

# mseed2ascii(1)

## Name

mseed2ascii - convert miniSEED data to various ASCII text format

## Synopsis

```
mseed2ascii [-v | --verbose] [--include-pattern=PATTERN]...  
             [--select-station=STATION]... [--select-channel=CHANNEL]...  
             [--output-dir=DIRECTORY] [--force-overwrite] [--force-concat]]  
             [--format=FORMAT]  
             [file | directory]...
```

```
mseed2ascii [-h | --help] [--version] [--sysinfo]
```

## Description

**Mseed2ascii** reads from one or more *files* (or standard input) and converts each miniSEED record in the input to ASCII text format. If a *directory* is given instead, **mseed2ascii** searches recursively for input files inside the directory. The search can be restricted to contain only files with a name also matched by a pattern given via one or more `--include-pattern` options.

The conversion result is written to standard output (i.e. console) or saved in an output directory (use option `--output-dir`). Different predefined output variants (data and/or header, ...) are available can be selected using the `--format` option.

## Options

The program pretty much follows expected Unix command line syntax. Some command line options have two variants, one long and an additional short one (for convenience). These are shown below, separated by commas. However, most options only have a long variant. The '=' for options that take a parameter is required and can not be replaced by a whitespace.

### **-h, --help**

Print a brief summary of all available command line options and exit.

### **--version**

Print the **mseed2ascii** release information and exit.

### **--sysinfo**

Provide some basic system information and exit.

## **-v, --verbose**

This option increases the amount of information given to the user during program execution. By default, (i.e. without this option) **mseed2ascii** only reports warnings and errors. (See the diagnostics section below.)

## **--include-pattern=*PATTERN***

Only process files whose filename matches the given *PATTERN*. Files with a name not matching the search *PATTERN* will be ignored. This option is quite useful to speed up recursive searches through large subdirectory trees and can be used more than once in the same command line.

You can use the two wild card characters (*\** and *?*) when specifying a *PATTERN* (e.g. *\*.pri?*). Or alternatively, you can also use a predefined filter called **GIPP** that can be used to exclude all files not following the usual GIPP naming convention for miniSEED files recorded by [Earth Data](#) loggers (e.g. message logging or status files, see examples section below).



The search *PATTERN* is only applied to the filename part and not to the full pathname of a file.

## **--select-station=*STATION***

Only process miniSEED records with a matching *STATION* identifier. Records that do not match the *STATION* expression are ignored/skipped.

This command line option understands the two wild card character *\** and *?* and can be used more than once in a command line.

## **--select-channel=*CHANNEL***

Only process miniSEED records with a matching *CHANNEL* identifier. Records that do not match the *CHANNEL* expression are ignored/skipped.

This command line option understands the two wild card character *\** and *?* and can be used more than once in a command line.

## **--output-dir=*DIRECTORY***

Save the resulting ASCII text files to this *DIRECTORY*. The directory must already exist and be writable! Already existing files in that directory will not be overwritten unless the option **--force-overwrite** is used as well.

## **--force-overwrite**

If this option is used, already existing files in the output directory will be overwritten without mercy!

The default behavior however is **not** to overwrite already existing files. Instead, a new file is created with an additional number in between filename and extension.

## **--force-concat**

Concatenate the ASCII text output creating as few new files as possible. This means that a new output file is only started when there is a (data) discontinuity in the miniSEED input. Without discontinuity the converted data is simply appended to the currently used output file.

By default, however, a separate new output file is created for every single miniSEED input file.

### **--format=FORMAT**

Select one of the predefined output formats:

#### **HEADER**

Write out the miniSEED header information for each record.

#### **DATA**

Write only the sample values contained in the miniSEED records.

#### **ALL**

The combination of the HEADER and DATA format. (This is also the default output format.)

#### **GMT**

Write the sample values of each miniSEED record in a two-column ASCII format that can be directly used as input to the **psxy** command. (The **psxy** command is part of the Generic Mapping Tools, GMT.)

## 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.

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.

## Diagnostics

Occasionally, the **mseed2ascii** utility will produce user feedback. In general, user messages are classified as *INFO*, *WARNING* or *ERROR*. The *INFO* messages are only displayed when the **--verbose**

command line option is used. They usually report about the progress of the program run, give statistical information or write a final summary.

More important are *WARNING* messages. In general, they warn about (possible) issues that may influence the outcome. Although the program will continue with execution, you certainly should check the results carefully. You might not have gotten what you (thought you) asked for. Finally, *ERROR* messages inform about problems that can not be resolved automatically. Program execution usually stops and the user must fix the cause of the error first.

## Exit codes

Use the following program exit codes when calling **mseed2ascii** from scripts or other programs to see if **mseed2ascii** finished successfully. Any non-zero code indicates an *ERROR*!

**0**

Success.

**64**

Command line syntax or usage error.

**65**

Data format error. (The input was not a valid miniSEED file.)

**66**

An input file did not exist or was not readable.

**70**

Error in internal program logic.

**74**

I/O error.

**99**

Other, unspecified errors.

## Examples

1. To see an ASCII representation of a single file you could use one of the following variations:

```
mseed2ascii input.mseed
```

```
mseed2ascii input.mseed | more
```

```
cat input.mseed | mseed2ascii > output.txt
```

The first command will write header and data of the miniSEED records to the console. The second will use the pager program **more** to display the result. And the last variant will save the content of the input.mseed file to the output.txt file.

2. You have a multiplexed miniSEED file with logging information embedded in a special **LOG** recording channel. The logging data is stored as ASCII encoded content inside "normal" miniSEED records. To extract the logfile use e.g.:

```
mseed2ascii --format=DATA --select-channel=LOG input.mseed > logfile.txt
```

Here, output format DATA is selected so that information about the miniSEED record is not intermixed with the logging file. In addition, the **LOG** recording channel is picked for processing with the **--select-channel** option.

## Files

**\$GIPPTOOLS\_HOME/bin/mseed2ascii**

The **mseed2ascii** "program". Usually just a copy of or symbolic link pointing to the standard GIPTools start script.

**\$GIPPTOOLS\_HOME/bin/gipptools**

The GIPTools start script. Almost all utilities of the GIPTools package are started from this shell script.

## See also

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

## Bugs and caveats

None so far.