

Alaska Earthquake Information Center

University of Alaska Fairbanks

Revised procedures for maintaining the AEIC master_stations database

AEIC Internal Report 2008-01

by Glenn Thompson and Martin LaFevers

July, 2008

Suggested citation:

Thompson, G, LaFevers, M. A. 2008, Revised procedures for maintaining the AEIC master stations database. Alaska Earthquake Information Center, University of Alaska Fairbanks, AEIC Internal Report 2008-01.

This version was last revised: September 2nd, 2008

This document, any updates to it, and any additional information are available at:
<http://www.aeic.alaska.edu/AEIC/internal/report/2008-01/>

The Alaska Earthquake Information Center is a cooperative program between the Geophysical Institute of the University of Alaska and the U. S. Geological Survey with support from the Earthquake Hazards Programme.

DISCLAIMER

This report has not been edited or reviewed for conformity with U. S. Geological Survey and State of Alaska standards and nomenclature. The data in this report are preliminary and subject to revision. This report is released on the condition that neither the U. S. Geological Survey, nor the Geophysical Institute, University of Alaska Fairbanks, may be held liable for damages resulting from its authorized or unauthorized use.

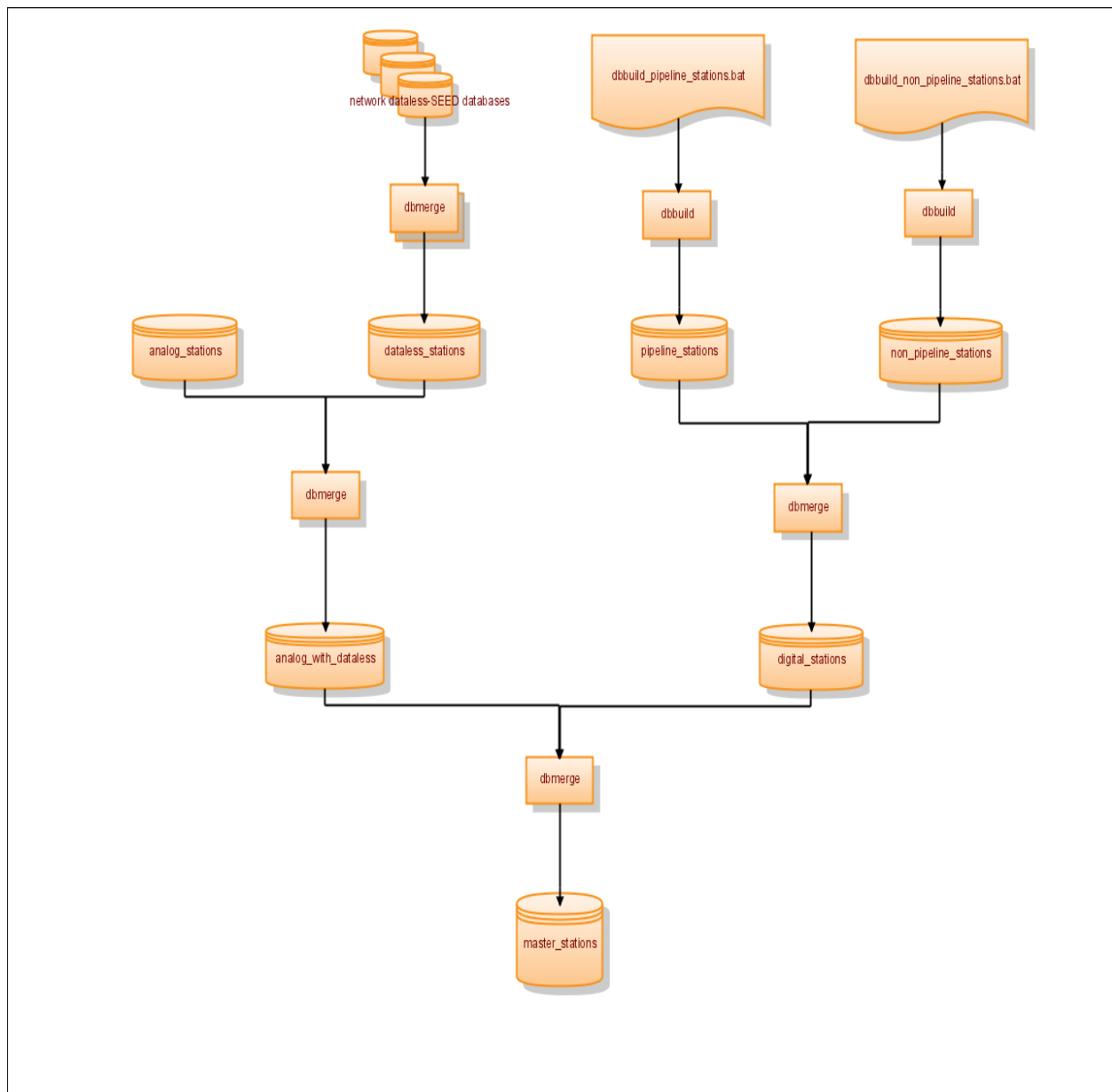
Table of Contents

1	Introduction	1
1.1	Hierarchy	1
1.2	Basic directory structure	2
1.3	Description of the databases	2
1.4	Useful man pages	3
1.5	Useful programs	3
1.6	Overview of steps for database update	3
1.7	Types of updates	3
2	Procedures	4
2.1	Before you start	4
2.2	Updating the Alaska analog network	4
2.3	Updating the imported stations via dataless SEED volumes	5
2.4	Merging the (dataless_stations) and (analog_stations) databases	7
2.5	Updating the Trans-Alaska Pipeline System (TAPS) stations	8
2.6	Updating other digitally-telemetred stations	8
2.7	Creating a database of all digitally-telemetred stations	9
2.8	Creating the final master stations database	10
	Appendix 1: Miscellaneous details	12
	Dbmerge	12
	dbverify	12
	dbbuild	12
	dbbuild comments	12
	Instrument table	13
	Guralp issues	13
	Dataless SEED files additions (these change all the time)	13
	Non-standard tables	14
	Upgrading a station from analog to digital:	14
	Changing a site's coordinates:	15
	Removing duplicate site rows:	15
	If all else fails:	15

1 Introduction

1.1 Hierarchy

This diagram gives a logical overview of how the (master_stations) database is constructed from network dataless-SEED databases, an analog database and dbbuild batch files (for digitally-telemetred stations).



1.2 Basic directory structure

The master stations database resides in a repository at /home/glenn/stations (which has been substantially re-organised from when it resided at /home/martin/work/stationdb). The crucial directory hierarchy is represented below (Datascopie databases are shown <like this>) :

```
(master_stations)
dataless_stations/
----- (dataless_stations)
-----Response/
digital_stations/
----- (non_pipeline_stations)
----- (pipeline_stations)
----- (digital_stations)
-----Response/
analog_stations/
----- (analog_stations)
-----Response/
analog_with_dataless/
----- (analog_with_dataless)
-----Response/
/usr/local/aeic/4.10/data/instruments
/Seis/databases/dbbuild/data/instruments
```

(need to investigate dbbuild more to consolidate latter 2)

Henceforth, we use the environment variable \$REPOSITORY to refer to the path /home/glenn/stations/.

1.3 Description of the databases

There are many databases that may be used in the update procedure. Primary databases are:

1. Dataless-SEED-derived network databases. These are constructed from dataless SEED volumes using seed2db. Currently there are 6 of these: atwc_datalessdb, cnsn_datalessdb, gsn_datalessdb, iu_datalessdb, kdak_datalessdb and usarray_datalessdb.
2. (analog_stations): This database contains all short period stations and broadband stations with response not handled by dbbuild. It might better be thought of as a database for stations that use analog telemetry and do not yet have dataless SEED volumes (which need poles & zeroes).
3. (pipeline_stations): This is built from a dbbuild batch file dbbuild_pipeline_stations.bat. It contains all the stations that form part of the Trans-Alaska Pipeline monitoring system, and they are kept separately as this database is expected to be used elsewhere.
4. (non_pipeline_stations): This is built from a dbbuild batch file dbbuild_non_pipeline_stations.bat. It includes all digitally-telemetred stations apart from the pipeline stations.

Secondary databases are then constructed by merging primary databases, these are:

1. (dataless_stations): This database contains is built from merging all the dataless-SEED-derived network databases together.
2. (analog_with_dataless): This database is built from merging (dataless_stations) with (analog_stations).
3. (digital_stations): This database is built from merging (pipeline_stations) with (non_pipeline_stations).
4. (master_stations): This database is the final product used in data acquisition. It is built from merging (analog_with_dataless) and (digital_stations).

1.4 Useful man pages

dbbuild
dbbuild_batch
dbbuild_examples

1.5 Useful programs

dbbuild
dbverify
dbmerge

1.6 Overview of steps for database update

- Make local copies of appropriate database(s).
- Maintain (analog_stations) database.
- Maintain (dataless_stations) database for imported stations.
- Maintain dbbuild batch files for pipeline and non-pipeline digital stations.
- Merge (dataless_stations) with (analog_stations) to create (analog_with_dataless).
- Merge (pipeline_stations) with (non_pipeline_stations) to create (digital_stations).
- Merge (analog_with_dataless) with (digital_stations) to create (master_stations).
- Hand edit tables, network, schanloc, snetsta, affiliation.
- Give Mitch the path to (master_stations) and the appropriate Response/ and stage/ directories. Mitch will install new (master_stations) database, Response/ and stage/ directories to the appropriate systems.
- Copy necessary files back to the database repository.

Note: BACKUP YOUR WORK AT EVERY STEP OF THE PROCESS!

1.7 Types of updates

There are three types of updates:

- An addition or change to the Alaska analog networks, not handled by SEED volumes (analog_stations).
- An addition or change to imported stations via dataless SEED volumes.
- An addition or change to the Alaska digital network, handled with a dbbuild batch file.

Depending on your update needs, you may be performing any number of these at the same time.

2 Procedures

The procedures below are written in the chronological order in which they should be followed. Depending on the updates required, each procedure will tell you when its OK to move onto the next procedure.

2.1 Before you start

- Before starting any database work, make sure there is already a tar backup of the last version (see procedure 2.8 step 7).
- Blow away the local copy of the master stations database, since you will be overwriting this (after already making sure you have a backup above!).

```
>> cd $REPOSITORY
>> rm master_stations*
>> rm -r Response
```
- Copy the non-standard tables from /iwrn/op/run/dbmaster/master_stations. These will have the latest updates make automatically by Antelope:

```
>> cp /iwrn/op/run/dbmaster/master_stations.network .
>> cp /iwrn/op/run/dbmaster/master_stations.affiliation .
>> cp /iwrn/op/run/dbmaster/master_stations.schanloc .
>> cp /iwrn/op/run/dbmaster/master_stations.snetsta .
```

2.2 Updating the Alaska analog network

If you do not need to make changes to the (analog_stations) database, go on to procedure 2.3. If you do, there are 5 steps:

Notes:

- *Entry for DDM in site table was modified 2008/08/14, ondate changed from 1977244 to 1977213. This was to remove an error that sensor ondate < site ondate.*

Revised procedures for maintaining the AEIC master_stations database

- *Errors in analog_stations are primarily different historical names for the same site, or secondarily they are stations whose coordinates have changed slightly (AUE, AUW, HUR).*
- *The site rows for KDAK, DLBC, PMR, SMY & COLA were identical to clashing site rows in dataless_stations, so delete_rows.csh was written to remove these duplicate rows.*
- *Offdates were added to site records for DAWY, HYT, NDB, PLBC, MID, SDPT and SIT because revised coordinates in dataless_stations commence the following day.*

1. Assuming you need to edit the (analog_stations) database, run dbedit in edit mode on your local copy of the (short-period) database to make the required changes:

```
>> cd analog_stations  
>> dbedit -e analog_stations
```

2. Run dbverify to check your modifications, redirecting the output to a file to capture errors and warnings. This output file will be needed later, so don't delete it:

```
>> dbverify analog_stations > dbverify_a.txt
```

3. Sift through the output file with the “more” (or “less”, or “vi”) command:

```
>> more dbverify_a.txt
```

4. Don't be surprised if there are of the order of 400-500 errors or warnings, but they are mostly of just two or three types. Expect errors about the stations for which site records were removed: PMR, KDAK, DLBC, SMY & COLA. Consult Appendix 1 and move forward only when you're convinced there are no unexpected errors. This output file will be required later, so don't delete it.

5. Congratulations! You now have an updated copy of the (analog_stations) database. Go to the procedure 2.3.

2.3 Updating the imported stations via dataless SEED volumes

These updates will be rather rare. And it's hard to even know when they are needed. The best you can hope for is that a network manager has let you know that there has been a response change or addition for an imported station. Or quarterly checks by downloading dataless SEED volumes from IRIS and comparing with installed databases. Always compare on and offdates of the database you create from the dataless seed volume with existing dates. Some networks do not keep historically accurate databases. If you are sure you the imported stations have not changed, jump to Procedure 2.4.

You can get dataless SEED volumes from IRIS at <http://www.iris.edu/data/webRequest.htm>. IRIS doesn't have historic information going back to the 1960s/70s – but our databases do. Make sure not to lose this important historic data! For Canadian stations, dataless SEED volumes can be got through a Canadian website.

Revised procedures for maintaining the AEIC master_stations database

NB: On 2008/08/14:

- *Duplicate entry (1995) for KDAK removed from iu_datalessdb.*
- *Set offdates for BBGH, SDDR, TGUH to -1 in site and sitechan in gsn_datalessdb (they were 2999365 which is beyond valid range).*
- *Set ondate for SMY in site in atwc_datalessdb to 1970321 to solve error that sensor ondate before site ondate.*
- *Edited atwc_dataless to remove duplicate entries in instrument table.*
- *Edited the last_id table in all network database where dbverify flagged a warning. Reset to last id + 1 for each table.*

(1) Make sure you are in the correct directory:

```
>> cd $REPOSITORY/dataless_stations/
```

(2) ***(this step needs further explanation !!!)***. Download all new/updated dataless SEED volumes for your network from webpages, and put them in appropriate subdirectories. Then convert each dataless seed volume to a database. It is also possible to download SEED seismic data from IRIS and extract the station metadata from the seed header file using seed2db with other options. Check the man page for seed2db if you want to use this method. Inspect each database well, not every network follows seed convention in their channel naming schemes. You are likely to have to make hand edits.

```
>> seed2db -respdir ./Response -stagedir ./Response/stage  
atwc_dataless atwc_db
```

(3) If any network databases have been modified, run dbverify on them:

```
>> dbverify atwc_dataless >& dbverify_atwc.txt
```

(4) Merge all the network databases into to one big target dataless SEED database (dataless_stations). The Response and stage directories are handled by dbmerge behind the scenes, as long as they are in your working directories. Duplicates are removed and the instrument table links properly.

```
>> rename dataless_stations dataless_stationsbu  
>> mv Response Responsebu  
>> cp atwc_dataless/atwc_databasedb* .  
>> cpdir atwc_dataless/Response Response  
>> rename atwc_datalessdb dataless_stations  
>> dbmerge cnsn_dataless/cnsn_datalessdb dataless_stations  
>> dbmerge gsn_dataless/gsn_datalessdb dataless_stations  
>> dbmerge kdak_dataless/kdak_datalessdb dataless_stations  
>> dbmerge iu_dataless/iu_datalessdb dataless_stations  
>> dbmerge usarray_dataless/usarray_datalessdb  
dataless_stations
```

(5) Run dbverify on the (dataless_stations) database, directing the output to a suitable output file:

```
>> dbverify dataless_stations > dbverify_dataless.txt
```

- (6) Move to non-standard tables to some backup directory so they don't interfere with (master_stations) later. Either:

```
>> remove_non_standard.csh
```

Or:

```
>> mv dataless_stations.affiliation non_standard_tables/  
>> mv dataless_stations.snetsta non_standard_tables/  
>> mv dataless_stations.schanloc non_standard_tables/
```

- (7) The site rows for ADK, AKUT, BILL, BNAB, MA2, PET, TIXI and YAK clash with site rows from analog_stations, but here have a narrow time range, so lets remove them:

```
>> delete_rows.csh
```

2.4 Merging the (dataless_stations) and (analog_stations) databases

1. Make sure you are in the appropriate directory & initialize it:

```
>> cd $REPOSITORY/analog_with_dataless  
>> rm analog_with_dataless*  
>> rm -r Response
```

2. Copy your (updated) analog_stations database into this analog_with_dataless directory:

```
>> cp ../dataless_stations/dataless_stations* .  
>> cpdir ../dataless_stations/Response Response
```

3. You should still be in the analog_with_dataless directory. Now you are ready to merge your local copy of the (dataless_stations) database with the (analog_stations) database, with the dbmerge command:

```
>> dbmerge ../analog_stations/analog_stations dataless_stations  
>> rename dataless_stations analog_with_dataless
```

4. The target database (analog_with_dataless) should now have the (analog_stations) database folded into it. Lets check with dbverify, again sending the output to a file:

```
>> dbverify analog_with_dataless > dbverify_awd.txt
```

Previously there would be warnings about duplicate rows for AKUT, MID, SDPT, SIT, SMY, PMR, DAWY, DLBC, PLBC, BNAB, HYT, NDB, KDAK, BILL, COLA, MA2, PET, TIXI, YAK, ADK. Now there should be none.

5. Again, there will be lots of warnings and errors since this contains all the database rows from the (analog_stations) and (dataless_stations) databases you verified earlier. But are there any new errors or warnings? If there are, consult Appendix 1.
6. Congratulations! You now have an updated copy of the (analog_with_dataless) database. Go to procedure 2.5.

2.5 Updating the Trans-Alaska Pipeline System (TAPS) stations

The TAPS stations are handled separately as it is AEIC has decided these are so important that they should be managed in a separate database. If no changes have been made to the TAPS stations, go on to procedure 2.6.

- 1) Change to the appropriate directory:

```
>> cd $REPOSITORY/digital_stations
```

- 2) Update the (pipeline_stations) dbbuild batch file with relevant entries. See the man pages dbbuild_batch and dbbuild_examples.

```
>> vi dbbuild_pipeline_stations.bat
```

- 3) Remove the existing copy of the (pipeline_stations) database:

```
>> rm pipeline_stations*
```

- 4) You may also need to modify the affiliation table of (master_stations) if this station has not previously been a bband instrument, or used to calculate magnitudes (it needs a response though). To do this:

```
>> dbe -e ../master_stations.affiliation
```

- 5) You may also need to create a new network (if you are defining one from scratch), or modify the snetsta or schanloc tables (if this is not an AK network station). Again, just edit the appropriate tables as above.

- 6) Rebuild the (pipeline_stations) database:

```
>> dbbuild -b -v pipeline_stations dbbuild_pipeline_stations.bat >&  
dbbuild_pipeline_stations.out
```

- 7) Blow away all the incorrectly built non-standard tables either with:

```
>> remove_non_standard.csh
```

Or:

```
>> rm pipeline_stations.schanloc
```

```
>> rm pipeline_stations.snetsta
```

```
>> rm pipeline_stations.affiliation
```

- 8) Verify:

```
>> dbverify pipeline_stations >& dbverify_ps.out
```

```
>> vi dbverify_ps.out
```

2.6 Updating other digitally-telemetred stations

Any Alaskan stations which are digitally telemetred and are not TAPS stations should be managed here. If there have been no changes, move on to procedure 2.7. The most common type of change is an analog station being upgraded to digital telemetry, in which case the site should

be closed off in the (analog_stations) database, and a new entry will be added here. Other common modifications are the replacement of a Guralp sensor with an STS-2, the addition of an STS-2 serial number, or the swap of an STS-2 sensor or Q330 datalogger with another. Again, seek the man pages dbbuild_batch and dbbuild_examples.

- 1) Change to the appropriate directory:

```
>> cd $REPOSITORY/digital_stations
```

- 2) Update the (non_pipeline_stations) dbbuild batch file with relevant entries. Edit the batch file to suit your needs, if any changes have occurred to the broadband seismic network. Be sure that your choice of editor does not put in any control characters or end-of-line characters (I like simple old vi). All sensor changes are tracked in the batch file. There is an example of every sensor somewhere in the database – best approach is to find something similar, and copy the syntax.

```
>> vi dbbuild_non_pipeline_stations.bat
```

- 3) You may also need to modify the affiliation table of (master_stations) if this station has not previously been a bband instrument, or used to calculate magnitudes (it needs a response though). To do this:

```
>> dbe -e ../master_stations.affiliation
```

- 4) You may also need to create a new network (if you are defining one from scratch), or modify the snetsta or schanloc tables (if this is not an AK network station). Again, just edit the appropriate tables as above.

- 5) Remove the existing copy of the (non_pipeline_stations) database:

```
>> rm non_pipeline_stations*
```

- 6) Rebuild the (non_pipeline_stations) database:

```
>> dbbuild -b -v non_pipeline_stations  
dbbuild_non_pipeline_stations.bat >& dbbuild_non_pipeline_stations.out
```

- 7) Blow away all the incorrectly built non-standard tables either with:

```
>> remove_non_standard.csh
```

Or:

```
>> rm pipeline_stations.schanloc  
>> rm pipeline_stations.snetsta  
>> rm pipeline_stations.affiliation
```

- 8) Verify:

```
>> dbverify non_pipeline_stations >& dbverify_nps.out  
>> vi dbverify_nps.out
```

2.7 Creating a database of all digitally-telemetred stations

If you updated either (pipeline_stations) or (non_pipeline_stations), you need to merge them to re-create (digital_stations). This is a very simple procedure:

- 1) Remove the existing copy of (digital_stations):
`>> rm digital_stations*`
- 2) Copy the (non_pipeline_stations) database to (digital_stations):
`>> dbcp non_pipeline_stations digital_stations`
- 3) Run dbmerge:
`>> dbmerge -v pipeline_stations digital_stations >& dbmerge.out`
- 4) Check through dbmerge for errors:
`>> vi dbmerge.out`
- 5) Verify:
`>> dbverify digital_stations >& dbverify_ads.out`
`>> vi dbverify_ads.out`

Three warnings were found here and to fix these, instrument row #31 was changed from 'e' to 's', lasted inid was changed from 32 to 39, and chanid was changed from 1116 to 1457.

2.8 Creating the final master stations database

If you updated either (analog_with_dataless) or (digital_stations), which you should have done if you've made any changes at all, you need to merge them to re-create (master_stations). Note that you should have already taken care of any changes in your network, affiliation, schanloc and snetsta tables.

- 1) Make sure you are in the appropriate directory & remove all the master_stations tables except for the non-standard tables:
`>> cd $REPOSITORY`
- 2) Remove all the master_stations tables except for the non-standard tables (you should already have done this in procedure 1: if so, ignore this step):
`>> rm -i master_stations*`
`>> rm -r Response`
- 3) Copy the (analog_with_dataless) database to (master_stations). DO NOT overwrite the non-standard tables network, affiliation, schanloc and snetsta.
`>> cp analog_with_dataless/analog_with_dataless* .`
`>> rename analog_with_dataless master_stations`
`>> cpdir analog_with_dataless/Response Response`
- 4) Run dbmerge:

Revised procedures for maintaining the AEIC master_stations database

```
>> dbmerge -v digital_stations/digital_stations master_stations
>& dbmerge.out
```

5) Check through dbmerge for errors:

```
>> vi dbmerge.out
```

6) Verify:

```
>> dbverify master_stations >& dbverify_ms.out
>> vi dbverify_ms.out
```

7) Send Mitch & Natasha an email telling them the location of your updated version of (master_stations) and the appropriate Response/ and stage/ directories.

8) Its probably good practice to backup your edits. Create a tar file, with today's date in the filename:

```
>> cd $REPOSITORY/..
>> tar -cvf stations_20080725.tar stations
```

That's it! Unless there are some tricky issues with **non-standard tables** (see below).

Appendix 1: Miscellaneous details

dbmerge

dbmerge merges two databases together. There is no independent output database, the second argument is the output. dbmerge also merges the Response and stage dirs as it does it's job. Kindly, dbmerge is nice about not doing anything if there is a problem with your database, it will point out the error and tell you to fix it before continuing. But still, backup your work before using!

dbverify

dbverify should be used before you consider any level update complete. dbverify gives a lot of error messages that are not significant. You will recognize them as being repetitive. dbverify tolerances for these errors can be customized to reduce the error output, consult the man page if you want to do this. You can redirect the dbverify output to a file with a command line like this:

```
>> dbverify analog_stations >& output_file
```

dbbuild

dbbuild has two modes. The gui interface is quite useful as a practice tool to create a batch file. However, the gui does not accomplish all of our needs. For example, the dbbuild gui will not close stations. It will close old channels when new channels are opened as long as the channel names stay the same (BHZ->BHZ). When channel names change or a station is closed, a close statement must be added manually to the dbbuild-batch file.

The batch mode is the most effective method of using dbbuild. It involves maintaining an input file. Proper syntax in the batch file is extremely important. Nearly all combinations of instruments are represented in the batch file. When making edits, search the file for the station you are interested in, then edit that section of the file to suit your needs.

The batch file command line will look something like:

```
>> dbbuild -b dbout dbbuild.bat >& dbbuild_out
```

where dbbuild_out will save the output to a file for you to proofread. This output is different than the dbverify output and should be inspected for errors.

dbbuild comments

Necessary dbbuild comments in the batch file include:

person editing the file,

date edits take place,

a reason such as "sensor change" or "site closed", "station installation"

Other comments are welcome when adding to the file, but not necessary.

instrument table

There are stations that do not have known response. But the sensor table must have entries for an analyst to make picks. So dummy entries exist in the sensor table for short period stations and broadbands. Reference these dummy rows when adding or editing stations without known response.

Guralp issues

The installed cmg5t response file is wrong in release 4.8. I used a 4.7 response file as a template and created my own. In the response directory is a file created by Martin that is called aeic_cmg5t.

In the sensor directory is a file created by Martin called cmg5t. Guralp equipment response can vary based on cabling between sensor and digitizer. A sensor directly coupled to the digitizer in the stacked configuration (no visible cables) uses “single ended” response. Sensors and digitizers connected via a cable are considered “differential”. Guralp manuals handle this on the sensor side with a 2x multiplier, an incompatibility with dbbuild. Instead /2 on the digitizer side is appropriate for dbbuild. This allows a sensor originally delivered for use as single ended to retain the same response when used as differential.

Dataless SEED files additions (these change all the time)

Periodic updates of response for imported stations should take place when response is known to have change or for checking purposes.

web address: <http://www.iris.washington.edu/data/DatalessRequest.htm>

station list:

US net, EGAK, WRAK

AT net, AKUT, CRAG, PMR, SDPT, SIT, SKAG, SMY, OHAK

Convert the dataless seed files to a database with seed2db, setting the proper path and names for stage and response directories. After converting to a database, remove network and affiliation tables, these are maintained by AEIC needs, not the originating seismic lab. Merge with analog_stations database. The merge will not properly handle existing entries in the site table with historic value, the added entries may need to be removed by hand.

1/22/2007 sta removed from dataless_stations.site: AKUT, SMY, PMR, SDPT, SIT

Be aware that US stations EGAK and WRAK have future offdates in the dataless seed files. For now, these dates are intact.

Delaney Park array stations in Anchorage are all under the official name 8040. DPA1, DPA2, DPA3, and DPA4 are technically channel names since the stations all have the same gps location. Response for these stations is not available as dataless seed, the parameters came from an E-mail and exist in the master_stations-dbbuild batch file.

ATWC stations may lag behind in the dataless SEED files from IRIS. Compare your database from the dataless seed file to the most current data in the database you are about to update.

****Some stations may have more current information coming from the dbbuild-batch file than exist in dataless seed! The source of the more current information is from the header file on a seed volume.**

1/21/2007 PMR and all CNSN stations.

Non-standard tables

schanloc, network, affiliation and snetsta tables are not standard tables and must be edited by hand. They are dynamically updated by Antelope is another network starts exporting new stations. For this reason it is important to always use a copy of the latest versions from /iwrn/op/run/dbmaster/master_stations.

- schanloc is a combination of hand-built and dbbuild. If a station has a loc code, then there must be an entry in this table. seed2db handles this well. But Antelope will dynamically update this table when a new channel is imported. This often makes trouble when the chan row has the loc code appended to the sta code. These rows must be hand removed. There are many sta entries that have no loc code, these cause no problem and are not needed in the table. This is a dynamically updated table by Antelope any time a new channel comes into data acquisition, but often needs hand edits when a channel contains a loc code that cannot be managed by dbloc for earthquake processing. loc code is becoming an IRIS standard. A lot of scripts can't handle loc code.
- network is used by all stations. There are many different net entries based on stations that share particular processing needs. Some net entries are very broad, such as the AK network, some are quite narrow such as the Fourpkd network on Fourpeaked volcano. Always blow away any version of this table produced by dbbuild.
- affiliation is used by all stations. Groups of stations sharing characteristics are mapped to the network table via affiliation. Subset a particular station by name to determine what networks it is affiliated with. If upgrading an analog station to digital, you will probably want to add it to bband and mag: use RAG as an example. Blow away any version of this table built by dbbuild.
- snetsta is used to map stations AEIC imports from other networks. This table is dynamically updated by Antelope any time a new station comes into data acquisition. snetsta will need the proper snet hand applied when a new station is imported from another network, such as CN or AV. If this is not done, Antelope will try to incorrectly map an imported station to the AK net. An update to snetsta is only needed if a new station is coming online that's not in AK network.

Upgrading a station from analog to digital:

- Close off the site record in the analog_stations database one day before the start date entered in the dbbuild batch file.

Changing a site's coordinates:

- If its an analog station, close off the site record in the analog_stations database one day before the new GPS coordinates were changed, and then add a new row for the new site coordinates, starting on the appropriate day.

Removing duplicate site rows:

- Use a modified version of delete_rows.csh.

If all else fails:

Call Martin LaFevers at CVO (1 360 993 8902). He was in charge of the master stations database for several years until May 2008.