GIPPtools — Release History

This file contains a high-level, user relevant description of the GIPPtools evolution back to 2008. The list is sorted in reverse chronological order (i.e. most recent GIPPtools release first).

> ℹ️ Releases are 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-of-year number of the software release date.

Release 2022.340

Please cite this release as:
Lendl, Christof (2022), GIPPtools, Version 2022.340, doi:10.5281/zenodo.7405772

Added:

- Added the command line options `--select-station` and `--select-channel` to the `mseedinfo` utility.

Changed:

- Updated the leap seconds list based on *Bulletin C number 64* (published by the Earth Orientation Center of IERS). The new leap second list will expire June 28, 2023.

Release 2022.171

Please cite this release as:
Lendl, Christof (2022), GIPPtools, Version 2022.171, doi:10.5281/zenodo.6668924

Fixed:

- None of the GIPPtools will crash anymore when encountering miniSEED records containing ASCII encoded (meta-)data.

Added:

- Two new command line options (`--select-station` and `--select-channel`) were added to the `mseed2ascii`, `mseed2pdas` and `mseedcut` utilities. Use them to select a subset of the miniSEED input for processing when working with multiplexed miniSEED files. Records containing data from other stations or recording channels are silently skipped/ignored during processing.

Changed:

- Updated the leap seconds list based on *Bulletin C number 63* (published by the Earth Orientation Center of IERS). The new leap second list will expire December 28, 2022.

Release 2021.327

Please cite this release as:
Lendl, Christof (2021), GIPPtools, Version 2021.327, doi:10.5281/zenodo.5643357

Fixed:

- Correcting a sub-sample timing inaccuracy in the postprocessing (conversion) of Cube data streams.

  To use data acquired with Cube recorders, the data has to be converted from the proprietary internal data format, including interpretation of the timing information contained in the data stream. We were notified that the recording time computation for the samples was imperfect.

  The necessary correction depends on the used sampling rate (*sps*) and can be computed as `(0.199 / sps + 0.00030)` seconds. For example a sampling rate of 100 samples per second requires an additional time correction of 2.29 milliseconds, which have not been considered so far.

  This (and all following) GIPPtool release(s) will automatically correct imperfect Cube timing during conversion to other data formats (using `cube2mseed`, `cube2segy` and `cube2ascii`).

  > 🛈 As the timing error is only about a quarter of the sample period it is probably not a grave problem for most applications!

  Many thanks to Jake Anderson from Boise State University for reporting the issue and contributing to the fix.

- Correctly warn the user about a missing JRE under Windows.
  Previously, Windows user were only receiving a rather misleading error message when no Java Runtime Environment (JRE) was found.

Added:

- Extended the `mseed2mseed` utility to handle the timing quality field located in blockette #1001 (inside the miniSEED header).

  Use one or more of the command line options `--equals-timing-quality`, `--less-than-timing-quality` or `--more-than-timing-quality` as condition to select individual miniSEED records based on their respective timing quality. The `--set-timing-quality` command line option can be used to modify the timing quality field.

Please see the manual page of the `mseed2mseed` utility for more details.

Changed:

- Updated the leap seconds list based on *Bulletin C number 62* (published by the Earth Orientation Center of IERS). The new leap second list will expire June 28, 2022.

Release 2021.168

Please cite this release as:
Lendl, Christof (2021), GIPPtools, Version 2021.168, doi:10.5281/zenodo.4972595

Changed:

- The GIPPtools now expect at least a working Java virtual machine (also sometimes called a Java Runtime Environment or simply JRE) supporting **Java 8** or newer!
  This change became necessary as it becomes increasingly difficult to generate Java class/jar files on up-to-date platforms that are still backward compatible back to Java Release 5!

- Improving the check for corrupted Cube files. The algorithm originally used when this check was introduced with the previous GIPPtools release produced to many false alarms when the GNSS reception is not optimal.

- Support Cube logger with build-in u-Blox GNSS receiver board as time source.

- Updated the leap seconds list based on *Bulletin C number 61* (published by the Earth Orientation Center of IERS). The new leap second list will expire December 28th, 2021.

Release 2020.352

Please cite this release as:
Lendl, Christof (2020), GIPPtools, Version 2020.352, doi:10.5281/zenodo.4707612

Fixed:

- Cube conversion tools will ensure that the time between consecutive Cube time tags "flows" in the right direction.
  This additional check was inspired by a corrupted Cube file read from a failing drive. It somehow contained fragments of an earlier recording and converting the data to miniSEED resulted in misleading error messages and other "surprising" results.

- The `--include-pattern=GIPP` filename filter now accepts month numbers larger than '12'!
  There is a new option in the Cube firmware that forces a strict increment of the month number for each consecutive month. If used in a recording that starts in December then January would become '13', February would be '14', and so on. The option is intended for long-running (> 1 year) continuous recordings where otherwise filenames might repeat. (Cube filenames do not include a year.) Obviously, Cube filenames beginning with such higher month numbers should nevertheless be included by the GIPP filename filter, hence the change.

- Corrected the handling of Cube "system event" block containing more than two bytes. Previously, `cubeinfo` would stop working when encountering one of those rare blocks. (This bug did not affect other Cube related utilities.)

Changed:

- Automatically add a blockette #1001 to the miniSEED record if necessary. A blockette #1001 adds (among other things) microsecond resolution to the start time of the miniSEED record. This will improve the timing precision in situations where the miniSEED output requires more resolution after processing

  An example would be a 300 Hz recoding, where some recording times contain repeating decimal numbers (e.g. 14:20:09.953333...; with an infinite number of "threes" in the fractional second part).

- Updated the leap seconds list based on *Bulletin C number 60* (published by the Earth Orientation Center of IERS). The new leap second list will expire June 28th, 2021.

Release 2020.161

Fixed:

- The `mseed2segy` utility no longer crashes when the miniSEED input is fragmented and contains gaps/overlaps. This only happened under very scarce circumstances.

Changed:

- Updated the leap seconds list based on *Bulletin C number 59* (published by the Earth Orientation Center of IERS). The new leap second list will expire December 28th, 2020.

Release 2019.332

Added:

- New `--bugfix-gps-epoch` command line option for the `mseed2mseed` utility. It is intended to be used to fix incorrect recording times that were caused by a GPS week number rollover (WNRO) problems. Please see the program documentation for more details on how to apply the corrections.

Changed:

- Updated the leap seconds list based on *Bulletin C number 58* (published by the Earth Orientation Center of IERS). The new leap second list will expire June 28th, 2020.

Release 2019.155

Fixed:

- The miniSEED utilities stumbled over partially read blockettes. The problem surfaced for the first time when the mandatory blockette *#1000* is placed after an optional blockette *#100* (i.e. parts of blockette *#1000* extend beyond the initially read 64 byte).

Added:

- Preliminary read support of blockette *#100* ("sample rate"). If a miniSEED record contains a blockette *#100*, the sample rate information from the blockette will supersede the sample rate information from the fixed header area.

Changed:

- Improved I/O speed when converting Cube data to miniSEED/ASCII.
- Updated the leap seconds list based on *Bulletin C number 57* (published by the Earth Orientation Center of IERS). The new leap second list will expire December 28th, 2019.

Release 2018.310

Fixed:

- Detect corrupted Cube files containing very large blobs (i.e. >8 MB) of "zero" bytes. The GIPPtools 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.
- Detect Cube file corruption 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.)
- GIPPtools incorrectly assumed that blockette *#1000* is always the first blockette following the fixed header section of a miniSEED record.
- 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.)
- Changed the Microsoft Windows start scripts to correctly set the `ERRORLEVEL` return code variable after the execution of a GIPPtools utility.

Changed:

- Updated the 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

Fixed:

- Corrected a bug in the Windows start scripts that caused the utilities not to start at all if the path to the GIPPtools installation directory contained spaces or other "problematic" characters.

Changed:

- 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!)
- Updated the 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

Fixed:

- 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.
- The `mseed2pdas` utility now correctly identifies miniSEED files recorded by an *EDR-209/210* units.
- 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%.

Added:

- 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".

  Only use this algorithm for worst case scenarios, 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 using it.

- 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 off using the `--no`

`-dedupe` command line option.

- New 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.

- Support 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.)

Changed:

- 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

Added:

- New `--sysinfo` command line argument for 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.

Changed:

- 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.

Release 2016.358

Fixed:

- The `mseedrecover` utility will (again) read directly from a Unix/Linux device file. (There was an overeager safety check complaining about a device file of "zero length".)

- 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.

- 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).

- Corrected a faulty range check when writing miniSEED files using IEEE single precision floating-point encoding.

- 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.

- 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.

Added:

- 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.

- Added a `--shift-time` command line option to the `mseed2mseed` utility moving the start time of miniSEED records by the given time span.

Changed:

- 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.

- 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.

- Renamed the `DUMP` output format of the `cubeinfo` utility to `DEBUG`, which more clearly indicates its intended use.

- Updated internal leap second table with the upcoming leap second event on 2016-12-31.

Release 2015.225

Fixed:

- 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).

- 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-encoded at all. (No data was lost, but the output files contained the following 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

Fixed:

- The `mseed2mseed` utility erroneously produced data gaps in the output when an input file containing more than 64 miniSEED records.
  This only happened while 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.

- 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

Fixed:

- Something went wrong while packaging the previous 2015.163 release. Unfortunately, Java classes were compiled without the correct backward compatible setting. The resulting GIPPtools release required at least a Java 1.7 runtime. A quick re-compilation with correct settings solved the problem.
  Release 2015.166 is functional identical to 2015.163 with the single exception that it requires only a Java 1.5 runtime environment!

Release 2015.163

Fixed:

- 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 it went undetected for such a long time.

- 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.

- 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.

- 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!)

- 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 are usually 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.

- Removed a bug, where an uninitialized internal variable accidentally would prevent the display of an error message.

Added:

- A new `cube2ascii` utility working just like the already existing `cube2mseed` but outputting the Cube trace in ASCII text format instead of miniSEED format.

- New IEEE double (`--encoding=FLOAT-64`) and single (`--encoding=FLOAT-32`) precision floating-point numbers to the supported miniSEED encodings!

Changed:

- 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!

- 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.

- 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

Fixed:

- Took care of a stupid decoding error when reading Steim compressed miniSEED files written by the new *EDR-209/210* logger from EarthData.

- 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!)

- Made `cubeinfo` more resilient to crashes when working with damaged or incomplete Cube data files.

Added:

- 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`.

- 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).

- 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.

- Finally, got around to write a proper `Readme` file and to start a Frequently Asked Questions (FAQ) list.

Changed:

- 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.

Release 2012.093

Fixed:

- 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 an issue when working with converted files that were originally recorded by a Cube.

Release 2012.088

Added:

- `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.

- 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.

- 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

Fixed:

- Added an extra test to `mseedrecover` to make sure 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 records were not detected reliably.

Added:

- New `--events` option for the `mseedcut` utility. Use this option to cut out more than one time window during a single program run.

Changed:

- Renamed EDL/miniSEED related GIPPtools by replacing the string *edl* with *mseed* in the respective command name:

  - `edl2ascii` → `mseedinfo` (to inspect files) + `mseed2ascii` (for conversion)

  - `edl2edl` → `mseed2mseed`

  - `edl2pdas` → `mseed2pdas`

  - `edlcut` → `mseedcut`

  - `edlrecover` → `mseedrecover`

  - `edlrename` → `mseedrename`

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

- The change introduced to the `cube2segy` and `mseed2segy` utility in the last 2011.166 release 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.

- 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.

- 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).

- 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.

- 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

Fixed:

- 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.)

Added:

- New options `--byte-order`, `--record-size` and `--encoding` in the `cube2mseed` utility so that the user can influence the miniSEED output format.

Changed:

- 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 files.)

Release 2011.119

Fixed:

- The `cube2segy` utility will no longer crash when the time window of the trace to read starts before the beginning of recording in the corresponding Cube file. (This only happened when there was no data for the beginning of the time window but data for the end of the time window was available.)

- Sped up the `cube2segy` utility to run two to five times as fast as before! Please note: The individual speed-up highly depends on the used Cube recorder settings, especially sampling rate and GPS cycling times. So, your mileage may vary...

Added:

- Upgraded `cube2segy` to work with the extended, multi-file Cube data format of the next generation three channel units (currently in development).

Changed:

- The `--include-pattern` option can now be used more than once in the same command line. The given include patterns are cumulative.

Release 2010.321

Fixed:

- The `cubeinfo` will no 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).

- The `edlrename` utility will no longer overwrite already existing files. Instead, a new filename with an additional number in between basename and extension will be used for writing. To get the previous behavior of overwriting existing files use the new command line argument `--force-overwrite`.

Added:

- New *OVERVIEW* output format to `edl2ascii` that will summarize miniSEED input streams consisting of many miniSEED records in an easy (for humans) readable format.

Changed:

- 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 beginning of the trace and the time of the first sample is documented in the SEG-Y trace header (in field "*lag A*").

- Improved `cubeinfo` to work with extended Cube file format of the next generation three channel units currently in development. Due to these changes the output format of the report generated by `cubeinfo` has changed.

Release 2010.202

Fixed:

- The `edl2pdas`, `edl2ascii`, `cube2mseed` and `cubeinfo` utilities will no 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`.

- 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.

- 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.

Added:

- New `cube2segy` utility working just like the already existing `mseed2segy` but reading Cube files as input instead of miniSEED files.

- Support the newer XDR style output format of Seismic Unix for the `cube2segy` and `mseed2segy` utility. (Before, only the old "native binary" format was supported.)

- Added a GPS output format to the `cubeinfo` utility reporting time and position of all GPS blocks contained in a Cube data stream.

- 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.

Changed:

- The `edl2segy` utility is now called `mseed2segy`.

Release 2010.120

Fixed:

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

- Improved tolerance of the Steim decoding subroutines. The GIPPtools utilities now can read more miniSEED variants than ever before.

- The `edl2segy` utility produced defect SEG-Y files under Solaris when using a Sun Java 1.6 virtual machine. (Strangely, it always worked with a Java 1.5 runtime.)

- Occasionally, `edl2segy` produced traces would contain one additional sample.

Release 2010.105

Fixed:

- `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

Fixed:

- A change in the sample rate between two consecutive miniSEED records was not always detected correctly.

- The `edlcut` and `edl2edl` utilities will no 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`.

- Completely rewrote the internal handling of miniSEED streams. Records are no longer decoded and encoded at every opportunity, which should speed up many GIPPtool utilities.

Added:

- Support 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.

- Support miniSEED blockette *#1001* in all GIPPtool utilities. This makes it possible to work with microsecond precision (instead of "only" with 0.0001s resolution).

- New *CHECKSUM* output format for 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.

- The sample rate (of a miniSEED file) was added to the list of possible template variables of the `edlrename` utility.

- 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.

- 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.

Changed:

- Error messages do no 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, which can be ignored.)

Release 2009.244

Fixed:

- 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.

Added:

- New `edl2edl` utility, which 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.

- Added *Steim-2* encoding support to all GIPPtool utilities.

- 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).

Changed:

- Improved the `HEADER` output format of the `edl2ascii` program to also report on record size, byte order and used data encoding scheme.

Release 2009.026

Added:

- New `edlrename` utility to systematically rename miniSEED files using a user provided "filename template".

- Created a separate Microsoft Windows distribution of the GIPPtools software.
  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.

Changed:

- 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.

Release 2008.351

Added:

- New `edlrecover` utility to recover miniSEED data from damaged files or corrupted file systems.

- Added Unix manual pages for `edl2ascii`, `edl2pdas`, `edlcut` and `edlrecover`.

Changed:

- 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.
- Added a new *SUMMARY* output format to `edl2ascii` that will summarize miniSEED input streams consisting of many miniSEED records.

Release 2008.291

Fixed:

- 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

Fixed:

- 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.

Added:

- New `edl2segy` program, which can be used to extract shot or receiver gathers from miniSEED files and save them as SEG-Y or Seismic Unix files.

Release 2008.037

Added:

- New `edlcut` utility to cut out data samples from miniSEED files based on a given start time and a trace length.

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