RT130 Data Processing

You've offloaded a service run and have data from each RT130. Follow the steps in this document to convert the data to miniSEED and reorganize it into station/channel/day volumes. Then, create a stationXML for your experiment using Nexus (see step 7) before submitting data to PASSCAL. **Unix commands (bold print)** and any command line arguments are highlighted in yellow. Input files are denoted by <*filename*>.

Additional documentation can be found on the PASSCAL website:
https://www.passcal.nmt.edu/content/passive-source-seed-archiving-documentation

1. Create an organized directory structure for your data. Start by creating a main directory for the project. Under your main project directory, make a first level directory "SVC1" for service run number 1. For each subsequent service run create a new directory, e.g. SVC2, SVC3. Create directories in the SVC1 directory for the raw data files and log files. For example: **mkdir** RAW and **mkdir** LOGS

Move the raw data files (either .ZIP or CF folders) into the RAW directory.

2. Create the parameter file(s). The parameter file is used by **rt2ms** to assign header information to the miniSEED files. **rt2ms** is a PASSCAL program that generates miniSEED formatted files from REF TEK RT130 raw files. In addition, **rt2ms** also modifies the headers. In the SVC1 directory, use a text editor and information from your field notes to create an ASCII parameter file (parfile) following the examples below.

Example 1: Parameter file for station ME42 with a broadband sensor and RT130 9306 recording 3 data streams, 40sps (refstrm 1), 1sps (refstrm 2), and 0.1sps (refstrm 9, the mass positions) with a gain of 1.

```
#das; refchan; refstrm; netcode; station; channel; samplerate; gain
9306;    1;      1;        ZJ;     ME42;    BHZ;      40;        1
9306;    2;      1;        ZJ;     ME42;    BH1;      40;        1
9306;    3;      1;        ZJ;     ME42;    BH2;      40;        1
9306;    1;      2;        ZJ;     ME42;    LHZ;      1;         1
9306;    2;      2;        ZJ;     ME42;    LH1;      1;         1
9306;    3;      2;        ZJ;     ME42;    LH2;      1;         1
9306;    1;      9;        ZJ;     ME42;    VM1;      0.1;       1
9306;    2;      9;        ZJ;     ME42;    VM2;      0.1;       1
9306;    3;      9;        ZJ;     ME42;    VM3;      0.1;       1
```

Example 2: Parameter file for station NB11 with a short period sensor and RT130 9BDE recording 1 data stream, 500sps (refstrm 1) with a gain of 32.

```
#das; refchan; refstrm; netcode; station; channel; samplerate; gain
9BDE;    1;      1;        X5;     NB11;    DHZ;      500;       32
9BDE;    2;      1;        X5;     NB11;    DH1;      500;       32
9BDE;    3;      1;        X5;     NB11;    DH2;      500;       32
```

NOTES:

- Station names must be 3-5 alphanumeric characters, all capitals.
- To determine the correct channel names, see the summary table from the SEED manual on the PASSCAL website (see link on previous page)
- PASSCAL encourages the use of 1 and 2 in place of N and E, respectively, in channel names to match current GSN convention. This also allows for updates to be made to the sensor orientations in case of incorrectly oriented sensors without requiring changing the channel names.
- The parfile only references the RT130 serial number and does not keep track of start/end times. If a RT130 is assigned to different stations at different times you must treat each epoch separately. Create separate parfiles for each epoch and process the data separately.
- Most broadband sensors (e.g, CMG-3T, TR-240, STS-2) will need mass positions described in the parfile (refstrm 9).
- For most broadband sensors (e.g, CMG-3T, TR-240, STS-2), the gain will be 1. For most high frequency sensors (e.g, L-22, L-28), the gain will be 32.
- Use of a plain text editor is highly recommended in creating the parfile. **rt2ms** will not recognize the file if unwanted characters are present in the parfile
- refstrm 1 is commonly the highest sample rate. If unsure how the RT130 was programmed, you can check the configuration using **logpeek**.

3. Convert your data into miniSEED files. In the service run directory, convert the raw RT130 data to miniSEED. Typing **rt2ms** -h shows a list of available options.

If raw data is in decompressed folders, use the following commands:
ls -d $PWD/RAW/*.cf > file.lst
rt2ms -F file.lst -Y -L -o MSEED -p <*parfile*> >& rt2ms.out
The (-F) flag will process all files in the named list, (-Y) puts the data in yearly directories, (-L) outputs .log and, if created, .err files, (-o) creates an output directory, MSEED, and (-p) points to your parfile.

If raw data is in ZIP files:
rt2ms -D RAW -Y -L -o MSEED -p <*parfile*> >& rt2ms.out
The (-D) flag will process all .ZIP files in a specified directory, instead of in a file list as in the previous example.

When **rt2ms** finishes, move all of your .log and .err files from the MSEED directory to the LOGS directory that you created in step 1.

After running **rt2ms** the MSEED directory structure should look something like the example below. In the MSEED directory there will be .log files and possibly .err files along with a

subdirectory for each year (left) that contains day directories for each stream (right):

```
2014.019.21.29.16.98EZ.log
2014.019.21.29.16.98EZ.err
Y2014/
```

```
R065.01/
R065.02/
R065.09/
```

4. Reorganize the miniSEED data into station/channel/day volumes. dataselect is an IRIS DMC program that allows for the extracting and sorting of miniSEED data (https://github.com/iris-edu/dataselect). This will read the data from the MSEED directory and convert them into day volumes with the required naming format:
dataselect -A DAYS/%s/%s.%n.%l.%c.%Y.%j MSEED/Y*/*/*
The (-A) flag writes file names in the specified custom format. The format flags are (s) for station, (n) for netcode, (l) for location, (c) for channel name, (Y) for year, and (j) for Julian date. See the help menu for more details on options (**dataselect** -h).

Depending on how much data you have, you may need to run **dataselect** in a loop that runs over the different days or stations in your experiment.

5. Confirm your station and channel names. In the DAYS folder just created by **dataselect**, check to see if you have folders for each of your stations. The data should be organized into those folders in station/channel/day volumes named STA.NET.LOC.CHAN.YEAR.JULDAY. For example: BA01.XR..HHZ.2018.039 (The .. after XR is where the location code would be if needed).

If your parfile was incomplete (i.e. missing stations or channels), there will be one or more folders named with the RT130 serial numbers (e.g. 9306) instead of the desired station name (e.g. ME42). To change any miniSEED headers to correct a station name, network code, etc., see the **fixhdr** doc on the PASSCAL website (see link on the first page). After you have modified the headers with **fixhdr**, rename the files so that the station-network-location-channel codes in the miniSEED file names match the corrected headers.

6. Perform quality control of waveforms and logs. Verify the data quality by reviewing the traces and log files (with **logpeek** and **pql**). Obvious signs of trouble include loss of GPS timing, overlaps, gaps, corrupted files, etc. Make a note of any problems. Use **fixhdr** to correct mark timing issues, and/or to convert the files to big endianess if they are not already. For more information on how to use these tools, refer to the appropriate documentation on the PASSCAL website (see link on the first page).

7. Create metadata for your experiment. Use Nexus to generate a stationXML file for your experiment metadata. See the "Metadata Generation with Nexus in a Nutshell" document on the PASSCAL website (see link on first page).

8. Send miniSEED data to PASSCAL. Please drop a note, with your PASSCAL project name in the subject, to *data_group@passcal.nmt.edu* before sending the data to PASSCAL so that we can set up a receiving area. Attach the stationXML created with Nexus to this email unless it is larger than 5Mb. Use our tool **data2passcal** to send the data:
data2passcal DAYS/
data2passcal will scan all subdirectories of the DAYS folder and send any miniSEED files that have the correct file names.