IRIS/PASSCAL INSTRUMENT CENTER

GENERATING SEED FROM Q330 DATA USING ANTELOPE FOR STAND-ALONE STATIONS

For PASSCAL Platforms: Mac OSX & Linux
Version 2017 - 115

This document was written and produced by the
PASSCAL Data Group

Last revision 4/25/2017 IRIS PASSCAL Data Group

Please send your feedback, questions, and comments to: data_group@passcal.nmt.edu

Table of Contents

Introduction

This detailed document will guide the data archiver through the process of data archiving using a Linux or Mac OSX operating system. This guide assumes the user has basic Linux/UNIX skills, which are essential for completing the archiving task. We begin this process with the data on a field or local computer and end with the submission of these data to the IRIS PASSCAL Instrument Center (PIC). The data undergo fundamental format verification checks prior to PIC submission of the data for archiving to the IRIS Data Management Center (DMC). The archiving of your data fulfills the principle investigator (PI) responsibilities defined in the PASSCAL Data Delivery Policy (https://www.passcal.nmt.edu/content/general-information/policy/data-delivery-policy).

You will use tools developed by PASSCAL and Boulder Real Time Technologies (BRTT) to create a valid dataless SEED volume (a.k.a. "dataless") and mini-SEED (mseed) station-channel-day files for archiving purposes. Examples of command line usage, short scripts, and definitions of Antelope parameter files (pf) to generate a dataless and manipulate mseed may be found throughout this guide and its appendices.

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

Please take a moment to thoroughly review this guide before you start. If you have any questions please contact: data_group@passcal.nmt.edu.

The steps described below for data processing and archiving (PASSCAL tools and Antelope) are supported on Mac's Tiger, (Snow) Leopard, and (Mountain) Lion as well as Linux Fedora Core platforms. Some help may be available for PASSCAL's tools for Solaris and other flavors of Linux, but BRTT does not support Antelope on Solaris after version 5.3.

Within this document:

Headers, general scripts and commands are in bold.
Command-line usage is highlighted yellow
GUI options or menus are highlighted turquoise
Standard output is italicized.
URLs and email addresses are blue.

List of materials and initial steps

Prior to starting this submittal process you should contact the Data Group and acquire, complete, and/or review:

1. On the PASSCAL website (http://www.passcal.nmt.edu) log in to your account. Once you have logged in, go to the PI home page (http://www.passcal.nmt.edu/pihomepage), select your current experiment and then click on the "Mobilize Project" button. Completion of this form sends a request to the DMC for a network code for your experiment.

Note: These forms alert the IRIS Data Management Center (DMC) of your temporary network and sets up the infrastructure needed for the DMC to accept your data.

2. Review the PASSCAL Data Delivery (http://www.passcal.nmt.edu/content/general-information/policy/data-delivery-policy).

3. Obtain a PASSCAL field laptop or contact the Data Group on how to install the latest PASSCAL software package (PASSOFT) for your platform. US educational organizations can also request their own copies of Antelope from Boulder Real Time Technologies (http://brtt.com/education_and_academic_research.html). If you have a PASSCAL experiment, but do not have a .edu address, please contact the Data Group for instructions on how to obtain Antelope for archiving your project. Please note that starting with Antelope 5.5, you must choose to install the contrib directory when you install Antelope to have access to the datalogger and sensor parameter files.

Notes:
- If you encounter any problems with PASSOFT, submit bug reports to http://www.passcal.nmt.edu/node/add/externalticket or email passcal@passscal.nmt.edu

- BRTT releases patches throughout the year and we recommended that you patch your version of Antelope by running **antelope_update** from the command line.

-If your Antelope license on your PASSCAL field computer has expired, request a new license: **register_antelope**
In the resulting GUI window, put your email address under "e-mail address" and data_group@passcal.nmt.edu under "2nd e-mail address". Also put your project name and the word PASSCAL in the "any comments or special requests" field at the bottom before clicking "Register". The license request will go directly to BRTT, but let the Data Group know if you don't hear back from them.

-The PIC will provide Antelope support for data processing for all PASSCAL experiments as required by an agreement between PASSCAL and BRTT. Please direct your Antelope questions to: data_group@passcal.nmt.edu. Questions regarding further processing such as location of events, etc. are beyond the scope of our data archiving support.

Steps in brief

Data Reduction and Timing Quality control

- Create an organized directory structure for your data
- Create a batch file
- Verify data quality using Q330 log files
- Split the multiplexed files into sta.net.loc.chan mseed files
- Modify headers using fixhdr/change endianness and flag timing (if needed)

Populating the Antelope Database

- Build the Antelope database
- View your database
- Add the mseed day volumes to your database
- Verify the integrity of your database
- Create the dataless SEED volume and verify
- Send the data to IRIS/PASSCAL

Data reduction and timing quality control

Once Antelope is installed you will begin the process of preparing your data and creating an Antelope database for producing a dataless seed volume. Your multiplexed waveform files need to be split into individual station-channel-day files. A station-channel-day file is a file containing a day of data for a given channel from one station.

Create an organized directory structure for your data

PASSCAL suggests using the following directory structure but feel free to customize your directories. Start by creating a main directory for the project. Once the project directory is made, create a subdirectory within it for each service run. Let's call your directory "SVC1" for service run number 1. For each subsequence service run you would simply increment the number to SVC2, SVC3, etc. Create the following directories under the SVC1 directory and organize the files accordingly:

>> mkdir SVC1/RAW – where the multiplexed files offloaded from the baler will be stored (i.e., the *.ALL files)

>> mkdir SVC1/day_volumes – where the station-channel-day de-multiplexed data will be stored

Each service of data should be processed in its own directory for ease of record keeping (i.e. new data versus data already archived), however it is recommended to have only one database to represent your entire experiment (more about the database may be found later in this document).

If you used B14 balers copy the multiplexed (*.ALL) files to the SVC1/RAW directory. If you used B44 balers, copy the BALER44 directory on the USB drive to the SVC1/RAW directory and rename it to

reflect your station.

>> **cp** –rp /Volumes/BALER44 SVC1/RAW/<*stationName*>

(The –p preserves the timestamp on the files). If your data used more than one USB drive you could call the second one <*stationName*>2.

Create a batch file

In the SVC1 directory, use a text editor to create a batch file describing every station in your network. The batch file is an ASCII file with specific keywords and details used to build the database. It is an effective way to keep a history of your experiment and also allows you to reproduce most of your database from scratch, if necessary. Use the following template (next page) as an example and edit the fields in green to reflect the equipment used in your experiment. As with many scripts, comments in a batch are denoted by #. The description for each field in the batch file (and how dbbuild works) may be found in the following man pages: dbbuild_batch, dbbuild, dbbuild_examples, on our web page http://www.passcal.nmt.edu/content/data-archiving/documentation/passive-source, or in the BRTT web manpages found in $ANTELOPE/antelope.html.

In the example below the fields in green are inputs that you need to provide, while the fields in black are parameters required by Antelope (for more details please check our appendices for Passive Source documentation and examples of the different channel naming conventions according to SEED found in the link in the paragraph above). Please refer to the Standard for the Exchange of Earthquake Data Reference Manual SEED Format Version 2.4 for complete details on SEED format: https://www.fdsn.org/seed_manual/SEEDManual_V2.4.pdf

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

net PI Pier database at PASSCAL

sta ME42 34.0745 -106.9247 1.430 Socorro, NM, USA
time 02/06/2011 00:00:00
datalogger  rt130  0984
sensor cmg3t 0 T4476
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 BH1
channel E BH2
samplerate 1sps
channel Z LHZ
channel N LH1
channel E LH2
add

close ME42 12/31/2013 23:59:59

sta ME101 -77.72591 162.26907 0.079 Socorro, NM, USA
time 02/29/2011 00:00:00
datalogger  rt130  0988
sensor l22 0 G071
axis Z 0 180 - 1 32
axis N 0 90 - 2 32
axis E 90 90 - 3 32
samplerate 100sps
channel Z EHZ
channel N EH1
channel E EH2
samplerate 1sps
channel Z LHZ
channel N LH1
channel E LH2
add

close ME101 12/31/2013 23:59:59
```

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

net network code network name

sta stacode lat long elev (km) city, state, country
time config start time – time when you power on or earlier
datalogger  code serial number (code from par file)
sensor code edepth (km) serial number (depth is below
surface)
axis label hang vang [sens [lead [pgain [pstage]]]]
axis label hang vang [sens [dlgain [pgain [pstage [lead]]]]]
axis label hang vang [sens [lead [pgain [pstage ]]]]
samplerate code (appropriate sample rate for the sta)
channel axis label
channel axis label
channel axis label
samplerate code (appropriate sample rate for the sta)
channel axis label
channel axis label
channel axis label
add

close sta time (VERY IMP! Closes sta at this time)

sta stacode lat long elev (km) city, state, country
time config start time – time when you power on or earlier
datalogger  code serial number (code from par file)
sensor code edepth (km) serial number (depth is below
surface)
axis label hang vang [sens [lead [pgain [pstage]]]]
axis label hang vang [sens [dlgain [pgain [pstage [lead]]]]]
axis label hang vang [sens [lead [pgain [pstage ]]]]
samplerate code (appropriate sample rate for the sta)
channel axis label
channel axis label
channel axis label
samplerate code (appropriate sample rate for the sta)
channel axis label
channel axis label
channel axis label
add
```

NOTE: We have not included location codes in the above example. If you need to use location codes use the format shown below:

```
samplerate 200sps
channel Z EPZ  01
channel N EP1 01
channel E EP2  01
```

In this example, 01 is the location code for the EP* channel designations.

However, please note that we discourage the use of locations codes and suggest that they be explicitly

defined only when necessary to avoid ambiguity (such as when operating a dense network of stations within 1 km) or when recording multiple streams at sample rates sharing a common band code (first letter) within the channel code.

NOTE: PASSCAL encourages the use of 1 and 2 in place of N and E in channel names (see 'channel' lines in the batch file example above) 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 batch file is the history of your entire experiment. It records any and all of the changes, additions, removals, etc. of channels, DAS, sensors, sample rates etc. for all the stations and thus MUST include all of these changes, covering the entire time window of your experiment. Yes, this does mean from the moment data acquisition is turned on to the moment it is turned off and all times in between….

We suggest using a start time for each station that is slightly before you turn on the acquisition to assure that all of the traces are included within the meta-data. This will prevent further errors and problems during archiving. The only caveat may occur when you are moving DAS units from one station to another during the course of your experiment. Make sure that the start time for one station/DAS combination does not overlap with the end time of a different-station/same-DAS combination. The same thing applies for sensors, as Antelope knows that a single sensor can't be in the two different places at the same time.

Congratulations! You have now set up the necessary directories and built your batch file. The next step is to verify the data quality using the Q330 log files and split the multiplexed files into station-channel-day mseed files.

Split the multiplexed files into station-channel-day mseed files

The data downloaded from a Q330-B14 datalogger using EZ_baler, or copied to a working directory on your computer from the Q330S/Q330-B44 is a multiplexed file. This multiplexed file is similar to a tar file that contains all the data you downloaded from each baler. Ideally, it should contain all of the waveforms, for all of the recorded sample rates, as well as the state of health (SOH) channels. These multiplexed files usually have the extension "ALL".

If you have not already done so, copy the raw data to the SVC1/RAW directory. If you used B14 balers copy the multiplexed (*.ALL) files to the SVC1/RAW directory. If you used B44 balers, copy the BALER44 directory on the USB drive to the SVC1/RAW directory and rename it to reflect your station.

>> **cp** –rp /Volumes/BALER44 SVC1/RAW/<*stationName*>

(The –p preserves the timestamp on the files). If your data used more than one USB drive you could call the second one <*stationName*>2.

Use the command **miniseed2days** to split the multiplexed baler data files into station-channel-day files. Miniseed2days will create miniseed day volumes utilizing a specific naming convention required for archiving purposes in station-named subdirectories beneath the day_volumes directory. Miniseed2days takes the multiplexed data from the .ALL files in the RAW directory, splits it into miniseed day volumes

and moves the files to station subdirectories in the day_volumes directory – see below.

In your service run directory (SVC1):
FOR a Q330-B14 type the following on the command line:

>> **miniseed2days** -v –w "day_volumes/%{sta}/%{sta}.%{net}.%{loc}.%{chan}.%Y.%j" RAW/*.ALL >& miniseed2days.out

FOR a Q330S/Q330-B44 type the following on the command line:

>> **miniseed2days** -v –w "day_volumes/%{sta}/%{sta}.%{net}.%{loc}.%{chan}.%Y.%j" RAW/<*stationName*>*/data >& miniseed2days.out

Where:

-v specifies verbose mode which will output information about all of the dropped blocks

-w specifies an alternate pattern for the output miniseed volumes. This pattern dictates the way data records are allocated to files. IRIS PASSCAL requires the following format for quality control purposes:

"day_volumes/%{sta}/%{sta}.%{net}.%{loc}.%{chan}.%Y.%j"''

The mseed headers read by **miniseed2days** are the source of information used to populate the database's waveform table and name the files using the information contained in the string above. This long string tells **miniseed2days** to organize the output into a day_volumes directory, and then, within the day_volumes directory to organize the files by station name. Within each station directory the data are organized by sta/network/channel/year/day of year.

Next, we review some example output in which **miniseed2days** did what it was supposed to do, but the output is unexpected. If the subdirectories and/or files written by **miniseed2days** are incorrectly named, you will need to correct the mseed headers using **fixhdr** and rename the files to reflect the corrected fields.

For example, let's say before you went to the field you forgot to program digitizer 0361 with the station name, STN05. This means that the station code written by the digitizer to the mseed headers is the digitizer serial number, 0361. Since **miniseed2days** utilizes the mseed header fields to name the station-channel-day files and subdirectories, either the files and/or the subdirectories may appear to be incorrectly named. Let's take a look at the directories created by **miniseed2days**:

>> **ls** day_volumes/*

0361/ *STN01/*
STN02/ *STN03/*
STN04/

Notice the 0361 directory. This isn't the station name we expected; we are missing the expected STN05 subdirectory, however all of the other directories in this example have the expected station names.

When you see the 0361 directory you realize that the datalogger serial number was assigned as the station code in the mseed headers. Let's take a look at a bit of the 0361 directory.

>> **ls** -d day_volumes/0361/*

*day_volumes/0361/**0361**.XV..01.HHZ.2005.305*
*day_volumes/0361/**0361**.XV..02.LHZ.2005.305*
*day_volumes/0361/**0361**.XV..01.HHZ.2005.306*
*day_volumes/0361/**0361**.XV..02.LHZ.2005.306*

The mseed headers, which define the data, must be corrected before you submit this data to the DMC. And, to avoid confusion later in the archiving process, you must also rename the files and subdirectory. While you can easily rename the files and subdirectory with Linux/UNIX commands and/or shell scripts, the mseed header correction requires specific software, such as **fixhdr**. We introduce **fixhdr** briefly in the next section.

Please see the section "Modify headers using **fixhdr"** (below), and our Appendix document "Fixhdr Help" for more information.

We recommend keeping a backup copy of the corrected station-channel-day files along with the original raw data.

Now let's take a look at how this directory should look after you fix the directory and file names and headers:

>> **ls** -d day_volumes/STN05/*

*day_volumes/STN05/**STN05**.XV..01.HHZ.2005.305*
*day_volumes/STN05/**STN05**.XV..02.HHZ.2005.306*
*day_volumes/STN05/**STN05**.XV..01.LHZ.2005.305*
*day_volumes/STN05/**STN05**.XV..02.LHZ.2005.306*

This partial directory listing shows that the station code STN05 is assigned as the station code in the mseed headers. The correct format for the mseed day volume files is:

Sta.NetCode.Loc_code.Chan.Yr.Julday

Where:

Sta= station name
NetCode= FDSN assigned network code for your experiment
Loc_code= Location code (if applicable) (- if there is no Location Code Antelope will add two dots as placeholders.)
Chan= Channel
Yr= Year the data recorded

Julday= Julian day data describes

A short but useful digression. What is a full miniseed day volume at PASSCAL?

For each day (DOY) of recorded data for each station - a day_volume will include a file for each recorded channel including log/State of Health channels.

Thus, for example, if there is description in the batchfile for a station using an STS-2 sensor and RT130 DAS recording 3 channels (Z,N,E) of 2 data streams (e.g. 40sps and 1sps) a full miniseed day volume for each DOY will include 10 files. The example below shows the day volumes for year 2013 day 056 (where NN will be replaced by your network code):

StationName.NN..BHZ.2013.056
StationName.NN..BHN.2013.056
StationName.NN..BHE.2013.056
StationName.NN..LHZ.2013.056
StationName.NN..LHN.2013.056
StationName.NN..LHE.2013.056
StationName.NN..VM1.2013.056
StationName.NN..VM2.2013.056
StationName.NN..VM3.2013.056
StationName.NN..LOG.2013.056

Verify Data Quality using Q330 Log files

The Q330 Baler creates State of Health (SOH) files, which allow you to ascertain the health of the station. These files may be viewed with **pql**, **pqlx**, and **qpeek**. See our document on "Q330 State of Health (SOH) Channels" within our Appendices section at https://www.passcal.nmt.edu/content/data-archiving/documentation/passive-source. Please refer to this document for more details on SOH channels, specific checks on data quality of waveforms, and other uses of the software such as:

- Timing quality
- Power problems
- System reboots
- Generation of mean station locations and elevations

Modify headers using fixhdr/ Change endianess and flag timing

The PASSCAL software, **fixhdr** allows users to make changes to mseed fixed header values. It also provides a means for you to modify the endianess (byte-order) of your files from little to big (if required). You can also apply global-timing shifts by setting flags for questionable timing, or apply time corrections when necessary. It also has a batch mode (-b option) that can be run with template files

created either by **fixhdr** or from scratch.

To launch **fixhdr** with a GUI (graphical user interface) you need to type on the command line:

>> fixhdr

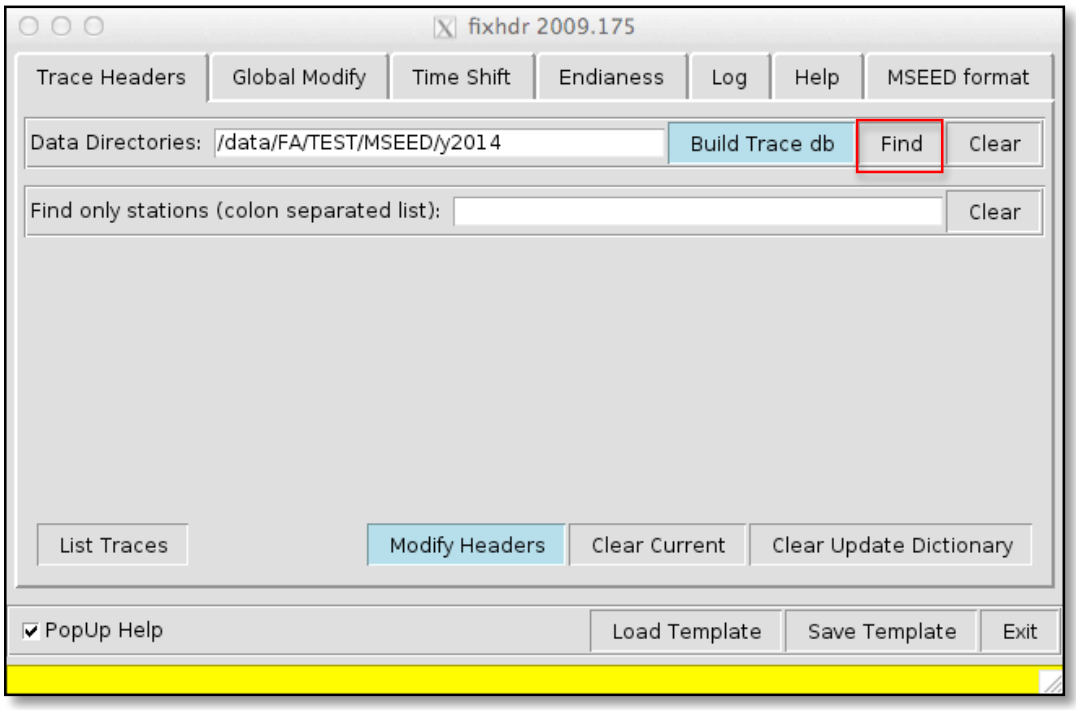
Header fields will need to be modified following the SEED format (see Appendices). Fields that you will be able to modify using **fixhdr** are:

- station name
- network code
- channel
- location code (optional, only if needed),

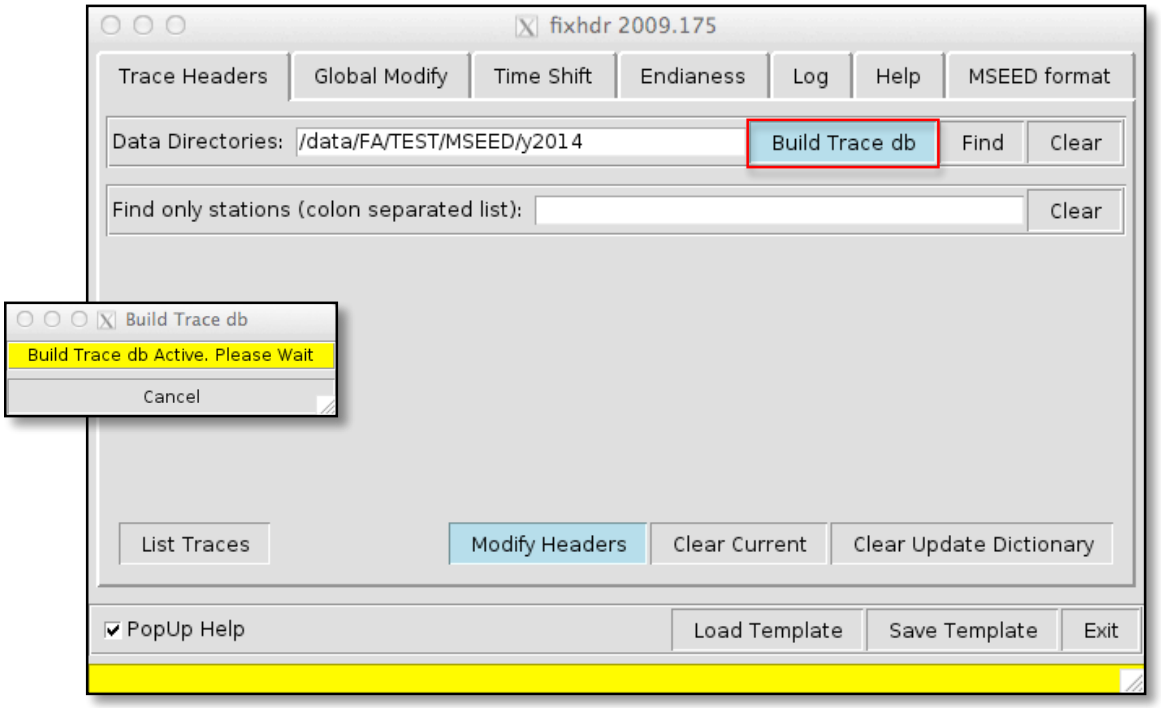
Please refer to our Appendices for suggested channel names for PASSCAL sensors. See also the Standard for the Exchange of Earthquake Data, Reference Manual, (SEED Format Version 2.4 https://www.fdsn.org/seed_manual/SEEDManual_V2.4.pdf) for complete details on the SEED format.

A Help button is available within the program when running the GUI or from the command line by typing **fixhdr** –h.

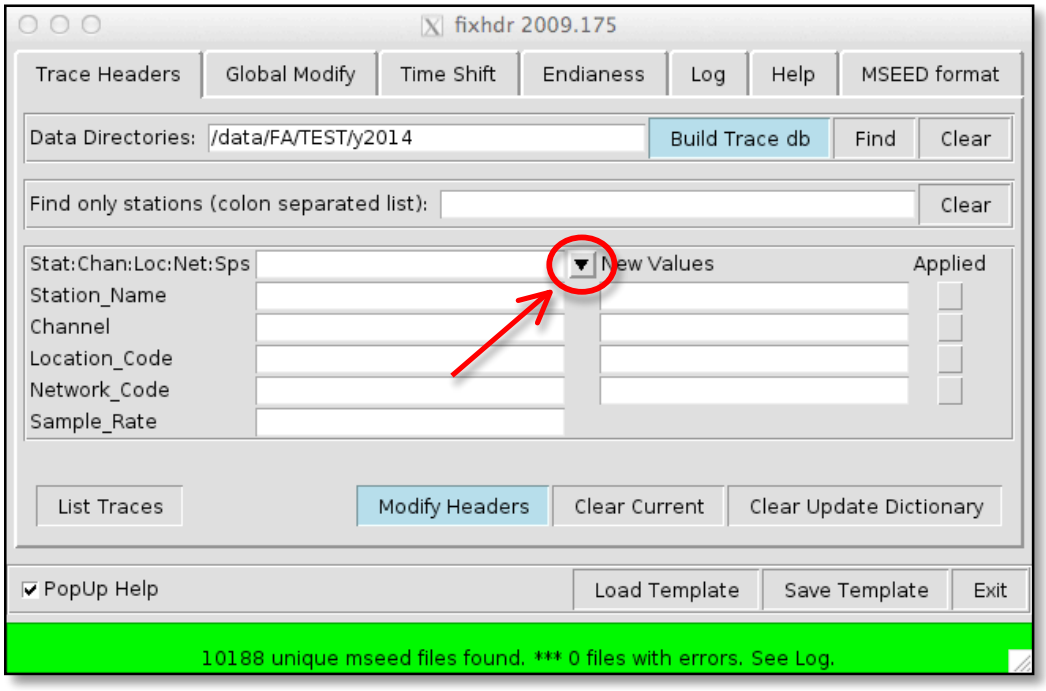
Once the **fixhdr** GUI has loaded, click on Find to navigate to the correct directory.



Build Trace db loads your *.m files. Once you select Build Trace db a window pops up telling you that the trace db is being built. This could take a few minutes depending on how many files you are loading.



Fixhdr has various tabs but the one you will want to use is the default tab, Trace Headers. Click on the downward pointing arrow (see figure below) to show the header values. If these values are not as expected you may change them by inputting the new values and selecting Modify Headers.



More help with **fixhdr** can be found here:
http://www.passcal.nmt.edu/content/data-archiving/documentation/passive-source

If you had to alter any of the header information with **fixhdr**, you'll need to re-run **miniseed2days.** This will correctly rename those files before adding them to the database. To do this you should make a new

output directory and keep the revised files separate from the originals; here we call it *day_volumes2*

>> **miniseed2days** –w "day_volumes2/%{sta}/%{sta}.%{net}.%{loc}.%{chan}.%Y.%j" day_volumes >&miniseed2days2.out

So far we have split the multiplexed files downloaded from the baler into station-channel-day files and fixed any pesky errors in the headers. Next we will build the Antelope database and at the same time populate the database with the metadata found in the headers.

Build the Antelope Database and Populate with Metadata

Using your batch file, run **dbbuild** to create your Antelope database. This program builds the database and populates it with the metadata contained in the batch file. This database is an important part of the archiving process and will be the basis for creating the dataless seed, a file containing all the metadata.

Since your batch file completely describes the configuration of each station in your network it will be the input for the Antelope database. Remember that this step adds only the metadata. The waveforms will be added later. Before running dbbuild you MUST be sure that your batch file is absolutely correct by checking station names, sensor orientation, start times, close statement, etc.

Don't forget, it is best to use a conservative start time for each of the stations (i.e. a little early rather than milli-seconds late). Feel free to turn stations on at 00:00:00 and off at 23:59:59. The time only needs to be more precise when equipment moves from one site to another on the same day.

In the SVC1 directory, build the Antelope database using the batch file you created previously:
>> **dbbuild** -b **<dbname> <batchfile>** >& dbbuild.out

The [–b] flag runs dbbuild in batch mode using the specified batch file.

Output from dbbuild -b for a station called EX01 will look something like the file snippet below.

Added 8 records for EX01 at 9/05/2013 (248) 0:00:00.000
10 calibration records
1 dlsensor records
2 instrument records
1 network records
10 sensor records
1 site records
10 sitechan records
82 stage records
close_station EX01 at 12/31/2015 (365) 23:59:59.900

This process creates a series of tables and two new directories, "response" and "nom_response". An example of these individual database tables and directories created by **dbbuild** are shown below for a

database called TEST.

Batchfile	*TEST*	*TEST.instrument*	*TEST.sensor*	*TEST.snetsta*
dbbuild.out	*TEST.calibration*	*TEST.lastid*	*TEST.sensormodel*	*TEST.stage*
nom_response/	*TEST-dbbuild*	*TEST.network*	*TEST.site*	
response/	*TEST.dlsensor*	*TEST.schanloc*	*TEST.sitechan*	

View your database

Now that you have built a database and filled it with metadata from your batchfile you will want to take a look at it. Use **dbe** (data base explorer) to check that the metadata is as expected.

You will want to use **dbe** to verify that all of your station and channel meta-data is correct. Start by taking a quick look at the site table to ensure that all of your locations are in the correct hemisphere and look at the site chan table to check that all of your channels and on/off dates are correct. There will be no wfdisc table until the waveforms are added a bit later in this tutorial. The database contains only metadata and information from the Antelope responses and instruments directories ($ANTELOPE/responses and $ANTELOPE/current_version/instruments/).

>> **dbe** <*dbname*>

Typing man **dbe** brings up a detailed user guide for the program.

Adding the waveform files to your database

To add your waveforms files to your database, use the command **miniseed2db.** This will add the waveform information to the database in the form of a wfdisc table.

>> **miniseed2db** -v day_volumes/* <*dbname*> >& miniseed2db.out

Verify the integrity of your database

Now you have a database containing all the metadata tables and a waveform table (the wfdisc). The next thing you will need to do is to run some checks between the traces and the database to verify the integrity of the database. The following programs ensure that the dataless completely describes the waveform data and that the database is error free.

In the SVC1 directory, correlate the channel ids between tables by running:

>> dbfixchanids <*dbname*>

Then verify the waveforms in the wfdisc table are described by the metadata:

>> **dbversdwf** –tu <*dbname*> >& dbversdwf.out

This program checks that the times in the wfdisc table agree with the metadata times. Typing **man** dbversdwf gives you a complete description of program usage.

A good result is:
0 bad files
0 bad records

This next step checks the validity of information between the different tables:

> > **dbverify** –tj <*dbname*> >& dbverify.out

Typing **man** dbverify gives you a complete description of program usage. The [-tj] flags specify that the program will perform only the two-table joins and test for the correct number of records. For example, **dbverify** will complain if the number of rows in the wfdisc and sensor tables do not agree or if the number of rows in the sensor and sitechan tables do not agree.

A good result is:
0 problems

If you have any questions about the output of these verifications please e-mail the Data Group at data_group@passcal.nmt.edu before submitting your data and dataless.

Create and verify a dataless SEED file

A dataless SEED file (or volume) is one of the key files that will be archived at the IRIS/DMC (Data Management Center). This file contains all the metadata for your experiment. Once you have verified the database with the programs listed above you are ready to create your dataless SEED volume. From the DB directory (where your database, batchfile, and parameter file reside):

>> **mk_dataless_seed** -o NN.YY.dbname.YYYYDOYHHMM.dataless <*dbname*>

Below is an example of the output you will see when running **mk_dataless_seed**.

perl: schema css3.0 dbpath ./{TEST}
Using existing TEST.snetsta table
Finished building dataless wfdisc
XX.14.TEST.20140571100.dataless truncated to 5128192 bytes

The [-o] flag specifies the name of the output file. IRIS PASSCAL has a naming convention for dataless SEED files that is specified in the command line above. *NN* represents the 2-character network code (in the example above the network code is XX), *dbname* is the name of your database (our example database is called TEST), *YYYY* is the 4-character year that the data was collected, *DOY* is the day of

year you create the file, *HHMM* are the hour and minute that you create the file. Typing **man** mk_dataless_seed gives you a complete description of program usage.

> *A small digression. You can use the following tools to convert from calendar day to day of year or the converse:*
>
> The utility **julday** converts from calendar day to day of year as in the example below.
>
> >> **julday** 03 01 2014
> Calendar Date 03 01 2014
> Julian Date 060 2014
>
> The utility calday converts a date from a day of year to a calendar day as in the example below.
>
> >> **calday** 304 2013
> Julian Date 304 2013
> Calendar Date 10 31 2013

This program generates a dataless using all the information from the database tables created when you ran dbbuild. The output dataless SEED file contains all of the information from your batch file, i.e., the history of your experiment, as well as instrument response information provided by Antelope when you ran dbbuild.

Congratulations! You have created the dataless but you still need to verify it.

To verify the dataless SEED file you will need to run **seed2db.** This program can either be used as a QC tool for verification of the dataless (which is what you will be doing next) OR you can use **seed2db** to generate a database from the dataless. (What a neat trick!).

From the DB directory:

>> **seed2db** –v NN.YY.dbname.YYYYDOYHHMM.dataless

The [-v] flag provides a more verbose output. Typing **man** seed2db gives you a complete description of program usage.

IMPORTANT NOTE: The dataless SEED file must describe the entire data set for your experiment. This means that the dataless must include all service runs of the experiment if the experiment is more than simply an install and remove instruments with no data downloading in the interim. The agreement, or lack thereof, between the dbbuild batch file, the resulting database, the dataless, and the waveforms will be reflected in the availability of the data at the DMC.

Once you have verified that your data are complete, please contact us by sending an email to data_group@passcal.nmt.edu with your network code, experiment name and dates so that we can expect your data. For example: XX, TEST data, 2013-2014. Also, include an estimate of how much data you are sending (number of Gb, number of files). Please include the dataless with the email as an attachment unless it is larger than 5 Mb.

Send Data to PASSCAL

The latest tool to send your data to PASSCAL is a program called **data2passcal**. Until the program is available in the PASSFOT release (it is not as of 10 June 2015), this program can be downloaded from the PASSCAL Cloud: https://cloud.passcal.nmt.edu/index.php/s/u2ISyAWjVqMqSC6
Install the program in your /opt/passcal/bin directory and make it executable.

>> **sudo mv** data2passcal /opt/passcal/bin/.
>> **sudo chmod** +x /opt/passcal/bin/data2passcal

Type **data2passcal** <*directory that has the miniseed files*> to send all the data in the specified directory to PASSCAL via ftp. A complete status of the ftp transfer is kept in a log file, *data2passcal.log,* in the current directory. Sample output from the *data2passcal.log* is shown below. Notice how the program scans the input directory and keeps track of what has been sent. If there is a connection problem the program simply picks up again where it left off once the connection has been reestablished. If the program is killed or died, restarting it will cause it only to send data that have not been sent yet. Type **data2passcal** –h to see a list of option flags.

[GCC 4.2.1 (Apple Inc. build 5664)]
2014-02-27 10:00:46 -0700 - DEBUG - CWD: /Test/SVC1/day_volumes
2014-02-27 10:00:46 -0700 - INFO - Version: 2014.008
2014-02-27 10:00:46 -0700 - INFO - Scanning: /Test/SVC1/day_volumes/XX01
2014-02-27 10:00:46 -0700 - INFO - Total Size = 944.191 MB
2014-02-27 10:00:46 -0700 - INFO - Total Files = 252
2014-02-27 10:00:46 -0700 - INFO - Total Dirs = 0
2014-02-27 10:00:46 -0700 - INFO - Scan time = 0.01s
2014-02-27 10:00:50 -0700 - INFO - MiniSEED files: 252
2014-02-27 10:00:50 -0700 - INFO - Other files: 0
2014-02-27 10:00:50 -0700 - INFO - Removing files already sent to PASSCAL
2014-02-27 10:00:50 -0700 - INFO - 0 miniSEED files have already been sent, not resending.
2014-02-27 10:00:50 -0700 - INFO - Sending MSEED files to PASSCAL
2014-02-27 10:00:50 -0700 - INFO - Sending 252, 944.191 MB files to PASSCAL
2014-02-27 10:00:50 -0700 - INFO - Connecting to FTP host qc.passcal.nmt.edu from 129.138.26.58. Attempt 1 of 10080
2014-02-27 10:00:50 -0700 - INFO - Success: Connected to PASSCAL FTP
2014-02-27 10:00:51 -0700 - ERROR - Failed to send file N24I.ZL..BHE.2012.292.
2014-02-27 10:00:51 -0700 - ERROR - [Errno 35] Resource temporarily unavailable
2014-02-27 10:01:21 -0700 - INFO - Connecting to FTP host qc.passcal.nmt.edu from 129.138.26.58. Attempt 1 of 10080
2014-02-27 10:01:21 -0700 - INFO - Success: Connected to PASSCAL FTP

Verifying archived data

Once **data2passcal** has delivered all of the data to our system, it will run through our verification software. The data are submitted to the IRIS/DMC only when the waveform data and the dataless file pass a series of checks. This process may take between one to two weeks depending on how much data

are flowing through PIC and to the DMC. Once the data are sent to the DMC, the waveforms and meta-data are read and loaded into an ORACLE database and the waveforms are archived. When we have confirmation from the DMC that your data have been archived we will send you an e-mail and a summary of what data has been archived. Please take a moment to ensure this summary agrees with your records and that all of the data you expect to have been archived is actually archived.

Adding more data: future service runs

A typical question from a data archiver is: "I have more data from the last service run. Is there a way to add the new data to the existing database? "

The answer is yes. You just need to be consistent, do the initial quality control on your data and follow the same steps previously described. If during this new service run changes have been made to the initial configuration of your stations, make sure those changes are also included in the batch file and, therefore, in your database. Below are some examples of what to do in each case:

To add new data to the existing database you will follow the same steps as before with some slight variations. Follow the same steps for data reduction and timing quality control as you did for previous services. Make sure to be consistent with the use of location codes, network and channel assignment, etc when fixing headers. Sending data to PASSCAL will be the same as well. Below you will find information to considering while populating the database during later services.

1. Data Reduction and timing quality control – same as before
2. Populating the Antelope Database for service runs

    a. Update the Batch File (if needed)

    At this point you already have a batch file. You may need to update or modify your batch file if any of the following situations apply:

        i. NEW STATIONS - you need to add each new station with its proper configuration to the batch file and re-run **dbbuild** in the same directory where you create the database the first time.

        ii. REMOVED STATIONS – if there is any existing data for this station you simply add a close statement (e.g. if the station NP00 was removed April 10 2006, use “close NP00 04/10/2006 23:59:59”). If data was never recorded for this station there’s no need to add it to the batch file.

        iii. CHANGED sensor, digitizer, sample rate, gain or fix orientation – in this case you will add an extra block describing the same stations with the modified fields below the first description. The start time of the second configuration will be the end time for the initial configuration.

IMPORTANT: IF THERE ARE NO MODIFICATIONS (different sensor type and/or serial number, digitizer, gain, sample rate, orientations): there is NO need to re-run dbbuild since your

stations are already accurately described in your database. There's also no need to build a new dataless as no new meta-data information has been added.

a. Building the Antelope Database

If any one of the three points above is applicable, then you will need to update your database with **dbbuild** as shown below:

>> **dbbuild** -b <*dbname*> <*updatedBatchFile*>

b. View your database – same as before

c. Add your waveforms to the database

Once you have the new data ready to add to the database (QC complete, timing issues evaluated, headers fixed, etc), you may add the waveform information to the database using miniseed2days (as before), however be sure to specify the location of the new service run's station-day-volume. Let's assume you have it under a service 2 folder named SVC2, then you will run:

>> **miniseed2days** –d <*dbname*> –w "SVC2/day_volumes/%{sta/%{sta}.%{net}.%{loc}.%{chan}.%Y.%j" SVC2/RAW/*.ALL >& miniseed2days_service2.out

Where SVC2/day_volumes delineates that these are the waveforms for your second service run.

Once you have added the waveforms to your database you will still need to run verifications on the database.

d. Verify the integrity of your database – same as before

e. Create a new dataless SEED volume if you ran **dbbuild** (revised or rebuilt the database)

f. Verify your dataless file and rename to conform to convention

Updating the meta-data without processing new data

All changes (changed/new stations/removed stations) in your network configuration must be described within your dataless. This dataless must be submitted to PASSCAL for review so the appropriate changes are visible for data and meta-data requests at the DMC. Meta-data/dataless changes may occur at any time including between service runs and after an experiment is complete.

There are a couple of ways to update your database and dataless. One clean way to add to or change a dataless is to simply create a temporary database in a separate directory and generate a dataless within it. The steps you should follow are:

1. Create a temporary directory to work on your new dataless (e.g. **my_new_dataless**)

>> **mkdir** my_new_dataless

2. Copy your existing batch file to the temporary directory and rename it.

>> **cp** <*batchFile*> my_new_dataless/<*modifiedBatchFile*>

3. Edit it.

4. Create a new database by using **dbbuild.**

>> **dbbuild** –b **<***modified_db*> <*modifiedBatchFile*> >& dbbuild.out

5. Check the database with **dbverify.**

>> **dbverify** –tj <*modified_db*> >& dbverify.out

6. Fix any errors, if you have any questions please e-mail data_group@passcal.nmt.edu

7. Create a new dataless, which will contain all of the information that needs to be incorporated into the database.

>> **mk_dataless_seed** -o NN.YY.dbname.YYYYDOYHHMM.dataless <*modified_db*>

8. Verify your dataless:

>> **seed2db** -v NN.YY.dbname.YYYYDOYHHMM.dataless

9. Contact the data_group@passcal.nmt.edu regarding how to submit the updated dataless.