



RT130 Data Processing In a Nutshell

11Mar2010



You've offloaded a data-set using Neo and have stacks of zip or tar files. Now what to do with them? Start by getting organized (steps 1-4). Then convert the data & log files to `_miniseed` (5-8). Build the database (9-13) and the `dataless` (14-15). Finally, send the day volumes and the `dataless` to PASSCAL.

Turquoise highlighting indicates a Unix command line, to be entered in a terminal window.

1. Create an organized directory structure for your data. For example: `mkdir raw_data mkdir mseed mkdir logs mkdir day_vol`.

2. Create a batch file using your preferred text editor: vi, emacs, etc. Use the template on page 3 to get started. Name it something obvious, for example "ab123.batch". Be very accurate with the metadata – this is the core of your database.

3. Convert the batch file to a parameter file, which our tool, `rt2ms`, can parse. Use `batch2par` to make the conversion: `batch2par batch_file > par_file`. The format of the final parameter file should look something like this:

| #das; | refchan; | refstrm; | netcode; | stn; | channel; | samplerate; | gain |
|-------|----------|----------|----------|-------|----------|-------------|------|
| 9306; | 1; | 1; | YH; | ME42; | HHZ; | 100; | 1 |
| 9306; | 2; | 1; | YH; | ME42; | HHN; | 100; | 1 |
| 9306; | 3; | 1; | YH; | ME42; | HHE; | 100; | 1 |
| 9306; | 1; | 2; | YH; | ME42; | LHZ; | 1; | 1 |
| 9306; | 2; | 2; | YH; | ME42; | LHN; | 1; | 1 |
| 9306; | 3; | 2; | YH; | ME42; | LHE; | 1; | 1 |

Please note: The 'refstrm' column needs to be revised by hand, because the initial output will not be useable for `rt2ms`. You could also make the parameter file from scratch, using the format above and the information in the batch file.

4. Make a list file showing the path and file names to be converted. This is to point `rt2ms` in the right direction. Name it whatever you wish. (You might have `.tar` files instead of ZIPs.)

```
raw_data/2008257.9E02.ZIP
raw_data/2008260.9E07.ZIP
raw_data/2008264.9E0F.ZIP
raw_data/2008285.9E00.ZIP
```

5. Convert the raw RT130 data to miniseed with `rt2ms`. First type `rt2ms -h` to see all the optional parameters you may wish to include. The actual command line should look something like this: `rt2ms -F list_file -p par_file -R -L -o mseed` This process can take a while, especially for large data sets.
6. Now that you have viewable files, **verify the data quality** by looking at the mseed and log files. Use `pqi` for trace files and `logpeek` for the logs. Look for obvious signs of trouble: loss of GPS timing, overlaps, gaps, corrupted files, etc. Make a note of any obvious problems. Use `fixhdr` to confirm the conversion is complete, and to correct any problem headers or timing issues.
7. Copy the local `log2miniseed` parameter file into your working directory by typing: `cp $ANTELOPE/data/pf/log2miniseed.pf`. Using your favorite text editor, change the default string in the `log2miniseed.pf` file from this: `wfname %Y/%j/%{sta}.%{chan}.%Y.%j` to this: `wfname day_vol/%{sta}/%{sta}.%{net}.%{loc}.%{chan}.%Y.%j` This long string of symbols specifies an organized directory path and filename format, as required by PASSCAL.
8. Convert the log files into miniseed format. You want to be sure the `log2miniseed` command is calling the parameter file you just revised, so (for `tsch`) type: `setenv PFPATH $ANTELOPE/data/pf`. Then: `log2miniseed -n PI -s AB00 logs/2005.128:15:09.0965D.log` where 'PI', 'AB00', and 'logs/*.log' are changed to your network, station, path and filenames. Do this for every file, or write a script for it.
9. Build the Antelope database by giving it a short name and calling the batch file you made in step 2: `dbbuild -b db_name batch.file.name >& dbbuild.out`
10. View your database with `dbe db_name` Scroll through a few tables to be sure it's healthy. If you see any glaring errors, chances are good there's a typo in the batch file. Best to find and fix it now, before you spend much more time and effort on the database. Regenerate the database as in step 9 in needed. At this point all you have is a descriptive framework; the trace files are added next.
11. Add the waveforms with this long command line: `miniseed2days -d db_name -w "day_vol/%{sta}/%{sta}.%{net}.%{loc}.%{chan}.%Y.%j" mseed/ >& miniseed2days.out` As with the log names in step 7, the long string of symbols specifies an organized directory path and filename format. This command also packs the files into day volumes.
12. Assign calibration values from the calibration table of the database to the just-created `wfdisc` table by typing: `dbfix_calib db_name`
13. Verify the integrity of your database: `dbversdwf -tu db_name` (a good result is: *0 bad files / 0 bad records*). Also run: `dbverify -tj db_name >& dbverify.out`
14. Create the dataless SEED volume. Use this naming convention: `mk_dataless_seed -v -o NN.YY.db_name.YYYYJJHHMM.dataless db_name` Where: NN is your network code, YY is the year of your data, and YYYYJJHHMM is the approximate *current* year-julian-date-hour-minute.

The "dataless" is the final manifestation of your metadata. It's like a menu, or an index, of the experiment. If any station or time range is missing from the dataless, the corresponding trace files will be orphaned – totally inaccessible by anyone. If this happens, you can always fix the problem with a new, corrected dataless.

15. Verify the dataless: `seed2db -v dataless_name`

16. Send the data to PASSCAL. We prefer you use `gui_DoFTP`. It will pack and send your files automatically, and trigger the automated system on our end. Please drop us note (data_group@passcal.nmt.edu) before sending, so we can be sure our system is up and ready.

```
#comment: This is a dbbuild batch file example.
net PI Pier database at PASSCAL

sta WH05 34.0740 -106.9188 1.43 Socorro NM
time 08/13/2009 00:00:00
net PI
datalogger rt130 9804
sensor cmg40t 0 T4K59
axis Z 0 0 - 1 1
axis N 0 90 - 2 1
axis E 90 90 - 3 1
samplerate 40sps
channel Z BHZ
channel N BHN
channel E BHE
samplerate 1sps
channel Z LHZ
channel N LHN
channel E LHE
add
close WH05 08/13/23:59:59

sta WH06 34.0740 -106.9188 1.43 Socorro NM
time 08/13/2009 00:00:00
net PI
datalogger rt130 980B
sensor cmg40t 0 T4O59
axis Z 0 0 - 1 1
axis N 0 90 - 2 1
axis E 90 90 - 3 1
samplerate 40sps
channel Z BHZ
channel N BHN
channel E BHE
samplerate 1sps
channel Z LHZ
channel N LHN
channel E LHE
add
close WH06 08/13/23:59:59
```

A few tips....

Many database errors can be avoided by rounding the start & close times in the batch file to 00:00:00 and 23:59:59, respectively.

Station changes, such as a new datalogger, sensor or sample rate, can be described in the batch file by studying the examples shown in: [man dbbuild_examples](#)

For useful information on any command, try typing `man <command>` (man -> manual) or `<command> -h` (h -> help).

To see a much more detailed processing guide, please see the PASSCAL document:

“Generating SEED From RT130 Data Using Antelope for Stand-Alone Stations”

available from

http://ftp.passcal.nmt.edu/passcal/docs/data_docs/

For more information, see also:

<http://www.passcal.nmt.edu>