NAME_SHORT** – short name of tool
- **NAME_LONG** – long name of tool
- **COPYRIGHT** – copyright paragraph of tool
• **AUTHOR** – full author text of tool
• **AUTHOR_NAME** – name of author
• **AUTHOR_EMAIL** – email address of author
• **FILE_TYPES** – list of implemented file types
• **HISTORY_LAST** – information about changes from last version
• **HISTORY** – list of change history
• **HISTORY_FULL** – full list of change history

The result can be adjusted using following options:

• **--json-result** – Get result on standard output (STDOUT) as JSON. (see section 3.9)
• **--no-info-key** – Do not print the key of an info in case of one requested info.

### 3.9 General options

Similar to other tools, the **-h,** **--help** option prints a full list of options with description texts (see appendix A).

Using the option **-v,** **--version** is the easily way to check which version a current gt92 installation has. It prints a general information about this program version, e.g.

```
$ gt92 -v
# Printed result to console starts with following line:
$ gt92 (GruanToolRS92) 0.4.11_03 (2019-04-02)
```

Sometimes it is easier if all answers or results on console would have a well defined structure like JSON. Using option **--json-result** activate such behaviour, so that all results will be printed on standard output (STDOUT) as JSON.

```
# without JSON result
$ gt92 -t
# Following lines are printed to console:
d:\Temp\LIN-Test\Lindenberg_20170321_104624.dc3db = OK
d:\Temp\LIN-Test\Lindenberg_20170321_104624.nc = OK

# with JSON result
$ gt92 -t --json-result
# Following lines are printed to console:
{
  "function" : "test",
  "timestamp" : "2019-04-12T19:12:18.985Z",
  "results" : [
    {
      "name" : "d:\Temp\LIN-Test\Lindenberg_20170321_104624.dc3db",
      "result" : "OK"
    },
    {
      "name" : "d:\Temp\LIN-Test\Lindenberg_20170321_104624.nc",
      "result" : "OK"
    }
  ]
}
```
4 Examples

4.1 Example 1 – Convert GNC-DATA files to LDF format

Convert all files of type GNC-DATA in directory “~/gruan-data” to the possible target format (in this case LDF). The result files are written in the local working directory.

```
gt92 -c -f GNC-DATA ~/gruan-data
```

4.2 Example 2 – Extract all data tables from DC3DB as TSV files

Extract all data-tables of DC3DB file “D:\Data\gruan-test\test1.dc3db” only as Vaisala TSV formatted text-files in local working directory.

```
gt92 -x --no-meta --no-csv --tsv -f DC3DB "D:\Data\gruan-test\test1.dc3db"
```

4.3 Example 3 – Extract meta-data only from GNC-RAW

Extract only the meta-data from all files of type GNC-RAW in both directories “/data/test1” and “/data/test2” and write these meta-data files in output directory “/data/meta”.

```
gt92 -x --no-tables -o "/data/meta" -f GNC-RAW "/data/test1" "/data/test2"
```

4.4 Example 4 – Extract a filtered part of meta-data from DC3DB

Extract only a filtered part of meta-data from all files of type DC3DB in current directory. Result meta-data file will only include parameter lines with adequate names.

```
gt92 -x --no-tables --meta-regex ".*(RsGroundCheck).*" -f DC3DB
```

4.5 Example 5 – Extract a filtered part of data tables from DC3DB

Extract only data tables which match given filter (start with “Gps”) from all files of type DC3DB in current directory.

```
# import (and extract) all tables stating with "Gps"
gt92 -x --no-meta --import-table-filter "Gps.*" -f DC3DB
```

4.6 Example 6 – Convert DC3DB files to GNC-RAW files

Convert all files of type DC3DB in local working directory to the possible target format (in this case GNC-RAW). The result files are written in local working directory.
4.7 Example 7 – Convert MWX files to GNC-RAW files

Convert all files of type \textit{MWX} in local working directory to the possible target format (in this case \textit{GNC-RAW}). The result files are written in local working directory.

\begin{verbatim}
gt92 -cf MWX
\end{verbatim}

4.8 Example 8 – Extract all data tables and meta-data from MWX files

Extract all data tables and meta-data from all files of type \textit{MWX} in local working directory. The result files are written in separate sub-directories ("\textit{<file\_name>_<extension>}") in local working directory.

\begin{verbatim}
gt92 -xsf MWX
\end{verbatim}

4.9 Example 9 – Extract and collect a filtered part of meta-data to store in one general file

Extract and filter meta-data of all input files. But store all collected meta-data in one general meta-data file instead of one meta-data file for each input file. That allows to extract and store meta-data of files from e.g. a time series.
gt92 -xf GNC-DATA \
--no-csv --no-tables --no-meta-root \
--meta-out-file="md-all-v1.txt" --no-meta-out-units \
--meta-regex "Ascent\.(BurstpointPressure|StartTime)" \
RS92-GDP.2/SOD/*

# following general meta-data file will be created:
md-all-v1.txt

# with this content:
"FileTitle","Ascent.BurstpointPressure","Ascent.StartTime"
"SOD-RS-02_2_RS92-GDP_002_20110116T113000_1-000-001","27710.5","2011-01-16T11:30:08.000Z"
"SOD-RS-02_2_RS92-GDP_002_20121123T110000_1-001-001","28890.0","2012-11-23T11:00:20.000Z"
"SOD-RS-02_2_RS92-GDP_002_20131104T105200_1-001-001","31448.0","2013-11-04T10:52:57.000Z"
# Appendix

## A Full list of options

The tool gt92 has a lot of options to help to customise regarding user requirements.

<table>
<thead>
<tr>
<th>List of options:</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-c, --convert</td>
<td>Convert all given file(s) to another file format. The possibilities are: DC3DB, MWX, GSFZ, STRATO-ZIP to GNC-RAW; GNC-DATA to LDF.</td>
</tr>
<tr>
<td>--checksum-type &lt;TYPE&gt;</td>
<td>The type of checksum, digest or hash to calculate. Possible types are CRC-32, MD2, MD5, SHA-1, SHA-224, SHA-256, SHA-384, SHA-512. If this option is not set, CRC-32 is used as default.</td>
</tr>
<tr>
<td>--copy-corrupt</td>
<td>Copy all corrupt source files which cannot be converted.</td>
</tr>
<tr>
<td>--corrupt-dir &lt;DIR&gt;</td>
<td>The directory to copy all corrupt source files.</td>
</tr>
<tr>
<td>--csv-delimiter &lt;DELIMITER&gt;</td>
<td>Use given delimiter within csv files.</td>
</tr>
<tr>
<td>-d, --distribute</td>
<td>Prepare all given file(s) for distribution.</td>
</tr>
<tr>
<td>--dump</td>
<td>Extract data tables in addition as binary dump files. Only possible with DC3DB files.</td>
</tr>
<tr>
<td>--export-col-filter &lt;REG-EX&gt;</td>
<td>Export (write) columns only if they match the regEx filter.</td>
</tr>
<tr>
<td>--export-table-filter &lt;REG-EX&gt;</td>
<td>Export (write) tables only if they match the regEx filter.</td>
</tr>
<tr>
<td>-f, --file-type &lt;FILE-TYPE&gt;</td>
<td>Read only file(s) with this specified file type. If this option is not set,</td>
</tr>
<tr>
<td></td>
<td>GNC-DATA is used as default. The possibilities are: DC3DB, GNC-RAW, GNC-DATA,</td>
</tr>
<tr>
<td></td>
<td>MWX, GSFZ, STRATO-ZIP, IGNC-RAW.</td>
</tr>
<tr>
<td>-h, --help</td>
<td>Print the help information and exit.</td>
</tr>
<tr>
<td>-i, --info &lt;KEY&gt;</td>
<td>Print a specific information (default is VERSION) and exit. The possibilities are: VERSION, VERSION_NUMBER, VERSION_DATE, NAME, NAME_SHORT, NAME_LONG, COPYRIGHT, AUTHOR, AUTHOR_NAME, AUTHOR_EMAIL, FILE_TYPES, HISTORY_LAST, HISTORY, HISTORY_FULL</td>
</tr>
<tr>
<td>--import-table-filter &lt;REG-EX&gt;</td>
<td>Import (read) tables only if they match the regEx filter.</td>
</tr>
<tr>
<td>--include-source</td>
<td>Include source file during converting a file in a NetCDF4 file (version 2) with compression. Only possible with DC3DB, MWX, GSFZ and STRATO-ZIP files.</td>
</tr>
<tr>
<td>--json-result</td>
<td>Get result on standard output (STDOUT) as JSON.</td>
</tr>
<tr>
<td>-k, --checksum</td>
<td>Compute checksums for all given file(s). CRC-32 is used if option --checksum-type is not specified.</td>
</tr>
<tr>
<td>-l, --logging &lt;LOG-LEVEL&gt;</td>
<td>Set the logging level to an other than</td>
</tr>
</tbody>
</table>
default (INFO). The possibilities are: SEVERE, WARNING, INFO, CONFIG, FINE, FINER, FINEST.

--meta-osep Extract meta-data items using original separator in names (not default one).

--meta-out-delimiter <DELIMITER> Extract meta-data using this delimiter (file name, meta-data name, meta-data value) for general output file.

--meta-out-file <FILE> Extract all meta-data of all files to this general meta-data output file.

--meta-regex <REG-EX> Extract meta-data only if it matches the regEx.

--meta-sep <SEPARATOR> Extract meta-data using specified separator for names (not default or original one).

--nc-old Convert a file in a NetCDF3 file without compression. Only possible with DC3DB files. This converts to the old version of GNC-RAW.

--no-csv Extract no data tables as default csv files.

--no-csv-nan Do not print NaN values in csv files.

--no-file-type-check Skip default file type check before converting or extracting. This option forces critical errors in case of wrong/corrupt files.

--no-info-key Do not print the key of an info in case of one requested info.

--no-meta Extract no meta-data.

--no-meta-group Extract no group items of meta-data.

--no-meta-name Extract only values of meta-data (not the names).

--no-meta-out-units Extract meta-data items without units (if available) to general output file.

--no-meta-root Extract meta-data items without root.

--no-meta-valid Extract meta-data items using original names (not valid one).

--no-msg-error Print no default error messages.

--no-msg-header Print no default header messages.

--no-msg-info Print no default information messages.

--no-silent Print messages also if silent mode is active.

--no-source Do not extract an included source file during extracting a file. Only possible with GNC-RAW files.

--no-tables Extract no data tables.

--no-xdata Do not decode xdata frames.

-o,--out-dir <DIR> The output directory for all exported and converted files. If not given the working directory is used as default.

--reverse-col-filter Reverse given column filter.

--reverse-table-filter Reverse given table filter.

-s,--separate Separate all extracted output files for every input file.

--separate-dir Separate all output files for every input directory.

--skip-convert Skip converting file if output file already exists.
B History of development of gt92

The gt92 is in constant development. This chapter briefly describes all software changes back to first functional version (v0.3).

B.1 History of Version 0.5.x

- 0.5.1_01 (2020-08-18) - change: improve MWX file definition for MW41 v2.15
- 0.5.1 (2020-08-04) - change: update file format GSFZ to GrawMet v5.15.02.02
- add handling of DFM-17 within GSFZ
- 0.5.0_05 (2020-04-29) - change: update file format IGNC-RAW to v1.2.0
- 0.5.0_04 (2020-04-28) - bug fixed: crash using MwxFile because not initied
- optimised and restructured 'DataFile' classes which allows easier adding of new config properties
- add option --no-meta-out-units
- 0.5.0_02 (2020-04-20) - add options --meta-out-file and --meta-out-delimiter
- # implemented for MWX, GNC-RAW (v1, v2), GNC-DATA, DC3DB, IGNC-RAW, STRATO-ZIP, GSFZ
- 0.5.0_01 (2019-12-11) - bug fixed: wrong calculation of OIF411 PumpTemperature
- 0.5.0 (2019-11-06) - new compiled only
- first release version 0.5

B.2 History of Version 0.4.x

- 0.4.11_08 (2019-10-23) - bug fixed: skip gridding part of DC3DB in case of --itemize
- change: extract SystemEvents as meta-data using last value in case of multiple events of same type
- 0.4.11_07 (2019-09-12) - bug fixed: crash during CSV export
- 0.4.11_06 (2019-08-28) - add options --export-col-filter and --reverse-col-filter and --csv-delimiter
* 0.4.11_05 (2019-07-31) - change: update MWX file definition for MW41 v2.15
* 0.4.11_04 (2019-05-07) - bug fixed: handling of wrong date format in GSFZ files
* 0.4.11_03 (2019-04-02) - bug fixed: add 'missing_value' to several columns of XDATA type 'SKYDEW_3F'
* 0.4.11_02 (2019-03-29) - bug fixed: repaired column names of XDATA type 'SKYDEW_3F'
- rename file type SGNC-RAW to IGNC-RAW
* 0.4.11_01 (2019-03-28) - bug fixed: decode abnormal package length 35 of XDATA type 'SKYDEW_3F'
* 0.4.11 (2019-03-28) - add handling of XDATA type 'SKYDEW_3F' (SKYDEW) packets
- compiled with Java 8
* 0.4.10_09 (2019-03-19) - add handling of file type SGNC-RAW
* 0.4.10_08 (2019-02-07) - better handling of maybe corrupt system-wide installed netcdf libraries (print messages)
- compiled with Java SE 11
* 0.4.10_07 (2018-12-17) - bug fixed: wrong XDATA match if both PCFH and CFH are flight together using one data stream (MWX file)
* 0.4.10_06 (2018-09-27) - change: update MWX file definition for MW41 v2.11
* 0.4.10_05 (2018-09-27) - change: SYNC_* columns added/activated for table EDT_XDATA_GRIDDED of file type MWX
* 0.4.10_04 (2018-07-24) - bug fixed: correct global attribute key 'g.Instrument.PCFH_MirNum' to 'g.Instrument.SerialNo' in case of handling of XDATA type 'PCFH_3C' (PCFH)
- bug fixed: add columns PCFH_Trsurf1, PCFH_Trsurf2, PCFH_2_Trsurf1, PCFH_2_Trsurf2 to table 'EDT_XDATA_GRIDDED' in case of handling of XDATA type 'PCFH_3C' (PCFH)
* 0.4.10_03 (2018-07-13) - change: handling of XDATA type 'PCFH_3C' (PCFH) packets 0x01 to 0x07, add TimestampXX columns
* 0.4.10_02 (2018-05-31) - add option --checksum-type
# with following possibilities: CRC-32, MD2, MD5, SHA-1, SHA-224, SHA-256, SHA-384, SHA-512
# with CRC-32 as default (like before)
* 0.4.10_01 (2018-05-31) - change: update MWX file definition for MW41 v2.9
* 0.4.10 (2018-05-19) - change: handling of XDATA type 'PCFH_3C' (PCFH) packets 0x00 to 0x07 (follow doc v1.2, 2018-04-05)
* 0.4.9_02 (2018-04-16) - bug fixed: handle 'NULL' value in column DataSrvTime, table AdditionalSensorData of MWX file
* 0.4.9_01 (2018-03-06) - add two hidden options for MWX
* 0.4.9 (2018-02-28) - add handling of XDATA type 'PCFH_3C' (PCFH) packets 0x00 to 0x07
* 0.4.8_03 (2018-01-05) - change: option --no-info-key is now also related to test messages (use of -t)
- change: option --no-info-key is now also related to identify messages (use of -y)
- change: option --no-info-key is now also related to checksum messages (use of -k)
- add option --no-source for combination with -x (extract) and file type GNC-RAW
* 0.4.8_02 (2017-12-08) - change: improved file check possibilities (value options added to attribute description)
- bug fixed: test messages with file path now (instead 'null')
* 0.4.8_01 (2017-11-01) - bug fixed: better error handling during identify DC3DB (in case of corrupt files)
<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>* 0.4.8</td>
<td>(2017-10-23)</td>
<td>- change: enable options --no-meta-root and --meta-sep for file type DC3DB</td>
</tr>
<tr>
<td>* 0.4.7_03</td>
<td>(2017-03-17)</td>
<td>- bug fixed: CFH optics temperature fixed</td>
</tr>
<tr>
<td>* 0.4.7_02</td>
<td>(2017-02-15)</td>
<td>- bug fixed: using attribute 'g_format_format' if type BYTE or SHORT during converting GNC-DATA to LDF (crash fixed)</td>
</tr>
<tr>
<td>* 0.4.7_01</td>
<td>(2016-11-24)</td>
<td>- bug fixed: decoding TINT and ILAMP of XDATA type 'FLASH_3D_0' (FLASH-B)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- add option --no-csv-nan for MWX, DC3DB, GNC-RAW, GNC-DATA</td>
</tr>
<tr>
<td>* 0.4.7</td>
<td>(2016-11-23)</td>
<td>- add handling of non-valid xml-chars (e.g. \x00;) for internal XML file 'AdditionalSensorData.xml' of MWX file</td>
</tr>
<tr>
<td>* 0.4.6_13</td>
<td>(2016-11-23)</td>
<td>- bug fixed: wrong frame length of XDATA type 'FLASH_3D_0' (FLASH-B)</td>
</tr>
<tr>
<td>* 0.4.6_12</td>
<td>(2016-11-22)</td>
<td>- change handling of XDATA type 'FLASH_3D_0' (FLASH-B)</td>
</tr>
<tr>
<td>* 0.4.6_11</td>
<td>(2016-11-16)</td>
<td>- add handling of XDATA type 'FLASH_3D_0' (FLASH-B)</td>
</tr>
<tr>
<td>* 0.4.6_10</td>
<td>(2016-11-16)</td>
<td>- add handling of XDATA type 'FLASH_3D' (FLASH-B)</td>
</tr>
<tr>
<td>* 0.4.6_09</td>
<td>(2016-11-10)</td>
<td>- bug fixed: handling of data type BYTE with file type MWX</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- change: update MWX file definition for MW41 v2.5</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- add option --no-info-key</td>
</tr>
<tr>
<td>* 0.4.6_08</td>
<td>(2016-11-09)</td>
<td>- bug fixed: handling of XDATA with file type STRATO-ZIP</td>
</tr>
<tr>
<td>* 0.4.6_07</td>
<td>(2016-11-08)</td>
<td>- bug fixed: wrong results with a list of INFO keys in combination with HISTORY, HISTORY_LAST, HISTORY_FULL</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- bug fixed: wrong results with -y in case of file types without version</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- change: include tool version in converted GNC-RAW files (gatts: history, g.Product.ProducerTool)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- add handling of XDATA type 'COVRAD_05_0110' (Cover of radiation chamber)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- change: improved automatical detection of number type for MWX columns (FLOAT, DOUBLE, LONG)</td>
</tr>
<tr>
<td>* 0.4.6_06</td>
<td>(2016-09-21)</td>
<td>- bug fixed: always add global attribute 'gAliases' during converting to GNC-RAW</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- add new INFO keys: HISTORY_LAST, HISTORY, HISTORY_FULL</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- bug fixed: convert STRATO-ZIP (SkySonde) to GNC-RAW</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- bug fixed: use correct data type INT for COBALD counters</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- add additional meta-data (time) for STRATO RAW decoding &amp; gidding</td>
</tr>
<tr>
<td>* 0.4.6_05</td>
<td>(2016-09-13)</td>
<td>- bug fixed: handling GSFZ files with extension *.zip</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- bug fixed: handling GNC-DATA files with missing ALIASES</td>
</tr>
<tr>
<td>* 0.4.6_04</td>
<td>(2016-09-12)</td>
<td>- bug fixed: handling several XData sensors of same type with MWX (CFH, CFH_2, ...)</td>
</tr>
<tr>
<td>* 0.4.6_03</td>
<td>(2016-09-07)</td>
<td>- bug fixed: handling GSFZ files</td>
</tr>
<tr>
<td>* 0.4.6_02</td>
<td>(2016-08-29)</td>
<td>- add BOM handling for internal XML files of MWX file</td>
</tr>
<tr>
<td>* 0.4.6_01</td>
<td>(2016-08-01)</td>
<td>- bug fixed: extract first 'correct' value with calculation</td>
</tr>
<tr>
<td>* 0.4.6</td>
<td>(2016-08-01)</td>
<td>- change options</td>
</tr>
<tr>
<td></td>
<td></td>
<td># add --nc-old (enable old version, default before)</td>
</tr>
<tr>
<td></td>
<td></td>
<td># remove --nc4, --nc4-v2 (--nc4-v2 is default now)</td>
</tr>
<tr>
<td></td>
<td></td>
<td># add --no-meta-valid, --meta-osep, --no-meta-root, --no-meta-group (use defaults for each file type)</td>
</tr>
<tr>
<td>* 0.4.5_11</td>
<td>(2016-07-29)</td>
<td>- add calculation to table columns (-store result as)</td>
</tr>
</tbody>
</table>
- change gridded table of MWX
  # add additional columns for GPS
  # remove 'meta-data' columns, e.g. CFH, OIF411
  # add calc to gridded table, e.g. CFH, OIF411
  (→ store result as global attributes)

* 0.4.5_10 (2016-07-26) - bug fixed: use XML version 1.1 for parsing internal
  MWX files

* 0.4.5_09 (2016-06-28) - bug fixed: use correct base time for all columns to
  grid (DC3DB and MWX)
  - add a gridded table to DC3DB

* 0.4.5_08 (2016-06-22) - bug fixed: use correct path to write LDF files
  - bug fixed: first try to handle too short formats for
    LDF converting (negative ascent speed)

* 0.4.5_07 (2016-06-10) - bug fixed: static reading of file descriptions of MWX,
  DC3DB, STRATO, GSFZ

* 0.4.5_06 (2016-05-23) - new compiled
  * 0.4.5_05 (2016-02-26) - add XData specific meta-data (MWX only)
    # g.Instrument.XData, g.Instrument.XDataSensors
  - rename XData instrument RSA411 to OIF411

* 0.4.5_04 (2016-02-25) - change some internal MWX things
  # e.g. add correct positions to a couple of tables
  - change base of gridded table of MWX to RawPtu

* 0.4.5_03 (2016-02-23) - change column names of XDATA columns of MWX
  # CfH_... to CFH_...
  # several identical instruments possible, like
  CFH_..., CFH_2_...
  - bug fixed: rename meta-data keys correctly to original
  once (if required)

* 0.4.5_02 (2016-02-23) - add functionality of ‘--json-result’ to option
  --info / -i
  - change print result (without --json-result) to
    <file/name> = <result>
  - add result line for option --extract / -x (EXTRACTED
    or NOT_EXTRACTED)

* 0.4.5_01 (2016-02-22) - bug fixed: data type changed from BYTE to SHORT of
  column ‘PumpCurrent’ of map ‘RawOzone’ of format ‘MWX’
  - bug fixed: unit ‘seconds since ...’ of column
    ‘RadioRxTimePk’ is now correct in case of converting
    MWX to GNC-RAW
  - add argument --export-table-filter
  - add argument --import-table-filter
  - add argument --reverse-table-filter
  - mark argument --table-filter as ‘deprecated’

* 0.4.5 (2016-02-18) - some changes at file descriptions (MWX, DC3DB,
  STRATO-ZIP)
  - with option -t (--test) a file can be tested against a
    file description
    # this is very helpful to test created GDP files
    (GRUAN data product NetCDF file)
  # all test result messages are printed like other
    results (at console ‘out’ without/with JSON format)

* 0.4.4_05 (2016-02-16) - add automatically extracting of relevant native
  libraries to sub-dir ‘gt92-native-libs’ of temp dir of
  system
  # in case of no rights to write in own program folder
* 0.4.4_04 (2016-02-16) - add argument --test-description=<FILE> (for -t)
  - bug fixed: use now argument --no-xdata (true/false) with --convert
* 0.4.4_03 (2016-02-03) - optimization of messages (reading MWX)
  - add new table 'StabilityIndex' to MWX file description
  - change start time for gridded MWX tables to Soundings.
    BeginTime or if not available Soundings.RadioResetTime
  - bug fixed: wrong name of one V7 value
* 0.4.4_02 (2016-02-02) - bug fixed: optimize gridded data map EDT_XDATA_GRIDDED
  of MWX file
  # set grid base to AdditionalSensors
  # add columns from RawPtu, GSupport,
    SynchronizedSoundingData
  # optimize messages if columns/maps are missing
* 0.4.4 (2016-01-13) - include native netcdf libraries
  # for os-arch: linux-x86, linux-x86-64, win32-x86, win32-x86-64
  # automatically extracting of relevant native
    libraries to sub-dir 'lib' (in case of bin to
    '../lib')
  - optimise netcdf logging (bind to log4j)
* 0.4.3 (2016-01-06) - add argument --itemize / -z
  - add argument --identify / -y
  - add argument --no-msg-header
  - add argument --no-silent (for -t, -z, -k, -y)
  - add argument --json-result
  - improved automatically silent mode for -t, -z, -k, -y
  - restructured code of processing routines
* 0.4.2 (2015-12-16) - add argument --corrupt-dir=<DIR>
  - add argument --copy-corrupt
  - add argument --no-msg-error
  - add argument --no-msg-info
* 0.4.1_01 (2015-12-15) - improve source documentation
  - bug fixed: remove tmp file in case of an error
  - add final info about corrupt files
* 0.4.1 (2015-12-14) - add argument --skip-convert
  - add argument -i <KEY> / --info=<KEY> with a couple of
    possible keys:
    # VERSION, VERSION_NUMBER, VERSION_DATE,
    # NAME, NAME_SHORT, NAME_LONG,
    # COPYRIGHT,
    # AUTHOR, AUTHOR_NAME, AUTHOR_EMAIL,
    # FILE_TYPES
  - new error message in case of missing required option
  - add argument -w <DIR> / --work-dir=<DIR>
  - add argument --no-file-type-check
* 0.4.0 (2015-12-11) - add argument --separate-dir
  - improve handling with DC3DB files (e.g. wrong SN,
    additional log info, ...)
  - use working directory to create NC files
B.3 History of Version 0.3.x

* 0.3.24_11 (2015-10-14) - bug fixed: read/use all metadata also with MWX source
* 0.3.24_10 (2015-10-07) - new compiled only (?)
* 0.3.24_09 (2015-09-30) - new compiled only
* 0.3.24_08 (2015-09-21) - new compiled only
* 0.3.24_07 (2015-09-11) - kleine Optimierung zu GSFZ
* 0.3.24_06 (2015-08-31) - kleine Optimierung GNC-RAW (v1+v2)
* 0.3.24_05 (2015-08-28) - NetCDF-Bibliotheken angepasst
* 0.3.24_04 (2015-08-25) - NetCDF-Bibliotheken angepasst
* 0.3.24_03 (2015-08-14) - add file converter for STRATO-ZIP to GNC-RAW (using GenericFileConverter)
* 0.3.24_02 (2015-07-20) - Datentyp STRATO-ZIP nun moeglich
  - test version 2
* 0.3.24_01 (2015-07-02) - test version 1
* 0.3.24 (2015-06-10) - add file converter for MWX to GNC-RAW (using GenericFileConverter)
* 0.3.23 (2015-06-09) - changed file converter for DC3DB to GNC-RAW v2 (now using GenericFileConverter)
* 0.3.22 (2015-06-05) - now with new NetCDF library v4.6
* 0.3.21_02 (2015-05-28) - Bug fixed?
* 0.3.21_01 (2015-05-21) - try option --nc4-v2
* 0.3.21 (2015-05-15) - add option --table-filter REG-EXT to filter data tables
* 0.3.20 (2015-05-08) - better handling of corrupt DC3DB files
* 0.3.19 (2015-05-07) - Bug fixed: crash if wrong xdata row
  - updated MWX file description
  - lesser error messages in case of undefined NaN values
* 0.3.18 (2015-04-24) - compiled with java-1.8.0-openjdk-1.8.0 (to JRE 1.7 features)
  - Add JEval library to jar
* 0.3.17 (2015-04-16) - Bug fixed: meta-data in converted LDF are 'null'
* 0.3.16 (2015-04-09) - MWX meta-data extraction includes full table 'SoundingParameters' now
* 0.3.15 (2015-03-06) - print MWX meta-data name without ROOT
* 0.3.14 (2015-01-13) - bug repaired
* 0.3.13 (2015-01-12) - add option --no-xdata
* 0.3.12 (2015-01-09) - MWX mit decodeXData=true
* 0.3.11 (2014-11-19) - Datentyp MWX nun moeglich (z.Z. nur bei -x)
* 0.3.10 (2014-01-24) - mit Moeglichkeit fr netCDF4
* 0.3.9 (2014-01-22) - mit neuer netCDF-Bibliothek v4.3
* 0.3.8 (2013-09-05) - besserer Umgang mit fehlendem Output-Verzeichnis
* 0.3.7 (2013-08-30) - einheitliche interne Config-Properties
  - Dc3DbFile kann nun mit RS80 umgehen
* 0.3.6 (2013-07-10) - Weniger Fehlermeldungen mit DC3DB
* 0.3.5 (2013-07-08) - Bug behoben, dass auch Binaer-Spalten aus den DAT-Tabellen (DC3DB) korrekt ausgelesen werden
* 0.3.4 (2012-10-02) - good release version
C Collection of tables describing XData columns

Definition of all additional data columns related to specific XData instrument are described in tables which are included in this chapter. These columns can appear as part of several tables:

- tables AdditionalSensorData and EDT_DATA_GRIDDED of file type MWX

Table 1: Definition of all additional data columns related to Xdata instrument V7, the Innovative System Designs V7 Ozone interface board (id=01). Column names follow naming ‘V7_<Name>’ (first instrument) or ‘V7_2_<Name>’ (second instrument).

<table>
<thead>
<tr>
<th>Name</th>
<th>Unit</th>
<th>Timeres</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OzoneCurrent</td>
<td>µA</td>
<td>1 s</td>
<td>Ozone current</td>
</tr>
<tr>
<td>PumpTemperature</td>
<td>°C</td>
<td>1 s</td>
<td>Temperature of pump</td>
</tr>
<tr>
<td>PumpCurrent</td>
<td>mA</td>
<td>1 s</td>
<td>Current of pump</td>
</tr>
<tr>
<td>PumpBatteryVoltage</td>
<td>V</td>
<td>1 s</td>
<td>Battery voltage of pump</td>
</tr>
</tbody>
</table>

Table 2: Definition of all additional data columns related to Xdata instrument OIF411, the Vaisala Ozone interface board (id=05). Column names follow naming ‘OIF411_<Name>’ (first instrument) or ‘OIF411_2_<Name>’ (second instrument).

<table>
<thead>
<tr>
<th>Name</th>
<th>Unit</th>
<th>Timeres</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PumpTemperature</td>
<td>°C</td>
<td>1 s</td>
<td>Temperature of pump</td>
</tr>
<tr>
<td>OzoneCurrent</td>
<td>µA</td>
<td>1 s</td>
<td>Ozone current</td>
</tr>
<tr>
<td>PumpBatteryVoltage</td>
<td>V</td>
<td>1 s</td>
<td>Battery voltage of pump</td>
</tr>
<tr>
<td>PumpCurrent</td>
<td>A</td>
<td>1 s</td>
<td>Current of pump</td>
</tr>
<tr>
<td>ExternalBatteryVoltage</td>
<td>V</td>
<td>1 s</td>
<td>External battery voltage</td>
</tr>
</tbody>
</table>

Parameter

<table>
<thead>
<tr>
<th>Name</th>
<th>Unit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SerialNumber</td>
<td>sporadic</td>
<td>Serial number of OIF411 interface card (in addition stored as meta-data ‘g.Instrument.OIF411.SerialNumber’ or ‘g.Instrument.OIF411_2.SerialNumber’)</td>
</tr>
<tr>
<td>OifVersion</td>
<td>sporadic</td>
<td>(in addition stored as meta-data ‘g.Instrument.OIF411.OifVersion’ or ‘g.Instrument.OIF411_2.OifVersion’)</td>
</tr>
</tbody>
</table>

This table is continued on the next page.
Table 2 – Continued from previous page

<table>
<thead>
<tr>
<th>Name</th>
<th>Unit</th>
<th>Timeres.</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>UnknownValue</td>
<td></td>
<td>sporadic</td>
<td>(in addition stored as meta-data ‘g.Instrument.OIF411.UnknownValue’ or ‘g.Instrument.OIF411_2.UnknownValue’)</td>
</tr>
<tr>
<td>Interface</td>
<td></td>
<td>sporadic</td>
<td>(in addition stored as meta-data ‘g.Instrument.OIF411.Interface’ or ‘g.Instrument.OIF411_2.Interface’)</td>
</tr>
</tbody>
</table>

Table 3: Definition of all additional data columns related to Xdata instrument CFH, the Cryogen Frostpoint Hygrometer (id=08). Column names follow naming ‘CFH_<Name>’ (first instrument) or ‘CFH_2_<Name>’ (second instrument).

<table>
<thead>
<tr>
<th>Name</th>
<th>Unit</th>
<th>Timeres.</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FrostPoint</td>
<td>°C</td>
<td>1 s</td>
<td>Frost point temperature (maybe better would be “MirrorTemperature”)</td>
</tr>
<tr>
<td>TuningVoltage</td>
<td>V</td>
<td>1 s</td>
<td>Tuning or alignment voltage of photodiodes</td>
</tr>
<tr>
<td>OpticTemperature</td>
<td>°C</td>
<td>1 s</td>
<td>Temperature of optics (will be changed in “OpticsTemperature” in future)</td>
</tr>
<tr>
<td>BatteryVoltage</td>
<td>V</td>
<td>1 s</td>
<td>Battery voltage</td>
</tr>
</tbody>
</table>

Parameter

<table>
<thead>
<tr>
<th>Name</th>
<th>Unit</th>
<th>Timeres.</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cfg1Key</td>
<td></td>
<td>sporadic</td>
<td>Parameter key 1 (possible keys are: PicVersion, CalD, CalC, CalB, CalA, CfhSerialNumber)</td>
</tr>
<tr>
<td>Cfg1Value</td>
<td></td>
<td>sporadic</td>
<td>Parameter value 1 (in addition stored as meta-data ‘g.Instrument.CFH.&lt;Cfg1Key&gt;’ or ‘g.Instrument.CFH_2.&lt;Cfg1Key&gt;’)</td>
</tr>
<tr>
<td>Cfg2Key</td>
<td></td>
<td>sporadic</td>
<td>Parameter key 2 (possible keys are: SetpointTemperature, DetectorSignal, AD630Offset, Specular, CodeVersion)</td>
</tr>
<tr>
<td>Cfg2Value</td>
<td></td>
<td>sporadic</td>
<td>Parameter value 2 (in addition stored as meta-data ‘g.Instrument.CFH.&lt;Cfg2Key&gt;’ or ‘g.Instrument.CFH_2.&lt;Cfg2Key&gt;’)</td>
</tr>
</tbody>
</table>
Table 4: Definition of all additional data columns related to Xdata instrument FPH, the Frost Point Hygrometer (id=10). Column names follow naming ‘FPH.<Name>’ (first instrument) or ‘FPH_2..<Name>’ (second instrument).

<table>
<thead>
<tr>
<th>Name</th>
<th>Unit</th>
<th>Timeres.</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frst</td>
<td></td>
<td>1 s</td>
<td>Frost coverage of the chilled mirror</td>
</tr>
<tr>
<td>FiltFrst</td>
<td></td>
<td>1 s</td>
<td>Frost coverage after sunlight removal filtering is applied</td>
</tr>
<tr>
<td>Sun</td>
<td></td>
<td>1 s</td>
<td>The ambient sunlight measured in the sensor housing</td>
</tr>
<tr>
<td>LowSun</td>
<td></td>
<td>1 s</td>
<td>Lowest sunlight measured over a short timespan</td>
</tr>
<tr>
<td>FptAdc</td>
<td></td>
<td>1 s</td>
<td>ADC measurement of the frostpoint thermistor</td>
</tr>
<tr>
<td>Opt</td>
<td></td>
<td>1 s</td>
<td>Optics block temperature, controlled to a stable level above room temperature at surface</td>
</tr>
<tr>
<td>OptHeat</td>
<td></td>
<td>1 s</td>
<td>PWM value used to heat the optics block</td>
</tr>
<tr>
<td>MirrHeat</td>
<td></td>
<td>1 s</td>
<td>PWM value used to heat the chilled mirror</td>
</tr>
<tr>
<td>Pres</td>
<td>mbar</td>
<td>1 s</td>
<td>Pressure</td>
</tr>
<tr>
<td>PresTemp</td>
<td>°C</td>
<td>1 s</td>
<td>Pressure sensor temperature</td>
</tr>
<tr>
<td>AvgFpt</td>
<td>°C</td>
<td>1 s</td>
<td>A long exponential moving average of the frostpoint temperature calculated onboard the hygrometer</td>
</tr>
<tr>
<td>VBat</td>
<td>V</td>
<td>1 s</td>
<td>Battery voltage</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Parameter</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>MirNum</td>
<td>sporadic</td>
<td>Serial number of mirror (in addition stored as meta-data ‘g.Instrument.FPH.MirNum’ or ‘g.Instrument.FPH_2.MirNum’)</td>
</tr>
<tr>
<td>Cal0</td>
<td>sporadic</td>
<td>The thermistor resistance at 0 °C (in addition stored as meta-data ‘g.Instrument.FPH.Cal0’ or ‘g.Instrument.FPH_2.Cal0’)</td>
</tr>
</tbody>
</table>

*This table is continued on the next page.*
### Table 4 – Continued from previous page

<table>
<thead>
<tr>
<th>Name</th>
<th>Unit</th>
<th>Timeres.</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cal45</td>
<td>sporadic</td>
<td></td>
<td>The thermistor resistance at $-45^\circ C$ (in addition stored as metadata ‘g.Instrument.FPH.Cal45’ or ‘g.Instrument.FPH_2.Cal45’)</td>
</tr>
<tr>
<td>Cal79</td>
<td>sporadic</td>
<td></td>
<td>The thermistor resistance at $-79^\circ C$ (in addition stored as metadata ‘g.Instrument.FPH.Cal79’ or ‘g.Instrument.FPH_2.Cal79’)</td>
</tr>
<tr>
<td>HygFirm</td>
<td>sporadic</td>
<td></td>
<td>Version of firmware (in addition stored as metadata ‘g.Instrument.FPH.HygFirm’ or ‘g.Instrument.FPH_2.HygFirm’)</td>
</tr>
</tbody>
</table>

### Table 5: Definition of all additional data columns related to Xdata instrument COBALD, the Compact Optical Backscatter and Aerosol Detector (id=18). Column names follow naming ‘COBALD.<Name>’ (first instrument) or ‘COBALD_2.<Name>’ (second instrument).

<table>
<thead>
<tr>
<th>Name</th>
<th>Unit</th>
<th>Timeres.</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SN</td>
<td></td>
<td>1 s</td>
<td>Serial number of COBALD instrument (in addition stored as metadata ‘g.Instrument.COBALD.SN’ or ‘g.Instrument.COBALD_2.SN’)</td>
</tr>
<tr>
<td>InternalTemperature</td>
<td>°C</td>
<td>1 s</td>
<td>Internal temperature of instrument</td>
</tr>
<tr>
<td>SignalRed</td>
<td></td>
<td>1 s</td>
<td>Signal of channel red</td>
</tr>
<tr>
<td>SignalBlue</td>
<td></td>
<td>1 s</td>
<td>Signal of channel blue</td>
</tr>
<tr>
<td>MonitorRed</td>
<td></td>
<td>1 s</td>
<td>Monitor of channel red</td>
</tr>
<tr>
<td>MonitorBlue</td>
<td></td>
<td>1 s</td>
<td>Monitor of channel blue</td>
</tr>
<tr>
<td>IntegrationTime</td>
<td></td>
<td></td>
<td><em>(old)</em> Integration time</td>
</tr>
<tr>
<td>Counter</td>
<td></td>
<td></td>
<td><em>(old)</em> Counter</td>
</tr>
</tbody>
</table>
Table 6: Definition of all additional data columns related to Xdata instrument PCFH, the (id=). Column names follow naming ‘PCFH_<Name>’ (first instrument) or ‘PCFH_2_<Name>’ (second instrument).

<table>
<thead>
<tr>
<th>Name</th>
<th>Unit</th>
<th>Timeres.</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>PCFH package 00 (configuration)</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SerialNo</td>
<td></td>
<td></td>
<td>Serial number of PCFH</td>
</tr>
<tr>
<td>Hardware</td>
<td></td>
<td></td>
<td>Version of hardware (month &amp; year)</td>
</tr>
<tr>
<td>TempPCB</td>
<td></td>
<td></td>
<td>Version of “temp” PCB (month &amp; year)</td>
</tr>
<tr>
<td>ContrPCB</td>
<td></td>
<td></td>
<td>Version of “contr” PCB (month &amp; year)</td>
</tr>
<tr>
<td>ReflexPCB</td>
<td></td>
<td></td>
<td>Version of “reflex” PCB (month &amp; year)</td>
</tr>
<tr>
<td>ContrFW</td>
<td></td>
<td></td>
<td>Version of “contr” firmware (month &amp; year)</td>
</tr>
<tr>
<td>ReflexFW</td>
<td></td>
<td></td>
<td>Version of “reflex” firmware (month &amp; year)</td>
</tr>
<tr>
<td><strong>PCFH package 01 (sub-sensor 1)</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Timestamp01</td>
<td></td>
<td>1 s</td>
<td>Timestamp (0..249), incremented at a rate of 10 Hz</td>
</tr>
<tr>
<td>Tmirr1</td>
<td>°C</td>
<td>1 s</td>
<td>Sub-sensor 1 frost point mirror temperature</td>
</tr>
<tr>
<td>Thot1</td>
<td>°C</td>
<td>1 s</td>
<td>Sub-sensor 1 Peltier hot side temperature</td>
</tr>
<tr>
<td>Tair1</td>
<td>°C</td>
<td>1 s</td>
<td>Sub-sensor 1 air temperature</td>
</tr>
<tr>
<td>Ttarg1</td>
<td>°C</td>
<td>1 s</td>
<td>Sub-sensor 1 anticipated frost point mirror temperature</td>
</tr>
<tr>
<td>Rmirr1</td>
<td>% FS</td>
<td>1 s</td>
<td>Sub-sensor 1 frost point mirror reflectance</td>
</tr>
<tr>
<td>Rrefr1</td>
<td>% FS</td>
<td>1 s</td>
<td>Sub-sensor 1 reference surface reflectance</td>
</tr>
<tr>
<td>Iheat1</td>
<td>% FS</td>
<td>1 s</td>
<td>Sub-sensor 1 warm side heating current</td>
</tr>
<tr>
<td>IPelt1</td>
<td>% FS</td>
<td>1 s</td>
<td>Sub-sensor 1 Peltier current</td>
</tr>
<tr>
<td><strong>PCFH package 02 (sub-sensor 2)</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Timestamp02</td>
<td></td>
<td>1 s</td>
<td>Timestamp (0..249), incremented at a rate of 10 Hz</td>
</tr>
</tbody>
</table>

This table is continued on the next page.
Table 6 – *Continued from previous page*

<table>
<thead>
<tr>
<th>Name</th>
<th>Unit</th>
<th>Timeres.</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tmirr2</td>
<td>°C</td>
<td>1 s</td>
<td>Sub-sensor 2 frost point mirror temperature</td>
</tr>
<tr>
<td>Thot2</td>
<td>°C</td>
<td>1 s</td>
<td>Sub-sensor 2 Peltier hot side temperature</td>
</tr>
<tr>
<td>Tair2</td>
<td>°C</td>
<td>1 s</td>
<td>Sub-sensor 2 air temperature</td>
</tr>
<tr>
<td>Ttarg2</td>
<td>°C</td>
<td>1 s</td>
<td>Sub-sensor 2 anticipated frost point mirror temperature</td>
</tr>
<tr>
<td>Rmirr2</td>
<td>% FS</td>
<td>1 s</td>
<td>Sub-sensor 2 frost point mirror reflectance</td>
</tr>
<tr>
<td>Refr2</td>
<td>% FS</td>
<td>1 s</td>
<td>Sub-sensor 2 reference surface reflectance</td>
</tr>
<tr>
<td>Iheat2</td>
<td>% FS</td>
<td>1 s</td>
<td>Sub-sensor 2 warm side heating current</td>
</tr>
<tr>
<td>IPelt2</td>
<td>% FS</td>
<td>1 s</td>
<td>Sub-sensor 2 Peltier current</td>
</tr>
</tbody>
</table>

*PCFH package 03 (slow sensor and housekeeping)*

| Timestamp03  |         | 5 s      | Timestamp (0..249), incremented at a rate of 10 Hz                          |
| Tsink1       | °C      | 5 s      | Sub-sensor 1 heat sink temperature                                          |
| Trsurf1      | °C      | 5 s      | Sub-sensor 1 reference surface temperature                                  |
| Tsink2       | °C      | 5 s      | Sub-sensor 2 heat sink temperature                                          |
| Trsurf2      | °C      | 5 s      | Sub-sensor 2 reference surface temperature                                  |
| Tref         | °C      | 5 s      | Thermocouple reference temperature                                          |
| Tres         | °C      | 5 s      | Reserved temperature                                                        |
| Ubat60       | V       | 5 s      | 6V supply battery voltage                                                    |
| Ubat45       | V       | 5 s      | 4.5V supply battery voltage                                                 |

*PCFH package 04 (experimental 1)*

| Timestamp04  |         |         | Timestamp (0..249), incremented at a rate of 10 Hz                          |
| Exp01        |         |         | Experimental test data value 1. The meaning of these placeholders is defined according to experimental requirements. |
| Exp02        |         |         | Experimental test data value 2                                              |

This table is continued on the next page.
Table 6 – Continued from previous page

<table>
<thead>
<tr>
<th>Name</th>
<th>Unit</th>
<th>Timeres.</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exp03</td>
<td></td>
<td></td>
<td>Experimental test data value 3</td>
</tr>
<tr>
<td>Exp04</td>
<td></td>
<td></td>
<td>Experimental test data value 4</td>
</tr>
<tr>
<td>Exp05</td>
<td></td>
<td></td>
<td>Experimental test data value 5</td>
</tr>
<tr>
<td>Exp06</td>
<td></td>
<td></td>
<td>Experimental test data value 6</td>
</tr>
</tbody>
</table>

**PCFH package 05 (experimental 2)**

<table>
<thead>
<tr>
<th>Name</th>
<th>Unit</th>
<th>Timeres.</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Timestamp05</td>
<td></td>
<td></td>
<td>Timestamp (0..249), incremented at a rate of 10 Hz</td>
</tr>
<tr>
<td>Exp07</td>
<td></td>
<td></td>
<td>Experimental test data value 7</td>
</tr>
<tr>
<td>Exp08</td>
<td></td>
<td></td>
<td>Experimental test data value 8</td>
</tr>
<tr>
<td>Exp09</td>
<td></td>
<td></td>
<td>Experimental test data value 9</td>
</tr>
<tr>
<td>Exp10</td>
<td></td>
<td></td>
<td>Experimental test data value 10</td>
</tr>
<tr>
<td>Exp11</td>
<td></td>
<td></td>
<td>Experimental test data value 11</td>
</tr>
<tr>
<td>Exp12</td>
<td></td>
<td></td>
<td>Experimental test data value 12</td>
</tr>
</tbody>
</table>

**PCFH package 06 (experimental 3)**

<table>
<thead>
<tr>
<th>Name</th>
<th>Unit</th>
<th>Timeres.</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Timestamp06</td>
<td></td>
<td></td>
<td>Timestamp (0..249), incremented at a rate of 10 Hz</td>
</tr>
<tr>
<td>Exp13</td>
<td></td>
<td></td>
<td>Experimental test data value 13</td>
</tr>
<tr>
<td>Exp14</td>
<td></td>
<td></td>
<td>Experimental test data value 14</td>
</tr>
<tr>
<td>Exp15</td>
<td></td>
<td></td>
<td>Experimental test data value 15</td>
</tr>
<tr>
<td>Exp16</td>
<td></td>
<td></td>
<td>Experimental test data value 16</td>
</tr>
<tr>
<td>Exp17</td>
<td></td>
<td></td>
<td>Experimental test data value 17</td>
</tr>
<tr>
<td>Exp18</td>
<td></td>
<td></td>
<td>Experimental test data value 18</td>
</tr>
</tbody>
</table>

**PCFH package 07 (experimental 4)**

<table>
<thead>
<tr>
<th>Name</th>
<th>Unit</th>
<th>Timeres.</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Timestamp07</td>
<td></td>
<td></td>
<td>Timestamp (0..249), incremented at a rate of 10 Hz</td>
</tr>
<tr>
<td>Exp19</td>
<td></td>
<td></td>
<td>Experimental test data value 19</td>
</tr>
<tr>
<td>Exp20</td>
<td></td>
<td></td>
<td>Experimental test data value 20</td>
</tr>
<tr>
<td>Exp21</td>
<td></td>
<td></td>
<td>Experimental test data value 21</td>
</tr>
<tr>
<td>Exp22</td>
<td></td>
<td></td>
<td>Experimental test data value 22</td>
</tr>
</tbody>
</table>

*This table is continued on the next page.*
### Table 6 – Continued from previous page

<table>
<thead>
<tr>
<th>Name</th>
<th>Unit</th>
<th>Timeres.</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exp23</td>
<td></td>
<td></td>
<td>Experimental test data value 23</td>
</tr>
<tr>
<td>Exp24</td>
<td></td>
<td></td>
<td>Experimental test data value 24</td>
</tr>
</tbody>
</table>

Table 7: Definition of all additional data columns related to Xdata instrument FLASH-B, the Fluorescence Lyman-Alpha Stratospheric Hygrometer for Balloon (id=3D). Column names follow naming ‘FLASH_<Name>’ (first instrument) or ‘FLASH_2_<Name>’ (second instrument).

<table>
<thead>
<tr>
<th>Name</th>
<th>Unit</th>
<th>Timeres.</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VER</td>
<td></td>
<td>1 s</td>
<td>Version of xdata frame definition</td>
</tr>
<tr>
<td>SIG</td>
<td></td>
<td>1 s</td>
<td>Photomultiplier counts</td>
</tr>
<tr>
<td>BKG</td>
<td></td>
<td>1 s</td>
<td>Photomultiplier background counts</td>
</tr>
<tr>
<td>TINT</td>
<td>°C</td>
<td>1 s</td>
<td>Temperatur of photomultiplier</td>
</tr>
<tr>
<td>UBAT</td>
<td>V</td>
<td>1 s</td>
<td>Battery voltage</td>
</tr>
<tr>
<td>ILAMP</td>
<td>mA</td>
<td>1 s</td>
<td>Current of VUV lamp</td>
</tr>
<tr>
<td>UPMT</td>
<td>V</td>
<td>1 s</td>
<td>PMT voltage</td>
</tr>
<tr>
<td>SN</td>
<td></td>
<td>1 s</td>
<td>Individual instrument (serial) number</td>
</tr>
<tr>
<td>ULAMP</td>
<td></td>
<td></td>
<td>(old)</td>
</tr>
<tr>
<td>AUX1</td>
<td></td>
<td></td>
<td>(old)</td>
</tr>
<tr>
<td>AUX2</td>
<td></td>
<td></td>
<td>(old)</td>
</tr>
</tbody>
</table>

Table 8: Definition of all additional data columns related to Xdata instrument SKYDEW, the Peltier-based chilled-mirror hygrometer “SKYDEW” (id=3F). Column names follow naming ‘SKYDEW_<Name>’ (first instrument) or ‘SKYDEW_2_<Name>’ (second instrument).

<table>
<thead>
<tr>
<th>Name</th>
<th>Unit</th>
<th>Timeres.</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Raw data</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PTAD</td>
<td></td>
<td>1 s</td>
<td>AD value for calculating a mirror temperature</td>
</tr>
<tr>
<td>DewAD</td>
<td></td>
<td>1 s</td>
<td>AD value for calculating scattered light level</td>
</tr>
</tbody>
</table>

*This table is continued on the next page.*
### Table 8 – Continued from previous page

<table>
<thead>
<tr>
<th>Name</th>
<th>Unit</th>
<th>Timeres.</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RefAD</td>
<td></td>
<td>1 s</td>
<td>Reference resistance, which is used for calculating a mirror temperature</td>
</tr>
<tr>
<td>BaseAD</td>
<td></td>
<td>1 s</td>
<td>Offset value, which is used for calculating a mirror temperature</td>
</tr>
<tr>
<td>PeltierAD</td>
<td></td>
<td>1 s</td>
<td>AD value for calculating Peltier current</td>
</tr>
<tr>
<td>HSAD</td>
<td></td>
<td>1 s</td>
<td>AD value for calculating heatsink temperature</td>
</tr>
<tr>
<td>CBAD</td>
<td></td>
<td>1 s</td>
<td>AD value for calculating Circuit board temperature</td>
</tr>
<tr>
<td>BattAD</td>
<td></td>
<td>1 s</td>
<td>AD value for calculating Battery voltage</td>
</tr>
<tr>
<td>PID</td>
<td></td>
<td>1 s</td>
<td>PID (unknown)</td>
</tr>
<tr>
<td>Coeff</td>
<td></td>
<td>1 s</td>
<td>Current parameter (0 to 9)</td>
</tr>
<tr>
<td>Cnt</td>
<td></td>
<td>1 s</td>
<td>Counter of parameter (0 to 9)</td>
</tr>
</tbody>
</table>

**Calculated values**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Unit</th>
<th>Timeres.</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Temp</td>
<td>°C</td>
<td>1 s</td>
<td>Mirror temperature</td>
</tr>
<tr>
<td>Dew</td>
<td>V</td>
<td>1 s</td>
<td>Detector signal (scattered light level)</td>
</tr>
<tr>
<td>Peltier</td>
<td>A</td>
<td>1 s</td>
<td>Peltier current</td>
</tr>
<tr>
<td>Batt</td>
<td>V</td>
<td>1 s</td>
<td>Battery voltage</td>
</tr>
<tr>
<td>HS</td>
<td>°C</td>
<td>1 s</td>
<td>Heatsink temperature (optional)</td>
</tr>
<tr>
<td>CB</td>
<td>°C</td>
<td>1 s</td>
<td>Circuit board temperature (optional)</td>
</tr>
</tbody>
</table>

**Parameter**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SN</td>
<td>sporadic Serial number</td>
</tr>
<tr>
<td>FwVer</td>
<td>sporadic Firmware version</td>
</tr>
<tr>
<td>CoeffB</td>
<td>sporadic Coefficient B</td>
</tr>
<tr>
<td>CoeffC</td>
<td>sporadic Coefficient C</td>
</tr>
<tr>
<td>CoeffD</td>
<td>sporadic Coefficient D</td>
</tr>
<tr>
<td>CoeffE</td>
<td>sporadic Coefficient E</td>
</tr>
</tbody>
</table>
D Example of file description

Here an example of a file description of **GNC-DATA** file of data product “RS92-GDP.2” to use during test is given. Presented are some parts of these file to demonstrate the structure and possibilities:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<fileDescription name="GNC-DATA" createdOn="2016-01-26T15:10:00Z" version="2.0"
createdBy="Michael Sommer, GRUAN Lead Centre, DWD"
description="GRUAN Data Product NetCDF of RS92-GDP.2"
defaultDataMap="DEFAULT" defaultBaseCol="time" defaultBaseColContentType="TIME"
defaultBaseColCfStdName="time" defaultBaseColCfUnits="seconds since %1$s"
defaultBaseColCfAxis="T">

<!-- global CF attributes -->
<attribute name="Conventions" dataType="STRING" cf="true" required="true"
global="true" value="CF-1.4" />
<attribute name="title" dataType="STRING" cf="true" required="true"
global="true" value="RS92 GRUAN Data Product (Version 2)" />

<!-- global GRUAN attributes -->
<!-- part Product -->
<attribute name="g.Product.ID" dataType="STRING" required="true"
gruan="true" global="true" />
<attribute name="g.Product.Code" dataType="STRING" required="true"
gruan="true" global="true" />
<attribute name="g.Product.Version" dataType="STRING" required="true"
gruan="true" global="true" />

<!-- part MeasuringSystem -->
<attribute name="g.MeasuringSystem.ID" dataType="STRING" required="true"
gruan="true" global="true" />
<attribute name="g.MeasuringSystem.Altitude" dataType="STRING" required="true"
gruan="true" global="true" />

<!-- part Ascent -->
<attribute name="g.Ascent.ID" dataType="STRING" required="true"
gruan="true" global="true" />
<attribute name="g.Ascent.StandardTime" dataType="STRING" required="true"
gruan="true" global="true" />

<!-- part Instrument -->
<attribute name="g.Instrument.SerialNumber" dataType="STRING" required="true"
gruan="true" global="true" />
<attribute name="g.Instrument.Type" dataType="STRING" required="true"
gruan="true" global="true" />

<!-- column CF attributes -->
<attribute name="standard_name" dataType="STRING" required="true" cf="true"
column="true" />
<attribute name="units" dataType="STRING" required="true" cf="true"
column="true" />
<attribute name="long_name" dataType="STRING" required="true" cf="true"
column="true" />
<attribute name="comment" dataType="STRING" required="true" cf="true"
column="true" columnExp="(?!time).+" />
<attribute name="related_columns" dataType="STRING" required="true" cf="true"
column="true" columnExp="(?!time).+" />
<attribute name="coordinates" dataType="STRING" required="true" cf="true"
```
<attribute name="g_column_type" dataType="STRING" required="true"
gruan="true" column="true" />
<attribute name="g_processing_flag" dataType="STRING" required="true"
gruan="true" column="true" columnExp="(?!time).*" >
  <valueOption value="raw" />
  <valueOption value="raw, smoothed" />
  <valueOption value="raw, smoothed, internal QC passed" />
  <valueOption value="raw, internal QC passed" />
  <valueOption value="raw, internal QC passed, additional QC passed" />
  <valueOption value="uncertainty calculated, smoothed" />
  <valueOption value="smoothed" />
</attribute>

<!-- only data map DEFAULT -->
dataMap name="DEFAULT" lastChange="2016-01-26T15:11:00Z" data="true"
description="" baseCol="time">
  <!-- time --><-- contentType="TIME" -->
  <column name="time" dataType="DOUBLE" unit="s"
    cfStdName="time" cfLongName="Time" cfUnits="seconds since %1$s"
    cfAxis="T" cfCalendar="gregorian"
    gFmtType="FLT" gFmtWidth="8" gFmtFormat="F8.1" gFmtNaN="NaN"
    gSrcDesc="FRAWPTU" gResolution="1.0 s (time)" gColType="original data"
    testValues="true" tvMin="0.0" tvMax="10000.0" tvLimitNaN="0"
    tvLimitZero="1" />
  </column>

  <!-- rh -->
  <column name="rh" dataType="DOUBLE" unit="-"
    cfStdName="relative_humidity" cfLongName="Relative Humidity" cfUnits="1"
    cfRelCols="u_std_rh u_cor_rh u_rh " cfCoords="lon lat alt"
    gFmtType="FLT" gFmtWidth="6" gFmtFormat="F6.2" gFmtNaN="NaN"
    gSrcDesc="FRAWPTU_U1-U2_cc_RC_TL" gResolution="see column res_rh"
    gColType="original data" testValues="true" tvMin="-0.050" tvMax="1.250"
    tvLimitNaN="100" tvLimitZero="10" >
    <!-- CF attributes -->
    <attribute name="comment" value="Relative humidity collated from U1 and U2
     based on the water vapor pressure formula of HylandWexler, corrected by GRUAN
     correction scheme" />
  </column>

  <!-- cor_rh -->
  <column name="cor_rh" dataType="DOUBLE" unit="-"
    cfStdName="relative_humidity correction" cfUnits="1"
    cfLongName="Correction of relative_humidity" cfCoords="lon lat alt"
    gFmtType="FLT" gFmtWidth="6" gFmtFormat="F6.2" gFmtNaN="NaN"
    gSrcDesc="FRAWPTU_U1-U2_cc_RC_TL" gColType="correction applied to the data"
    testValues="true" tvMin="-0.5" tvMax="0.5" tvLimitNaN="100"
    tvLimitZero="10" >
    <!-- CF attributes -->
    <attribute name="comment" value="Bias corrections applied to
     relative_humidity by the GRUAN correction scheme" />
  </column>
</dataMap>

<!-- column GRUAN attributes -->
<attribute name="g_column_type" dataType="STRING" required="true"
grau="true" column="true" />'
<attribute name="g_processing_flag" dataType="STRING" required="true"
grau="true" column="true" columnExp="(?!time).*" >
  <valueOption value="raw" />
  <valueOption value="raw, smoothed" />
  <valueOption value="raw, smoothed, internal QC passed" />
  <valueOption value="raw, internal QC passed" />
  <valueOption value="raw, internal QC passed, additional QC passed" />
  <valueOption value="uncertainty calculated, smoothed" />
  <valueOption value="smoothed" />
</attribute>

<!-- only data map DEFAULT -->
dataMap name="DEFAULT" lastChange="2016-01-26T15:11:00Z" data="true"
description="" baseCol="time">
  <!-- time --><-- contentType="TIME" -->
  <column name="time" dataType="DOUBLE" unit="s"
    cfStdName="time" cfLongName="Time" cfUnits="seconds since %1$s"
    cfAxis="T" cfCalendar="gregorian"
    gFmtType="FLT" gFmtWidth="8" gFmtFormat="F8.1" gFmtNaN="NaN"
    gSrcDesc="FRAWPTU" gResolution="1.0 s (time)" gColType="original data"
    testValues="true" tvMin="0.0" tvMax="10000.0" tvLimitNaN="0"
    tvLimitZero="1" />
  </column>

  <!-- rh -->
  <column name="rh" dataType="DOUBLE" unit="-"
    cfStdName="relative_humidity" cfLongName="Relative Humidity" cfUnits="1"
    cfRelCols="u_std_rh u_cor_rh u_rh " cfCoords="lon lat alt"
    gFmtType="FLT" gFmtWidth="6" gFmtFormat="F6.2" gFmtNaN="NaN"
    gSrcDesc="FRAWPTU_U1-U2_cc_RC_TL" gResolution="see column res_rh"
    gColType="original data" testValues="true" tvMin="-0.050" tvMax="1.250"
    tvLimitNaN="100" tvLimitZero="10" >
    <!-- CF attributes -->
    <attribute name="comment" value="Relative humidity collated from U1 and U2
     based on the water vapor pressure formula of HylandWexler, corrected by GRUAN
     correction scheme" />
  </column>

  <!-- cor_rh -->
  <column name="cor_rh" dataType="DOUBLE" unit="-"
    cfStdName="relative_humidity correction" cfUnits="1"
    cfLongName="Correction of relative_humidity" cfCoords="lon lat alt"
    gFmtType="FLT" gFmtWidth="6" gFmtFormat="F6.2" gFmtNaN="NaN"
    gSrcDesc="FRAWPTU_U1-U2_cc_RC_TL" gColType="correction applied to the data"
    testValues="true" tvMin="-0.5" tvMax="0.5" tvLimitNaN="100"
    tvLimitZero="10" >
    <!-- CF attributes -->
    <attribute name="comment" value="Bias corrections applied to
     relative_humidity by the GRUAN correction scheme" />
  </column>
<!-- GRUAN attributes -->
<attribute name="g_processing_flag" required="false" />
<attribute name="g_resolution" required="false" />
</column>

<!-- u_rh -->
<column name="u_rh" dataType="DOUBLE" unit="-"
  cfStdName="relative_humidity standard_error" cfUnits="1"
  cfLongName="Uncertainty of relative_humidity" cfCoords="lon lat alt"
  gFmtType="FLT" gFmtWidth="6" gFmtFormat="F6.2" gFmtNaN="NaN"
  gSrcDesc="FRAWPTU_U1-U2_cc_RC_TL" gColType="total uncertainty"
  testValues="true" tvMin="0.0" tvMax="0.1"
  tvLimitNaN="100" tvLimitZero="10" >
  <attribute name="comment" value="Standard uncertainty (k=1) of
  relative_humidity calculated by the geometric sum of the correlated and
  random uncertainties" />
  <attribute name="related_columns" required="false" />
</column>
</dataMap>
</fileDescription>

---

**Acronyms**

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GRUAN</td>
<td>GCOS Upper-Air Network</td>
</tr>
<tr>
<td>gt92</td>
<td>GruanToolRs92</td>
</tr>
<tr>
<td>JRE</td>
<td>Java Runtime Environment</td>
</tr>
</tbody>
</table>

**References**

Sommer, M., Brief Description of GRUAN NetCDF Radiosonde Raw Data Files (GNC-RAW), 2020a, DRAFT, in preparation.

Sommer, M., Brief Description of Vaisala DigiCORA 3 DataBase File Format (DC3DB), 2020b, DRAFT, in preparation.


Vaisala, *DigiCORA III MW31 - Technical Reference*, Vaisala Oyi, P.O. Box 26 FIN-00421 Helsinki Finland, 2015, document number M210489EN-M.
Vaisala, *Vaisala DigiCORA Sounding System MW41 - Technical Reference*, Vaisala Oyi, P.O. Box 26 FI-00421 Helsinki Finland, 2019, document number M211415EN-W.


Vömel, H., *Description of the Strato and Balloon - Data output*, 2014b, version 1.2.