```
#comment: This is a batch file example.

net PI Pier database at PASSCAL

sta WH05 34.0740 -106.9188 1.43 Socorro, NM, USA
time 08/13/2009 00:00:00
datalogger  Passcal_q330_linear  0984
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 04/25/2010 23:59:59

sta WH06 34.6577 -106.9244 1.43 Socorro, NM, USA
time 08/27/2009 00:00:00
datalogger  passcal_q330_linear  0988
sensor cmg40t 0 T4059
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 04/25/2010 23:59:59
```

Last revised: November 12, 2012

Q330 Data Processing In a Nutshell

You've offloaded a service run and have stacks of multiplexed files. Now what to do with them? Start with organization and quality control (steps 1-4). Build the database (5-10) and the dataless (11-12). Then send the day volumes and dataless to PASSCAL (13). Turquoise highlighting indicates a Unix command line entry.

1. Make a directory for the raw data. If you used B14 balers: place the multiplexed (*.ALL) files there.
For example you might start with: `mkdir service-1-raw` then:
`mv /some/path/*.ALL service-1-raw/`
If you used B44 balers then you want to copy the BALER44 directory on the USB drive to your raw directory and rename it to reflect your station, e.g.
`mv –rp /Volumes/BALER44 service-1-raw/station-name.1`
If your data used more than one USB drive you could call the second one station-name.2 (The –p preserves the timestamp on the files.)

2. Create a dbbuild batch file describing every station in your network; use your preferred text editor and name it whatever you like. See the template on page 4 to get started. Be very accurate with your entries – small typos now can cause big headaches later.

3. Verify the data quality by looking at the trace and log files. Use `pql` or `pqlx` for the traces, and `qlog` for the logs (see 'Q330 State of Health (SOH) Channels' at the web link at the end of this document for more info). Look for obvious signs of trouble: incorrect times or dates, loss of GPS timing, gaps, overlaps, or corrupted files. Make a note of any problems.

4. Split the multiplexed files into station subdirectories with the following command line:

```
miniseed2days –v –w
"day_volumes/%{sta}/%{sta}.%{net}.%{loc}.%{chan}.%Y.%j" service-1-
raw/*.ALL >& my_miniseed2days.out
```

Substitute service-1-raw/station-name.?/data for service-1-raw/*.ALL if you have B44 data (the '?' after station-name is a wildcard to grab both USB drives' data).
The above line specifies an organized directory path and the required filename structure. This command also packs the files into station-channel-day-volumes.

5. Check and correct any faulty header information with fixhdr. If each Q330 was programmed to write the headers correctly, you can move on to the next step.
Some actions you can take with fixhdr:
-> view and edit channel names & other header field
-> change byte order to big endianess
-> set the 'timing questionable' flag
-> work in batch mode for large numbers of changes

Please note: If you had to alter any of the headers with fixhdr, you'll need to re-run miniseed2days so as to rename those files before adding them to the database in step 8. You should make a new output directory to keep the revised files separate from the originals; here we call it day_volumes2

```
miniseed2days –v –w
"day_volumes2/%{sta}/%{sta}.%{net}.%{loc}.%{chan}.%Y.%j"
day_volumes >& my_miniseed2days2.out
```

6. Build the Antelope database using dbbuild. Make up a short nickname for the database and call the batch file you made in step 2:

```
dbbuild  -b dbname your-batch-file  >& my_dbbuild.out
```

7. View your database: dbe dbname. Where dbname is the name of your database named in step 6. Scroll through a few tables to be sure it's healthy. If you find problems or inaccuracies, correct the batch file and repeat step 6. At this point you have only a descriptive framework-the trace files are attached next.

8. When you're certain the database is error-free, link in the waveforms:

```
miniseed2db –v day_volumes2/* dbname >& my_miniseed2db.out
```

Please note: If your headers were correctly formatted to begin with, run miniseed2db on your day_volumes folder.

9. Assign calibration values from the calibration table to the wfdisc (created in step 8) by typing: dbfix_calib dbname

10. Verify the correlation of your data and database: dbversdwf –dtu dbname >& my_dbversdwf.out (a good result is: 0 bad files/ 0 bad records). Also run: dbverify –tj dbname >& my_dbverify.out. Check the resulting dbverify.out file for errors. If necessary, fix the batch file and repeat step 6. Hint: The most common errors by far are due to trace or log files with times outside the station description in the batch file.

11. Create the dataless SEED volume with the following naming convention:

```
mk_dataless_seed –v –o NN.YY.db_name.YYYYDOYHHMM.dataless  dbname
```

Where: **NN** is your network code, **YY** is the year of your data, and **DOYHHMM** is the approximate **current** day-of-year-hour-minute.

12. Verify the dataless: seed2db –v NN.YY.dbname.YYYYJJJHHMM.dataless
Note that the dataless is the final manifestation of your meta-data. It's like a menu or an index that you and future users can use to see what data are available. If any station or time range is missing from the dataless, the corresponding data are orphaned and totally inaccessible by anyone. If this should happen, find and correct the error in the batch file, then make another database and create a new dataless.

13. 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. We prefer you use our tool gui_DoFTP. It will pack and send your files automatically, trigger the automated system on our end to begin processing, and save us all a little time and trouble.

A few tips…

Many database errors can be avoided by rounding the start & close times in the batch to 00:00:00 and 23:59:59, respectively. It's better to start early and close late (even by a day or two) to be certain that all data & log files are described in the dataless.

Station changes, such as a new datalogger, sensor or sample rate, **must** be documented in the batch file as shown in: man dbbuild_examples.

For a much more detailed processing guide, please see the PASSCAL document: "Generating SEED From Q330 Data", available from our web page: http://www.passcal.nmt.edu/content/passive-source-seed-archiving-documentation

```
#comment: This is a batch file example.

net PI Pier database at PASSCAL

sta WH05 34.0740 -106.9188 1.43 Socorro, NM, USA
time 08/13/2009 00:00:00
datalogger  Passcal_q330_linear  0984
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 04/25/2010 23:59:59

sta WH06 34.6577 -106.9244 1.43 Socorro, NM, USA
time 08/27/2009 00:00:00
datalogger  passcal_q330_linear  0988
sensor cmg40t 0 T4059
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 04/25/2010 23:59:59
```