

# Reading MiniSEED data into SH/SHM

The MiniSEED format is a subformat of the commonly used SEED data format (get a description of the SEED data format from [?IRIS](#)). It is well suited for continuous data sets or for archiving large time spans of data. SH accesses MiniSEED data by start time and read length and reads only part of the file rather than accepting a filename and reading it completely (like in other formats e.g. GSE or Q). SEED and even MiniSEED is a very flexible data format and allows a great variety of subformat types. SH/SHM is not able to read general MiniSEED, several restrictions apply:

- the data must be coded in Steim1 or Steim2 compression
- the data must be organised in separate files for each data stream
- the data must not contain records other than data records (that's what I understand as MiniSEED)

The record length of the SEED records should be a power of 2. We work with 4096 and 512 byte records, other record lengths are supported as well. There are no naming conventions to be matched for such MiniSEED files. Data are read in by specifying start time, read length, station list, channel code and components. In order to find the appropriate data file(s) for reading SH needs a directory file which holds necessary information about location, filename and content of a set of MiniSEED files. This file is usually called *sdf*file.sfd and resides on the data directory or on any other location ('sfd' stands for 'seed file directory'). SH/SHM can read only data which are listed in such a file. Processing of online data in MiniSEED format with SH needs an update of the sdf file before. The SH package contains a program to create *sdf*file.sfd. Its syntax is

```
sfdlist <wild> <dataroot> [<outfile>] [<quiet>] [<reclth>] [<byteoff>]
```

where

<wild>

wildcard to match available data files in the data directory tree (e.g. "\*" or "\*.bh?"). Please mind the quotes.

<dataroot>

path to the root directory of the data files. It is recommended to specify here an absolute path (e.g. *\$PWD* instead of *./*), since otherwise SH cannot find the data files if it is called from another directory. Also the resulting sdf file cannot be moved to another location.

<outfile>

name of the output sdf file, usually sdf file.sfd (default)

<quiet>

if equal to quiet the processed files are not written to stdout

<reclth>

record length of SEED records, default is automatic detection

<byteoff>

byte offset in MiniSEED file, the first <byteoff> bytes in the file are ignored, default is automatic detection.

The *sfdlist* program searches the specified directory and all its subdirectories for data files matching the given wildcard. If too many data files exist in the data directory tree the program aborts with an error message. In this case use the (slower) program *sfdlist\_l*. The commands *sfdlist* and *sfdlist\_l* are aliases to the programs *\$SEED\_PROG/sfdlist.csh* and *\$SEED\_PROG/sfdlist\_l.csh*, respectively. After the sdf file.sfd has been created SH can read in the data files listed. The appropriate SH command (reads) for reading MiniSEED data needs the location of the sdf file.sfd. Because of its VAX/VMS origin, SH does not distinguish between upper and

lowercase by default and has its own interpretation of slashes / in the command line. Therefore it is a good idea to define an environment variable pointing to your sfdfile.sfd before using SH. This makes it a easier to specify the required pathname within SH. E.g. if your sfdfile path is /home/gonzo/data/sfdfile.sfd use

```
setenv DATA /home/gonzo/data
```

to define the environment variable *DATA*. Please mind the uppercase variable name and do not include the sfdfile.sfd itself to the definition. From within SH you may read in MiniSEED data now using the command

```
reads <sfdpath> <start-time> <seclth> <stations> <comp> <chan>
```

where

- <sfdpath>  
path to the sfdfile.sfd used. If a environment variable has been defined for this, give this name and append a colon : to it (e.g. data:). As mentioned before SH is not case sensitive (all characters are read as uppercase).
- <start-time>  
start time for reading in SH time format
- <seclth>  
number of seconds to read in
- <stations>  
station list. Separate the names by commas if more than one station. Do not use blanks between stations (e.g. bfo or bfo,brg,wet).
- <comp>  
component list. One or more component can be specified. Just concatenate the component names or separate with commas (e.g. z or zne or z,n,e)
- <chan>  
channel to read (e.g. bh)

SHM has no command line, all parameters are controlled by the Read? dialog box. The <sfdpath> parameter is specified in the textfield in the lower right corner. The colon after the environment variable name may be omitted here. All other settings are done by mouse clicks and should be self-explaining.

Complete SEED volumes may be read in using the program \$SEED\_PROG/seed2sfd, provided in recent SH/SHM distributions. It creates an sfdfile.sfd appropriate to read in data directly from the SEED volume (encoded in Steim1 or Steim2). Older SH/SHM versions need to split up SEED volumes into MiniSEED files using the program \$SEED\_PROG/split\_seed <seedvolume>. The resulting miniseed\_\* files are MiniSEED formatted and can be read in as described above. The split\_seed program can be used also for multiplexed MiniSEED files.

SH/SHM read in the data from MiniSEED which is usually given in counts. To have access to true amplitudes you need to apply a calibration gain multiplied to the sample counts. From version SHM-2.4 on this information is provided in a sensitivity file specified in the global setup file (by default this is \$SH\_INPUTS/sensitivities.txt). For details about this file see the syntax of the sensitivity file?.