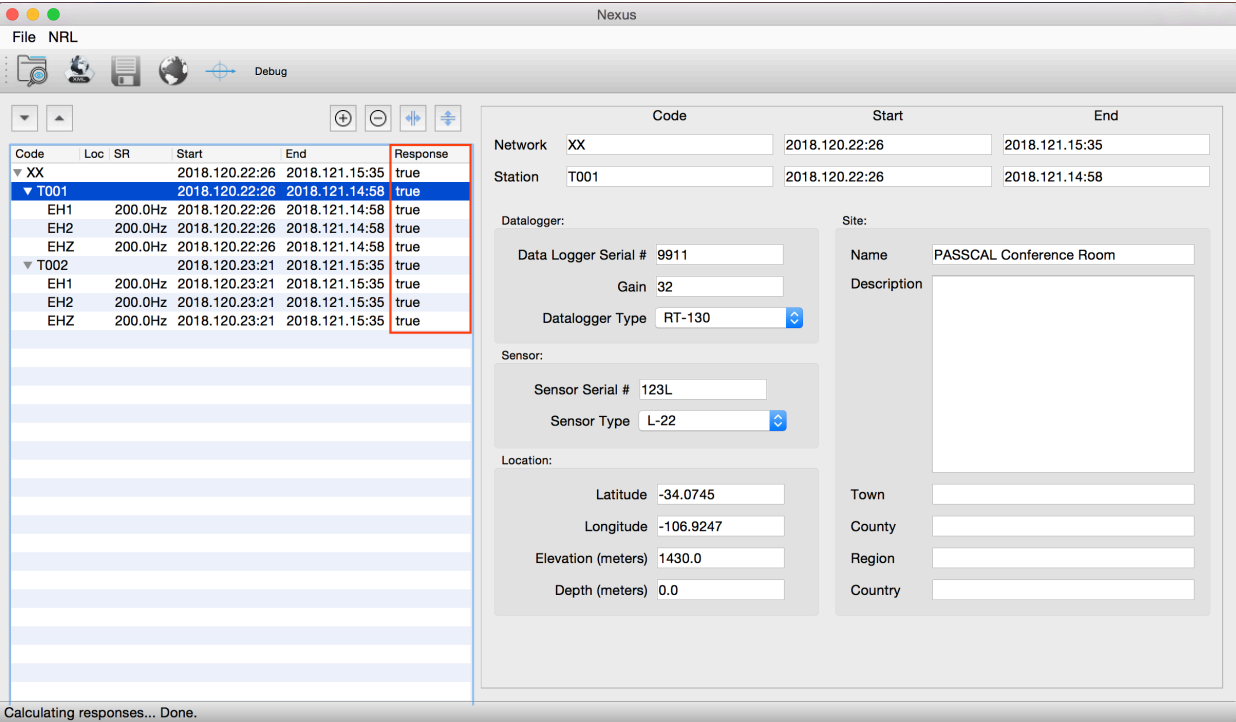For a much more detailed processing guide, please see the PASSCAL document: "Generating SEED From RT130 Data", available from our web page: http://www.passcal.nmt.edu/content/data-archiving

An example of station T001 with complete information filled out. The XML is only complete for use when the Network line contains the word 'true' in the Response column. This means that all responses have been calculated for all stations (highlighted in red).

Last revised: March 19, 2019

RT130 Data Processing with Nexus In a Nutshell

You've offloaded a service run and have stacks of zip files. Now what to do with them? Start by getting organized (Steps 1-2). Then convert the data & log files to miniseed (3-5). Build the metadata in stationXML format (6). Finally, send the day volumes and the dataless to PASSCAL (7). **Unix commands (bold print)** and any command line arguments are highlighted in yellow. Input files are denoted by *<filename>*. GUI options and menus are highlighted gray.

1. Create and maintain an organized directory structure for your data. Start by creating a main directory for the project. If you will have multiple service runs, we recommend creating a series of service run directories within your main project directory. Once the main project directory or service run directory is made, create subdirectories within it for your raw data (RAW) and log files (LOGS). For example: **mkdir** RAW (move your raw files here), **mkdir** LOGS (for your log files).

2. In the main directory, use a text editor to create a parameter file that our tool, **rt2ms**, can parse. Use a text editor and the format to the right to create one from information in your field notes. This example describes three data streams for DAS 9306 at station ME42. The data are 40sps (refstrm 1), 1 sps (refstrm 2), and 0.1sps (refstrm 9, the mass positions).

```
#das; refchan; refstrm; netcode; station; channel; samplerate; gain
9306;   1;        1;      PI;      ME42;  BHZ;     40;           1
9306;   3;        1;      PI;      ME42;  BHE;     40;           1
9306;   2;   1;            PI;      ME42;  BHN;     40;           1
9306;   1;   2;            PI;      ME42;  LHZ;     1;      1
9306;   3;   2;            PI;      ME42;  LHE;     1;      1
9306;   2;   2;            PI;      ME42;  LHN;     1;      1
9306;   1;   9;            PI;      ME42;  VM1;     0.1;          1
9306;   2;   9;            PI;      ME42;  VM2;     0.1;          1
9306;   3;   9;            PI;      ME42;  VM3;     0.1;          1
```
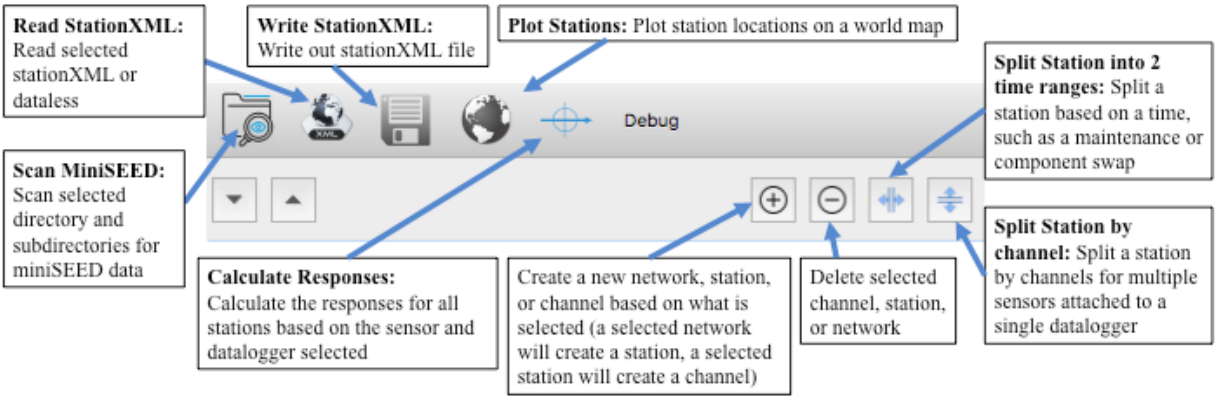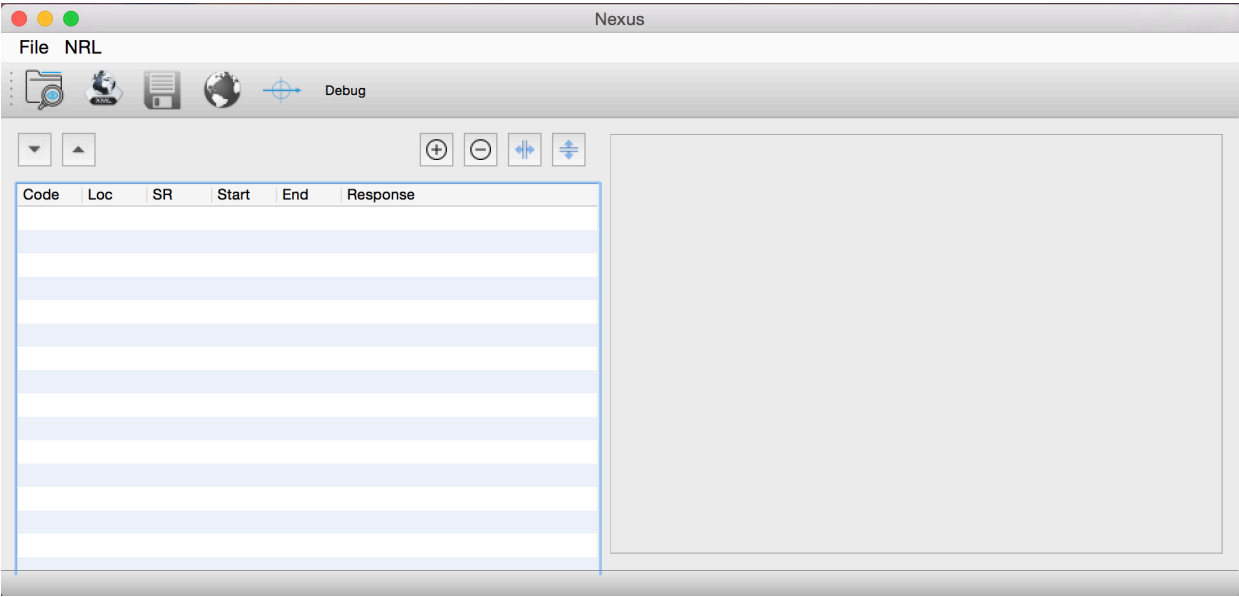
3. In the main project directory, convert the raw RT130 data to miniseed. Typing **rt2ms –h** shows a list of available options. The command shown below is the most general usage:

rt2ms -D RAW –Y –L –o MSEED -p *<parfile>* >& rt2ms.out

The (-D) flag will process all .ZIP files in a specified directory, (-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. Note: The data conversion may take many hours for large data sets. When **rt2ms** finishes, move all of your .log and .err files from the MSEED directory to your LOGS directory.

4. **dataselect** is an IRIS DMC program that allows for the extracting and sorting of miniseed data. This will read the data from the MSEED directory and convert them into day volumes with the accepted naming format:

dataselect -A DAYS/%s/%s.%n.%l.%c.%Y.%j MSEED/Y*/*/*

5. Verify the data quality by reviewing the traces in the DAYS directory (with **pql**) and log files in the LOGS directory (with **logpeek**). Obvious signs of trouble include loss of GPS timing, overlaps, gaps, corrupted files, etc. Make a note of any problems. Use **fixhdr** to confirm the conversion is complete and to correct any problem headers, mark timing issues, and/or to convert the files to big endianess if they aren't already. For more information on how to use these tools, refer to the Appendices at: http://www.passcal.nmt.edu/content/data-archiving/documentation/passive-source and http://www.passcal.nmt.edu/content/pql-II-program-viewing-data

6. Use Nexus to create a stationXML for your experiment. Start by opening the Nexus GUI by typing **nexus** in a terminal window. This will bring up the Nexus GUI:

Buttons in the main Nexus GUI. The Split Station buttons can be used to create new time epochs (for a service run that involved a sensor or datalogger swap) or separate out channels for a station with multiple sensors.

After opening Nexus, scan the miniseed files created by **dataselect** by either selecting the Scan MiniSEED button or going to File > Scan MiniSEED. Navigate to the DAYS folder created by **dataselect** and select Open.

Select each station block and specify station location, datalogger and sensor type and serial numbers, datalogger gain, and station long name. Confirm that all stations listed are named correctly and have the correct channel names. If not, this can be fixed in **fixhdr**.

Once you have finished inputting all station level information including the datalogger and sensor types, select the Calculate Responses button. If the instrument information has been correctly populated for all stations, the values in the Response column will all change to "true". Verify the station locations by selecting the Plot Stations button to bring up a map of the station locations.

After all values in the Response column are true and you have verified your station locations, select the Write StationXML button to save your stationXML file. Name the stationXML file with the network code, experiment name, and today's date. For example: PI.PasscalTestNetwork.2018172.xml

7. Last step: Please drop a note to *data_group@passcal.nmt.edu* before sending the data to PASSCAL so that we can set up a receiving area. Attach the stationXML to this email unless it is larger than 5Mb. You can use our tool **data2passcal** to automatically send the data:

data2passcal *<directory containing miniseed day volumes>*

A few tips…
…Coming soon…