IRIS/PASSCAL INSTRUMENT CENTER

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

PASSCAL Platforms: Mac OS10 & Linux

Version 2012-286

This document was written and produced by the

PASSCAL Data Group

Please send your feedback, questions, and comments to:

data_group@passcal.nmt.edu

Table of contents

List of appendix for data & software

Name/Short description

1. ***Data Specific***

- ***Data Delivery Policy:*** *Requirements on Data Delivery for PASSCAL*
- ***SEED Format:*** *Brief description of seed format for archiving of passive source data*
- ***Q330 State of Health Channels (SOH):*** *Description of SOH in a Q330*
- ***PQL,qlog:*** *Evaluation of ACE,OCF,LOG files using PQL,qlog (Q330)*
- ***Troubleshooting:*** *Common problems when processing data for archiving*

2. ***Software Specific***

- ***Antelope:*** *How to update and install Current Version Notes*
- ***How to build a Batch file –PASSCAL'S Antelope guide & examples:*** *Details on how to generate batch file in antelope*
- ***PASSCAL & contributed software guide:*** *Links to main software and brief detail on most used tools for data processing/archiving*
- ***FIXHDR Help:*** *Detail instructions on using fixhdr*
- ***Gui_DoFTP:*** *Detail Instructions on using gui_DoFTP to send data to PASSCAL*
- ***Troubleshooting some basic antelope tools during data processing***
- ***Other useful tools:*** *Common problems software related & other useful tools*

Note: These documents are available online http://www.passcal.nmt.edu/content/passive-source-seed-archiving-documentation or by request to the data group data_group@passcal.nmt.edu

GENERATING SEED FROM Q330 DATA USING ANTELOPE
(Version 2012-286)

Introduction

This detailed document serves to guide the data archiver through the process of data archiving utilizing a Linux or Mac OSX operating systems. 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 submitted data undergo fundamental quality assurance checks prior to PIC submitting the data for archiving at the IRIS Data Management Center (DMC). Individuals utilizing the Solaris operating system will find this guide helpful though specific details, such as software installation, for example, may differ. The archiving of your data fulfills the principle investigator responsibilities defined in the PASSCAL Data Delivery Policy (APPENDIX_DATA).

You will use tools developed by PASSCAL and Boulder Real Time Technologies (BRTT: Antelope) to create a valid dataless SEED volume (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.

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) work on most platforms.

Notice that:

- 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.
- Important notes are **brown**.

List of materials and initial steps

Prior to starting this submittal process you should contact the data_group@passcal.nmt.edu and acquire, complete, and/or review:

1) Principal Investigator web page (new) http://www.passcal.nmt.edu/pihomepage
   Create an account and follow your experiment.
2) Mobilization and Network code request form, best done through PI Home Page
3) Demobilization form, also done through PI Home Page
   (To be completed by the end of the experiment)

4) PASSCAL Data Delivery Policy is found online at:
   http://www.passcal.nmt.edu/content/general-information/policy/data-delivery-policy

5) ANTELOPE – Notes on the current release are online as well as the guide for updating and installing new versions.

6) Install the latest PASSCAL software package for your platform, which may be found at:
   http://www.passcal.nmt.edu/content/software-resources

7) Submit a bug report or problem to: http://www.passcal.nmt.edu/node/add/externalticket

   Note: The forms in step 2 & 3 alert the DMC of your temporary network and setup the infrastructure needed for the DMC to accept your data.

PASSCAL field computers loaned to the PI are shipped with PASSCAL software, with the most current version of Antelope pre-installed. However, you may need to update the version of Antelope on the computer if it has been in field for more that one year. Additionally, BRTT releases patches throughout the year and it is recommended that you patch your version of Antelope by running **antelope_update** from the command line. For more information about Antelope visit http://www.brtt.com/ and/or read the man pages.

New versions of Antelope are usually released in the spring or summer each year. You should check for the most recent version at http://www.brtt.com/. If you have an old version of Antelope please fill out the proper form under http://www.iris.edu/manuals/antelope_irismember.htm. If your institution is not an IRIS member and you will process data from a PASSCAL experiment, please contact data_group@passcal.nmt.edu and we will process your license request.

NOTE: the PASSCAL Instrument Center will provide Antelope support for all PASSCAL experiments, as required by an agreement between IRIS and BRTT. Please direct all Antelope questions related to data archiving to data_group@passcal.nmt.edu. Questions on further processing such as location of events, etc are beyond our main experience with Antelope.

Some of the software you will use originates at PASSCAL, thus please refer to our documentation web page http://www.passcal.nmt.edu/content/passive-source-seed-archiving-documentation for more detailed information.

Directions

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
- Create mseed day volumes and add them to your database
- Assign calibration values from calibration table to wfdisc
- Verify the integrity of your database
- Create the dataless SEED

Verify the dataless

Send Data to IRIS/PASSCAL

Steps in detail

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

You may generate your own directory structure and organize it as you see fit. The following is PASSCAL's suggested structure. Let's call "EXPT" the directory where you have all the data. Create the following directories under the EXPT directory and organize the files accordingly:

<my_cpu: EXPT > mkdir RAW –where the multiplexed files offloaded from the baler will be stored (i.,e., the *.ALL files)
<my_cpu: EXPT> mkdir 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).

Create a batch file

The batch file is an ASCII file with specific keywords and details used to build the database without the use of the GUI. It is an effective way to keep a history of your experiment and allows you to reproduce most of your database from scratch, if necessary. Effectively, the batch file is the history of all the changes, editions, removals, etc. done to the stations in your network, so it MUST include all of the details of each station from the first sample rate on any channel, to the day the station closed.

Use the following template below as an example and edit accordingly the fields in green. A batch file may have comments denoted by #. The description for each field in the batch file (and how dbbuild works) may be found in the man pages **dbbuild_batch**, **dbbuild** and **dbbuild_examples** or in our appendices.

In the example below the fields in green are inputs that you need to provide, while Antelope recognizes the fields in black. Header fields will need to be modified following SEED format. For more details please check our appendices for Passive Source documentation and example of different channel naming conventions according to SEED. Please refer to the Standard for the Exchange of Earthquake Data Reference Manual, SEED Format Version 2.4 (<http://www.irisedu/manuals/>) for complete details on SEED format.

NOTE: In the example below we don't include location codes. If you prefer to use location codes you should have something like:

samplerate 200 sps
channel Z EPZ 01
channel N EPN 01
channel E EPE 01

where: 01 is the location code for data stream 1. For more information, see the dbbuild man page for the proper use of location codes.

```
#comment: This is a dbbuild batch file.
net PI Pier database at PASSCAL

sta WHO5 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

sta WH06 34.6577 -106.9244 1.43 Socorro, NM, USA
time 02/29/2004 02:57:57
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 200sps
channel Z HHZ
channel N HHN
channel E HHE
samplerate 1sps
channel Z LHZ
channel N LHN
channel E LHE
add

close WHO5 04/25/2010 23:59:59
close WH06 04/25/2010 23:59:59
```

```
#comment: This is a dbbuild batch file example
net network code network name

sta stacode lat long elevation (km) city, state, country of sta
time config start time ← time when you power on
datalogger code serial number <--code from par files
sensor code edepth serial number ← depth below surface
axis label hang vang [sens [lead [pgain [pstage]]]]
axis2 label hang vang [sens [dlgain [pgain [pstage [lead]]]]
axis3 label hang vang [sens [lead [pgain [pstage]]]]
samplerate code ← appropriate sample rate for your sta
channel axis label
channel axis label
channel axis label
samplerate code ← appropriate sample rate for your sta
channel axis label
channel axis label
channel axis label
add ← adds the current configuration to the database

sta staname lat long elevation (km) city, state, country of
sta
time config start time ← time when you power on
datalogger code serial number <--code from par files
sensor code edepth serial number ← depth below surface
axis label hang vang [sens [lead [pgain [pstage]]]]
axis2 label hang vang [sens [dlgain [pgain [pstage [lead]]]]
axis3 label hang vang [sens [lead [pgain [pstage]]]]
samplerate code← appropriate sample rate for your sta
channel axis label
channel axis label
channel axis label
samplerate code ← appropriate sample rate for your sta
channel axis label
channel axis label
channel axis label
add ← adds the current configuration to the database

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

We discourage the use of location 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.

It is important to always remember that your batch file is the history of all the changes, editions, removals, etc done to the stations on your network, so it MUST include all of these changes, covering the time frames from the very first sample rate on any channel, to the day the station is closed.
We suggest a slightly earlier time for the start time (second line after station in the batch file) to assure that all of the traces are included within the meta-data. This will prevent further errors and problems during archiving.

Please read this manually carefully and for more details on specifications of each parameter please refer to our appendix on "How to Build a Batch File and Examples" located on our website.

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 **qlog**. See our document on "Q330 State of Health (SOH) Channels" within our Appendices section at www.passcal.nmt.edu/content/passive-source-seed-archiving-documentation. 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

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

If you are working with a standard Q330-B14 datalogger and have downloaded the data using EZ_baler, or if you are using a Q330S/Q330-B44 you have copied your directory onto your working computer, then most likely you have your data as a multiplexed file. This multiplexed file is sort of like a tar file that contains all the data you requested to download from each baler. Ideally, it should contain all the waveforms, for all recorded sample rates in addition to the state of health (SOH) channels. These multiplexed files usually have the extension "ALL".

For better organization, please move your multiplexed baler data, the .ALL files, into the directory RAW (where my_path is the directory where you have downloaded the raw data files).

<my_cpu: EXPT> mv my_path/*.ALL RAW/

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.

FOR a Q330-B14 on the command line type the following:

<my_cpu:EXPT>**miniseed2days -v –w "day_volumes/%{sta}/%{sta}.%{net}.%{loc}.%{chan}.%Y.%j"** RAW/*.ALL >& my_miniseed2days.out

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

<my_cpu:EXPT>**miniseed2days -v –w "day_volumes/%{sta}/%{sta}.%{net}.%{loc}.%{chan}.%Y.%j"** data/* >& my_miniseed2days.out

Where:

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

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

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

RAW Is the name of your input directory where your multiplexed data file(s) live

data Is the name of your input directory where your multiplexed data file(s) live

NOTE: If you would like to avoid typing the long option for the file description with the –w option, please check details on the appendix documentation on how to modify the miniseed2days.pf .

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 not named what you expected, then you will need to correct the mseed headers using **fixhdr** and appropriately rename the files to reflect the corrected fields.

For example, let's say before you went to the field you unintentionally did not program digitizer 0361 with the desired station name, STN05. Therefore 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 review the output of a few **ls** commands:

<my_cpu:EXPT> **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.

Using **ls again**, this time to view the contents of the 0361 directory, also determines that the datalogger serial number is assigned as the station code in the mseed headers. Below there is the command line and a subset of the **ls** output.

<my_cpu:EXPT **ls -d** day_vol/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 submission to the DMC. The files

and subdirectory must also be renamed to avoid confusion later in the archiving process. While the file names and subdirectory may be renamed with some Linux/UNIX commands and shell scripts, the mseed header correction requires specific software, such as **fixhdr**. We introduce **fixhdr** later in this document after a review of the LOG files generated by each station.

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

It is recommended to keep a backup copy of the corrected station-channel-day files along with the original raw data.

For a comparison, lets use another ls, but this time to view the contents of a correct directory, say for STN02. Typing the following command we see:

<my_cpu:EXPT **ls -d** day_vol/STN02/*

*day_volumes/STN02/**STN02**.XV..01.HHZ.2005.305*
*day_volumes/STN02/**STN02**.XV..02.HHZ.2005.306*
*day_volumes/STN02/**STN02**.XV..01.LHZ.2005.305*
*day_volumes/STN02/**STN02**.XV..02.LHZ.2005.306*

Above we see that for the station code STN02 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)
Chan= Channel
Yr= Year the data recorded
Julday= Julian day data describes

> **NOTE:** Notice the –w option is used to define the directory, subdirectory structure and filename convention for each new station/network//location_code/channel/year/julian day. You could customize the parameter file miniseed2days.pf to use, by following the steps described in the Appendices section on our website.

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). In addition to this you can apply global-timing shifts by setting flags for questionable timing, or to 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. Typing **fixhdr** on the command line

launches the program.

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

<my_cpu: EXPT >**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, channel, location code (optional, only if needed), and network code. 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 http://www.iris.edu/manuals/) for complete details on the SEED format.

Help is available within the program and may be viewed by choosing the help button in the GUI or by running **fixhdr** with the "–h" option. For additional information on fixhdr, please see our document "Fixhdr Help" in our Appendices section at https://www.passcal.nmt.edu/content/passive-source-seed-archiving-documentation.

Populating the Antelope Database

Build the Antelope Database

Now that you have a batch file you may run **dbbuild** to create your Antelope database.

<my_cpu:EXPT **dbbuild -b your-database your-batch-file** >& my_dbbuild.out

IMPORTANT NOTES:

- The configuration for each station in your batch file must agree with the mseed headers.
- The batch file filename should not end with a ".pf" suffix.
- Before running dbbuild, please make sure that your batch file is absolutely correct by checking station names, location codes (if you have used any), sensor orientation, start times, close statement, etc.
- It is best to start your stations a few seconds or minutes early, rather than milli-seconds late.

Below is a subset of output from **dbbuild -b** (in the above example written to dbbuild.out).

loading your-batch-file_bf
Added 20 records to calibration
Added 2 records to instrument
Added 1 record to network
Added 20 records to sensor
Added 1 records to site
Added 20 records to sitechan
Added 38 records to stage

By running **dbbuild,** a series of tables and a new directory "response" directory are created (there is a second directory created by Antelope 5.0 and later called "nom_response"). These tables and

directories are the constituents of the database.

<my_cpu:EXPT> ls

my_db.instrument
my_db.sensor
my_db.stage
my_db.lastid

my_db.site
my_db.network
my_db.sitechan
my_db.calibration

my_db.schanloc
my_db.snetsta
response/

View your database

Using **dbe** (a viewing and editing GUI), you may visualize the contents of your database. At this point there are no waveforms, only the information populated into the database from the batch file and the responses directory (/opt/antelope/current_version/responses and opt/antelope/current_version/instruments). To see the contents of your database type the following:

<my cpu> **dbe** my_db

dbe is a general purpose tool for examining, exploring, and editing Antelope relational CSS databases. For a detailed description on how to use **dbe** please see the manual page for **dbe** (“**man dbe**”)

Create mseed day volumes and add them to your database

By running **dbbuild**, you now have a database that describes your network, however you have not associated any waveforms with the meta-data.

When adding the data to the database you can use two tools depending on the output files when running **miniseed2days** the first time (page7), when de-multiplexing the files.

- If you had to modify the header of the files using fixhdr you will need to run **miniseed2days** on those files so the new files have the proper headers and file name convention.
-

<my_cpu:EXPT >**miniseed2days –v –d my_db –w "day_volumes2%{sta}/%{sta}.%{net}.%{loc}.%{chan}.%Y.%j"** day_volumes/*m >& my_miniseed2days2.out

Where:
day_volumes is the directory where you have the miniseed files with proper headers.

-d Runs **miniseed2db** on the output files to create the wfdisc table and thus add your waveforms to the database

You DO NOT need to run this step IF your headers were correct and the filename is already in this format!

- If the headers of your files were correct already (meaning the Q330 was programmed with the proper net code, station name, channel and location code, and therefore the miniseed day volumes have the right filename convention and are ready to be added to the database) then you can run **miniseed2db** as specified below:

To add your waveforms details to your database, use the command **miniseed2db.** This will create the miniseed day volumes (from your header-corrected mseed files in the ref_mseed directory) and create an extra table for your database with the information regarding the waveforms called *my_db.wfdisc*:

<my_cpu:EXPT > **miniseed2db -v** day_volumes/* my_db >& my_miniseed2db.out

IMPORTANT**:**

- 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 data for the station (sta), network (net), location code (loc), channel name (chan), year of the data (Y) and julian day (j) as in: day_volumes/%{sta}/%{sta}.%{net}.%{loc}.%{chan}.%Y.%j"
- You must ensure the mseed headers in the station-channel-day files produced by **miniseed2days** are correct. If the database (and its batch file) does not describe all of the data then errors will result when we check the consistency of the database.
- If you would like to avoid typing the long option for the file description with the –w option, please check our additional documentation in the appendixes on how to modify the miniseed2days.pf .

Assign calibration values from calibration table to the wfdisc table

To ensure that the calibration values are incorporated into the newly created wfdisc table, please run **dbfix_calib**. This step is performed for the integrity of the database and correlates information from the data (waveforms) with the dataless.

<my_cpu:EXPT > **dbfix_calib** my_db

Verify the integrity of your database

Before you create a dataless you will want to ensure your meta-data completely describes the waveform data and your database is free of errors. **Read** the man page on **dbversdwf** and **dbverify** for details regarding tests you may run on your database. Examples of suggested tests are:

<my_cpu: EXPT > **dbversdwf -dtu** my_db >& my_dbversdwf_output
0 bad files
0 bad records

<my_cpu: EXPT > **dbverify –tj my_db** >& my_dbverify.out
0 problems

Common errors associated with these commands can be found in our appendix called "Troubleshooting and identifying errors in basic data processing with antelope-Archiving oriented".

Please read the outputs and if you still have any questions about them, feel free to e-mail the datagroup at data_group@passcal.nmt.edu before submitting your data and dataless.

Create the dataless SEED volume

The dataless SEED volume, often referred to as a "dataless", contains the meta-data describing the station and instrumentation of your experiment. To generate the dataless SEED volume, run **mk_dataless_seed**, which builds the dataless from the contents of your experiment's database. You will submit this file along with the waveforms to PASSCAL.

<my_cpu:my_experiment> **mk_dataless_seed –v –o** PI.04.my_db.20042082000.dataless my_db
Using existing my_db.snetsta table
Finished building dataless wfdisc
PI.04.my_db.20042082000.dataless truncated to 24576 bytes

Using the option -o you can provide the name of the dataless-seed filename to use. Please use the following naming convention:

NN.YY.dbname.YYYYJJJHHMM.dataless

Where:

NN is your network code
YY is the year of your data
YYYYJJJHHMM is the approximate current time – year-julian-day-hour-minute
dbname is the name of your database
my_db is the name of your database

To convert from calday to Julian day, for example March 1, 2007:

<my_cpu> **julday** 03 01 2007
Calendar Date 03 01 2007

To find the current julday:

<my_cpu> **julday**
Calendar Date 03 01 2007

To convert from Julian day to calendar day, for example day 150 of 2006:

<my_cpu> **calday** 150 2006

Calendar Date 05 30 2006

Verify the dataless

Now you may check the structure of the dataless with **seed2db:**

<my_cpu: EXPT > **seed2db –v** my_db_dataless_seed

Where:

my_db_dataless_seed is the name of your dataless, i.e., PI.04.my_db.20042082000.dataless (from the example above)

IMPORTANT: the dataless must describe the entire data set, including all service runs of data. The agreement, or lack thereof, between the dbbuild batch file, the resulting database, the dataless, and waveforms will be reflected in the availability of the data at the DMC.

Send Data to IRIS/PASSCAL

stadayvols

When ready to submit the data to PASSCAL, please make sure that you all of your data are complete and the dataless fully describes the network. One simple way to verify all data for each station/Julian day is complete is by running the PASSCAL tool called **stadayvols,** which is part of the PASSCAL software release.

<my_cpu:EXPT> stadayvols -d my_db -f day_volumes_S >& stadayvols.out

What is a full miniseed day volume?

At PASSCAL, a full miniseed day volume contains every data stream and state of health channel defined on the dataless per station and Julian day. Thus, if in a dataless there is description for a STS2/Quanterra - Q330 station with 2 data streams (e.g. 40sps and 1sps) recorded in 3 channels each (Z,N,E => BHZ,BHN,BHE & LHZ,LHN and LHE respectively) from day 001 2010 until day 365 2010, a full miniseed day volume expected for each Julian day will include:

Full miniseed day volume STS2/Q330 = BHZ + BHN +BHE + LHZ+LHN+LHE +SOH channels (LCE LCQ VCO VEA VEC VEP VKI VM1 VM2 VM3 VPB OCF LOG ACE).

*What is the purpose of running **stadayvols** before sending the data to PASSCAL?*

stadayvols - Determines what files in the day_volumes directory (or your input directory name) can create complete station day volumes (vols). The channels needed for completeness are taken from the css db "db" which must have sitechan, snetsta, and chanloc tables. If outdir is given, complete station day volumes will be created in directory outdir. If -i is used incomplete station day volumes will be created as well.

When using stadayvols to send data there is NO need to use the –i option, the purpose of this tool is for verification of what is complete or what is missing before sending data to PASSCAL for archiving. By typing in the command line stadayvols –h you will get the usage of this tool:

usage: StaDayVols.py [options]
options:
-h, --help show this help message and exit
-v Verbose; will also describe complete Vols
-d DB CSS database to use for station description
-f INDIR Dir to search for MSEED files
-o OUTDIR Dir to place created vols, will be created if doesn't exist
-i Used with -o, will also create incomplete vols from files that exist

Once you have verified that your data are complete, please contact us by sending an email with your experiment name and network code to data_group@passcal.nmt.edu. For example: XO –Terra data 2004-2005.

Sending data – gui_DoFTP and con_DoFTP

There are two options for data submission via FTP: using the command line or a GUI. We recommend the use of **gui_DoFTP** to submit data to the PIC (current version 2008.038 or later version). See the appendix for more details on **gui_DoFTP**, which is a python-based package available from PASSCAL as part of our software release: http://www.passcal.nmt.edu/content/software-resources

<my_cpu: EXPT > **gui_DoFTP**

DoFTP will:

- Descend the specified directory path, identify, and pack ALL mseed files found
- Create .tar and .md5 (similar to check sums) files of the data
- Send the dataless and its .md5 file
- Build a report (list) of all data files sent and its md5
- Start an FTP session to PIC and send the data

Note:

- Be as specific as possible when specifying the path to the data, so unintended files are not packed
- The software requires at least as much free disk space as the size of the dataset to be sent. That is, if you have 100 GB of data to send, **DoFTP** will need at least another 100 GB of free space to build the tar files.

To use in the command line option, type **con_DoFTP** .

<my_cpu: EXPT> do_ftp_path/**con_DoFTP**
(do_ftp_path is where you have installed **DoFTP**.)

-# Print version of this program
-a force ACTIVE FTP mode (default is PASSIVE mode)

-f ftp the tarred data or resume ftp from the last broken pt.
-r gives an integer from 1 to 366 (default is today's julday)
-t set FTP timeout with a positive integer (default: no timeout)
-help print help information

A typical command line usage may look like this:

<my_cpu:EXPT> **./con_DoFTP** -a -f -r 366 -t 15 /Users/kxu/FA_tremor

Adding more data: future services

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. Here 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 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 may find some points to consider while populating the database during later services.

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

    a. Update the Batch File (if needed)

At this point you already have a batch file. You may need to update or modify it 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 10:15:59”). 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.

a. Building the Antelope Database

If none of the above 3 points come up, there is no need to re-run **dbbuild** since your stations are already described on your database or build a new dataless. If one of the three points were required then you will need to update your database with **dbbuild** as shown below:

<my_cpu:EXPT > **dbbuild -b my_db** your_updated_batch_file

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 the same command 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 service2, then you will run:

<my_cpu:EXPT > miniseed2days –d my_db –w "day_volumes_service2/%{sta/%{sta}.%{net}.%{loc}.%{chan}.%Y.%j" RAW/*.ALL >& my_miniseed2days_service2.out

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)

g. 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 the PIC for review and archiving at the DMC so the appropriate changes are visible for data and meta-data requests. Meta-data/dataless changes may occur at any time including between service runs and after an experiment is complete.

There are a couple of way 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:

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

<my_cpu> **mkdir** my_newdataless

b. Copy your existing batch file to the temporary directory.

<my_cpu> **cp** your_updated_batch_file my_newdataless/modified batch file

c. Edit it.

d. Create a new database by using **dbbuild.**
e. Check the database with **dbverify.**
f. Fix any errors, if you have any questions please e-mail data_group@passcal.nmt.edu

<my_cpu> **dbbuild –b** modified_db modified_batch_file
<my_cpu> **dbverify –tj** modified_db

g. Create a new dataless, which will contain all the information that needs to be incorporated into the database you have with all your waveforms.

<my_cpu> **mk_dataless_seed –v –o** PI.04.my_db.20072201700.dataless modified_db

h. Verify your dataless:

<my_cpu> **seed2db –v** PI.04.my_db.20072201700.dataless

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

Verifying archived data

Usually once the data makes it to our system, it will run through verification software. If the data and the dataless pass all of the checks in the Quality Control System (QCS), the data are prepared for submission as station-day volumes to the DMC. This process may take between one to two weeks depending on how much data volume is flowing through the 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. Once we confirm the data have been archived we will send you an e-mail with a summary of the data archived for your experiment. Please take a moment to ensure this summary agrees with your records of data you expect to be archived.