gipptools(1)

Name

gipptools - a collection of programs for (seismic) data conversion and pre-processing

Synopsis

gipptools [--run-tool=*TOOL* [*TOOLPARAM*]...]

gipptools [-h | --help] [--version] [--sysinfo]

Introduction

The *Geophysical Instrument Pool Potsdam* (*GIPP*) provides seismic and magnetotelluric recorders as well as sensors for academic research. It is hosted by the *Helmholtz-Zentrum Potsdam Deutsches GeoForschungsZentrum GFZ*. The GIPPtools main motivation is to provide a collection of software utilities that aid with initial pre-processing of recorded data to support users that borrow instruments from the GIPP.

Hopefully, it is also useful for others that work with miniSEED or Cube data files.

Description

The programs in the GIPPtools collection are designed and tested to work with files produced by Data-Cube^3 recorders (produced by DiGOS Potsdam GmbH) and with miniSEED files written by PR6-24 or EDR-210 portable recorder (produced by Earth Data). However, there is absolutely no reason why the utilities should not work with miniSEED or Cube files produced by other instruments or software as well.

The programs will help you to manage your recorded data and to prepare the dataset for import into whatever processing software you use for further scientific analysis. You can use the utilities, among other things, to convert, re-organize and cut out (seismic) data from miniSEED and Cube data files.

The following is a summary of all utilities that are part of the GIPPtools collection as well as a one sentence synopsis of their intended use. Detailed information about each program can be found in the separate manual pages.

cube2ascii

Convert Cube files to various ASCII text format.

cube2mseed

Convert Cube files to miniSEED format.

cube2segy

Convert Cube files to SEG-Y format.

cubeevent

List events captured by the Cube "event recorder" hardware.

cubeinfo

Inspect and summarize the content of Cube recordings.

mseed2ascii

Convert miniSEED data to various ASCII text formats.

mseed2mseed

Modify header fields of miniSEED records.

mseed2pdas

Convert miniSEED data to PDAS format.

mseed2segy

Convert miniSEED files to SEG-Y format.

mseedcut

Cut out the miniSEED input into pieces.

mseedinfo

Inspect and summarize the content of miniSEED files.

mseedrecover

Recover miniSEED records from damaged files and disks.

mseedrename

Batch rename miniSEED files using a template.

Usage

All utilities of the GIPPtools collection are written in the Java programming language. Usually they are run from the command line by executing a little start script named after the respective GIPPtool utility. This script will start up the Java Runtime Environment (JRE) that is then used to execute the code of the requested tool.

Alternatively, you can also use the "universal" start script called **gipptools** and specify the desired utility via the `--run-tool` command line option. The **gipptools** start script supports the following options:

-h, --help

Print a brief help text and an up-to-date list of all GIPPtool utilities available. The help and utility list is also displayed if **gipptools** is called without command line arguments.

--version

　　Print the GIPPtools release information and exit.

--sysinfo

　　Provide some basic system information and exit.

--run-tool=*TOOL*

　　Select the *TOOL* from the list of GIPPtools utilities that you want to execute. All following command line options (displayed as *TOOLPARAM* in the synopsis above) will be forwarded to the selected utility. (Please see the examples section below.)

Examples

1. Executing the **gipptools** start script without any command line argument will display a brief help text including an up-to-date list of all GIPPtool utilities supported by this release.

```
gipptools
```

2. To obtain basic system information about the currently used GIPPtools installation use the `--sysinfo` command line option:

```
gipptools --sysinfo*
```

This will display information about the used GIPPtools release, the Java Runtime Environment (JRE), the operating system, the source of the leap seconds list as well as the values of selected environment variables.

3. Display the online help of the **cubeinfo** utility:

```
gipptools --run-tool=cubeinfo --help
```

This is equivalent to calling the start script for the cubeinfo utility directly:

```
cubeinfo --help
```

4. Use the **cubeinfo** utility to show a short summary of the content of the `input.cube` file.

```
gipptools --run-tool=cubeinfo --format=summary input.cube
```

All command line options after `--run-tool=cubeinfo` are directly passed on to the **cubeinfo** utility. This makes the above command functionally equivalent to calling:

```
cubeinfo --format=summary input.cube
```

Environment

The following environment variables can optionally be used to influence the behavior of the GIPPtool utilities.

GIPPTOOLS_HOME

This environment variable is used to find the location of the GIPPtools installation directory. In particular, the Java class files that make up the GIPPtools are expected to be in the `java` subdirectory of **GIPPTOOLS_HOME**.

GIPPTOOLS_JAVA

The utilities of the GIPPtools are written in the programming language Java and consequently need a Java Runtime Environment (JRE) to execute. Use this variable to specify the location of the JRE which should be used.

GIPPTOOLS_OPTS

You can use this environment variable for additional fine-tuning of the Java runtime environment. This is typically used to set the Java heap size available to GIPPtool programs.

GIPPTOOLS_LEAP

The GIPPtools require up-to-date leap second information to correctly interpret Cube files. Usually, this information is obtained from the `leap-seconds.list` file located in the config subdirectory of the GIPPtools installation directory. This environment variable can be used to provide a more up-to-date leap second list to GIPPtool programs.

It is usually not necessary to define any of those variables as suitable values should be selected automatically. However, if the automatic detection build into the start script fails, or you need to choose between different GIPPtool or Java runtime releases installed on your computer, these environment variables might become quite helpful to troubleshoot the situation.

Files

$GIPPTOOLS_HOME/bin

Directory containing the start scripts for the various GIPPtool utilities. Usually, the scripts are simply a copy of or a link to the central **gipptools** start script in the same directory.

$GIPPTOOLS_HOME/java

Contains the actual java classes that make up the GIPPtools.

$GIPPTOOLS_HOME/config

Default settings and other supporting files (e.g. leap-seconds.list).

$GIPPTOOLS_HOME/doc

    Documentation and example files.

$GIPPTOOLS_HOME/man

    Directory containing the GIPPtools manual page hierarchy.

See also

cube2ascii(1), **cube2mseed**(1), **cube2segy**(1), **cubeevent**(1), **cubeinfo**(1), **mseed2ascii**(1), **mseed2mseed**(1), **mseed2pdas**(1), **mseed2segy**(1), **mseedcut**(1), **mseedinfo**(1), **mseedrecover**(1), **mseedrename**(1)

Home page

http://www.gfz-potsdam.de/gipp

Contact

The GIPPtools are written and maintained by *Christof Lendl* (lendl@gfz-potsdam.de). If you are having trouble installing and/or running the software, feel free to e-mail me. However, please study the documentation **before** contacting me. Im not your ghostreader!

When reporting a bug, please be sure to include the GIPPtools release number and the computer platform you are using.

    All GIPPtool utilities have a build-in `--sysinfo` command line option that can be used to obtain this as well as other helpful information.