Interim Documentation: RT130 Data Processing (rt2ms 2021.090 and later)

The version of **rt2ms** released on the newest PASSOFT (late 2020, requires Python 3.6 or above) is modified and streamlined compared to the previous versions. The major changes include direct conversion from REF TEK format to miniSEED day volumes, time dependency of the parameter file (parfile), option to automatically generate a parfile, and removal of support for .ZIP and .ref files. Running the new **rt2ms** also automatically creates a verbose output file, rt2ms.msg. Use **rt2ms** -h to see the flags available. Please see https://www.passcal.nmt.edu/content/software-resources for instructions on updating **rt2ms** to the latest release.

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:
http://www.passcal.nmt.edu/content/data-archiving/documentation/passive-source

1. Organize the raw data. Create a main project directory with service run directories under it (SVC#), and create a RAW directory for each service run. Copy the CF cards to RAW using the Drag and Drop procedure received during training. The MSEED and LOGS directories will be automatically created by **rt2ms** during step 4.

2. Create the parfile. Run **rt2ms** in exploratory mode. For most experiments, create a list of the CF card folders. For example, from the SVC1 directory:
For multiple RT130s:

```
ls -d $PWD/RAW/*.cf > dir.list
rt2ms -D dir.list -e
```

For a single RT130:

```
rt2ms -d RAW -e
```

If the folder has the expected REF TEK directory structure, rt2ms will read it; if not, it will print a warning message and move to the next folder.

This automatically generates a parfile in the directory where **rt2ms** was run called parfile.txt. The parfile, and any error, warning, or information messages will also be output into rt2ms.msg. Here is an example of the new parfile format:

```
das; refchan; refstrm; netcode; station; channel; samplerate; gain; implement_time
9388; 1; 1; 6E; SH05; EHZ; 100; 1; 2015-07-10T17:54:58.000000Z
9388; 2; 1; 6E; SH05; EH1; 100; 1; 2015-07-10T17:54:58.000000Z
9388; 3; 1; 6E; SH05; EH2; 100; 1; 2015-07-10T17:54:58.000000Z
9388; 1; 2; 6E; SH05; LHZ; 1; 1; 2015-07-10T17:54:58.000000Z
9388; 2; 2; 6E; SH05; LH1; 1; 1; 2015-07-10T17:54:58.000000Z
9388; 3; 2; 6E; SH05; LH2; 1; 1; 2015-07-10T17:54:58.000000Z
9388; 1; 9; 6E; SH05; VM1; 0.1; 1; 2015-07-10T17:54:58.000000Z
9388; 2; 9; 6E; SH05; VM2; 0.1; 1; 2015-07-10T17:54:58.000000Z
9388; 3; 9; 6E; SH05; VM3; 0.1; 1; 2015-07-10T17:54:58.000000Z
```

3. Prepare the parfile for use in converting the raw data to miniseed. You will likely have to modify one to three columns.

- Network code: this is populated as XX by default, unless you mobilized the experiment before going to the field and the RT130 configuration file included the network code. Please change this column to the 2-character network code assigned after you completed the mobilization form on the PASSCAL website.
- Station name: This may be correct, depending on how the RT130 was programmed. If needed, change this column to the correct station name. Use the DAS SN and field notes as a reference.
- Channel name: **rt2ms** assigns the channel codes based on gain and sample rate programmed into RT130 as well as SEED and GSN conventions. They should not need to be changed for most instruments. Please see the SEED channel naming document at https://www.passcal.nmt.edu/content/passive-source-seed-archiving-documentation to verify that the channel names were assigned correctly. Note that if you are using short period sensors (L22, L28, etc.) at gain 32, you will need to change the middle letter from L to H.

Please do not change the other columns. They are dictated by RT130 programming, and/or are needed for the new **rt2ms** to run properly.
As mentioned above, unlike parfiles used with the previous **rt2ms**, the new version is time-dependent and can be used with equipment changes, instead of needing to create a separate parfile after equipment changes.

4. Convert the raw data to miniseed day volumes. For most experiments, use the list of CF card folders created in step 2. This is necessary if data from more than one RT130 is in the RAW directory, so **rt2ms** has the structure to correctly parse the directories and create output files. The miniSEED day volumes output by **rt2ms** will be named in STA.NET.LOC.CHAN.YEAR.JULDAY format, as required for sending the files to PASSCAL.

For multiple RT130s:

```
rt2ms -D dir.list -p parfile.txt
```

For a single RT130:

```
rt2ms -d RAW -p parfile.txt
```

This will create miniSEED day volume files in the MSEED directory as well as a log file in the LOGS directory for each RT130's CF card (e.g., if there are two cf folders for RT130 9BC5, two log files will be created). The rt2ms.msg file, similar to rt2ms.out on previous versions, will also be populated with warnings, errors and informational messages (e.g., progress in parsing state of health files, writing day volumes, skipped data packets, etc.). The messages are more verbose than previous versions of **rt2ms**. In most cases, if a data packet causes an error, **rt2ms** will skip to the next data packet instead of exiting.

Notes:

- The log files are named like: RT130_9BC5_2015240_2015294.log instead of 2015.240.21.59.28.9BC5.log as they were in previous versions of **rt2ms**, and no .err files are generated. The log files will still be read as normal by **logpeek**.
- If you have not used the CF card folder list when it is required in running **rt2ms** (or: if you do not use the -D/-d options when running rt2ms), this will cause it to exit with the following error - "ERROR - Path to CF card directory(ies) not provided! Required input! Try --help option."

5. Confirm your station and channel names. In the MSEED folder just created by **rt2ms**, 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 the parfile generated by **rt2ms** exploratory mode was incomplete (i.e. missing stations or channels), there will be one or more folders named with the RT130 serial numbers (e.g. 9388) instead of the desired station name (e.g. SH05). To change any miniSEED headers to correct a station name, network code, etc., use 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 or mark timing issues, and/or to convert the file headers 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 MSEED/
```

data2passcal will scan all subdirectories of the MSEED folder and send any miniSEED files that have the correct file names.

Alternate scenario 1: Quick troubleshooting in the field (check on state of health of station)

In this case, only creation of the logs is required. If needed create a directory list with instructions above. Then run:

```
rt2ms -D dir.list -X
```

for multiple RT130s or

```
rt2ms -d RAW/<cf folder> -X
```

for one RT130. This will create log file(s) in the LOGS directory for you to read with **logpeek**.

Alternate scenario 2: I have a parfile already!

We strongly recommend that you use exploratory mode to generate a parfile, but if you have one you want to keep using, it is possible to adapt it. If you made alterations to the RT130 configuration (e.g., sample rate) and did not document the time, please use exploratory mode. However, if you documented the alterations, or did not make any, you can apply the following changes to a parfile used with previous **rt2ms** versions:

- Remove the # before column labels in the first row
- Add implement_time column label to the label line, and

populate the implement_time column with UTC start time in the format YYYY-MM-DDThh:mm:ss.ssssssZ, for example: 2015-07-10T17:54:58.000000Z