Database processing – Antelope based
Version 2010.237

Objective: Generating a db, visualizing the components of your db and what to do with it.

_____________________________________________________
SUMMARY

a) Generate database
   seed2db -v TA_DATALESS* TA_DB
   miniseed2db TA_DATA/* TA_DB
b) Run dbdetect to execute STA/LTA detectors on the waveforms
   *can use the default dbdetect.pf file
   dbdetect -v -arrivals -pf dbdetect.pf TA_DB
c) Run ttgrid to make a grid file
   *copy the ttgrid pf file to local directory and edit
   ttgrid -pf ttgrid.pf -time all TA_grid
d) Run dbgrassoc to produce origin and event tables
   copy the the dbgrassoc.pf pf file to local directory and edit
   dbgrassoc -pf pf/dbgrassoc.pf TA_DB TA_DB TA_grid
e) Run dbloc2 on the database to modify picks and calculate earthquake locations
   dbloc2 TA_DB
f) Calculate magnitude, associate events, etc
gh) Convert to other formats
h) Have fun with it! (but don’t mess it up ;)

Conventions:
Command line : > foreach file (‘ls TA_DATALESS/DATA*’)
Man page or help for command: (italic and size font 10)
Explanatory text or additional comments – regular size 12
Main Step: Antelope database generation (bold letters)
Secondary processing

1) Antelope database generation:

Creation of a database with metadata and data

a) metadata – populates tables
   • calibration,
   • instrument,
   • lastid
   • network
   • schanloc
• sensor
• site
• sitechan
• snetsta

```bash
> foreach file (`ls TA_DATALESS/DATAL*`) > foreach? seed2db -v $file TA_DB > foreach? end
```

*seed2db* will scan a complete SEED volume (including both control blockettes and waveform data) and make or merge them into a Datascope database using the CSS3.0 schema.

b) Data (waveforms) populates table
   - wfdisc

```bash
> miniseed2db -v TA_DATA/* TA_DB
```

c) View your database and understand it (dbe)
d) View your wfdisc table and play with it
e) Assign calibration values – what for? How?

*miniseed2db* reads input seed or miniseed files and creates wfdisc records which reference those files in the output database db. If an argument is a directory, then all files below that directory are examined.

```bash
> dbfix_calib TA_DB
```

Assign calib values from calibration table to wfdisc

Let’s try to make picks and calculate earthquake locations.

2) Location of events

a. Use dbdetect

“*dbdetect* runs STA/LTA algorithm on seismic data to determine the arrival times of seismic energy. This procedure results in marking the time where the amplitude of the trace rises significantly above the background noise level. This is necessary to determine where the earthquakes are located in time in the data.

First you will need to setup a parameter file (pf file) for use as input to dbdetect. This file is located in the directory `/opt/antelope/current_version/data/pf`, and the file name is dbdetect.pf. Copy this file into the working directory and edit it from the working directory. Begin by running dbdetect using the default parameters in the dbdetect.pf file on three days worth of data. For the example database, use the following command”

```bash
• cp parameter file on local directory pf
```

```bash
> cp $ANTELOPE/data/pf/dbdetect.pf pf
```
• set up
> PFPATH $ANTELOPE/data/pf: # to read default and local pf

> dbdetect -arrivals -pf pf/dbdetect.pf dbin dbout

  b. View your picks – dbpick

> dbpick TA_DB
  say yes to display database
  o fe
  o sd on
  o sc :.*Z
  o ts 2010057
  o tw 86000  # since in the pf file we only ask to detect Z

You should see a marker labeled “D” next to each of the arrivals similar to the figure below.
Note that some arrivals have more than one detection flag. This is because the detection algorithm was run on more than one frequency band so it is possible to trigger a detection at more than one frequency band. This is OK as these extraneous arrivals will be handled by dbgrassoc. Also note that sometimes there are detection flags in undesired locations. This is also OK because they will be ignored later in dbgrassoc. The most important thing is to try and get detection flags at ALL the times of the known arrivals.

c. ttgrid - compute a travel time grid file

Once the detection times have been generated we can search for events/arrivals based on these detection times using the program dbgrassoc. First, however, one must generate a travel time grid for dbgrassoc to search over. To generate this grid use the antelope command ttgrid.

- cp parameter file on local directory pf

\texttt{> cp$ANTELOPE/data/pf/ttgrid.pf pf}

- Customize ttgrid.pf

- Set path

\texttt{>set up PFPATH$ANTELOPE/data/pf/: # to read default and local pf files}

Parameter file - ttgrid.pf

\begin{verbatim}
params &Arr{
  local &Arr{
    mode edp   # defines an equal-distance projection regular 3-D mesh
    #ndist 10000
    latr 40 # reference latitude (origin of grid) – mid point
    lonr -109.0 # reference longitude (origin of grid) mid point
    nx 300 # Number of X-axis distance grid nodes
    ny 300 # Number of Y-axis distance grid nodes
    xmin -10.0 # Minimum value of X-axis distance grid in degrees
    xmax 10.0 # Maximum value of X-axis distance grid in degrees
  }
}
\end{verbatim}
if teleseismic events - alter the other tables

Now you can generate the grid by typing the following command:

```bash
> ttgrid -pf pf/ttgrid.pf -time all TA_DB > TA_grid
```

To display the grid that you just generated please type:

```bash
> displayttgrid TA_grid local
```

You can again check your db using dbpick
Customize your pf file as needed, considering your station’s location, and if you want to calculate grids for regional and/or teleseism signals.
The green circles represent the locations of the travel-time grid nodes. The yellow squares represent the station locations. The grid should cover the entire region where earthquakes are likely to occur. However, there is a tradeoff. If the grid is coarse, dbgrassoc will run fast, but the preliminary eq locations will be less accurate. If the grid is fine, dbgrassoc will run slower, but the initial earthquake positions will improve.

d. dbgrassoc

Spatial grid search based associator/locator dbgrassoc is the Datascope equivalent of orbassoc(1). The user should reference the orbassoc(1) man page for most program and configuration information. In fact, the dbgrassoc executable image file is the same exact file as the orbassoc executable image. Although dbgrassoc is 99% the same as orbassoc, there are a few differences which are explained here. The biggest difference is that there is no longer a dbtrigger program.

Once you have created a gridfile using ttgrid, run dbgrassoc to determine the time and preliminary locations of the earthquakes. Dbgrassoc reads the detection times from the detection table, then searches for a best fit solution to the earthquake’s location problem on the travel time grid. The output of dbgrassoc is four tables:

1) origin - contains the hypocenter parameters (lat, lon, depth, origin time, etc.)
2) event - lists the preferred origin record for an event
3) arrival - lists the phase, time of the seismic arrivals
4) assoc – connects arrival records to origin records, also list travel time residuals, etc.
• cp parameter file on local directory pf

> cp SANTELOPE/data/pf/dbgrassoc_pf

• Customize dbgrassoc pf

• Set path

> set up PFPATH SANTELOPE/data/pf:

```plaintext
# Parameter file for dbgrassoc

process_time_window 500.0    # Main detection processing time window
process_ncycle 20            # how often to do detection processing, in detections
process_tcycle 400.0         # how often to do detection processing, in delta time

grid_params &Arr{
local &Arr{
  nsta_thresh 4        # Minimum allowable number of stations
  nxd 11             # Number of east-west grid nodes for depth scans
  nyd 11             # Number of north-south grid nodes for depth scans
  cluster_twin 1.5    # Clustering time window
  try_S no             # yes = Try observations as both P and S
  # no = Observations are P only
  associate_S yes        # yes = Try to associate observations as both P and S
  reprocess_S yes       # yes = Reprocess when new S-associations found
  # phase_sifter 1
  auth orbassoc
  priority 5
  use_dwt yes
  dwt_dist_near 2.0
}
```

Since this example is for a local array, ignore the regional and tele tables. Also, you may need to comment out the line that has the parameter phase_sifter as in the above example. Probably the most important parameter here is nsta_thresh. This parameter defines a cutoff for the number of arrivals at stations required to generate an earthquake location. If this number is small, dbgrassoc will find a large number of earthquake solutions (the solutions may or may not be real earthquakes). If this number is large, dbgrassoc will find a smaller number of solutions, but the results should be more robust.

Recommendations:

- **Multiple iterations**
- **Visual check**
- If there are more events in the origin table than found by hand, try increasing nsta_thresh in dbgrassoc.pf and/or increasing the parameter thresh in ddbdetect.
- If there are more events found by hand than in the origin table, try decreasing nsta_thresh in dbgrassoc.pf and/or decreasing the parameter thresh in ddbdetect.

e. View your picks – dbpick

**Command line**

> dbgrassoc -pf pf/dbgrassoc.pf TA_DB TA_DB TA_grid
dbpick TA_DB

say yes to display database

- sd on
- sc :*::*Z # since in the pf file we only ask to detect Z
- ts 2010057
- tw 86000

A list of useful dbpick commands is given here.

<table>
<thead>
<tr>
<th>MOUSE BUTTON ACTION</th>
<th>MOUSE BUTTON ACTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>Left-Click Time Scroll to the left</td>
<td></td>
</tr>
<tr>
<td>Right-Click Time scroll to the right</td>
<td></td>
</tr>
<tr>
<td>Middle-Drag Time scroll left or right</td>
<td></td>
</tr>
<tr>
<td>Shift-Left-Click Time zoom in</td>
<td></td>
</tr>
<tr>
<td>Shift-Right-Click Time zoom out</td>
<td></td>
</tr>
<tr>
<td>Left-Click-on-Station-Label Select/Deselect station</td>
<td></td>
</tr>
<tr>
<td>Left-Drag-across-Station-Label Select/Deselect station</td>
<td></td>
</tr>
<tr>
<td>Left-Click-on-Arrival-Flag Adjust arrival time</td>
<td></td>
</tr>
<tr>
<td>Shift-Left-Click-on-Arrival-Flag Adjust arrival time uncertainty</td>
<td></td>
</tr>
<tr>
<td>Control-Shift-Left-Click-on-Arrival-Flag Adjust arrival amplitude-period</td>
<td></td>
</tr>
<tr>
<td>Control-Left-Click-on-Arrival-Flag Show arrival measurements</td>
<td></td>
</tr>
<tr>
<td>Middle-Click-on-Arrival-Flag Phase code menu</td>
<td></td>
</tr>
<tr>
<td>Right-Click-on-Arrival-Flag Arrival menu</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>COMMAND</th>
<th>KEY ACTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>fw</td>
<td>F first waveform</td>
</tr>
<tr>
<td>nw</td>
<td>N next waveform</td>
</tr>
<tr>
<td>pw</td>
<td>P previous waveform</td>
</tr>
<tr>
<td>fa</td>
<td>f first arrival</td>
</tr>
<tr>
<td>na</td>
<td>n next arrival</td>
</tr>
<tr>
<td>pa</td>
<td>p previous arrival</td>
</tr>
<tr>
<td>pp phase</td>
<td>phase next phase</td>
</tr>
<tr>
<td>fe</td>
<td>e first event</td>
</tr>
<tr>
<td>ne</td>
<td>e next event</td>
</tr>
<tr>
<td>pe</td>
<td>previous event</td>
</tr>
<tr>
<td>Fe</td>
<td>find event</td>
</tr>
<tr>
<td>dw</td>
<td>delete blank traces</td>
</tr>
<tr>
<td>swd</td>
<td>show only all traces with detections</td>
</tr>
<tr>
<td>swa</td>
<td>show only all traces with arrivals</td>
</tr>
<tr>
<td>swda</td>
<td>show only all traces with detections or arrivals</td>
</tr>
<tr>
<td>tfit</td>
<td>t toggle time fit</td>
</tr>
<tr>
<td>sfit</td>
<td>s station fit</td>
</tr>
<tr>
<td>a</td>
<td>Repaint arrivals</td>
</tr>
<tr>
<td>r</td>
<td>Repaint window</td>
</tr>
<tr>
<td>R</td>
<td>Redraw and repaint window</td>
</tr>
<tr>
<td>sc sta:chan</td>
<td>display matching station/channels</td>
</tr>
<tr>
<td>rec</td>
<td>arrange stations by increasing distance.</td>
</tr>
<tr>
<td>ts time</td>
<td>start display at this time</td>
</tr>
<tr>
<td>tw time</td>
<td>change time window</td>
</tr>
<tr>
<td>cw start number</td>
<td>change channel zoom window</td>
</tr>
<tr>
<td>ph phase</td>
<td>change default phase</td>
</tr>
</tbody>
</table>
gp {segment|zero|interp|none} change telemetry gap processing

oa dbname open arrival database
oe dbname open event database
od dbname open detection database
se orid select event from event database
sp phase_list select predicted arrival phases
tc time_corr specify overlay arrival time correction
tse time scroll to the current event.
ae associate event with arrivals in display.
wa write associations to input database.
sa on/off show/hide arrivals
soa on/off show/hide overlay arrivals
sd on/off show/hide detections
sw on/off show/hide waveforms
sf on/off show/hide first motions
sv sta/off show/hide velocity annotations
cts on/off amplitude units in couts or physical units.
mg on/off amplitude units in mg or nm/s**2 for acceleration.
gr on/off show/hide background st grids
pal on/off time align on P arrivals on/off
filter index set filter index
sm {bilin|block} set grid display mode
fc hue light sat set hue, light and sat of foreground color.
ps plotfile Make PostScript window dump to plotfile.
pr Make PostScript window dump to printer.
exec command execute a UNIX command
quit quit dbpick
help print this menu

To make arrival picks:

1. Click on add arrivals button.
2. Move the cursor until it is aligned on the arrival time and left click.
3. Select the phase by middle clicking on the arrival flag.
4. Hold shift and left click an arrival flag to add error bars.

Once all of the picks have been made, use dbloc2 to locate the event.

f. dbloc2 - interactive hypocenter location

$dbloc2$ is a collection of several programs which run together under the control of a perl script, $dbloc2$. It facilitates the location of hypocenters from previously picked trace data, while allowing interaction with the location program and the ability to edit arrival picks. It can also attempt to associate groups of arrivals with other catalogs, for instance the PDE or REB catalogs

> dbloc2 TA_DB

This will bring up a window that looks something like this:
Event location

1. Select arrivals to be used in location by left clicking them in the dbloc2 window. Bold letters have been selected, gray letters have not. Initially start with all.
2. Choose location algorithm and velocity model (located just right of the word “Locate” in the lower left corner of the dbloc2 window). I recommend dbgenloc w/ the taup/iasp91 travel-time calculator/velocity model.
3. Choose location options below algorithm select. I recommend using either S-P time or a rectangular grid search for the initial location method.
4. Click the locate button.

dbloc2 Troubleshooting

1. Do not close the dbpick window when running from dbloc2. It will crash the program. Just minimize to get out of the way. The program will close when you file>quit dbloc2.
2. Do not file>quit w/out saving. This will leave a corrupt database.
3. If dbloc2 doesn’t work, try running with the –r option - dbloc2 –r ‘dbname’. This will clear the tmp/trial database.
4. If you get a message that says 'Save: arrivals have moved since location orid ##' you will have to reassociate. To do this click the button in the above figure that says 'keep' (middle left) until it says 'reassoc'. Then click the 'regroup' button near the top of the window. The regroup button is like the refresh button on your internet browser.

5. Sometimes dbloc2 will ask you to unregister buttons when trying to start. Copy this line and paste it on the command line, hit return, and try again.

6. If dbloc2 arrival flags turn red after locating, it means that your new calculated location is so far off from the previous location that the program thinks it is a new event. This may or may not be true. If it is true, use the new evid. If it is not, click on the blue button that says 226 in the above figure (middle left). Select 'set evid' and choose the previous evid.

7. Sometimes arrivals that are present in dbpick do not show up in the dbloc2 window. First try clicking the button that says predicted (middle left) and choose the option time. This will place the dbloc2 arrival flags at their absolute times. If this doesn't work, increase the time window scroll bar (near the top) and click the 'regroup' button.

3) Just for fun

   a. Converting from other formats to mseed

   http://www.iris.edu/software/downloads/conversion/
   http://www.orfeus.eu.org/Software/conversion.html

Waveform format conversions _ FROM ORFEUS
Last update: December 12, 2008
**Warning!** Seismological format conversion is tricky. A format may not contain all relevant information for other formats. Consequently, the programs listed below may not solve your complete conversion problem. Please contact [Torild van Eck](mailto:torild.vaneck@iris.edu) on mistakes, suggestions, additions.

<table>
<thead>
<tr>
<th>Conversion</th>
<th>Program</th>
<th>Documentation</th>
<th>Programmer</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>AH2ASCI</td>
<td>ah2asc.c</td>
<td>README</td>
<td>IRIS/DMC</td>
<td></td>
</tr>
<tr>
<td>AH2CSS</td>
<td>ah2css.c</td>
<td>README</td>
<td>IRIS/DMC</td>
<td></td>
</tr>
<tr>
<td>AH2SAC</td>
<td>ah2sac.c</td>
<td>README, PASSCAL guide</td>
<td>IRIS/DMC, IRIS/PASSCAL</td>
<td></td>
</tr>
<tr>
<td>AH2SEED</td>
<td>datascope</td>
<td>see program</td>
<td>IRIS/JSPC</td>
<td>CSS3.0; solaris</td>
</tr>
<tr>
<td>AH2STD</td>
<td>ah2std.c</td>
<td>README</td>
<td>IRIS/DMC</td>
<td></td>
</tr>
<tr>
<td>AH2XDR</td>
<td>ah2xdr.c</td>
<td>README</td>
<td>IRIS/DMC</td>
<td></td>
</tr>
<tr>
<td>ASC2AH</td>
<td>asc2ah.c</td>
<td>README</td>
<td>IRIS/DMC</td>
<td></td>
</tr>
<tr>
<td>CSS2AH</td>
<td>css2ah.c</td>
<td>README</td>
<td>IRIS/DMC</td>
<td></td>
</tr>
<tr>
<td>Command</td>
<td>Description</td>
<td>Format</td>
<td>Authors</td>
<td>Notes</td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
<td>--------</td>
<td>---------</td>
<td>-------</td>
</tr>
<tr>
<td>CSS2GSE</td>
<td>codeco_3.3c Documentation</td>
<td>Urs Kradolfer</td>
<td>CSS2.8/3.0</td>
<td>GSE1.0/2.x</td>
</tr>
<tr>
<td>CSS2MSEED</td>
<td>codeco_3.3c Documentation</td>
<td>Urs Kradolfer</td>
<td>CSS2.8/3.0</td>
<td></td>
</tr>
<tr>
<td>CSS2SAC</td>
<td>codeco_3.3c Documentation</td>
<td>Urs Kradolfer</td>
<td>CSS3.0; solaris</td>
<td></td>
</tr>
<tr>
<td>CSS2SEED</td>
<td>datascope see program</td>
<td>IRIS/SPC</td>
<td>CSS3.0; solaris. Reported problems</td>
<td></td>
</tr>
<tr>
<td>DMUX2AH</td>
<td>SUDS see program</td>
<td>Peter Ward, USGS</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>EQ2ASC</td>
<td>convseis manual (postscript)</td>
<td>Oncescu &amp; Rizescu</td>
<td>PCEQ (IASPEI) -&gt; ASCII</td>
<td></td>
</tr>
<tr>
<td>ESSTF2SAC</td>
<td>sismoutil howto.txt</td>
<td>Sebastien Judenherc</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>EQ2GSE</td>
<td>convseis manual (postscript)</td>
<td>Oncescu &amp; Rizescu</td>
<td>PCEQ (IASPEI) -&gt; GSE1.0 (uncompressed)</td>
<td></td>
</tr>
<tr>
<td>EQ2SEGY</td>
<td>convseis manual (postscript)</td>
<td>Oncescu &amp; Rizescu</td>
<td>PCEQ (IASPEI) -&gt; PC-SEGY</td>
<td></td>
</tr>
<tr>
<td>EQ2SEIS</td>
<td>convseis manual (postscript)</td>
<td>Oncescu &amp; Rizescu</td>
<td>PCEQ (IASPEI) -&gt; SEIS89 format</td>
<td></td>
</tr>
<tr>
<td>EQ2SG</td>
<td>convseis manual (postscript)</td>
<td>Oncescu &amp; Rizescu</td>
<td>PCEQ (IASPEI) -&gt; SeisGram binary</td>
<td></td>
</tr>
<tr>
<td>EQ2SUDS</td>
<td>convseis manual (postscript)</td>
<td>Oncescu &amp; Rizescu</td>
<td>PCEQ (IASPEI) -&gt; PC-SUDS</td>
<td></td>
</tr>
<tr>
<td>GCF2AH</td>
<td>gcf for UNIX User manual SAM/DM software.</td>
<td>ISTI</td>
<td>Guralp conversion software</td>
<td></td>
</tr>
<tr>
<td>GCF2SEGY</td>
<td>gcf for UNIX User manual SAM/DM software.</td>
<td>ISTI</td>
<td>Guralp conversion software</td>
<td></td>
</tr>
<tr>
<td>GCF2SAC</td>
<td>gcf for UNIX User manual SAM/DM software.</td>
<td>ISTI</td>
<td>Guralp conversion software</td>
<td></td>
</tr>
<tr>
<td>GDSN2SAC</td>
<td>evt2sac README</td>
<td>ORFEUS</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>GDSN2AH</td>
<td>gdsn2ah.c README</td>
<td>IRIS/DMC</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>GSE2CSS</td>
<td>codeco_3.3c Documentation</td>
<td>Urs Kradolfer</td>
<td>GSE1.0/2.x</td>
<td>CSS2.8/3.0 UNIX HP/SUN, Linux</td>
</tr>
<tr>
<td>GSE2GSE</td>
<td>codeco_3.3c Documentation</td>
<td>Urs Kradolfer</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>GSE2GSE</td>
<td>convseis manual (postscript)</td>
<td>Oncescu &amp; Rizescu</td>
<td>GSE1.0 &lt;-&gt; GSE2.0 DOS/Windows</td>
<td></td>
</tr>
<tr>
<td>GSE2MSEED</td>
<td>codeco_3.3c Documentation</td>
<td>Urs Kradolfer</td>
<td>GSE1.0/2.x UNIX HP/SUN, Linux</td>
<td></td>
</tr>
<tr>
<td>GSE2MSEED</td>
<td>gse2mseed</td>
<td>IRIS, Chad Trabant</td>
<td>GSE 2.x/IMS 1.0, INT or CM6 Solaris, Linux, Mac OSX and Windows</td>
<td></td>
</tr>
<tr>
<td>GSE2SAC</td>
<td>codeco_3.3c gssac Documentation</td>
<td>Urs Kradolfer/Hugues Dufumier</td>
<td>GSE1.0/2.x, SAC(A/B)</td>
<td></td>
</tr>
<tr>
<td>Library</td>
<td>Description</td>
<td>Author</td>
<td>Version</td>
<td></td>
</tr>
<tr>
<td>-------------------</td>
<td>------------------------------------</td>
<td>-------------------------------</td>
<td>---------------</td>
<td></td>
</tr>
<tr>
<td>MSEED2AH</td>
<td>ms2ah</td>
<td>OUG UCB README</td>
<td>IRIS/PASSCAL</td>
<td></td>
</tr>
<tr>
<td>MSEED2CSS</td>
<td>codeco_3.3c</td>
<td>Documentation</td>
<td>Urs Kradolfer</td>
<td></td>
</tr>
<tr>
<td>MSEED2GSE</td>
<td>codeco_3.3c</td>
<td>Documentation</td>
<td>Urs Kradolfer</td>
<td></td>
</tr>
<tr>
<td>MSEED2PITSA</td>
<td>HAM</td>
<td>HAM</td>
<td>Ernesto Del Prete</td>
<td></td>
</tr>
<tr>
<td>MSEED2SAC</td>
<td>codeco_3.3c</td>
<td>Documentation</td>
<td>Urs Kradolfer</td>
<td></td>
</tr>
<tr>
<td>MSEED2SEGY</td>
<td>PASSCAL software</td>
<td>PASSCAL guide</td>
<td>IRIS/PASSCAL</td>
<td></td>
</tr>
<tr>
<td>PDAS2AH</td>
<td>p2ah</td>
<td>README</td>
<td>Guenther Asch</td>
<td></td>
</tr>
<tr>
<td>PDAS2MSEED</td>
<td>pdas2mseed</td>
<td>README</td>
<td>Paul Denton</td>
<td></td>
</tr>
<tr>
<td>PDAS2SEGY</td>
<td>p2segy</td>
<td>README</td>
<td>Guenther Asch</td>
<td></td>
</tr>
<tr>
<td>PING2AH</td>
<td>pina2ah</td>
<td>README</td>
<td>IRIS/DMC</td>
<td></td>
</tr>
<tr>
<td>REF2SEGY</td>
<td>PASSCAL software</td>
<td>PASSCAL guide</td>
<td>IRIS/PASSCAL</td>
<td></td>
</tr>
<tr>
<td>REF2MSEED</td>
<td>PASSCAL software</td>
<td>PASSCAL guide</td>
<td>IRIS/PASSCAL</td>
<td></td>
</tr>
<tr>
<td>SAC2AH</td>
<td>sac2ah.c</td>
<td>README</td>
<td>IRIS/DMC</td>
<td></td>
</tr>
<tr>
<td>SAC2CSS</td>
<td>codeco_3.3c</td>
<td>Documentation</td>
<td>Urs Kradolfer</td>
<td></td>
</tr>
<tr>
<td>SAC2ESSTF</td>
<td>sismoutil-0.9b</td>
<td>README</td>
<td>Sebastien Judenherc</td>
<td></td>
</tr>
<tr>
<td>SAC2GSE</td>
<td>codeco_3.3c</td>
<td>Documentation</td>
<td>Urs Kradolfer</td>
<td></td>
</tr>
<tr>
<td>SAC2LOCAL</td>
<td>sac2local.zip</td>
<td></td>
<td>Meijian An</td>
<td></td>
</tr>
<tr>
<td>SAC2MSEED</td>
<td>codeco_3.3c</td>
<td>Documentation</td>
<td>Urs Kradolfer</td>
<td></td>
</tr>
<tr>
<td>SACSUN2PC</td>
<td>sac.appl</td>
<td>README</td>
<td>Tom McSweeney</td>
<td></td>
</tr>
<tr>
<td>SACPC2SUN</td>
<td>sac.appl</td>
<td>README</td>
<td>Tom McSweeney</td>
<td></td>
</tr>
<tr>
<td>SAC2SH</td>
<td>sac2sh</td>
<td></td>
<td>Raphael Garcia</td>
<td></td>
</tr>
<tr>
<td>SAM2SUDS</td>
<td>Guralp GSLConv</td>
<td>Guralp Software pages</td>
<td>Guralp</td>
<td></td>
</tr>
<tr>
<td>SAM2SAC</td>
<td>Guralp GSLConv</td>
<td>Guralp Software pages</td>
<td>Guralp</td>
<td></td>
</tr>
<tr>
<td>SAM2SEGY</td>
<td>Guralp GSLConv</td>
<td>Guralp Software pages</td>
<td>Guralp</td>
<td></td>
</tr>
</tbody>
</table>
b. Converting to other formats
   i. db2sac
ii. db2sd
iii. db2segy
iv. db2ah
v. db2sd
vi. db2sync

c. Moving around
   i. db2sql
   ii. db2pqc
   iii. db2neic
   iv. db2mac
   v. db2perl

d. Antelope established interfaces
   i. Command line
   ii. C
   iii. Fortran
   iv. Perl
   v. TCL/TL
   vi. PHP - contributed
   vii. Python - contributed
   viii. Matlab contributed
   ix. Java - contributed

By: Eliana Arias-Dotson at IRIS/PASSCAL (eliana@passcal.nmt.edu)