SEIS-UK

Data Management Procedures

Version 1.26

Alex Brisbourne & Anna Horleston

May 2007

E-mail: seis-uk@le.ac.uk
http://www.le.ac.uk/seis-uk/

seis-uk@le.ac.uk
www.le.ac.uk/seis-uk/

SEIS-UK Guralp Instrumentation – Data Management

1 Introduction to the SEIS-UK data management system

The minimum deliverable at the end of a fieldwork visit is two copies of the raw GCF format data on DLT or DAT tape, i.e. the output from gcfconv or dfdextract as a single gcf format file for each station for each download. However, it is possible to fulfil all the data processing described in this document on any of the field Sun workstations. It is assumed that data QC has already been carried out following the SEIS-UK guidelines.

The data processing workstation in Leicester (lochside.geol.le.ac.uk) can be accessed by external users via secure-shell packages (ssh/PuTTY) once users have been assigned usernames and granted remote access permission (users' I.P. addresses required).

The following list outlines the step-by-step process of data archive and event extraction. Steps 1 to 3 reformat the data such that it is ready for permanent archive at a data centre. The latter steps are for extracting events from known event lists, scanning data for local events, and also location of local events.

1. gcfconv – convert from GCF format to miniseed format
2. Move data to storage point
3. Generate dataless seed volume
4. weed – summary file production for event extraction
5. rdseed – teleseismic event extraction
6. datascan_2 – scanning data for local events with an STA/LTA trigger algorithm
7. SHM – event location and magnitude calculation

A number of bugs currently exist in the routines used. In these cases, work-around methods are presented.

A standard file and directory structure is presented here for which scripts are provided. Any deviation from this standard structure and any subsequent script updating required is the responsibility of the user.

1.1 Acquiring a Network Code

Prior to any data conversion all experiments require a unique network code which is assigned by FDSN (passive arrays) or IRIS (controlled source). It is the responsibility of the PI to request the codes from:
Passive: http://www.fdsn.org/getcode.html
Controlled source: http://www.iris.washington.edu/data/assembled_ID.htm.

To assist the data archiving process the PI should also complete the mobilisation form at:
http://www.iris.washington.edu/stations/mob.htm (Data shipment - usually 6 months after end of experiment).
At the end of the experiment, the PI should complete the demobilisation form at:
http://www.iris.washington.edu/stations/demob.htm

1.2 Use of the SEIS-UK data management server

Each project will be provided with a unique username and password. The user must provide the IP address of the machine you will be working from, where you will need ssh installed. It is the responsibility of the user to negotiate the security or firewall at your institution (contact your systems manager). SEIS-UK will endeavour to provide sufficient disk space for the data conversion. However, users must consider the amount of disk space available and ensure that they do not fill disk partitions (use the "df –k" command during conversion to miniseed).

2 Conversion from GCF to miniseed

The following procedures are identical for all data from SEIS-UK Guralp instruments extracted directly from the SAM or DFD disk with either gcfdisk or dfdextract respectively.

2.1 Directory Structure

Each project will be assigned data directories by SEIS-UK along the following lines:

Home directory (scripts/nft files/weed etc):	`/home/username`
GCF directory (raw data):	`/gcf[1-3]/project_name`

MSEED/SEGY directory (output from gcfconv): `/archive[1-8]/project_name`
Working directory (SAC event files etc): `/data[1-2]/project_name`
Archive working directory (for completed datasets): `/data1/ARCHIVE/project_name`

- Users are requested to abide by this directory structure.
- It is prudent to work on a single service-run of data at a time in the gcf directory (see description of the NFT file below):
  - `mkdir /gcf1/project_name/SERVICE_1`
- The disk images from the SCSI disk or the lacie disk should be read from the tapes to, for example, /gcf1/project_name/SERVICE_1.
- All scripts etc generated by the user should be installed in the users' home directory, not on the data partitions. Only miniseed data on the data partitions will backed up so any non-data files on the data partitions will not be recoverable in the event of disk failure. Home partitions will be backed up on a weekly basis.
- By default, SEIS-UK computers use the k-shell environment.

2.2 Conversion to miniseed - gcfconv

In order that the miniseed data headers are fully populated a text file termed the "NFT file", linking sensor serial number to site name is required. This file must be correct for the download currently being converted, and therefore will require updating in cases of instrument swaps. More information is available from the gcfconv man page. File format is as follows (USE AN EXACT COPY OF THIS FORMAT!). For example,

```
START YO

 SITE       SENSOR       DM     POSITION LOCATION

 ALBT       CMG3T/6051   DM24/6051
 BAYT       CMG3T/6037   DM24/6037
 CAPT       CMG3T/3A18   DM24/3A18
 WITT       CMG40T/4A94  DM24/4A94
 YORT       CMG3T/3A21   DM24/3A21

END
```

- For station names use a maximum of 5 alphanumeric characters. If this relates to the geographical location then this can be helpful when working with the data. However, for large number of stations it is easiest to use numerical sequences, generally with a prefix related to the experiment name. For example, for an experiment in Chixculub it may be helpful to use the site IDs CX001, CX002 and so on. Do not use 1, 2, 3, 4 ... 10 as this only leads to confusion (it helps to be consistent with the number of characters).
- YO is the network code
- Space or tab delimited entries
- 6TD data – the routine gcfconv does not currently accommodate 6TD instruments in the NFT file, therefore use CMG3T instead of CMG6T in the NFT file. This will not affect the data output.
- Position and location are not used here but must be in the header line as in this example.
- The NFT file should be kept in the users' home directory and checked carefully before each service conversion as it is these data which are populate the miniseed headers.

Data conversion can be done manually by following the steps outlined below or by running the script lacie2mseed.csh, which works on a lacie directory. This concatenates all the 'dd' chunks into one file, runs dfdextract and then gcfconv to produce the daylong output directory structure outlined below. This is run with:

```
lacie2mseed.csh <lacie.dir> <nft.file> <service_no> <lacie_no>
Eg. lacie2mseed.csh /gcf2/ITALY04/SERVICE_1/LACIE_2051
      /home/italy04/italy_serv_1.nft 1 2051
```

This will also detect any instruments that have defaulted to GURALP CMG6 and will save their dfdextract output files to a directory with the service name and lacie disk number – these must be processed separately – see section 2.21 below.

To run the data conversion manually, an environment variable must point to the NFT file:

ksh: export GURALP_NFT=/home/fieldusr/nft_filename.nft
csh: setenv GURALP_NFT /home/fieldusr/nft_filename.nft

The routine gcfconv, with the following flags, is then used to form hour-long Steim-1 format miniseed data files.

```
cd /archive?/project_name/MSEED/SERVICE_1
gcfconv –i –m –l 3600 –F /archive?/proj_name/GCF/SERVICE_1/file.data
```

If a number of gcf format files exist in the current directory then the routine can be run on each file in turn without user interaction, using the C-shell routine "foreach", for example,

```
cd /archive?/proj_name/MSEED/SERVICE_1
csh
setenv GURALP_NFT /home/username/proj_name.nft
foreach gcf_file(/archive?/proj_name/GCF/SERVICE_1/*.gcf)
      gcfconv –i –m –l 3600 –F $gcf_file
end
# use "exit" to return to K-Shell
# N.B. For 6TDs use (*_??.dd) instead of (*.gcf)
```

- Day directories of the form `G2002.211` referring to the year and julian day are created in the working directory.
- Each day directory contains hour-length miniseed format data files for each stream recorded.
- Hour-long data files are not truncated at day-boundaries, but relative to the power-up time of the instrument. There may therefore be up to one hour of data per component from one day directory in the previous day directory.
- Status stream data are written to text files, creating a single ".log" file per gcf file, in the working directory. This file is named according to the start time of the gcfconv process.

The success of the conversion can be ascertained by listing the data from one stream of each instrument in turn to check that all gcf format have been converted, e.g.:

```
cd /archive?/project_name/MSEED/SERVICE_1
ls G200?.???/*3A18*Z2* | more
```

- 24 files are expected per component per day – any deviation from this may indicate a problem with the data.
- Hundreds of files result from gaps in the data or the presence of bad blocks.
- At this stage, overlaps in the data must be removed e.g. resulting from multiple reads of SAM disks or the use of flushall with the DFD.

2.2.1 Potential problems with DFD and 6TD data

A 6TD can potentially reset to factory default settings (Sensor Serial: CMG600 / ID: GURALP / 100sps / Continuous GPS). These data may still be used but will require slightly more file and header manipulation. In such cases, users should contact SEIS-UK directly for assistance in this process.

These gcf files must be moved to an empty directory (one directory per file) and converted as normal with gcfconv. The script "rename_cmg6.csh" is then used to update the miniseed headers:

```
gcfconv –i -m –l 3600 –F CMG6.file
rename_cmg6.csh <serial-no> <Site-ID> <Net-code>
```

Data must then be moved to the directories containing the rest of the data.

2.2.2 Julian Day - Calendar Day Conversion

To convert from calendar day to Julian day, for example 17th July 2001:

```
julday 07 17 2001              # Month Day Year
```

And from Julian Day 198 of 2001 to calendar day:

```
calday 198 2001                # Julday Year
```

2.3 Renaming log files and extracting lat-lon data

Output from gcfconv: By examining the first few lines of the log file (`head filename.log`) one can derive the instrument serial number, and hence the station name. Log files can then be manually renamed and moved to a separate directory:

```
mkdir /archive?/project_name/LOGS/SERVICE_1
mv filename.log LOGS/STAT_INSTRUMENT_SERVICE_1.log
```

Output from Gusdav: Log files from Gusdav are labelled with the relevant serial number, therefore renaming is not necessary.

2.3.1 Automated log file renaming (gcfconv output only)

Alternatively, you can use the script:

```
rename_logfiles.csh <nft-file> <service-#>
e.g. rename_logfiles.csh ~/myproj_serv_4.nft 4
```

This will rename log files to something like <STAT_SERIAL_SERV_#.log> and put them into a "LOG" directory. This script is designed for log files output by *gcfconv* only.

2.3.2 Extracting average lat-lon data from log files

Average Lat-Lon data can be extracted from log files with one of the following scripts:

If you have used gcfconv (log files end in .log):

```
latlon_from_log_6td.csh 2005.01
latlon_from_log_sam.csh 2006.31
latlon_from_log_dcm.csh
```

If you have used Gusdav (log files end in .txt):

```
latlon_from_gusdav.csh
```

Simply move to the directory where the log files reside and run the relevant script. The first two scripts require the user to define a 10 day window, e.g. "2005.01" refers to julian days 2005.010 to 2005.019. The script then calculates an average of the lat-lon data within that window. This allows log files to be used which have, for example, huddle test data at the beginning. The DCM log files do not require this as they are much shorter and the user can select which data to use manually.

2.4 Miniseed headers

The miniseed header contains the minimum amount of data: Time / Station / Network / Channel.

2.4.1 Listing miniseed headers - mseedhdr

mseedhdr is used to list miniseed data header variables:

```
mseedhdr filename.m | more
```

2.4.2 Updating miniseed headers - mseedmod / qmerge

mseedmod can be used to change miniseed data headers. For example, to change the Network code to YY and station name to S02 for all 3A18 data files in all day directories:

```
cd /archive?/project_name/SERVICE_1
foreach directory(G200?.???)
      cd $directory
      mseedmod –q –N YY –S “S02  “ *3A18*m
      cd ..
end
```

- The –q flag runs quiet mode.
- Miniseed station names are 5 characters long, left-justified and padded with spaces.
- See man pages of mseedhdr and mseedmod for further details.

If mseedmod does not change the header values, use the routine qmerge (use qmerge –h for more information and an explanation of the flags required).

To change site ID to SO2 and network to YY in the file “infile”:

qmerge –o outfile –S SO2 –N YY infile

2.4.3 Miniseed file duration - mseed_start_stop

Lists start-stop times of miniseed data files:

```
mseed_start_stop example_3A18.m
```

2.4.4 Further miniseed header tools - seedsniff

The utility “seedsniff” from IRIS can be used to access the more detailed 48byte fixed section of the data record header (run seedsniff without arguments for details). If a timing correction has been applied to a miniseed file it will be reported here in units of 0.0001second. For example, a 1 second correction has been applied to the following miniseed data file.

```
seedsniff 4096 32768 4 < GCF.2005:293:15:44:42.3433Z2.m | more

================================================================================
STATION  LOCATION  CHANNEL  NETWORK  TIME

VAULT               HHZ      ZZ       2006,134,23:00:01.0000   [0]
         # samples in record: 1886
                 sample_rate: 200
                  multiplier: 1
              activity flags: (1) Time Correction Applied
         I/O and clock flags:
          data quality flags:
             # of blockettes: 1
             time correction: 10000
           begin data offset: 64
        begin blkette offset: 48
        BLOCKETTE 1000:
                    encoding format: STEIM 1 Compression (val:10)
                         word order: 68000/SPARC word order
                  data record length: 12
                            reserved: 0
```

2.4.5 Concatenating miniseed data

It is sometimes useful to form single network-day volumes of miniseed data, i.e. the data from a single day from all channels and all stations in one data file. All miniseed data (all components / all stations) are concatenated to a single network-day file using standard unix commands. For example, to form a single station-day file for the vertical component of sensor 3A18 on day 313 of 2002:

```
cd /archive?/project_name/SERVICE_1
qmerge –o outfile.msd –a G2002.313/*3A18Z2*.m
```

N.B. qmerge may only merge a single component at one time. A message "Warning: unable to open leapsecond file" can be ignored. The unix command "cat" can be used to merge different components, for example:

```
cat G2002.313/*3A18*.m > outfile.msd
```

2.4.6 Demultiplexing miniseed data

Multiplexed (data coincident in time from more than one channel in a single file) miniseed data can be demultiplexed (one channel per file) with the utility sdrsplit:

```
sdrsplit <infile>
```

Use the –P flag to output Passcal format filenames of the format 06.229.23.00.00.XX.ESK.B0N.00.

2.5 Channel Identifiers in SEED data

Channel Identifiers of the velocity outputs, e.g. BHZ / MHE, are set according to a SEED standard format. The following information is taken from the SEED manual of IRIS. The values most commonly used in SEIS-UK data streams are highlighted in bold.

2.5.1 Band Code

The first letter specifies the general sampling rate and the response band of the instrument. (The "A" code is reserved for administrative functions such as miscellaneous state of health.)

Band code	Band type	Sample rate (Hz)	Corner period (sec)
E	Extremely Short Period	>= 80	< 10 sec
S	Short Period	>= 10 to < 80	< 10 sec
H	**High Broad Band**	**>= 80**	**>= 10 sec**
B	**Broad Band**	**>= 10 to < 80**	**>= 10 sec**
M	**Mid Period**	**> 1 to < 10**	
L	**Long Period**	**= 1**	
V	Very Long Period	= 0.1	
U	Ultra Long Period	= 0.01	
R	Extremely Long Period	= 0.001	
A	Administrative		
W	Weather/Environmental		
X	Experimental		

2.5.2 Instrument Code and Orientation Code

The second letter specifies the family to which the sensor belongs. The third letter specifies the physical configuration of the members of a multiple axis instrument package or other parameters as specified for each instrument.

Code	Instrument
H	**High Gain Seismometer**
L	Low Gain Seismometer
G	Gravimeter

M	Mass Position Seismometer
N*	Accelerometer

* *Historically, some channels from accelerometers have used instrument codes of L and G. The use of N is the FDSN convention as defined in August, 2000.*

2.6 Verifying data conversion and data continuity

The continuity of miniseed data volumes can be verified by substituting into the following example:

1. Create a goat sync file by running the following command from your home directory:
   `make_goat_file.csh <mseed_base_dir> <net-code> <outfile-prefix>`
   `e.g. make_goat_file.csh /archive6/CBP05/6T/SERV_1 YG Serbia_Serv1`
   Or, for Taurus data use: `make_goat_file_taurus.csh`
2. Then in a web browser (firefox) go to:
   http://www.iris.washington.edu/SeismiQuery/goat/goatstart.pl
3. Browse-for and then upload the relevant goat output file from ~/GOAT:
   `e.g. ~/GOAT/YG_Serbia_Serv1_goatsync.txt`
4. Select the uploaded sync file (the first few lines of the file will appear on the right-hand side. Then select "GOAT Display".
5. Specify the channels and time interval required in julian days, then select "Show Graph".
6. Gaps in the data are highlighted in red, overlaps in blue. Continuous data are green.
7. Multiple sync files can be uploaded and will be removed automatically after a period.

3 Dataless seed volume production with make_dlsv

Dataless seed volumes are files that contain all the meta-data for the experiment. These files are used in parallel with the miniseed volumes which are the raw seismic data with very little header information.

The user must generate two text files, one of the station information and one describing the network information. It is recommended that the user create the dataless volume in a directory /home/username/dataless.

FILE 1: /home/username/dataless/network.txt

```
net_abbr net_code net_name
```

for example,

```
EAGLEBB XX EAGLE_Broadband_Array
```

The three items required are (space separated, use "underscore" to indicate a space in a string):

1. Network Abbreviation
2. IRIS Network Code
3. Full Network Description

FILE 2: /home/usrname/dataless/stations.txt

```
STAT_ID SERIAL_NO LAT LON ELEV_M START_DATE STOP_DATE SPS1 SPS2 SITE_NAME
```

for example,

```
LEME 3A13 9.87652 39.0727 1024.0 2001:030:10:30:00 2003:088:18:05:00 100 0 Lemon
E32 6123 9.202 38.888 204.8 2002:364:00:00:00 2004:001:23:59:59 50 5 High_School
```

Required items for each station are:

1. STAT_ID – Station code (maximum 5 characters)
2. SERIAL_NO – 4 character sensor serial number e.g. 6110, 4A99, 3513
3. LAT – Latitude (decimal)
4. LON – Longitude (decimal)
5. ELEV_M – Elevation (meters)
6. START_DATA – Installation date and time (year:julday:hour:min:sec)

7. STOP_DATE – Pull-out date and time (year:julday:hour:min:sec)
8. SPS1 – Primary sample rate (e.g. 100sps)
9. SPS2 – Secondary sample rate
    - e.g. E32 in the above example - 5sps.
    - use 0 if only archiving primary e.g. LEME in the above example.
10. SITE_NAME – Full site name (use underscore to denote spaces)
    - When setting the start/stop date and time, generally a start time of 00:00:00 can be used, and an end time of 23:59:59 – to encompass all available data. However, where a station swap occurs in a single day (pull-out and reinstallation) care must be taken to use the correct times (GMT) such that data from each station are described uniquely by the dataless volume.

Once the two input files have been generated, the dataless volume can be created in two simple steps:

```
prepare_dlsv.csh network.txt stations.txt
```

This script will verify the format of the station.txt file and generate a template file called **copy_seed.cfg** which is required by make_dlsv. At this stage, volume comments can be added to the dataless by editing the "Comment" lines in the file copy_seed.cfg (see notes in this file for further explanation). These comments may describe data with poor timing or orientation for example. Generally, no comments are required and this step can be ignored.

Once the file **copy_seed.cfg** is ready, the dataless volume can be produced as follows:

```
export SEED_STUFF_HOME=.
make_dlsv
```

This will generate a dataless seed volume called "dataless.seed" which can be verified with the routine "verseed" or "rdseed" (see respective man pages), e.g.

```
verseed –4 dataless.seed
```

Data are now ready to archive at IRIS. The data transfer will be carried out by SEIS-UK. However, to ensure that this process runs as smoothly as possible:

- The user should ensure that all information in the dataless seed volume is correct and encompasses all available data.
- Once all service runs have been converted, they should be merged into a single data directory containing all the day directories.
- All non-array data should be moved from the data directory or deleted, e.g. huddle test data, corrupt data.
- Ensure that all station moves or swaps are accounted for in the dataless volume.
- Verify that all miniseed headers are correct using the script "qc_mseed4iris.csh":
    - `cd /archive?/project_name/MSEED`
    - `qc_mseed4iris.csh ~/dataless/stations.txt`
        - Most errors in the data headers or dataless will be flagged and written to a text file named "~/qc_mseed4iris.out"
    - The script "mseed4iris.csh" will not account for station swaps in a single day – the miniseed headers of these data must be verified by hand.

4 Extracting teleseismic event data with weed and rdseed

Two packages from IRIS are used to extract seismic events from pre-existing event lists. "rdseed" is used to extract the data from minseed data volumes, used in conjunction with the dataless seed volume. This package may be run interactively at the command line to extract data windows, or with summary files to extract around known events. "weed" can be used to generate a summary file for use with rdseed. Weed calculates phase-arrival windows from iasp91 traveltime tables. If a summary file is used, the output format data headers are fully populated with the station and event information.

4.1 Generating a summary file from a pre-existing event list

weed is used to generate summary files which allow the extraction of windows of data from seed volumes around selected phases, according to the iaspei traveltime tables. The weed manual or man page should be used in conjunction with the following notes.

4.1.1 Event lists and Station files

An up-to-date event file is required in weed format. Weed format lists can be obtained from the IRIS website (www.iris.washington.edu/SeismiQuery/events.htm). The "`preferred list`" option should be used to obtain the most up to date information for each event, and to prevent any repetition of events in the list. Event files must be called "`_____.events`".

N.B. The SeismiQuery website may not supply the full event list. Comparison should be made with the NEIC/USGS list in case any events are missing. If the NEIC list is used, reformatting of the NEIC PDE event list will be required using the script "pde2weed.csh" (N.B. Event times must be in the format: 00:00:00.0000 – note the 4 required characters after the decimal point).

Alternative arrangements may be required for local event lists and awk scripts may be required to reformat local lists to weed format.

A station file is also required which describes the network, of the format:

STN NET LAT LON ELEV STAT_NAME COMPS START STOP

For example:

STA1 YO -39.1503 +174.1256 +210.0 "Station 1" "HHE HHN HHZ" 2001,052,00:00:00 002,274,00:00:00
LEME YO -39.1503 +174.1256 +210.0 "Station 2" "BHE BHN BHZ LHE LHN LHZ" 2001,174,00:00:00 2002,331,22:15:00
E32 YO -39.2499 +174.8039 +120.0 "Station 3" "HHE HHN HHZ" 2001,157,00:00:00 2002,274,00:00:00

N.B. A station file (rdseed.stations) can be extracted from the dataless seed volume using the "S" option in rdseed:

```
export ALT_RESPONSE_FILE=/home/username/dataless/dataless.seed
rdseed –S –f <any-miniseed-file>
```

This will export a "rdseed.stations" file. A global substitution must be made to replace all double spaces with single spaces – for example, in the vi editor use the following command:

```
vi my_array.stations
:1,$s/  / /g
```

- repeat the substitution line until no more substitutions.

To create a summary file which can be used by rdseed to extract windowed data:

1. Launch weed using the script:
   myweed
   - this will create a working directory ~/WEED and create a symbolic link to the relevant iaspei tables.
2. Select Station file
3. Select Event file(s)
4. Create and select a Data Window Definition file
   a. Use the "Create/Edit Phases" window to produce a .phases file for windowing the data.
   b. Select which phases around which the data will be cut.
   c. The bias refers to how many seconds before (-ve values), or how many seconds after (+ve values), the predicted arrival the data should be cut.
   d. Channels should be set to ??? to extract all those available, - click "Add to list"
5. Set Extraction Parameters

```
        a. Event lists may be filtered for magnitude / location etc
  6. Click "Make Summary file"
        a. Give the file a useful name - weed automatically attaches
           ".summary".
        b. The summary file is generated in the directory indicated by
           "Dir"
```

This summary file may then be used with rdseed to extract the desired event files. At present, rdseed cannot be run directly from the weed gui.

4.1.2 ".phases" files (Data Window Definition)

On certain screen sizes, due to window resizing problems, the ".phases" files cannot be built using the weed gui. Existing files must be copied and edited manually (contact SEIS-UK for assistance if the DWD gui is not completely visible). The following format is required:

`weed_version 2.80`	-	weed version no.
`2: First_P`	-	Start Phase
`-30`	-	bias to Start Phase (sec)
`0`	-	(Surface wave velocity if selected)
`1`	-	SW arrival index
`1:Surface_Wave`	-	End phase
`600`	-	Bias to End Phase
`2`	-	Surface Wave Velocity
`1`	-	SW arrival index
`1`	-	Number of components selected
`???`	-	Components selected (wild cards ?)
`/home/username/WEED`	-	weed_home

The example above will truncate the data 30 seconds before the predicted first P-wave arrival, and 600 seconds after the first arriving 2km/s surface wave arrival. The phase list is in the order it appears in the weed gui. By not selecting a start phase, the event time will be used to cut the data.

4.2 Extracting data from miniseed volumes

rdseed is a Unix utility for managing, extracting and converting seed format data. rdseed may be used in conjunction with network-day miniseed data and dataless seed volumes to produce event files in a range of seismic data formats e.g. SAC, AH, SEGY. Rdseed may be run from the command line to extract any window of data to the desired user format.

Rdseed will read only one miniseed file at a time. Therefore to extract data from a number of stations at a time, all relevant miniseed data must be concatenated to a single file prior to extraction with rdseed. The miniseed format allows this to be carried out with the unix "cat" command. For example, to form a single file called temp_2001_321.msd of all the vertical component tap-2 data for day 321 of 2001 use:

```
cat G2001.321/*Z2* > ./temp_2001_321.msd
```

This will produce large files which are duplicates of existing data. These temporary files should therefore be removed when no longer in use. If the temporary file volume exceeds around 2Gbyte in size the routine will fail due to insufficient memory.

When running rdseed the location of the dataless seed volume (i.e. when running with miniseed data), is defined by the `ALT_RESPONSE_FILE` environment variable. To set this, use for example:

```
export ALT_RESPONSE_FILE=/home/username/dataless/dataless.seed
```

RESP files output by rdseed represent displacement responses, with the sensor and digitiser gains combined. If the poles and zeros in the dataless volume are in radians and not Hz (check your dataless volume carefully), these files are in omega format (as required by SAC) and can be used directly to convert from velocity to normalised displacement using the "transfer" function in SAC.

4.3 Extracting fully populated event files

In order to populate all the headers of rdseed output files with the event information a summary file from weed must be referred to when extracting the event data.

Rdseed cannot run simultaneously with summary files and miniseed data and a dataless seed volume. To work around this problem, a script is run which outputs full network-day SEED volumes prior to running rdseed for event extraction.

4.3.1 extract_events_weed.csh

The script "extract_events_weed.csh" should be run in an empty directory where an event-based directory structure will be produced. Separate SAC files will be produced for each component. Alternatively, if the user requires one seed volume per event then replace the "sac" flag with "seed" (these seed volumes do not include event data in the headers).

An environment variable "MSD4RDSEED" must point to the data directory where all day directories (G20??.???) reside e.g.

```
export MSD4RDSEED=/archive1/project_name/MSEED

extract_events_weed.csh <project.summary> <project.dataless> sac
```

or, for seed output

```
extract_events_weed.csh <project.summary> <project.dataless> seed
```

The summary file and dataless file arguments must be the full path, or copies/symbolic links must exist in the working directory, e.g. to create a symbolic link to the dataless:

```
ln –s /home/username/dataless/dataless.seed ./project.dataless
```

The script will work through the summary file, one event at a time, forming full seed volumes around the event time using qmerge, and then extracting the fully populated event files with rdseed.

4.3.2 Update SAC data headers for external codes

The script `insert_sac_headers.csh` will insert further headers into the SAC data which are required by other seismic analysis codes, such as the VanDecar cross-correlation code.

```
insert_sac_headers.csh project_name.summary
```

- This script uses the directory name to derive the event time, and then using this time scans the summary file for the event data which it then writes to the SAC header.

5 Trigger Detection and Event Location with Continuous Data

A routine called `datascan_2` is provided for running miniseed data through an STA/LTA picking routine. This software runs in conjunction with the Seismic Handler software (shm) and has been adapted from the routines of Klaus Stammler of SZGRF.

5.1 Reading miniseed data with Seismic Handler

The motif version of Seismic Handler, SHM, is used for processing of miniseed data and local event detection. Also, filter files for the datascan_2 routine are generated with SHM. It is therefore useful at this stage for the user to familiarise themselves with the software.

An introductory manual is available at: http://www.szgrf.bgr.de/sh-doc/index.html.

5.1.1 Set up of Seismic Handler for new networks

- A number of files must be updated for any new networks to be analysed. Existing files should be edited to ensure the correct file format is used.
- All newly generated files must have global read permissions.
- These sh files are being used by other users so please do not change things which will affect others i.e. append your information to existing files (files STATINF.DAT and filter_lookup.txt), or add your files to the end of the lists (STATLIST files).
- If in doubt contact SEIS-UK for advice.

Station locations – describes station lat/lon/elev (append your station info. to the end of the file).
Edit file: /usr/local/sh/inputs/STATINF.DAT

Example:
```
ADEE   lat:+7.7909 lon:+39.9068 elevation:2485
```

Filter Lookup Table – describes every station component (append your station info. to the end of the file)
Edit file: /usr/local/sh/inputs/filter_lookup.txt

Example:
```
ADEE-BH-Z SEISUK_4A94_E
ADEE-BH-N SEISUK_4A94_N
ADEE-BH-E SEISUK_4A94_Z
SHEE-HH-N SEISUK_3A13_N
BUTE-HH-E SEISUK_6111_Z
```

Network Description – lists stations assigned to an array (30 max per file). Start your STATLIST file links after any existing user links (do not delete existing links unless told otherwise be SEIS-UK).
Edit files:
```
/usr/local/sh/globals/STATLIST_DEFAULT.STX
/usr/local/sh/globals/STATLIST_01.STX
/usr/local/sh/globals/STATLIST_02.STX … and so on …
```

Example:
```
set1: 0 29 EAGLE
set2: 14 18 Picking

*LEME
*KOTE
*BUTE
… and so on …
```

These .STX files are symbolic links to the actual network files. The user must therefore edit their own network description files and point these symbolic links to the relevant file.

To point the first .STX file to the file `project_name_1.STX` which has already been created:

```
cd /usr/local/sh/globals
rm STATLIST_DEFAULT.STX
ln –s project_name_1.STX STATLIST_DEFAULT.STX
```

Network description files contain up to 30 stations which may be examined at any one time in a SHM window. When the “Read” option is selected in SHM, a gui appears listing the stations in STATLIST_DEFAULT.STX. Selecting the NEXT button in the read gui will move the station list on to the contents of STATLIST_01.STX and so on. If a network contains more than 30 stations, more than one .STX file will be required. Any new .STX files created by the users must be given global read permission (-rw-r--r--).

The “set” strings refer to buttons in the read window which allow rapid selection of stations. In this case “set 1” refers to all stations (use the exact number of stations) and “set 2” is used to quickly select the 5 best stations for picking phases (14-18)

5.1.2 Installing Seismic Handler on external Unix workstations

If a user wishes to use SEIS-UK data with SHM on their own system they are free to install SHM and LocSAT which are freely available from: http://www.szgrf.bgr.de/. For event magnitude analysis they will also need to install copies of the following Wood-Anderson filter files from a SEIS-UK workstation:

```
/usr/local/sh/filter/SEISUK*_S+WOODAND.FLF
```

5.1.3 Starting SHM and setting user preferences

SHM is launched at the command line with:

```
shm &
```

The users' preferences are created initially by running `Specials/Configure` in SHM. The miniseed data path (Item 13) MUST be set to "PWD". All other preferences are set in the same manner e.g. Filter defaults / LocSAT defaults. As the user becomes more familiar with SHM, more of these may be set to the commonly used values. At this stage there is no need to update any other preferences.

Upon exit from the Configure window (q) a new start-up file is written to:

```
~/sh/private/SHM_USER_STARTUP.SHC
```

SHM must be restarted to initiate the miniseed data path variable. This file sets the start up parameters of SHM for all subsequent use.

5.1.4 Reading miniseed data

In order that miniseed data can be read into SHM a file called `sfdfile.sfd` must be created in the working directory. This file is simply a text file which lists the path to the miniseed files and the header values. A script (`form_sfd.csh`) is used to create `sfdfile.sfd`. This script accounts for: (1) Data not being truncated at exactly the day boundary (gcfconv), and (2) the desire to read data from the following day when an event occurs just before midnight.

1. `form_sfd.csh <wild-card> <data-directory-path> <dir1> <dir2> <dir3>`

```
e.g.form_sfd.csh    *.m    /archive1/PROJ/MSEED    G2001.101    G2001.102
   G2001.103
```

   a. Produces the text file sfdfile.sfd which points to the data files and lists the headers.
   b. Always use the original miniseed files rather than network-day files.
   c. Examples of use the of `<wild-card>` are:
      i. Read all data - `*.m`
      ii. Only the vertical components - `*Z*.m`
      iii. All velocity channels from tap 0 - `*0.m`
   d. The `<data-directory-path>` must specify the full path (not relative) to the base directory of the data (i.e. the directory which contains the G2???.??? directories).
   e. Symbolic links should be used if `<data-directory-path>` is more than 40 characters long, otherwise the filenames will be truncated in the sfdfile.sfd
      i. After creation the sfdfile.sfd should be examined for any truncation of the filename due to a long directory path (e.g. "Z2.m" missing off the end of the filename)

2. Launch SHM

3. Select File-Read (shortcut keystroke "r") to launch the Read gui

4. Select data to be read
   a. Select individual stations or "Sets"
      i. "Next" scrolls to the next available .STX file
   b. Select component(s) Z, N or E

c. Select Channel (BH / HH / MH …)
    i. Use "ed" and enter the channel name in the box if it is not already listed, e.g. LH
d. Select date and time of interest
e. Select Length of data window
    i. Use the "drag bar" or the enter value in the box.
    ii. Enter the value manually to select more than 60 minutes
f. The miniseed data path box (points to the sfdfile.sfd location) in the bottom right-hand corner should be set to "PWD"
g. Click "Read New" (if first read since launch or cancellation of parameters)
    i. Alternatively, click "Read Again"

5. The full functionality of SHM is now available to the user. The online manual at http://www.szgrf.bgr.de/sh-doc/index.html should be referred to at this stage. A beginner's manual is also available.

IMPORTANT – Errors in SHM are reported to the command tool from where SHM was launched. This gives useful diagnostics for solving problems when working in SHM. The most common problem encountered is with the sfdfile.sfd and data paths (if problems reading data are experienced - check the sfdfile.sfd carefully).

5.1.5 Creating filter files in SHM

Whenever data are filtered in SHM a text file describing the filter is created in the working directory. The routine "datascan" makes use of these filter files output by SHM. There are two types of filter used in SHM and datascan

Recursive – Time Domain (.FLR)
FFT – Frequency Domain (.FLF)

The filter type used in SHM is set in "Specials-Setup-Filter Type". The text file is created describing the filter coefficients, with a filename ending in the respective .FLR or .FLF.

5.2 Trigger detection with datascan

The routine datascan_2 is used for identifying LOCAL events on continuous data i.e. events which do not appear on published event lists. The routine scans each station individually, checking for STA/LTA triggers above the specified threshold. If enough sequential triggers occur on the stream then it is flagged as an event. Once all stations have been scanned, events across all stations are compared. If an event occurs on more than one station within the specified window then it is flagged as an event. The product is an event list which must then be scanned manually to eliminate spurious triggers. Initially, care must be taken to parameterise the routine correctly. With quiet sites the method has been used successfully as an event detection algorithm in a number of different seismic settings.

5.2.1 Parameterising the trigger algorithm

Before the automatic picker is applied the user needs to develop an understanding of the optimum filter coefficients to use, the optimum window lengths for the trigger algorithm and which stations will cause the lowest number of false triggers.

1 Select ONE day of data
2 Create sfdfile.sfd and read into SHM
3 Scan through the full day of data with a 10 minute window length.
    a. Scan the data for local events and note the times.
    b. Optimise the filters for local event discrimination
    c. Determine which four stations provide the quietest data
    d. Note all local events for the day in question
    e. The duration of the common seismic events can be used to guide the window lengths of the STA/LTA
4 For the optimum filters, create both .FLF and .FLR files (see Section 5.1.5).
    a. SHM writes text files describing all filters used

b. FLR format is used for the trigger detection algorithm
c. FLF format is used for plotting the data.

The datascan_2 routine should now be run on the same day of data with the optimum filter coefficients. The window lengths and threshold levels can then be adjusted to optimise the algorithm and minimise the number of flase triggers, and also to ensure that no events are missed. Once the parameterisation is optimised, the routine can be run on the rest of the data.

5.2.2 Input file formats and running datascan_2

Firstly, set the environment variable SFD to the current directory using:

```
export SFD=.
```

A number of input files are required for the program. Example files are produced if datascan_2 is run at the command line and the required files are not found.

```
datascan_2
```

Create the following files in the working directory (see below for format):

`station_params.dat` – individual station parameters
`miniplot_statlist.dat` – list of stations for postscript plotting
`fastdet.cfg.template` – parameter file

Any number of stations can be scanned and plotted. A maximum of 6 stations is recommended for plotting.

Create a symbolic link from the "stalta" and "plotting" filters to your chosen FLR/FLF filter respectively, for example, in the working directory:

```
ln -s seisuk_10-20Hz.FLR seisuk_stalta_filter.FLR
ln -s seisuk_10-20Hz.FLF seisuk_plotting_filter.FLF
```

i. datascan_2 requires these specific filenames for filters:
    1. seisuk_stalta_filter.FLR – for the trigger algorithm
    2. seisuk_plotting_filter.FLF – for the postscript plots

ii. FLR and FLF filters are SH format and are created in SHM as described above.

If SAC or AH data output is required the "s" or "a" flags are used respectively:

A full seed volume must be created before running datascan_2 using the `create_seed.csh` script (see Section 4.3.1). Also, the environment variable SEED_DATA_LOC must be set to the relevant full seed volume:

```
export SEED_DATA_LOC=temp_2001_321.seed
```

SAC or AH header information will relate to only station location and timing. No event information is included as none has been derived at this stage.

station_params.dat

```
! station_params.dat
!
! Lines which begin with ! are comments
S12   11  10.0   HH  Z  100
S19   18  10.0   HH  Z  100
S15   14  10.0   HH  Z  100
S16   15  10.0   HH  Z  100
! ^    ^     ^    ^  ^  ^
```

```
! |    |     |    |  |  Sample rate
! |    |     |    |  Component
! |    |     |    Instrument-type
! |    |     sta/lta-threshold
! |    station number - must vary between stations
! station name
```

miniplot_statlist

```
S12,S15,S16,S19
```

fastdet.cfg.template

```
* DETECTION PARAMETERS FOR FASTDET
* CHANGE THEM CAREFULLY!!!
* STA/LTA detection parameters
* LTA is calculated outside of detected events only
*
* ITEMS WHICH SHOULD BE CHANGED ARE UNDERLINED:
*
xxxCHANxxx > selected channels - taken from station_params.dat
seisuk_stalta_filter.FLR > Filename for filter coefficients
    E  > Method for adaptation of STA and LTA values ( R(ecursive) or E(xact) )
  1.0  > STA length in sec. (max. 10)
 10.0  > LTA length in sec.
xxxTHRESHxxx  1.5 > Threshold (in STA/LTA) and min. coherence for onset detection on Z
xxxTHRESHxxx  > Threshold (in STA/LTA) for onset detection on each component
 0.25  > Step for Trigger test in sec.
    2  > Number of sequential true trigger tests for onset confirmation
    4  > Number of sequential true trigger tests for decreasing of the actual number of
   true triggers
 60    > Time window in sec. for checking if onsets fall into one group
    1  > Switch between full trigger testing (0) and trigger testing only on winner
                                                 channel (1) (no effect on outcome)
```

To run datascan_2 the date of interest must be input in a specific format, e.g.

```
01-JAN-2001                    07-JUL-2001
02-FEB-2001                    08-AUG-2001
03-MAR-2001                    09-SEP-2001
04-APR-2001                    10-OCT-2001
05-MAY-2001                    11-NOV-2001
06-JUN-2001                    12-DEC-2001
```

Therefore, run the program as follows:

```
datascan_2 dps 06-FEB-2001 1.0

      d flag - to detect triggers – produces the event list
      p flag - produces postscript output
      s flag – produces SAC output
      a flag – produces AH output
```

The above example will produce postscript plots of the triggers and also SAC format 6 minute windows about the triggers (use the “a” flag instead of “s” to output AH format data). The third argument (1.0) is the window used to compare arrivals across different stations. If an event occurs on more than one station, within the window specified here, it is classed as a true trigger. Otherwise the event is deleted. To run datascan_2 without outputting SAC or AH format files, use:

```
datascan_2 dp 06-FEB-2001 1.0
```

Running datascan_2 produces event-list files for each stations (f*.tim files). These are then compared with each other to determine events occurring on more than one site. For example, if the routine is run on data from 01-JAN-2001, the output event list file will be called:

```
fgrsn_disc_010101
```

These event files are simply trigger times. These may be P arrivals or just noise. A manual check of event lists is therefore required. Once the event-list files have been produced, and possibly edited, datascan_2 may be run to output SAC or AH data files windowed about the events, without having to re-run the trigger algorithm:

```
SAC:  datascan_2 s 06-FEB-2001 1.0      # If fgrsn_disc_060201 already exists
AH:   datascan_2 a 01-JAN-2001 1.0      # If fgrsn_disc_010101 already exists
```

5.3 Locating Local Events in SHM

A routine called LocSAT, a least-squares location algorithmn can be run from within SHM for local event location. This utility is currently only available on talisker (Solaris 7). It will run on lochside (Solaris 9) but the output files will be unreadable. Contact SEIS-UK to have your username made available on talisker.

In the directory where the script was run, execute

1. shm &
2. read
   a. Select Station
   b. Select Z
   c. Select Components (HH/BH …)
   d. Click on the event list box
      i. Double click on fgsrn_disc_??????
   e. scroll through the event list
      i. as each event is selected, the arrival time of the phase is automatically picked up by SHM
3. Then, for each event:
   a. Reduce the arrival time by 1 min before accepting
      (otherwise the event will be at the very start of the window)
4. Trace List; demean
5. Apply a filter if required
   a. This remains on for all subsequent reads unless “None” is selected in the filter window.
6. Highlight Pn in the Phase Selection box
   a. Pick Phase Pn (vertical component)
   b. Use left mouse button to pick
   c. Use right mouse button to display zoomed data in top window
   d. Picks can also be made in the top window
7. Read all components Z, E and N
8. Highlight Sn in the Phase Selection box
   a. Pick Phase Sn (horizontal component)
9. Run Locate-LocSAT with desired parameters
   a. If successful, a text-editor window appears with the calculated event location. The data are also displayed in the Parameter Box
      i. Save this data to a text file with a relevant name
   b. If errors involving locsat_r.txt appear the location routine has failed and the algorithm has not converged. In this case, the picks should be checked for errors. Some events may not be locatable with LocSAT due to their origin in relation to the array.
   c. Picks can also be saved to a text file:
      i. Work-Show Parameters
         1. Brings up a text editor – save the text file with a relevant name.
         2. Remember – files on the /data partitions are not backed-up.
   d. These output files can later be reformatted with awk scripts if the user wishes to import the data into a different package.
10. Before moving on to the next event in the trace list, delete all the picks

a. Highlight Pn in the Phase Selection box
    i. Press "Del" in Phase Selection box
b. Highlight Sn in the Phase Selection box
    i. Press "Del" in Phase Selection box

11 Although the capability is included in SHM, the data currently read into SHM may NOT be saved as miniseed files using File-Write miniSEED – these output files are very difficult to read.
12 Use ctrl^Q followed by ctrl^Z to exit SHM
13 If the location routine gives errors, exit and run clean_shm.

5.4 Determining Local (Richter) Magnitude

If an event is successfully located the local magnitude ml may also be calculated. The Wood-Anderson method is used, whereby the known instrument response is deconvolved from the data and the velocity output is converted to displacement. The data are then convolved with the response of a Wood-Anderson instrument, and an empirical relationship used to estimate event magnitude.

For this procedure, the Wood-Anderson filters for the instruments used in the network must be present, e.g.

```
/usr/local/sh/filter/SEISUK_4A99_E_S+WOODAND.FLF
/usr/local/sh/filter/SEISUK_3A13_N_S+WOODAND.FLF
/usr/local/sh/filter/SEISUK_6111_Z_S+WOODAND.FLF
```

To determine the local magnitude:

1. Locate the event as described above with LocSAT
2. Set the filter type to FFT (Specials – Setup - Filter type FFT)
3. Apply a Wood-Anderson filter (under Work – Filter)
4. If all traces become flat, check the errors output to the command terminal
5. May not be able to see the Wood-Anderson filters
6. Incorrect filter type may be selected
7. Run Amplitude - ml (keystroke 1)
8. The next two phase picks (clicks with left mouse button) should be made on a horizontal trace about P and S (body waves)
9. The program then calculates the magnitude and writes it to the terminal window from which shm was started.
10. See the manual pages for a full description of the method.
11. See Appendix B for an outline of the derivation of the SEIS-UK WA filters in SHM.

6 Extracting SEGY format data from GCF files

Generally, controlled source data, and occasionally passive data recorded on a linear array, are required in SEGY format. A number of scripts are used which utilise the PASSCAL routine txn2segy to convert from the raw GCF format to PASSCAL SEGY. In this way, data compatible with other organisations is produced.

N.B. Scripts are currently configured to extract data from taps 0 and 2 only. This is generally a safe assumption as higher sample rates are usually required for experiments which use the segy format.

6.1 CONVERSION TO SEGY

6.1.1 LINUX CONVERSION TO SEGY

Convert to SEGY and reformat headers / filenames, insert dummy headers:

```
gcf2txnsegy-robust.csh <gcf-directory-table.txt>
```

Then insert SEGY station headers with:

```
insert-segy-station-headers.csh <station-time-file>
```

Where the "station-time-file" format is:

```
6149 SITE1 2007:001:01:01:01 2007:022:00:00:00
```

- Sensor serial numbers must be numerical, therefore, all alpha-numeric serial numbers must be changed to fit this format. The following convention is used:

- Substitute 0 for A, 1 for B, 2 for C and so on, e.g.
  - 3A18 becomes 3092
  - 4B92 becomes 4192
  - 4C88 becomes 4288

You are now ready to form gathers with txn2segy or segygather as outlined below.

6.1.2 SOLARIS CONVERSION TO SEGY

Firstly, to convert to SEGY format:

```
gcf2txnsegy.csh <gcf-data-path>
```

For example, to convert the data from service-run 1:

```
gcf2txnsegy.csh /data/project_name/gcf/SERVICE_1
```

At this stage only one input file is required. A template file will be produced if the script is run without any arguments. The following utilities will crash if you have a mixture of sample rates in your segy data. Remove any data which does not relate to your experiment, such as huddle test data.

- `dasfile1` – Receiver station name / Sensor serial number / Channel list

```
1000   6098  1
1010   6100  1
1040   6092  1
1050   6109  1
```

  - **Important:** Receiver names may be alpha-numeric but all letters must be upper case. Sensor serial numbers must be numerical, therefore, all alpha-numeric serial numbers must be changed to fit this format. The following convention is used:
    - Substitute 0 for A, 1 for B, 2 for C and so on, e.g.
      - 3A18 becomes 3092
      - 4B92 becomes 4192
      - 4C88 becomes 4288
    - Care should be taken that no name is repeated – if this is to be the case (should not occur with SEIS-UK equipment), contact SEIS-UK
  - The Channel list "1" refers to the vertical component only (2 = N / 3 = E)
    - Use "123" to select all 3 components for the selected "tap" (care must be taken when plotting)
    - Alternatively, run 3 times, once for each component

The script gcf2txnsegy.csh performs the following steps:

1. Run gcfconv to output SEGY format data
2. SEGY format data are output to a unique directory, the name of which is derived from the time of creation.
3. Rename directories and files to make them consistent with txn2segy
4. Update segy headers to make them consistent with txn2segy
5. Insert sensor/digitiser calibration scalars (scale_fac) and UTM time flag to SEGY header

6.2 GENERATION OF SHOT GATHERS – SOLARIS OR LINUX

At this stage, after the user has run the conversion script, all the data are in individual hour-long SEGY format files. Next, the script txn2segy must be run to create shot gathers i.e. a single SEGY output file with data from all the recorders for the shot. Two more text files are required at this stage

- `surveyfile.txt` – Shot / receiver sites with grid coordinates end elevation in metres (only receiver data required for the gcf2txnsegy stage, hence the data conversion can be carried out before navigation data are available).

```
01 6199651 2843827 34
03 6250329 2811239 764
1000 6196994.102684 2845857.336884      25
1010 6197284.051272 2845052.005930      15
1040 6199562.808576 2843057.326086      222
1050 6209562.808576 2842057.326086      222
```

- Site numbers must be in ascending order
- Coordinates are Northing / Easting

- `eventfile.txt` – Shot site number / Shot time / Receiver file list used (it is recommended that the user extracts only one shot at a time)

  1 2002:126:21:45:00 dasfile1

To output 2 minute long traces:

```
txn2segy –D <SEGY-Directory> -s surveyfile.txt –e eventfile.txt –l 120 <outfile>
```

By default SEGY output from txn2segy is *IBM Floating Point format* which may then be plotted with standard routines e.g. ProMAX / SeismicUnix / Plotsec.

Plotsec is provided on SEIS-UK field Suns and linux PCs. To use plotsec:

```
run_plotsec.csh <segy-shot-file>
```

Although in theory up to 300sec long gathers can be exported, plotsec will struggle with anything greater than 120sec. A manual page for plotsec (viewable in netscape) is available at:

```
/usr/local/plotsec-3.4/html/index.html
```

6.3 Generation of Receiver Gathers

The process is essentially the same as that outlined in the previous section. The data must be converted to SEGY in the same way and all headers updated using the dasfile1 text file. The user then generates the receiver gathers one at a time. For example, for instrument serial number 6111, vertical component (1) -

```
ls R20??.???/*6111.1 > segyfiles.list
segygather –i segyfiles.list –x –d A01 –s shot_geom.txt –g receivers.txt –l shot_time.txt –
n 120.0 –o outfile_Z
```

Then repeat for the N component data:

```
ls R20??.???/*6111.2 > segyfiles.list
segygather –i segyfiles.list –x –d A01 –s shot_geom.txt –g receivers.txt –l shot_time.txt –
n 120.0 –o outfile_N
```

File formats are as follows (see man pages for more details):

`shot_geom.txt`

```
shot   time                lon        lat
190001 05:042:16:52:45.016 -89.8109 21.5078
190002 05:042:16:53:05.016 -89.811 21.5082
```

`receivers.txt`

```
number DAS/C   lon  lat  elevation
CX107 6012/3 -89.4 21.2297 5.0
CX068 6013/3 -89.6222 21.1826 5.0
```

`shot_time.txt`

```
shot   time
190001 05:042:16:52:45.016
```

```
190002 05:042:16:53:05.016
```

… and repeat for all components and stations (in a script). See the man page for segygather for further information about file structure (use the exact structure recommended – essentially we are fooling the code into thinking that the data were collected on a DAS. In this example, the station name is A01 (the –d flag) and a 120 second record is required (-n flag). The routine will produce a single receiver gather which contains data from all shots. Output format and the plotting method are identical to shot gathers.

7 Seismic Noise Analysis - PSD PDF Generation

A recent addition to the SEIS-UK software suite is a routine called PSDPDF (Power Spectral Density Probability Density Function) courtesy of Dan McNamara (for more information see http://geohazards.cr.usgs.gov/staffweb/mcnamara/Pubs/Reports/PDFSA_OFR2.0.pdf). ***Please note, the plots produced are of acceleration, not velocity. This is to allow easy comparison to the LNM.*** PSDPDF produces .png files (and using our script .pdf files too) of the averaged power spectral density of the waveform. This may be used to show the noise level at sites. It requires a minimum of 3 days continuous data in miniseed format. Please read these instructions through completely, including the notes at the bottom, before you proceed.

The first step in PSDPDF generation is the creation of a number of files – fortunately we now have a script set up to do this for you: setup_PSDPDF_dirs.csh This script is run with the full path-name to your data directory and your dataless.seed volume as inputs, e.g.

>setup_PSDPDF_dirs.csh /data3/TORFA05 /home/torfa05/dataless/dataless.seed

This script creates the following directories:

- In your data directory: PSDPDF directory
- In this new PSDPDF directory: data, RESP and STATS directories
- In PSDPDF/RESP: all the response files for the stations listed in your dataless volume
- In PSDPDF/data: NETWORK and NETWORK/STATION directories for all stations in your dataless volume
- In PSDPDF/STATS: NETWORK.STATION.-- and NETWORK.STATION.--/COMPONENT directories
- In your home directory: PSDPDF directory (this will be empty but the output plots from the second script will go here).

The second stage is to run the script 'run-psdpdf.csh'. This script requires the following as inputs (where all pathnames must be the full pathname):

- Your miniseed directory (where all the G***.*** directories are – should be /archive*/USERNAME/MSEED
- Your data directory – where you work on your data – should be /data*/USERNAME
- A text file containing the days (as they are in the day directory names) you want the PSDPDF performed for, one per line, e.g.
  G2005.200
  G2005.201
  G2005.202
- Your nft file – if you've got this far you must have one of these, if not, go back to the beginning of the manual to find out how to make one.
- The sample rate of the data you want processed (only one at a time please!). Even if you only recorded at one sample rate you have to tell the script what it is.

Example:

```
run_psdpdf.csh /archive4/IRAN04/MSEED /data3/IRAN04
/home/iran04/PSD_days.txt /home/iran04/nft.txt 100
```

This script works through every RESP file in your RESP directory (if you just ran step 1 this will mean it runs through every component for every station you have) and creates a .png and a .pdf plot of the PSDPDF for the

days you asked for. The plot files will be put in your /home/username/PSDPDF directory and can then be viewed using ghostview or similar packages (remember to export the display.) The .png files can also be directly imported into powerpoint presentations.

7.1 Points to note:

1) **You need to list at least 3 consecutive days in your days.txt file.** However you'll probably find that the last day you asked for is often not included in your plots, for example, having asked for days G2005.200, G2005.201 and G2005.202 your final plot may say something like '96 PSDs : 2005:200 – 2005:201'. This is normal with this code so do not be alarmed.
2) While the script is checking that the sample rate you asked for exists it will report 'No match' occasionally – this can be ignored. Any fatal errors will be obvious.
3) **The second script runs through every station and component that you have a RESP file for.** The default therefore is for it to run through every component for every station you have in your dataless (which should be every station you had). To change this, remove RESP files from the <your_data_directory>/PSDPDF/RESP directory that you don't want noise plots for. (You can always recreate these files using rdseed or by re-running the script in step 1 above – don't worry when it complains that lots of directories exist already.)
4) Following on from the last point, the script will abort if you have asked for a station for which you have no data. Please, therefore check carefully that you have data for all requested stations for the time period you want to create noise plots for.
5) **Ensure that there are no sample rate changes, station changes or any large gaps in the data in the time period you want.** (Don't choose a time when you were servicing). There should only be one record per serial number in your nft file.
6) When you look at your plots you may find the data looks cut-off at the left hand end (short periods) this is a result of the sample rate and instrument type used and is normal. See the example plots in the McNamara reference given above for examples of distinctive data anomalies as seen in the PSDPDF plots (gaps in the data, earthquakes etc.)
7) Please don't ask for too many days/stations worth of data – 3 days at 50sps takes about 5 minutes to do one component of one station.
8) The PSDPDF code can cope with actual gaps in the data but telemetry drop-outs – periods where the signal just goes to zero for a number of seconds – will provide odd functions as demonstrated in the McNamara reference.

Appendix A Dataless seed volume production with PDCC

Dataless seed volumes are files that contain all the meta-data for the experiment. These files are used in parallel with the miniseed volumes which are the raw seismic data with very little header information.

Dataless seed volumes are produced using PDCC, which is simply a gui front-end to a mySQL database. A dataless seed volume (a file) can be imported to this database through PDCC to populate a new database. The dataless seed volume files can themselves be created with PDCC.

The easiest way to produce a dataless seed volume that describes the current network is to import a dataless seed volume from a previous experiment (provided at `~/dataless/start.dataless`). The user can then edit the appropriate fields in the database, e.g. station names, locations and instrumentation, for the current network, and then output their own dataless seed volume file.

IMPORTANT: Username root and a password required on talisker
- use just the username root without a password on field Suns.

1. Create a new mySQL database for the experiment (created in the database, not the working directory)

```
create_db <database_name_db>
```

2. Import the existing dataless seed volume to the database:

On oban/scapa/glenesk/bladnoch:

```
cd ~/dataless
importDataless ./start.dataless  <database_name_db> –u root
```

On talisker:

```
cd ~/dataless
importDataless ./start.dataless  <database_name_db>  –u  root  –p  _____
```

3. Run pdccToolkit:

```
cd ~/dataless
pdccToolkit &
```

4. Edit **Database Tables** and Update *relevant fields*:
   i. **generic_abbrev** (Forms drop-down boxes for text fields when filling out the rest of the tables)
   ii. station
      1. *netid* – Name of network (from generic_abbrev)
      2. *net* – Assigned Network Code
      3. *sta* – Station ID
      4. *lat* – Latitude (South is negative)
      5. *lon* – Longitude (West is negative)
      6. *elev* – Elevation in km
      7. *num_chan* – Number of velocity channels
      8. *sta_name* – Station Name
      9. *word_order32* - 3210
      10. *word_order16* - 10
      11. *start_time* – Station installation time (approx)
      12. *end_time* – Station pull-out time (approx)

   iii. channel
      1. *net* - Assigned Network Code
      2. *sta* – Station ID
      3. *chan* – Channel Identifier
      4. *instrument_id* – Sensor Type

5. *proto_id* - 0
6. *cal_input_units* – V - Volts
7. *resp_input_units* – M/S – Velocity in Meters Per Second
8. *lat* – latitude
9. *lon* - Longitude
10. *elev* – Elevation (km)
11. *edepth* – Sensor depth (m)
12. *azimuth* – Z = 0.0 / N = 0.0 / E = 90.0
13. *dip* – Z = 90.0 / EN = 0.0
14. *data_format* - 1
15. *record_length* - 12
16. *srate* – Samples per second
17. *clock_tolerance* – 0.005
18. *start_time* – Station install time
19. *end_time* – Station pull-out time

5. At the command line copy the response "stage_xxx.txt" files to your working directory from -
ftp://talisker.geol.le.ac.uk/pub/RESPONSES/PDCC

6. Next build the instrument responses in PDCC
   a. Select (2) Setup Responses –
      i. Run - Build prototype Responses
         1. For each response sequence there are 3 stages required:
            a. Stage 1 – Sensor PAZ;
            b. Stage 2 – ADC Gain;
            c. Stage 0 – Summary of Gains
         2. Assign a prototype name e.g. SEIS-UK 3TD
         3. Look in: Working directory
            a. Single click on 3T_stage1.txt
            b. Click "Add to Prototype"
            c. Single click on 3T_stage2.txt
            d. Click "Add to Prototype"
            e. Single click on 3T_stage0.txt
            f. Click "Add to Prototype"
            g. Stages 1-2-0 are now listed on the right – hand side
            h. Click on "OK"
         4. A prototype named "SEIS-UK 3TD" has now been built with these 3 stages; repeat for 40TD and 6TD as necessary.

7. Assign the relevant prototype to each channel
   a. (1) Edit Database Tables
      i. proto_names
         1. Note the proto_id of the created proto_names.
      ii. Channel
         1. Assign relevant *proto_id* for each channel

8. The database is now ready for building a dataless seed volume
   a. (5) Build SEED volumes
      i. Enter Network Code
      ii. Enter Volume Name – this will be the filename assigned
      iii. Select "Dataless SEED volume"
      iv. Click "Build" – Each station will be scrolled in the command window and a dataless seed volume created in the start-up directory.

Once the dataless seed has been written there is no need to update the PDCC database or the dataless seed volume unless the network data are changed. In this case, the PDCC database is updated with the new meta-data, and the existing dataless seed volume is overwritten.

The dataless seed volume should be written to `~/dataless/seed.dataless` and the environment variable `ALT_RESPONSE_FILE` set to the appropriate dataless seed volume path.

To remove a database use:

```
delete_db <database_name_db>
mysqlshow          – to list available databases
```

Due to the complexity and instability of the software, it is not recommended that users undertake the production of a dataless seed volume before receiving training on the use of PDCC.

Tips for using PDCC:

- Check the format of the database name entered in PDCC at start up.
- Save the tables at regular intervals.
- Exit and restart a table after making a few edits to make sure fields have been updated correctly.
- If a field is updated in a table, move the cursor to another field before saving the table.
- Never try to save a table if a new line is incomplete.
- If a database becomes corrupted, create a new database and import the most recent dataless seed volume as outlined above.
- Create a new dataless seed volume on a regular basis.
- Create a dataless seed volume before trying to create prototype responses.
- Standard PC cut and paste can be used to copy data fields (3 rapid clicks to highlight, ctrl-C to copy; ctrl-V to paste).
- It is easy to make typing errors even with small numbers of instruments – take care.
- Start and stop times accurate to the day of install are sufficient (generally just use time 00:00:00.0) unless a station swap has occurred within one day.

Appendix B Instrument response removal and the derivation of Wood-Anderson filters for SHM

The removal of the instrument response is often a cause of much confusion. The following section presents an overview of how the correct parameters can be derived. However, the reader is referred to a much more comprehensive discussion of the process in the volume "Of Poles and Zeros: Fundamentals of digital seismology", Scherbaum, F. (2001), from which much of this section is derived.

B.1 Conversion of instrument response from Hz to Radian

To convert poles and zeros in Hz format to omega (radian) format:

```
Pole (rad) = Pole (Hz) * 2.pi
Zero (rad) = Zero (Hz) * 2.pi
```

To convert A0 from radian to Hz:

```
A0 (rad) = A0 (Hz) * (2.pi)^(No. Pole - No. Zero)
```

A0 is the normalisation constant which scales the amplitude of the transfer function to unity. Therefore this constant must also be scaled to account for the difference in the number of 2.pi factors on the numerator and denominator used in the conversion to radian. If the user plots the amplitude response of the poles and zeros they will observe that without this scaling the response is not unity over the bandwidth of the instrument in question.

This conversion to radian (and the A0 scaling) can be verified using the digital seismology tutor at http://www.geo.uni-potsdam.de/Software/software.html.

B.2 Conversion of instrument response from velocity to displacement

The frequency response of a system is given as the Fourier transform of the output divided by the Fourier transform of the input. In terms of velocity or displacement, this is represented as:

$$T_{vel}(jw) = \frac{Output(jw)}{Input_{vel}(jw)} \qquad \text{or} \qquad T_{disp}(jw) = \frac{Output(jw)}{Input_{disp}(jw)}$$

The velocity input is obtained by differentiation of the displacement input in the time domain, equivalent to multiplication of the displacement input spectrum with jw in the frequency domain.

$$Input_{vel}(jw) = jw.Input_{disp}(jw) \qquad \text{giving} \quad T_{vel}(jw) = \frac{Output(jw)}{jw.Input_{disp}(jw)}$$

$$\text{hence} \quad T_{vel}(jw) = \frac{T_{disp}(jw)}{jw} \qquad \text{or} \quad T_{disp}(jw) = T_{vel}(jw).jw$$

That is, to convert the velocity response to the displacement response in the frequency domain, multiplication with jw is used. This is equivalent to one more zero in the pole-zero representation.

With an output signal amplitude of A_0 and an input ground motion of amplitude A_i^{disp}, the calibration gain, g_d, which would scale the output to unity at the calibration frequency would be calculated as:

$g_d = \frac{A_{Input}}{A_{Output}} = \frac{A_i^{disp}}{A_0}$ in m/count, or in terms of velocity $g_v = \frac{A_i^{vel}}{A_0}$ in ms^{-1}/count.

For a harmonic signal at w_{cal}, $A_i^{vel} = w_{cal}.A_i^{disp}$ and hence $g_d = \frac{g_v}{w_{cal}}$.

Assuming gain values for the 6TD of 1100 V/ms^{-1} and $4x10^6$ count/V, at a calibration frequency of 1Hz, the calibration gains are equivalent to:

$g_v = \frac{1}{4x10^6 x1100}$ ms^{-1} per count, and $g_d = \frac{1}{4x10^6 x1100x2\pi}$ m per count.

6TD POLE-ZERO INSTRUMENT RESPONSE		
VELOCITY		***DISPLACEMENT***
Hz	**Radian**	***Radian***
A0 = 1983000	**A0 = 491883573**	***A0 = 78285702***
Zeros (3)	**Zeros (3)**	***Zeros (4)***
(-5.032, 0)	(-31.6, 0)	*(-31.6, 0)*
(0, 0)	(0, 0)	*(0, 0)*
(0, 0)	(0, 0)	*(0, 0)*
		(0, 0)
Poles (6)	**Poles (6)**	***Poles (6)***
(-0.02365, 0.02365)	(-0.148597, 0.148597)	*(-0.148597, 0.148597)*
(-0.02365, -0.02365)	(-0.148597, -0.148597)	*(-0.148597, -0.148597)*
(-39.3011, 0)	(-2469.3609, 0)	*(-2469.3609, 0)*
(-7.4904, 0)	(-47.06357, 0)	*(-47.06357, 0)*
(-53.5979, -21.7494)	(-336.7655, -136.656)	*(-336.7655, -136.656)*
(-53.5979, -21.7494)	(-336.7655, 136.656)	*(-336.7655, 136.656)*

B.3 Instrument response removal in SAC

SAC format pole-zero files may be exported from rdseed when the SAC output option is selected. Alternatively, the user can create pole-zero files observing the following format:

```
ZEROS 4
-31.6       0.0
POLES 6
-0.148      0.148
-0.148      -0.148
-2469.36    0.0
-47.36      0.0
-336.766    -136.656
-336.766    136.656
CONSTANT 2.1619e+18
```

SAC pole-zero files are always in omega (i.e. radian) format. Zeros of value (0, 0) need not be represented in the file. SAC pole-zero files are exported from rdseed as displacement responses.

For a 6TD with gain values of 1100 V/ms^{-1} and 4x10^6 count/V and the normalisation constant A0(vel-Hz) = 1,983,000, the SAC PAZ constant is calculated as:

$$CONSTANT = A0xSensorGainxDigitiserGainx2\pi$$

$$CONSTANT = 1983000x(2\pi)^2 x1100x4x10^6 x2\pi = 2.16x10^{18}$$

In SAC, the response is removed with the command line:

```
SAC> transfer from polezero subtype <filename> to none
```

If using the pole-zero file in the above format to remove the transfer function, output data will be displacement in units of metres.

Demo data from an experiment with a 6TD on a moving drill bench are available at /usr/local/seis-uk/sac_paz_demo.tar (also a README file). These data can be used to verify the derivation of the sac PAZ files.

B.4 SHM Wood-Anderson filters

To calculate ml magnitudes, SHM requires inverse filters to transform the data recorded on SEIS-UK instruments to the equivalent recordings on a Wood-Anderson torsion seismograph. Empirical methods are then used to estimate earthquake magnitude from the amplitudes on the horizontal components. For this purpose, filters must be derived to remove the Guralp instrument response (sensor and digitiser) and apply a WA response. All responses must be DISPLACEMENT and in omega format.

DISPLACEMENT POLE AND ZERO VALUES USED IN WOOD-ANDERSON FILTERS			
Wood-Anderson:	**Guralp 3TD**	**Guralp 40TD**	**Guralp 6TD**
A0 = 2800	**A0** = 90958274	**A0** = 90958274	**A0** = 78285702
Zeros (2)	**Zeros (3)**	**Zeros (3)**	**Zeros (4)**
(0,0)	(0,0)	(0,0)	(-31.6,0)
(0,0)	(0,0)	(0,0)	(0,0)
	(0,0)	(0,0)	(0,0)
			(0,0)
Poles (2)	**Poles (5)**	**Poles (5)**	**Poles (6)**
(-6.283185, -4.712)	(-1005.31, 0)	(-1005.31, 0)	(-0.148597, 0.148597)
(-6.283185, 4.712)	(-502.6548, 0)	(-502.6548, 0)	(-0.148597, -0.148597)
	(-1130.973, 0)	(-1130.973, 0)	(-2469.3609, 0)
	(-0.037008, 0.037008)	(-0.148597, 0.148597)	(-47.06357, 0)
	(-0.037008, -0.037008)	(-0.148597, -0.148597)	(-336.7655, -136.656)
			(-336.7655, 136.656)

From the SHM manual pages (http://www.szgrf.bgr.de/sh-doc/fil_simulation.html): "A simulation filter is created by taking the transfer function of the recording instrument, inverting it (take reciprocal value of the normalization, exchange poles and zeros), multiply it with the transfer function of the simulated instrument and shorten all possible values of the rational expression."

SHM filter parameters for conversion from 6TD to Wood-Anderson	
CONSTANT = 1.2937E-12	
Zeros (6)	**Poles (4)**
(-0.148597, 0.148597)	(-6.283185, -4.712)
(-0.148597, -0.148597)	(-6.283185, 4.712)
(-2469.3609, 0)	(-31.6, 0)
(-47.06357, 0)	(0, 0)
(-336.7655, -136.656)	
(-336.7655, 136.656)	

The gains of the sensor and digitiser must also be removed by inclusion in the final constant.

$$CONSTANT = \frac{A0(WoodAnd.)}{\left(A0(6TD)xSensorGainxDigitiserGainx2\pi\right)} x1000mm$$

$$CONSTANT = \frac{2800}{90958274x3000x4x10^{6}x2.\pi} x1000$$

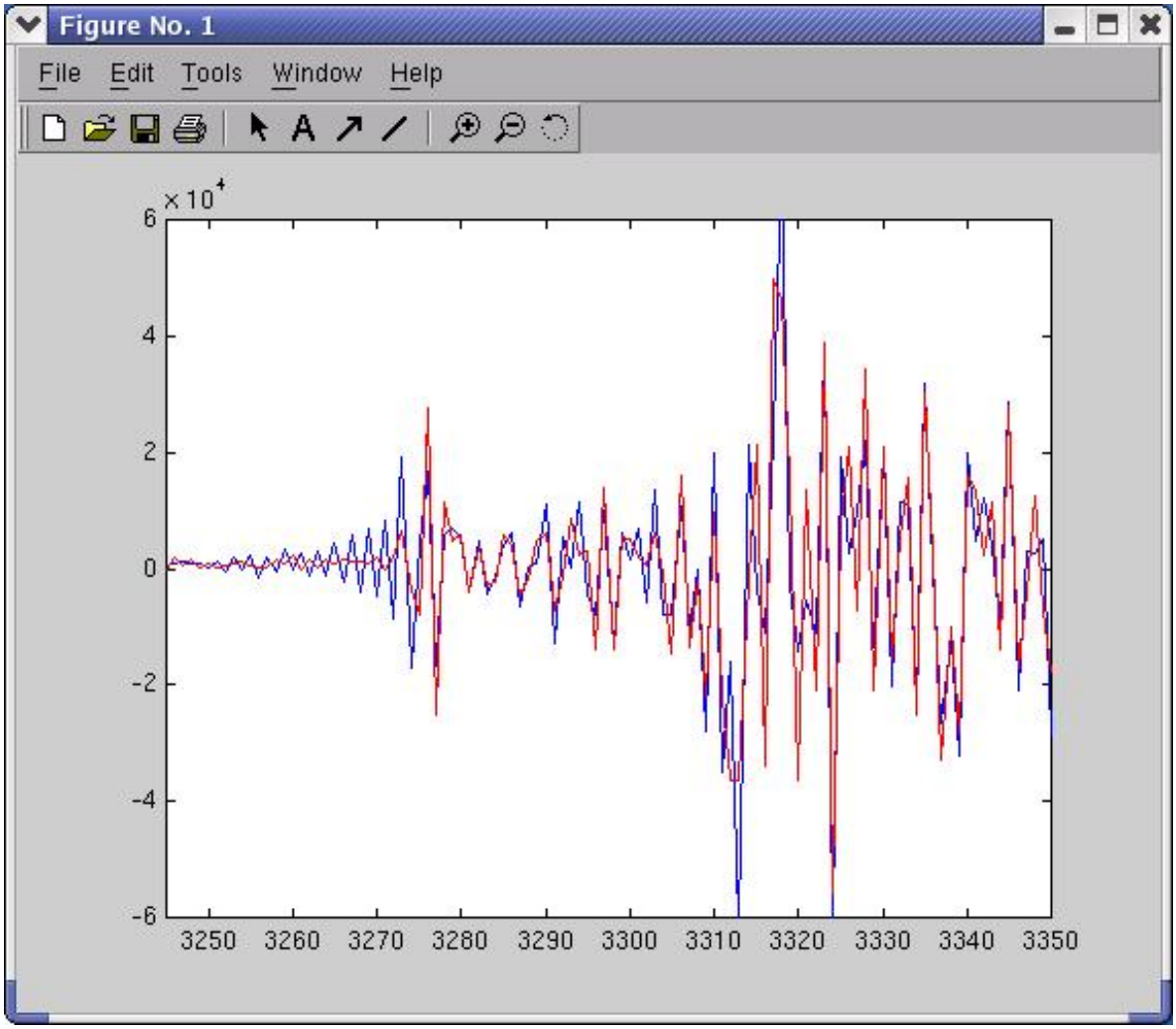
- A factor of 1000 is used to convert SEIS-UK instrument output to mm
- During instrument response removal, data are divided by the constant. Therefore, divide the transfer function constant by 1000 to give output in mm.
- This is equivalent to multiplying by 1000 in the WA filter constant, as above.
- SHM filter files are located in /usr/local/sh/filter and have been derived for each component separately (see SHM documentation for WA magnitude estimation).

Appendix C Removing FIR filter effects from SEIS-UK data

A script can be used to remove the ringing introduced by the digitiser FIR filters. The precursory signals can be observed when a high amplitude impulsive signal with frequency content close to that of the FIR filters is recorded. A script for removal of the fir effects can be applied to the SAC format event data as follows, for example where digitser-tap=2, sps=50:

```
remove-fir-effect.csh evt_2001.034.22.42.59.5600 6TD 2 50
```

This creates files in the event directories with the suffix .NO_FIR. The user can then compare the original with the filtered. Not all sample-rate / tap combinations are available at the present time. Contact SEIS-UK for further information. Care must be taken when using the corrected data as the waveform is altered. Corrected data should be used for first arrival picking only.



Example of FIR removal from 50sps 6TD data. Raw data are displayed in blue and the data with the FIR effects removed are in red.

N.B. If used the filter must be acknowledged to Paul Minchinton of Alice Designs and Guralp Systems Ltd.

Appendix D Viewing ascii dumps of dataless volumes (experts only)

To view an ascii dump of your dataless seed volume you can use the utility “cdscan” courtesy of George Helffrich, Bristol University:

```
cdscan -d /usr/local/dataless/05_06.ZB.Simuelue.dataless | more
```

An example output follows. Refer to the SEED manual for a further explanation.

```
  Vol. header: Type: 010, Version 02.3, recl=  4096
  Time span: 1970,001,00:00:00.0000 2038,001,00:00:00.0000
 Rec,Posn      1,   8      Type  10 Length  113
    0 010011302.3121970,001,00:00:00.0000~2038,001,00:00:00.0000~2006,
   64 227,12:47:18.0000~SEIS-UK Data Centre Leicester~~
 Rec,Posn      1, 121      Type  11 Length   98
    0 0110098008LEWK 000003LFAK 000017LUAN 000031PUTR 000045DEHI 00005
   64 9MAUD 000073BATU 000087LABU 000101
 Rec,Posn      1, 219      Type  12 Length   11
    0 01200110000
 Rec,Posn      2,   8      Type  30 Length  146
    0 0300146Steim-1 Integer Compression Format~000105006F1 P4 W4 D C2
   64  R1 P8 W4 D C2~P0 W4 N15 S2,0,1~T0 X W4~T1 Y4 W7 D C2~T2 Y2 W2 D
  128  C2~T3 N0 W4 D C2~
 Rec,Posn      2, 154      Type  33 Length   19
    0 0330019001Simeulue~
 Rec,Posn      2, 173      Type  33 Length   37
    0 0330037002Guralp CMG-3TD Seismometer~
 Rec,Posn      2, 210      Type  33 Length   38
    0 0330038003Guralp CMG-40TD Seismometer~
 Rec,Posn      2, 248      Type  33 Length   37
    0 0330037004Guralp CMG-6TD Seismometer~
 Rec,Posn      2, 285      Type  34 Length   32
    0 0340032001COUNTS~Digital Counts~
 Rec,Posn      2, 317      Type  34 Length   44
    0 0340044002M/S~Velocity in Meters per Second~
 Rec,Posn      2, 361      Type  34 Length   18
    0 0340018003V~Volts~
 Rec,Posn      2, 379      Type  34 Length   20
    0 0340020004A~Amperes~
 Rec,Posn      2, 399      Type  34 Length   35
    0 0340035005COUNTS/V~Counts per Volt~
 Rec,Posn      2, 434      Type  31 Length   73
    0 03100730001STiming correction of 1000 ms applied to all traces u
   64 ntil ~000
 Rec,Posn      2, 507      Type  31 Length   72
    0 03100720002S2006, December 31, 23:59.59.999.  This timing correc
   64 tion~000
 Rec,Posn      2, 579      Type  31 Length   70
    0 03100700003Sfixes a known timing error of the Trimble GPS system
   64 . ~000
 Rec,Posn      2, 649      Type  31 Length   63
    0 03100630004STiming of all traces should be correct but note~000
 Rec,Posn      2, 712      Type  31 Length   74
    0 03100740005Sthat this correction relies on information supplied
   64 by the~000
 Rec,Posn      2, 786      Type  31 Length   73
    0 03100730006Sinstrument maker and could not be verified independe
   64 ntly.~000
# To explain the characters printed:

Rec,Posn      3,   8      type  50 length   92
    0 0500092LEWK 002.9235800095.80407700037.30003000Lewak~00132101020
   64 05,346~2006,055,23:59:59~NZB
# 3=Record Number ; 8=Position in record of the character preceding
blockette  data (data start at Char #9); 50=Blockette Type ; 92=Blockette
Length
# 0=Position in record of the character preceding blockette data followed
by the data up to character 64
# 64=Position in record of the character preceding blockette data
followed by the data to the end of the blockette

Rec,Posn      3, 100      type  52 length  130
    0 0520130  BHZ0000004~002003002.9235800095.80407700037.3000.0000.0
```

```
   64 -90.00001125.0000E+015.0000E-030000CG~2005,346~2006,055,23:59:59
  128 ~N
Rec,Posn      3, 230      type  53 length  478
    0 0530478A01002003 4.91139E+08 1.00000E+00003-3.16000E+01 0.00000E
   64 +00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
  128 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00006-1
  192 .48597E-01 1.48597E-01 0.00000E+00 0.00000E+00-1.48597E-01-1.485
  256 97E-01 0.00000E+00 0.00000E+00-2.46936E+03 0.00000E+00 0.00000E+
```

 # Also look out for record delimiters in raw seed. A record is 4096 characters and is initiated with 8 characters (000001S* for example). See Seed manual "How to Assemble Control Headers".
V=Volume Header ; A=Dictionary ; S=Station ; T=Time Span
****=Record Continued from previous, otherwise use a blank character***
 # Each blockette starts with 3 characters indicating type, then 4 indicating length.

Appendix E Manipulation of miniseed data

E.1 Changing miniseed data blocksize from 512 to 4096byte

Certain packages, including the Nanometrics Apollo suite, produce miniseed data which have a default block size of 512byte. SEIS-UK use a block size of 4096 as default. Should you use any 512 size data with qmerge, it will by default repack the data to a block size of 4096. However, it does not do this in a sensible way, it simply pads the blocks from 512 to 4096 with zeros. This results in a significantly higher data volume than expected. We therefore recommend that any 512byte data be repacked with the utility "msrepack" of IRIS.

```
msrepack -R 4096 -o <outfile-4096> <infile-512>
```

E.2 Moving miniseed data created with linux to Solaris

Miniseed data are endian sensitive, i.e. the way they are written is dependent on the architecture on which they are created. Big Endian = Sun (Sparc) ; Little Endian = PC (Intel x86).

The final archive version used by SEIS-UK is big-endian, or Sun format. We therefore recommend that any data format conversions (e.g. GCF to miniseed) are carried out on the Sun server. However, any data which are produced on PCs can still be used on the Sun, but require a byte swap.

We therefore recommend that the users run the data through qmerge to rectify the problem:

```
qmerge -o $file.q $file
```

- Users should note that although "PC order" data will display in pql on a Sun, the headers will not make any sense when viewed with mseedhdr.
- Care must be taken as not all PC based packages will produce little-endian data. For example, the Scream software of Guralp produces Sun/big-endian miniseed data automatically, and therefore does not require a byte-swap.
- Scripts are available to facilitate this process.