IRIS/PASSCAL INSTRUMENT CENTER

GENERATING SEED FROM RT130 DATA USING ANTELOPE FOR STANDALONE STATIONS

PASSCAL Platforms: Mac OS10 & Linux

Version 2008.260

Data Group

Questions/comments: data_group@passcal.nmt.edu

Table of contents

GENERATING SEED FROM RT130 DATA USING ANTELOPE

(V2008-260)

APPENDICES - Table of contents

GENERATING SEED FROM RT130 DATA USING ANTELOPE

(V2008-260)

1 Introduction

This detailed document serves to guide the data archiver through the process of data archiving, utilizing 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 A).

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. Please refer to Appendix B for specifics on platforms and/or limitations for Antelope version 4.10.

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**.

2 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) Network code request: http://www.iris.edu/scripts/getcode.html
2) Mobilization form: http://www.iris.edu/stations/mob.htm
3) Demobilization form: http://www.iris.edu/stations/demob.htm, to be completed by the end of the experiment.
4) PASSCAL Data Deliver Policy found in Appendix A or at: http://www.passcal.nmt.edu/information/Policies/data.delivery.html,
5) ANTELOPE – Details on release 4.10 (Appendix B) and Guide/Request and Installation/How To, found in Appendix C,
6) Install the latest PASSCAL software package for your platform, which may be found at: http://www.passcal.nmt.edu/software/software.html.

   Note: The forms in step 3-4 alert the DMC of your temporary network, resulting in the experiment's assignment of a network code and setup the infrastructure needed for the DMC to accept your data.

PASSCAL field computers loaned to the PI are shipped with PASSCAL software and 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 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 to** data_group@passcal.nmt.edu.

Some of the software you will use originates at PASSCAL. Please find the RELEASE NOTES here: ftp://ftp.passcal.nmt.edu/passcal/software/RELEASENOTES. This page describes our program.

If you'd like to be on our mailing list for the next release, please send a note to passcal@passcal.nmt.edu. Refer to Appendix D for a list of software, which may be used during the data processing procedure.

3 Directions

3.1 Steps in brief

Data Reduction and Timing Quality Control

a. Back up the raw data from the RT130 compact flash cards
b. Create an organized directory structure for your data
c. View logs and waveforms: quality control
d. Modify headers using fixhdr
   - ➢ Change Endianess and Flag/Shift Timing
   - ➢ Modify default fields on your traces to MSEED format
e. Convert Reftek log files into mseed
f. Edit the local log2miniseed.pf to define directory structure and file name convention.
g. Set up the environment
h. Run log2miniseed

Populate the Antelope Database

a. Create a Batch File
b. Build the Antelope Database
c. View your database
d. Creating mseed day volumes and adding them to your database
e. Assign calibration values from calibration table to wfdisc
f. Verify the Integrity of Your Database
g. Create the dataless SEED volume

Send Data to IRIS/PASSCAL

3.2 Steps in detail

3.2.1 Data Reduction and Timing Quality Control

a. Back up the raw data from the RT130 compact flash cards

We encourage PASSCAL and USArray/FlexArray users to follow one of the suggested procedures in Appendix E to backup the raw images, and simultaneously generate zip files of the raw images where you can extract the mseed, ref, or log files for quality control and further processing. Instructions on this step vary depending on the platform you are working on. Please find below a suggested guide and comments about how to back up your data when working from one of the PASSCAL field machines (LINUX, MAC OS) or a Solaris machine

The software programs you will use are called **neo, chunky** and **unchunky.** These are PASSCAL scripts and part of the PASSCAL software release.

Neo	: is a method of extracting the data from the flash cards
chunky	: bundles the files together for transport (FTP)
unchunky	: unpacks them by running rt130cut and ref2mseed on the ZIP files created by **chunky.**

If you cannot find the programs listed above or have any trouble please let us know by writing to passcal@passcal.nmt.edu.

b. Create an organized directory structure for your data

You can generate your own directory structure and name 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 EXPT and organize the files accordingly:

<my_cpu:EXPT > mkdir raw_data

<my_cpu:EXPT > mkdir unchunky_ouput directory – the output or destination directory for *ref files

<my_cpu:EXPT > mkdir ref_mseed– location for mseed files obtained after extracting with unchunky

<my_cpu:EXPT > mkdir run_logs – location for *.log, *.err , *.run files

<my_cpu:EXPT > mkdir day_volumes – where the day volumes will go after running miniseed2days

Note that a "response" directory will be automatically created by **dbbuild.**

IMPORTANT – Place mseed day volumes of waveforms and log files (after running miniseed2days and log2mseed) in a directory called, for example, "day_volumes". The name is optional, call it what you like, but be consistent throughout the process.

In addition to the mseed files (e.g. R001.01/07.001.00.01.45.9764.1.m), **ref2mseed** (run optionally by **unchunky**) will generate a series of logs, *.run, *error files. Look at the log files to check for any time corrections (gps), phase errors, etc., in the respective log files for each station.

c. View logs and waveforms: Quality Control

Using **unchunky** you can extract the log files and/or waveforms to view and evaluate the station performance (log files are viewed with **logpeek** and waveforms with **pql**)

<my_cpu> unchunky -h

Examples:

To extract REFTEK logs from ZIP files for viewing with **logpeek** run **unchunky**:

<my_cpu> unchunky -d /path/to/zip/files -l -i das-number

To extract REFTEK mseed_files, from ZIP files for viewing with **pql** run **unchunky***:*

<my_cpu> unchunky -d /path/to/zip/files -m -i das-number

This is a good time to do a preliminary quality control on your data and log files. Quality control on mseed data can be extensive. We highly recommend a data evaluation before you continuing archiving. Use **pql** for waveform QC, and **logpeek** for log file evaluation**.** Both are part of the PASSCAL software release.

If errors larger than one half of the highest sample rate occur, then it may be best to flag the data as 'timing questionable'. Set the data quality bit in the miniseed header. Use **logpeek** (Appendix F) to identify timing issues and **fixhdr** (Appendix G) to set flags if the timing is questionable, if no timing issues please continue to the next section.

Basic checks include:

- Timing quality – GPS locks/unlocks, GPS clock lock gaps
- Power problems and system reboots
- Large phase errors, time jumps/jerks, and unexpected gaps
- Station GPS location (average given)
- Voltage drops
- CPU version

d. Modify headers using fixhdr

The PASSCAL software, **fixhdr,** ("fix-header") allows users to make changes to mseed fixed header values, change the endianess of the mseed headers, and apply bulk-timing shifts. 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.

➤ *Change Endianess and Flag/Shift Timing*

fixhdr also provides a means for you to modify the endianess (byte-order) of your files from little to big (if required) and setting flags for questionable timing, or to apply time correction when needed. To launch **fixhdr** with a GUI (graphical user interface) you need to type on the command line:

<my_cpu:EXPT> fixhdr

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.   A separate document detailing the use of fixhdr is available from PASSCAL.

➤ *Modify default fields on your traces to MSEED format*

Header fields will need to be modified following SEED format (Appendix H). Fields to modify using **fixhdr** are: station name, channel, location code (optional, only if needed), and network code. Appendix I**,** shows several examples on proper channel naming. Please refer to the Standard for the Exchange of Earthquake Data, Reference Manual, SEED Format Version 2.4 (http://www.iris.edu/manuals/) for complete details on SEED format.

To build your batch file using **fixhdr** please refer to the help for **fixhdr.** Below is an example of a batch file built by saving the template with **fixhdr:**

```
# Header Changes
hdrlist{
# sta:chan:loc:net:sps          new sta:chan:loc:net
0965D:1C1::XX:100.0             PA01:EHZ::YW
0965D:1C2::XX:100.0             PA01:EHN::YW
0965D:1C3::XX:100.0             PA01:EHE::YW
09511:1C1::XX:100.0             PA02:EHZ::YW
09511:1C2::XX:100.0             PA02:EHN::YW
09511:1C3::XX:100.0             PA02:EHE::YW
0969D:1C1::XX:100.0             PA03:EHZ::YW
```

Note that in this example we are not using the location code. If you decide to use location code (let's say you use 00) it should look something like:

```
# Header Changes
hdrlist{
# stat:chan:loc:net:sps        new sta:chan:loc:net
   0965D:1C1:00:XX:100.0   PA01:EHZ:00:YW
   }
```

e. ***Convert Reftek log files into mseed***

To complete this step you need to copy the log2miniseed.pf file into your local directory, or edit it in the default pf file located under $ANTELOPE/data/pf/

<my_cpu:EXPT> cp $ANTELOPE/data/pf/log2miniseed.pf .

f. ***Edit the local log2miniseed.pf to define directory structure and file name convention.***

Change the default string*:* **wfname %Y/%j/%{sta}.%{chan}.%Y:%j**
to:
wfname day_volumes/%{sta}/%{sta}.%{net}.%{loc}.%{chan}.%Y.%j

The word "**wfname"** is part of the parameter file format so you should keep it. **"day_volumes"** is the directory where the log files will go under a station subdirectory. Make sure this directory is the same where you will write the mseed day-volumes when running minseed2days –d (3.2.2, section d).

g. Set up the environment

<my_cpu:EXPT> setenv PFPATH $ANTELOPE/data/pf:.

h. Run log2miniseed

For each station (log file) you will need to run **log2miniseed,** or write a simple script to run log2miniseed for all your log files. For one log file the command line will be for example:

<my_cpu:EXPT> log2miniseed –n PI –s NP00 run_logs/2005:128:15:09.0965D.log

When running log2miniseed you are modifying:

Network code:	-n PI (PI is an example),
Station name:	-s NP00 (should correspond to the station name associated with the log file *2005:128:15:09.0965D.log)*
run_logs:	the directory where you have the log files; will depend on the directory structure you have set up, this is just an example)

3.2.2 Populate the Antelope Database

a. Create a Batch File

The next step is to create an Antelope database that defines your network and station configurations. You will use the tool **dbbuild** in batch mode (**dbbuild –b**) to construct a CSS3.0 database. The batch file is an ascii file with specific keywords and details used to build the database with out the use of the GUI. 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 as an example and edit accordingly the fields in green. A batch file may have comments (denoted with #). The description for each field in the batch file (and how **dbbuild** works) may be found in the man pages dbbuild_batch and **dbbuild**.

If you have questions about dbbuild or the batch file, please refer to Appendixes H & I. The fields in green are inputs that you need to provide.

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

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

(Where 01 is the location code for data stream 1.)

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

sta NP00 -77.72237 162.27354 0.042 Socorro, NM
time 02/06/2004 02:50:53
datalogger  rt130 0984
sensor l28 0 0001
axis Z 0 180 - 1 1
axis N 0 90 - 2 1
axis E 90 90 - 3 1
samplerate 200sps
channel Z EPZ
channel N EPN
channel E EPE
samplerate 1sps
channel Z LPZ
channel N LPN
channel E LPE
add

sta NP02 -77.72591 162.26907 0.079 Socorro, NM
time 02/29/2004 02:57:57
datalogger  rt130 0988
sensor l28 0 0015
axis Z 0 180 - 1 1
axis N 0 90 - 2 1
axis E 90 90 - 3 1
samplerate 200sps
channel Z EPZ
channel N EPN
channel E EPE
samplerate 1sps
channel Z LPZ
channel N LPN
channel E LPE
add

close NP00 12/31/2007 23:59:59
close NP01 12/31/2007 23:59:59
```

We discourage the use of locations codes and suggest they be explicitly defined only when necessary to avoid ambiguity (such as when operating a dense network (stations within 1 km) or when recording multiple streams at sample rates sharing a common band code (first letter) within the channel code.

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 them covering the times frames from the very first sample rate on any channel, to the day the station is closed.

We suggest a slightly earlier time for start time (second line after station in the batch file) to assure all the traces are included on the metadata. This will prevent from further errors and problems during archiving.

Please read this manual carefully and refer to the appendixes for detailed information on several steps in this guide.

b. Build the Antelope Database

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

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

NOTE: The configuration for each station in your batch file agrees 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 use a start time for each the station which is conservative (i.e. a little early rather than milli-seconds late).

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

: loading 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" are created. These tables and directories are the constituents of the database.

<my_cpu:EXPT> ls

my_db.instrument	*my_db.site*	*my_db.schanloc*
my_db.sensor	*my_db.network*	*my_db.snetsta*
my_db.stage	*my_db.sitechan*	*response/*
my_db.lastid	*my_db.calibration*	

You may find more detailed descriptions of common errors/warnings when running **dbbuild** in Appendix J, Tables 1 & 2.

c. View your database

Using **dbe** you can visualize the current information in your database. At this point there are no waveforms.

<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 read man dbe.

d. Create mseed day volumes and add them to your database

Now you have a database that describes your network (by running dbbuild), but you have not associated any waveforms with the meta-data.

To add your waveforms details to your database, use the command **miniseed2db.** <u>This will create the miniseed day volumes</u> (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> miniseed2days –d my_db –w
“day_volumes/%{sta}/%{sta}.%{net}.%{loc}.%{chan}.%Y.%j" ref_mseed/ >&
my_miniseed2days.out

Where:

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

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

  Note: the mseed headers read by miniseed2db are the source of information used to populate the database's waveform table. You must ensure the mseed headers in the station-channel-day files produced by miniseed2days are correct. If the database (and its batch file) do not describe all of the data then errors will result when we check the consistency of the database.

e. Assign calibration values from calibration table to wfdisc

To assure that the calibration values are incorporated into the just created wfdisc table, please run dbfix_calib:

<my_cpu:EXPT> dbfix_calib my_db

f. Verify the Integrity of Your Database

Before you create a dataless you will want to be sure that your meta-data completely describes the waveform data, and that your database is error free. **Read** the man page on *dbversdwf* and *dbverify* for other tests you can run on the database. Examples of suggested tests are:

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

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

Please refer to Appendix J, Table 3 for possible scenarios you may have when running these tests.

g. *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:EXPT > 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 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

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

h. *Verify the dataless*

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

<my_cpu:EXPT> seed2db –v my_db_dataless_seed

Please refer to **APPENDIX K, Table 4,** for possible cases you may run into when running *seed2db*.

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

3.3 Send Data to IRIS/PASSCAL

When you are ready to submit the data to PASSCAL, please contact us by sending an email with your experiment name and network code to data_group@passcal.nmt.edu. For example: data submission for XO –Terra data 2004-2005.

There are two options, using the command line or a GUI. We recommend use of **GUI_DoFTP** to submit data to the PIC (current version 2008.038 or later version). See Appendix L 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/software/software.html.

<my_cpu:EXPT> my_path Gui_DoFTP

(Where my_path is where you have installed **DoFTP**.)

DoFTP will:

- Descend the specified directory path, identify, and pack ALL miniseed 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
- 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 data set 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> my_path con_DoFTP
(Where my_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
```

e.g.: ./con_DoFTP -a -f -r 366 -t 15 /Users/kxu/FA_tremor

4. Adding more data: future services

A typical question from a data archiver: "I have more data from the last service. 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 service 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. Data reduction and timing quality control will remain the process as before 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 service runs

a. Update the Batch File (if needed)

At this point you have already 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 never was recorded for this station 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.

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.

b. 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 batch_mynet

i. View your database as described before.
ii. Add Your Waveforms to the Database

Once you have the new data ready to add to the database (QC done, timing issues evaluated, headers fixed, etc), you can add it to the database using the same command but pointing to the directory where you have your new service (let's say 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" ref_mseed/* >& my_miniseed2days_service2.out

iii. Verify the Integrity of your Database
iv. Create a dataless SEED volume if you ran dbbuild (revised or rebuilt the database)
v. Verify your dataless file and rename to conform with convention

5. Verify archived data

Usually once the data makes it to our system, it will run through verification software. If the data and dataless pass all 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 data volume 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 has 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.

6. Updating the meta-data without processing new data.

All changes change/addition/removal in your network configuration must be described on 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 generating a dataless within it. The steps you should follow are:

a. Create a temporal 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** my_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 in 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.

7 Other antelope tools that may be helpful

Please request the man page for these tools:

- **dbplotcov** - dbplotcov reads the wfdisc table from the specified database, determines the periods of time for which waveform segments exist for each station-channel and prints and plots this information. A PostScript version of the coverage plot, named dbplotcov.ps, is created in the current directory. This tool has several caveats but for small databases it works to give a visual display of the coverage for each station and all channels in your db.

  dbplotcov usage: dbplotcov dbname stachan tstart tend [-h [ntrigs]] [-wftar]

  Example:
  dbplotcov cafe_join '*:*' '07/09/2006 00:00:00.000' '06/10/2007 00:00:00.00' > plot

- **db2sync** is designed to create a Synchronization file in the format constructed by the IRIS DMC from an existing Datascope database.

  Example: *db2sync USAGE: db2sync [-s start_time] [-e end_time] [-S subset_expression] [-p pf_file] [-a] [-l] [-d] [-h] [-o] [-w] [-v] dbin fileout*

- **BRTTPLOT**
  viewport, axes, grid, ptext, polyline, polypoint, map - BRTT tk canvas item extensions These are all special tk canvas item extensions available through the Brttplot package in the Antelope tcl/tk extensions. All of these canvas item widgets act as normal tk canvas items, including such functionality as the ability to display these in scrolled canvases and the ability to generate PostScript output, and they should be thought of as extensions to the various items that are described in canvas(n).

- **dbsnapshot** collects some information and records from a database into a single tar file. This can be helpful when trying to resolve a database problem.

-

8 Using DMC tools to View/Requests your archive data

a) IRIS/DMC Meta-data Aggregator

Using the meta-data aggregator from DMC (http://www.iris.edu/mda/) you can view the complete list of assigned FDSN network codes, including all the networks that submit data to the DMC. Therefore, this is a good portal that summarizes data collected from PASSCAL experiments since 1986. Parametric information that is extracted from all submitted dataless SEED volumes for each network (location, time span, type of data –Real-time (R) -or Archive A)-, station names, number of stations, channels, instrument response plots, etc) can be found for each network available, as well as a link to the DMC's Google map service (http://www.iris.edu/gmap) that the locations for each network that has submitted meta-data to the DMC. Using the network code and the years of your experiment you can see the information stored about your network.

b) VIRTUAL NETWORKS

Currently the DMC has available 18 virtual nets (http://www.iris.edu/mda#vnetlist) including 2 from PASSCAL (_PASSCAL & *PAS-OPEN),* all EarthScope stations *(*US_ALL), USArray Flexible Array (_US-FA), and USArray Transportable Array (_US-TA), among others.

_PASSCAL Virtual Net – contains over three thousand stations with analog and real time data since 1990 to the present. Data from all PASSCAL experiments with archived data and currently deployed and/or submitting data.

_PAS-OPEN Virtual Net – Data from stations that have been made available to the public from principal investigators from PASSCAL experiments following the data delivery policy (since January 2005) explained in APPENDIX A.

c) BUD_stuff, Monitor, QUACK and others

BUD is the IRIS DMC's acronym for the online data cache from which we distribute our near-real time miniSEED data holdings prior to formal archiving.

d) VASE

VASE is a Java-based client application designed for viewing and extracting seismic waveforms from the DHI waveform repository via BUD.

e) JWEED

JWEED is a Java update of WEED allowing users to access event and station data through an interactive map.

You can find some interesting links under: http://www.iris.edu/

IRIS/PASSCAL Documentation- Created by Eliana Arias (eliana@passcal.nmt.edu, 2006,2007)
Revised Bruce Beaudoin (2006,2007,2008)
Revision 1 by Eliana Arias, May 22, 2008
Revision 2 by George Slad, June 5, 2008.
Revision 3 by Lisa Foley, June 18, 2008
Revision 4 by Lisa Foley, September 17, 2008
Revision 5 Eliana Arias, September 17, 2008