
The GIPTools Release History

\$Revision: 2618 \$

2018-11-06

This file contains a high-level, user relevant description of the GIPTools evolution. The list is sorted in reverse chronological order (i.e. most recent GIPTools release first). Releases are simply named after the date of their release. The first number indicates the release year and the three digit number after the dot is the respective day number of the date the software was released.¹

Release 2018.310

- [FIX] Detect corrupted Cube files containing very large blobs (i.e. >8MB) of "zero" bytes. The GIPTools will attempt to convert as much data as possible up to the location of the first corrupt area in the file and then skip ahead to the next input file.
- [FIX] Detect Cube file corruptions resulting from a sample block containing a few bytes to many. (Previously, only the more common case of a few missing bytes in a Cube sample block was detected reliably.)
- [FIX] GIPTools incorrectly assumed that blockette #1000 is always the first blockette following the fixed header section of a miniSEED record.
- [FIX] Huge time spans are now parsed correctly. This affected e.g. the `--shift-time` option of the `mseed2mseed` command when shifting the start time of a miniSEED record by more than 68.1 years. (Cause of the problem was an undetected integer overflow.)
- [FIX] Changed the Microsoft Windows start scripts to correctly set the `ERRORLEVEL` return code variable after the execution of a GIPTools utility.
- [UPG] Updated leap seconds-list based on *Bulletin C number 56* (published by the Earth Orientation Center of IERS). The new leap-second list will expire June 28th, 2019.

Release 2018.155

- [FIX] Corrected a bug in the Windows start scripts that caused the utilities not to start at all if the path to the GIPTools installation directory contained spaces or other "problematic" characters.
- [UPG] Improved error message when the Cube header block contains illegal (non-ASCII) characters like German "umlaute" or French accents. (Note: Non-ASCII characters are still illegal in the Cube header!)
- [UPG] Updated leap seconds-list based on *Bulletin C number 55* (published by the Earth Orientation Center of IERS). The new leap-second list will expire Dec 28th, 2018.

Release 2017.340

- [FIX] Corrected an obscure "array index out of bounds" error in the date/time computations. The code parsing the GPS time information did not correctly handle the situation of a leap second event being recorded on Dec 31st of any year (opposed to events occurring on Jun 30th, which worked flawless). This error only shows if the Cube GPS unit is actually powered on during the leap second event.

¹Keys used to categorize changes between GIPTool releases:

Key	Description
[CHG]	Changes previous functionality
[FIX]	Bug fix
[NEW]	New feature or utility
[OPT]	Optimization
[UPG]	Enhancement of existing functionality

- [FIX] The **mseed2pdas** utility now correctly identifies miniSEED files recorded by an EDR-209/210 units.
- [OPT] Reduced the memory consumption (RAM) of the **cube2ascii** and **cube2mseed** utilities by roughly 30% at the price of a slightly increasing runtime by about 3%.
- [UPG] Added detection of duplicate miniSEED records when rescuing data from corrupted media with the **mseedrecover** utility. This is the new default behaviour but can be switched of using the `--no-dedupe` command line option.
- [UPG] Added a new timing "quality control" algorithm (`--timing-control=FAKE`) to the utilities **cube2ascii**, **cube2mseed** and **cube2segy** that replaces all Cube timing information by a made-up "fake time".

Important! This algorithm is intended for worst case scenarios only, where a user ends up with a Cube data stream that cannot be processed due to total lack of (recorded) timing information! By adding fake time information to the Cube file becomes "processable" again, although at the price of a completely made-up time information. Please see the documentation for details before use.

- [UPG] Add a command line option to fix certain faulty record start times encountered in miniSEED files that were recorded by an EDR-209/210 data logger running a specific, buggy firmware. Please see the **mseed2mseed** manual page for details.
- [NEW] Added support for the new (and still experimental) event recorder units designed to record the trigger time of seismic sources. Specifically, the new utility **cubeevent** can be used to read detected / recorded events from files. In addition, the other Cube related GIPPTools (**cubeinfo**, **cube2ascii**, etc.) were upgraded as well to correctly handle event recorder files. (This works because event recorder write a variant of the Cube file format.)
- [NEW] Added a 'config' subdirectory to the GIPPTools distribution. At the moment it only contains a copy of the latest 'leap-seconds.list' file. In the future it will also contain things such as default settings or user preferences.

Release 2017.013

- [CHG] Added an expiration date to the internal leap second table so that GIPPTool utilities will not accidentally apply outdated leap second corrections. As a consequence, the Cube conversion tools (**cube2ascii**, **cube2mseed** and **cube2segy**) will refuse to process a file if the available leap second information is outdated (with respect to the Cube input). The **cubeinfo** program will output a warning. Utilities processing miniSEED formatted (input) data are not affected.
- [NEW] Added a `--sysinfo` command line argument to all GIPPTool utilities. Using it will generate a brief report of the (GIPPTools relevant) computing setup you are using. The system summary should be helpful for diagnosing configuration problems or when reporting bugs to the developer.

Release 2016.358

- [FIX] The **mseedrecover** utility will (again) read directly from a Unix/Linux device file. (There was an overeager safety check that complained about a device file of "zero length".)
- [FIX] Implemented a workaround for the "Could not parse TAIP message..." error message. This bug only affected users that (still) use an outdated Java 1.6 runtime to convert files recorded by Cubes with build-in "newer" GPS board. This bug did not surface when using Java 1.7 or newer.
- [FIX] Added a proper error message when reading from miniSEED files that contain an uninitialized "record size" header field (i.e. claim an impossible miniSEED record size of zero).
- [FIX] Corrected a faulty range check when writing miniSEED files using IEEE single precision floating point encoding.

- [FIX] Repaired the `--record-size` command line option of the **cube2mseed** utility, which was not accepting any number at all. Cause was an internal integer overflow when comparing the given record size with a build-in, upper limit.
- [UPG] Updated internal leap second table with the upcoming leap second event on 2016-12-31.
- [UPG] Upgraded the checksum calculation (option `--format=CHECKSUM`) of the **mseedinfo** utility to use the MD5 algorithm provided by the Java runtime. If accessing the MD5 algorithm fails for some reason, it will automatically fall back to SHA-1 or the previously used CRC-64 algorithm.
- [UPG] Added a `--shift-time` command line option to the **mseed2mseed** utility that will move the start time of miniSEED records by the given time span.
- [OPT] Updated the **cube2segy** utility to use the same Cube I/O subroutines as **cube2ascii** and **cube2mseed**. Due to the (internal) change the utility will be more resilient to bad GPS reception. This should also improve timing precision on sub-sample level.
- [CHG] The Cube conversion tools (**cube2ascii**, **cube2mseed** and **cube2segy**) contain a new (default) algorithm to better quality control the recorded timing information (option `--timing-control=LLS`). The main advantage of the LLS algorithm is its flexibility. It was designed to adapt to different situations and to handle different time keeping hardware as well. However, you can use the command line option `--timing-control=RULE` to fall back to the previously used timing quality control method.
- [CHG] Slightly modified the GPS output format (`--format=GPS`) of the **cubeinfo** utility. The reported leap second information now clearly indicates the origin of that information. The string 'gps-leap' stands for leap second information obtained from the GPS satellite broadcasts, while 'iers-leap' indicates leap seconds as officially announced by the International Earth Rotation and Reference Systems Service (IERS). The later are kept in an internal table.
- [CHG] Renamed the DUMP output format of the **cubeinfo** utility to DEBUG, which more clearly indicates its intended use.

Release 2015.225

- [FIX] Handle the case of a misspelled input file or directory more gracefully (i.e. with a proper warning message instead of just skipping the non-existing file silently).
- [FIX] The **mseed2mseed** utility erroneously did not correctly re-encode an input file containing more than 64 miniSEED records. Everything worked fine for the first 63 records, however, the following 63 records were not re-encoding at all. (No data was lost but the output files contained the later records in their original miniSEED encoding!) And yes, this is a follow-up to the (apparently incomplete) bug fixed in the previous release. Sorry!

Release 2015.196

- [FIX] The **mseed2mseed** utility erroneously produced data gaps in the output when an input file containing more than 64 miniSEED records was encountered while at the same time re-encoding the data (using at least one of the options `--record-size`, `--encoding` and `--byte-order`). Basically an internal buffer was not flushed (completely) when the end of an input file was reached, resulting in missing data in the output.
- [FIX] Handle the case of a zero length input file more gracefully, (i.e. with a proper warning message instead of a null pointer exception).

Release 2015.166

- [FIX] Something went wrong while packaging the previous 2015.163 release. The Java classes were apparently compiled without the correct backward compatible setting. Unfortunately, the resulting GIPPTools release required at least a Java 1.7 runtime. A quick recompilation with correct settings solved the problem.

Release 2015.166 is functional identical to 2015.163 with the single exception that it only requires a Java 1.5 runtime environment!

Release 2015.163

- [NEW] There is a new **cube2ascii** utility that works just like the already existing **cube2mseed** but outputs the Cube trace in ASCII text format instead of miniSEED format.
- [UPG] The **cube2mseed** utility was completely rewritten from scratch! The new version will correctly handle Cube data streams spanning over several files, long(er) periods without GPS reception and compensate for drifting clocks by resampling the Cube input.
- [CHG] Due to some internal changes, the checksum calculation for miniSEED files (command **mseedinfo --format=CHECKSUM**) had to be changed. Do not compare checksum calculated by an earlier GIPPTools release with checksums calculated with the **mseedinfo** tool of this (and future) releases! All checksum must be re-calculated!
- [NEW] Added IEEE double (`--encoding=FLOAT-64`) and single (`--encoding=FLOAT-32`) precision floating point numbers to the supported miniSEED encodings!
- [FIX] Took care of a problem, where creating files of equal length (e.g. when running **mseedcut --file-length=DAY**) would occasionally "forget" a single sample value in the data stream. This error occurred only when a very specific combination of sample amplitudes and miniSEED record layout was encountered, which is probably the reason why it went undetected for such a long time.
- [FIX] Took care of another problem while creating files of equal length (e.g. when running **mseedcut --file-length=DAY**). Once in a while a file of twice the desired length would be generated instead of two separate files. This error happened when the first sample of a new file was also located at the end of a miniSEED record in the input. Chances for this to happen depend on the used miniSEED encoding but are typically in the range from 1:1000 to 1:2000.
- [FIX] Utility **mseedcut** now correctly handles event files (option `--events=file`) as exclusive source for user requested time windows. Before, it would only process an event file when an additional time window was requested via other command line parameters.
- [FIX] All Cube related utilities now correctly handle the error case of two time tags following directly after each other in the recorded data stream. (Expected is approximately one second of samples in between time tags!)
- [FIX] Due to a firmware bug in (some) Trimble GPS boards, affected Cube recorder erroneously compensate a future leap second the moment it is announced via satellite and not starting at the time when the new leap second will happen. Unfortunately, pending (new) leap seconds usually are announce via GPS satellite many weeks in advance of the actual change... A workaround was added to the GIPPTools to compensate for the premature leap second correction.
- [FIX] Removed a bug, where an uninitialized internal variable accidentally would prevent the display of an error message.
- [UPG] Updated all Cube related GIPPTools so that they can handle the recent additions to the Cube data format. Especially the **cubeinfo** program now reports on auxiliary channels as well as event and debug blocks when called with option `--format=SUMMARY` or `--format=DUMP`.

Release 2013.268

- [FIX] Took care of an stupid decoding error when reading Steim compressed miniSEED files written by the new EDR-209/210 logger from EarthData.
- [UPG] Added a new output format called QUALITY to **mseedinfo** that provides information about the data quality flag and (if available) the timing quality value stored in a miniSEED record. The same information was also added to the HEADER output format of **mseed2ascii**.
- [UPG] Updated the **cubeinfo** tool so it knows about the newest enhancements to the Cube hard- and software (such as e.g. auxiliary recording channels and the new GPS board).

- [FIX] Ignore GPS information read from Cube files when the latest corresponding GPS fix is older than 10 seconds. (Note: This patch is only relevant for Cube recorders with the new GPS board actually providing that information. Cube files written by recorders with first generation GPS boards cannot provide the necessary GPS information in the first place. No need to reprocess old files!)
- [UPG] Expanded the GPS output format of the **cubeinfo** tool (command line option `--format=GPS`) to give an idea about the GPS reception and the quality of the Cube time information. Additional values (elevation and temperature) become available when the used Cube recorder contains one of the new GPS boards providing that information.
- [NEW] Added a new environment variable `GIPPTOOLS_OPTS` that can be used to fine-tune the Java runtime. This would typically be used to set the Java heap size available to GIPPTool programs.
- [FIX] Made **cubeinfo** more resilient to crashes when working with damaged or incomplete Cube data files.
- [NEW] Finally got around to write a proper Readme file and to start a Frequently Asked Questions (FAQ) list.

Release 2012.093

- [FIX] Took care of a problem where **mseedcut** failed to produce files of equal length (option `--file-length`) if no sample was recorded precisely on the hour. This became a problem when working with converted files that were originally recorded by a Cube.

Release 2012.088

- [UPG] **Mseed2mseed** now understands the keyword 'RESET' when parsing actions in a rule file or at the command line. Use the new keyword to clear/remove id values in miniSEED header fields.
- [NEW] The utilities **cube2segy** and **mseed2segy** can now cache the (internal) file index that is build from the input files. Using the new `--index-cache` option is especially helpful If you plan to use one of the programs repeatedly on the same, large dataset.
- [UPG] Improved syntax checking of the project file (utilities **cube2segy** and **mseed2segy**) enables better user feedback (i.e. error messages) when the format of the project file is almost but not quite correct. Also added a few integrity checks to catch obvious typing errors.

Release 2011.321

- [CHG] Renamed EDL / miniSEED related GIPPTools by replacing "edl" with "mseed" in the command name:

Old	New
edl2ascii	mseedinfo (to inspect files) and mseed2ascii (for conversion)
edl2edl	mseed2mseed
edl2pdas	mseed2pdas
edlcut	mseedcut
edlrecover	mseedrecover
edlrename	mseedrename

This change was done as the current incarnation of the GIPPTool collection is not longer limited to only process miniSEED files created by the EarthData Loggers (EDL) in use at the GIPP. This should be reflected by the name of the utilities as well.

- [UPG] The **mseedrename** utility will now try to preserve the "last modified" time of a file while renaming it.

- [UPG] Added an `--events` option to the **mseedcut** utility. Use this option to cut out more than one time window during a single program run.
- [FIX] Added an extra test to **mseedrecover** to make sure that all samples (and not only the header fields) of a recovered miniSEED record are readable as well. Before this bug fix some only slightly damaged (but nevertheless defect) records were not reliably detected.
- [CHG] The change introduced to the **cube2segy** and **mseed2segy** utility in the last release (2011.166) was reverted after user protest. By default both tools behave again as accustomed when writing seismic sections, creating a separate file for every shot/receiver gather requested.

Use the new command line option `--force-concat` when all resulting seismic sections should be written to a single output file.

- [CHG] Instead of automatically concatenating the results from some operation to obtain as few files as possible, the new default file creation strategy for GIPPTools is to start a new output file for every read input file. Judging from user feedback, this seems to be the expected behaviour by the majority.

So far, this change has been implemented for **mseedcut**, **mseedrecover**, **mseed2ascii**, **mseed2mseed** and **mseed2pdas**. You can use the new command line option `--force-concat` to return to the former, concatenating output behaviour.

A nice side effect of the re-written output routines is that the utilities need (much) less memory and are more stable when applied to very large datasets.

- [CHG] Renamed the `--segment-length` command line option of **mseedcut** to `--file-length`. The new name better describes the effect the option (writing files containing data of given length).
- [CHG] Renamed the `--output-sort` command line option of **mseedrecover** to `--force-sort`. The new name aligns better with the other output options `--force-overwrite` as well as `--force-concat` and should be easier to remember by the user.
- [CHG] Renamed the `--rule-file` command line option of **mseed2mseed** to `--rules` and renamed the `--event-file` command line option of **cube2mseed** to `--events`. The new names are shorter and better describe what the respective (file) arguments contain.

Release 2011.166

- [FIX] The utility **edlcut** no longer crashes when the miniSEED input does not contain any relevant data for the requested time window. (This happened only under certain rare conditions.)
- [NEW] Added the options `--byte-order`, `--record-size` and `--encoding` to the **cube2mseed** utility so that the user can influence the miniSEED output format.
- [CHG] Instead of creating a new output file for every shot/receiver gather requested by the user, the **cube2segy** and **mseed2segy** utility will write all seismic sections into one single file. This was changed to ease the data import into other software packages. (You only need to import one single file instead of many different files.)

Release 2011.119

- [UPG] The `--include-pattern` option can be used more than once in the same command line. The given include patterns are cumulative.
- [UPG] Improved **cube2segy** to work with the extended, multi-file Cube data format of the next generation three channel units (currently in development).
- [OPT] Sped up the **cube2segy** utility to run two to five times as fast as before! Please note: Speed-up highly depends on the used Cube recorder settings, especially sampling rate and GPS cycling times. So, your mileage may vary...

- [FIX] The **cube2segy** utility will not longer crash when the time window of the trace to read starts before the begin of recording in the corresponding Cube file. (This only happened when there was no data for the begin of the time window but data for the end of the time window was available.)

Release 2010.321

- [UPG] Improved **cubeinfo** to work with extended Cube file format of the next generation three channel units currently in development. Due to this changes the output format of the report generated by **cubeinfo** has changed.
- [UPG] Added a new OVERVIEW output format to **edl2ascii** that will summarize miniSEED input streams consisting of many miniSEED records in an easy (for humans) readable format.
- [CHG] The **edlrename** utility will not longer overwrite already existing files. Instead a new filename with an additional number in between basenname and extension will be used for writing. To get the previous behavior of overwriting existing files use the new command line argument `--force-overwrite`.
- [OPT] Improved the utilities that convert to SEG-Y (**cube2segy** and **mseed2segy**). Instead of simply beginning the traces with the first sample after the respective start time, the sample closest to the start time is used. This reduces the timing error down to at most plus/minus half a sampling period. However, it also produces traces where the first sample lies before the requested window start time!

The difference between the requested begin of the trace and the time of the first sample is documented in the SEG-Y trace header (in field "lag A").

- [FIX] The **cubeinfo** will not longer produce "NullPointerException" errors in the default INFO mode when reading from Cube files that were cut off prematurely (e.g. running out of power while recording).

Release 2010.202

- [CHG] The **edl2segy** utility is now called **mseed2segy**.
- [CHG] The **edl2pdas**, **edl2ascii**, **cube2mseed** and **cubeinfo** utilities will not longer overwrite already existing files by default. Instead a new filename with an additional number in between basenname and extension will be used for writing. To get the previous behaviour of overwriting existing files use the new command line argument `--force-overwrite`.
- [NEW] There is a new **cube2segy** utility that works just like the already existing **mseed2segy** but reading Cube files as input instead of miniSEED files.
- [UPG] Added support for the newer XDR style output format of Seismic Unix to the **cube2segy** and **mseed2segy** utility. (Before, only the old "native binary" format was supported.)
- [UPG] Added a GPS output format to the **cubeinfo** utility reporting time and position of all GPS blocks contained in a Cube data stream.
- [UPG] Made **edlrecover** more robust when recovering "almost-miniSEED files" (i.e. files that closely resemble valid miniSEED but still contain only binary garbage should no longer confuse **edlrecover**). Changed the output subdirectory name to contain the "year" as well as the "day-of-year" when using the `--output-sort` option.
- [FIX] Times given with more than six digits for the fractional part of the second (e.g with nanosecond resolution) are now rounded to the closest microsecond instead of truncating the fractional part completely.
- [FIX] Got rid of an internal integer overflow when extracting very large time windows (e.g. one year) from a miniSEED stream. This problem affected all utilities that use time windows to select data.

Release 2010.120

- [FIX] The **cube2mseed** utility no longer writes miniSEED files that contain gaps or overlaps. Instead a new file will be started if an discontinuity in the data stream is detected.

- [UPG] Improved tolerance of the Steim decoding subroutines. The GIPPtools utilities now can read more miniSEED variants than ever before.
- [FIX] The **edl2segy** utility produced defect SEG-Y files under Solaris when using a Sun Java 1.6 virtual machine. (Strangely, it worked with a Java 1.5 runtime.)
- [FIX] Occasionally, **edl2segy** produced traces would contain one additional sample.

Release 2010.105

- [FIX] **Edl2segy** works again! The previous release contained a subtle error that rendered **edl2segy** completely unusable! Fortunately, none of the other utilities was affected.

Release 2010.074

- [NEW] Added support for the Cube data loggers. Use the new **cubeinfo** tool to get a textual summary about the content of a Cube file and **cube2mseed** to convert Cube files to miniSEED format.
- [NEW] Added handling of miniSEED blockette #1001 to all GIPPtool utilities. This makes it possible to work with microsecond precision (instead of "only" with 0.0001s resolution).
- [OPT] Completely rewrote the internal handling of miniSEED streams. Records are not longer decoded and encoded at every opportunity, which should speed up many GIPPtool utilities.
- [UPG] Added a new CHECKSUM output format to the **edl2ascii** program. Use it to calculate a checksum over the samples of a continuous trace. MiniSEED header values are not included into the checksum calculation so that different miniSEED encoding scheme should still result in the same checksum, provided that the time series are identical.
- [UPG] The sample rate (of a miniSEED file) was added to the list of possible template variables of the **edlrename** utility.
- [FIX] A change in the sample rate between two consecutive miniSEED records was not always detected correctly.
- [CHG] The **edlcut** and **edl2edl** utilities will not longer overwrite already existing files by default. Instead a new filename with an additional number in between basename and extension will be used for writing. To get the previous behaviour of overwriting existing files use the new command line argument `--force-overwrite`.
- [UPG] The **edlcut** utility can now split a long miniSEED data stream into segments of given length. This can be used to produce e.g. "day files" out of a collection of input files.
- [UPG] New command line parameters (`--stop-time`, `--trace-offset`) were added to the **edlcut** utility to simplify the specification of a time window for cutting out samples from miniSEED data streams.
- [CHG] Minor change to the user interface. Error messages are not longer begin with the text SEVERE. Instead they start with ERROR! After all, that is what they are. (Several users assumed that SEVERE is just some sort of fancy warning that can be ignored.)

Release 2009.244

- [NEW] Added the new **edl2edl** utility. **Edl2edl** can be used to selectively modify miniSEED header fields such as e.g. station, channel and network id. You can also use it to change the record size, byte order or encoding scheme of a miniSEED file.
- [NEW] Added Steim-2 encoding support to all GIPPtool utilities.
- [CHG] Improved the HEADER output format of the **edl2ascii** program to also report on record size, byte order and used data encoding scheme.

- [UPG] Added a new GMT output format to the **edl2ascii** program. Files created using the GMT parameter can directly be used as "multiple segment" input to the **psxy** command of the Generic Mapping Tools (also known as GMT, hence the format name).
- [FIX] Changed an internal filename filter that falsely discarded the dot ('.') as an invalid (input) directory name. It is again possible to use the current working directory (aka '.') as starting point for a recursive search for input files.

Release 2009.026

- [UPG] The **gipptools** start script (jointly used by all GIPPTool utilities) will detect a set GIPPTOOLS_JAVA environment variable and use the respective Java Runtime Environment (JRE) instead of the automatically detected one.
- [NEW] Added the **edlrename** utility that can be used to systematically rename miniSEED files using a user provided "filename template".
- [NEW] Created a separate Microsoft Windows distribution of the GIPPTools software. (Note! The main difference between the Windows and Unix distribution is the used start script. For Unix a Bourne shell script is used while the Windows distribution relies on a Windows command file to start the various GIPPTool utilities.)

Release 2008.351

- [NEW] Added the **edlrecover** utility that can be used to recover miniSEED data from damaged files or corrupted file systems.
- [CHG] The **edl2ascii** output format SUMMARY was renamed to INDEX, which is more descriptive and frees the term SUMMARY to be used for the new output format.
- [UPG] Added a new SUMMARY output format to **edl2ascii** that will summarize miniSEED input streams consisting of many miniSEED records.
- [UPG] Added Unix manual pages for **edl2ascii**, **edl2pdas**, **edlcut** and **edlrecover**.

Release 2008.291

- [OPT] Significantly reduced the memory requirements of **edl2segy**. The program now runs comfortable in less than 512 MByte RAM and does not longer crash on larger jobs.

Release 2008.280

- [NEW] Added the **edl2segy** program to the GIPPTools collection. Use **edl2segy** to extract shot or receiver gathers from miniSEED files and save them as SEG-Y or Seismic Unix files.
- [FIX] Removed a command line parsing error from the program **edlcut** to make it possible to cut out time series data of less than ten seconds length. Before the fix **edlcut** demanded erroneously at least a two digit number for the `--trace-length` command line argument.

Release 2008.037

- [NEW] Added the **edlcut** utility to cut out data samples from miniSEED files based on a given start time and a trace length.

Everything before the year 2008 does not deserve the name "release", was non-public, preliminary or only in use locally at the GFZ. More detailed information is shrouded in the fog of history...