CONVERTING DATA – The AASPI_util SEGY to AASPI format conversion tab

Contents

Data Conversion SEGY to AASPI format

Data conversion will be the first and most difficult step in the AASPI workflow. Most oil companies have a small army of exploration technologists whose most time-consuming jobs in importing and exporting data in and out of workstation software. Their salaries are well earned! All subsequent steps using the AASPI software will be a lot easier.

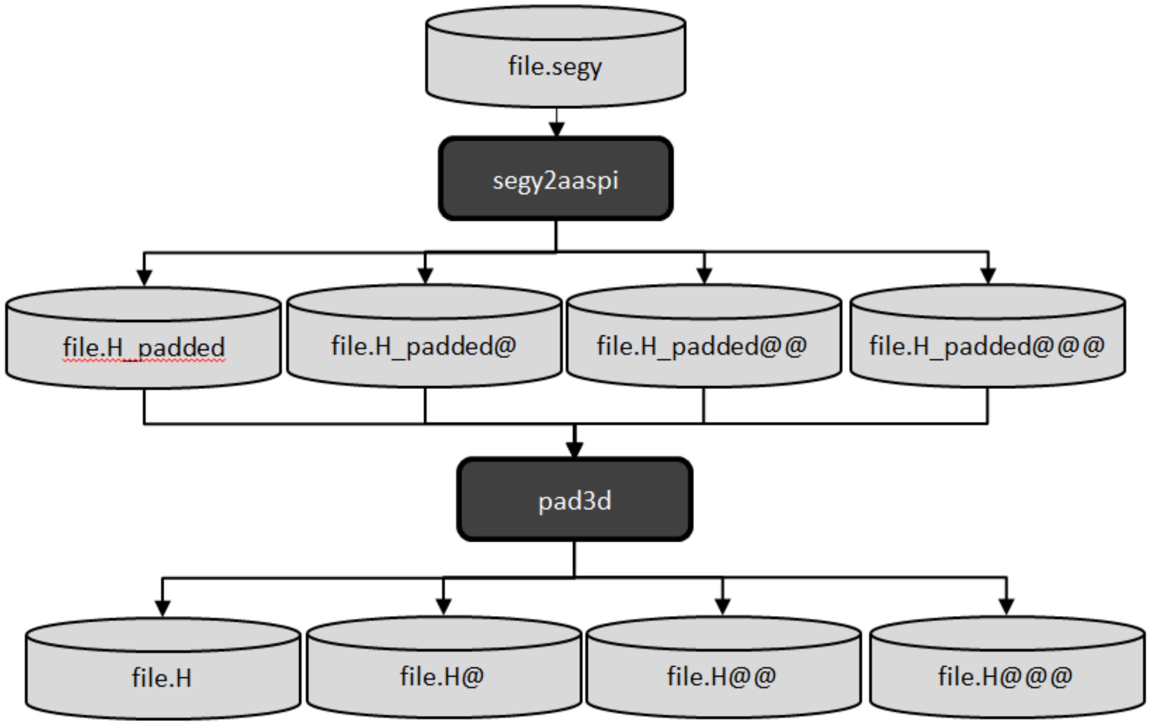
The SEGY standard should be considered to be a convenient way to transfer seismic data from one application to another. As such, the structure of the data is minimal, simply consisting of a line header and a suite of seismic traces, each of which has a seismic trace header and a seismic data component. Without examining these trace headers, there is nothing in the SEGY line header that says whether a data volume is 2D, 3D, or higher dimension (e.g. migrated data gathers containing traces with different offsets and azimuths, giving 5D data). Indeed, most SEGY data does not have any structure, with dead traces from a 3D interpretation project often being eliminated to minimize the space needed for data transfer and storage.

Much of the AASPI software assumes the data is stored as a 3D, 4D, or 5D hypercube of data, where dead traces have been padded with zeroes. Such padding allows us to construct simple computational stencils. For example if the data axes have n_1 samples in time, n_2 cdps, and n_3 lines, then the six voxels surrounding voxel (j_1,j_2,j_3) would be (j_1-1,j_2,j_3), (j_1+1,j_2,j_3) , (j_1,j_2-1,j_3) , (j_1,j_2+1,j_3) , (j_1,j_2,j_3-1) and (j_1,j_2,j_3+1).

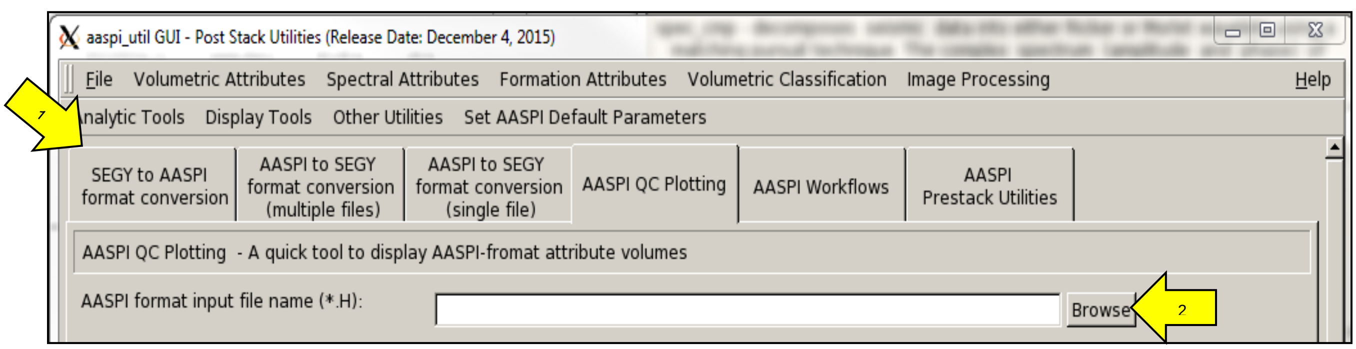
Conversion Flow Chart

The first step is to simply convert the SEGY format data, in this image called **file.segy** to AASPI format, as is, without padding. Since this file is not the one we need, we add the value **_padded** to the end. Once padding is completed, the python script erases this temporary file. As discussed in the description of the AASPI data format, the header names are will be stored in **file.H_padded@@** and the header values in **file.H_padded@@@.**

Data Conversion: The **AASPI_util** SEGY to AASPI format conversion tab

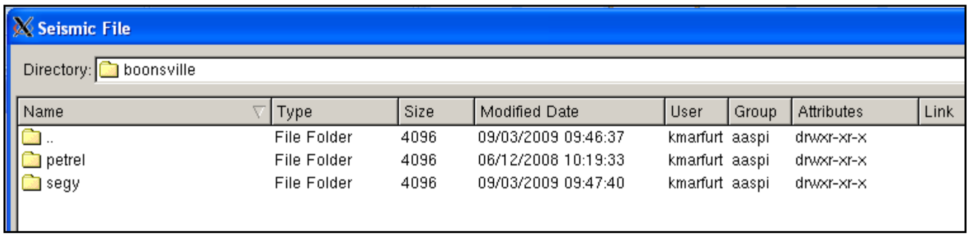


In the AASPI utilities GUI, (1) click on the **SEGY to AASPI** tab on the upper left.

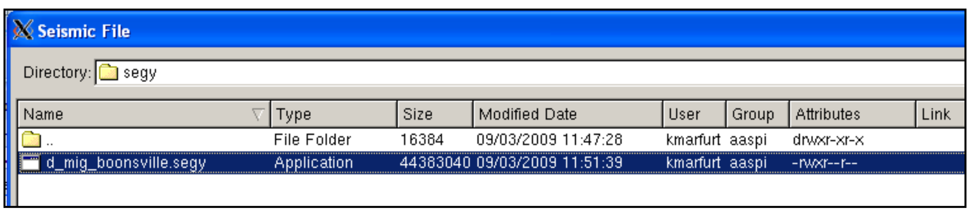


This tab will convert your seismic data from the SEG-Y standard (we think of it as the SEG-Y 'suggestion' since there are so many variations) to the somewhat simpler, more flexible AASPI format. The AASPI data structures are identical to those designed by the Stanford Exploration Project (SEP) and adopted in most part by the Madagascar work group. While the format is nearly identical (with all AASPI volumes readable by SEPLib and Madagascar), the internals are different. First, **aaspi_io** can read all the 2003 SEGY formats, such as 32-bit IEEE format, 16-bit integer format, ASCII headers, as well as the more common 32-bit IBM format and EBCDIC headers. **aaspi_io** circumvents all the big-endian/little-endian headaches, as well as limitations to the length of the file name. Most important, **aaspi_io** compiles and runs on 64-bit Linux as well as Windows 7, 8, and 10 operating systems. For this reason, the main difference between **aaspi_io** and SEP i/o is that **aaspi_io** will *not* support Linux pipes, since pipes is not supported by Windows. All input and output seismic data and attribute files will be in AASPI format, which at the end of the day, need to be converted back to SEG-Y format (the 2nd and 3rd tabs near the top of the panel).

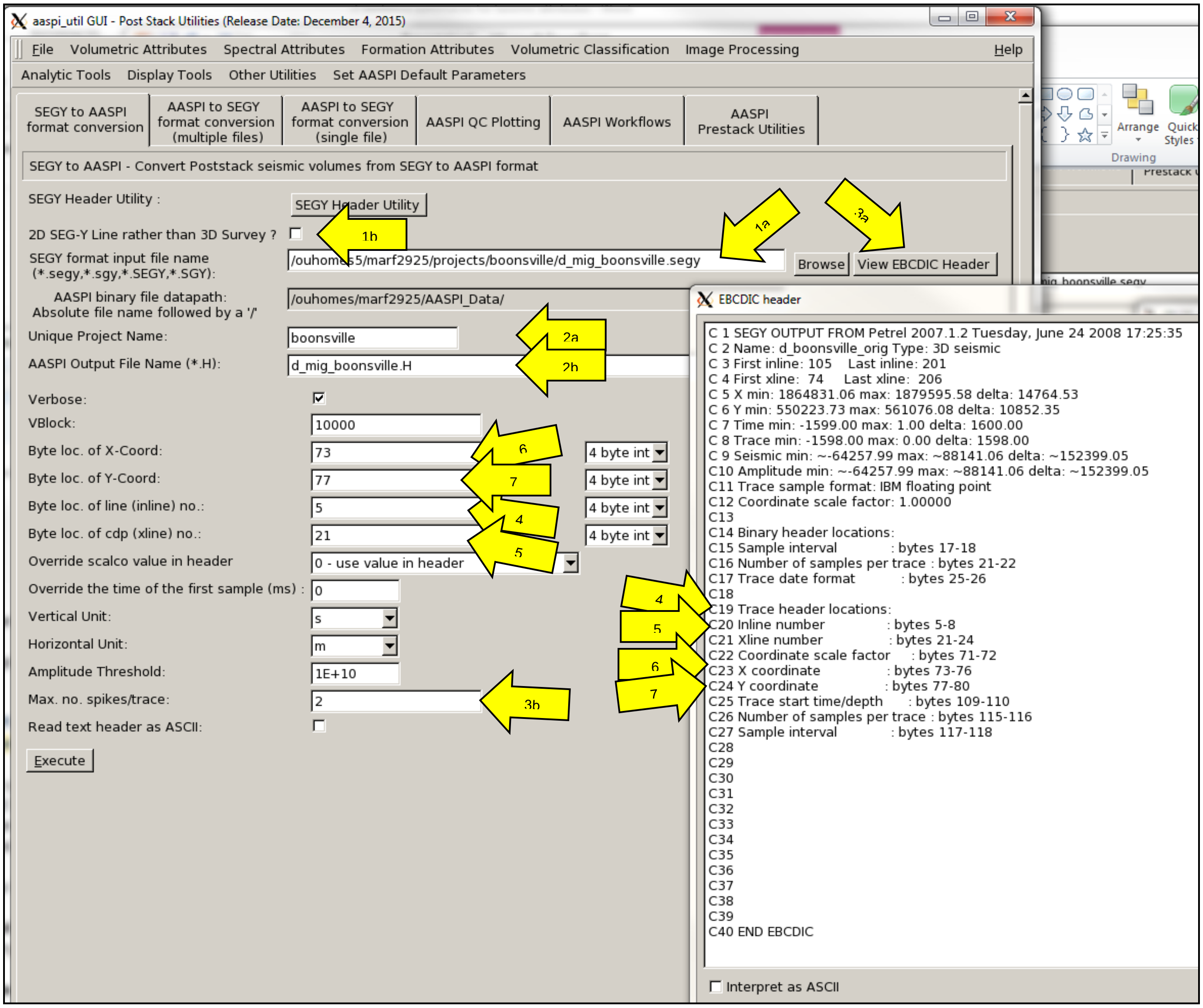
To enter the file name to be convert, simply (2) click the *Browse* button, a window will pop up containing all files that end in either '.segy' or '.sgy' as well as any folders with those name that lie below the project level. In my case I previously copied my segy input data to a subdirectory called 'segy' immediately below the directory in which I invoked **aaspi_util**.



Next, click the 'segy' folder to obtain



Then click the input segy seismic amplitude data file you want to use, **d_mig_boonsville.segy** (you can also highlight it and click OK).



Reexamining the **aaspi_util** GUI, note that the file name (1a) appears in the previously blank box. If the input data are a 2D vs. a 3D line, (1b) indicate so with a checkmark. Each seismic volume will carry along a unique project name, commonly the survey name. (2a) Type that particular name in, then (2b) type in the name of your output AASPI-format seismic amplitude data file. The simplest AASPI workflow is to use a common name with a given project, with one project per directory. In this case we will use the name *boonsville* and then use prefixes and suffixes to define attribute types and version numbers. A good workflow is to name your AASPI-format seismic amplitude data file **d_mig** to indicate it is amplitude data and migrated. Then tack on the project name (in this case *boonsville*). Finally, the AASPI (and SEP) convention is to add the suffix *.H to the end. With this entire preamble, next to the arrow (2) type in **d_mig_boonsville.H** .

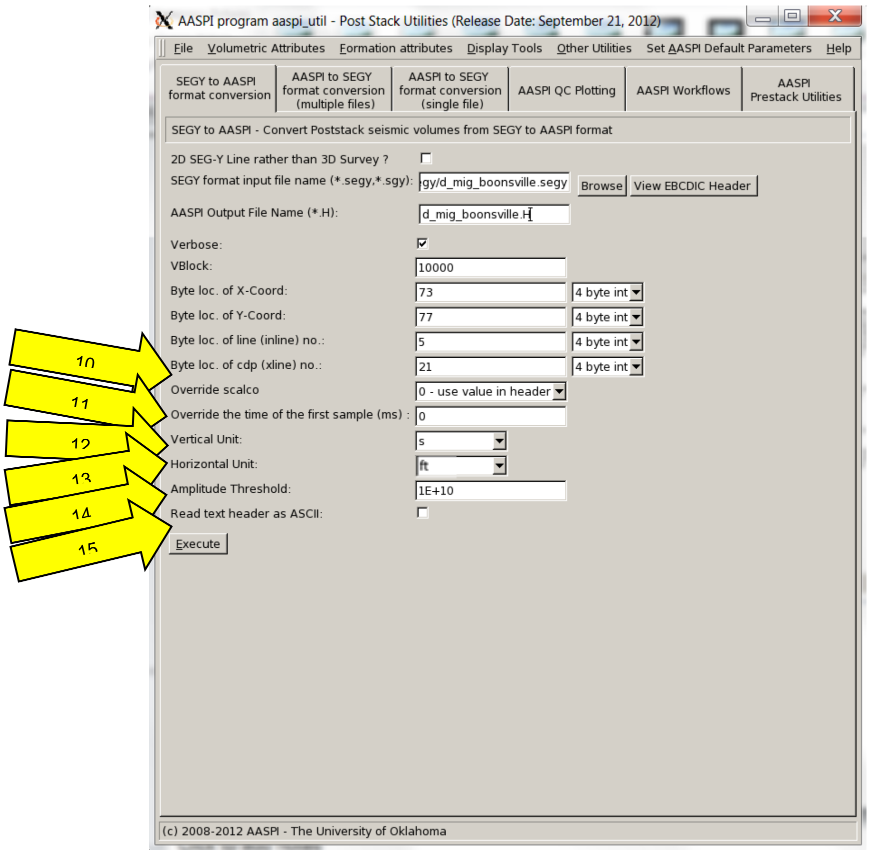
Next, you will want to click (3a) 'View EBCDIC Header' to determine where the various trace headers are defined for this seismic amplitude volume. The Extended Binary Coded Data

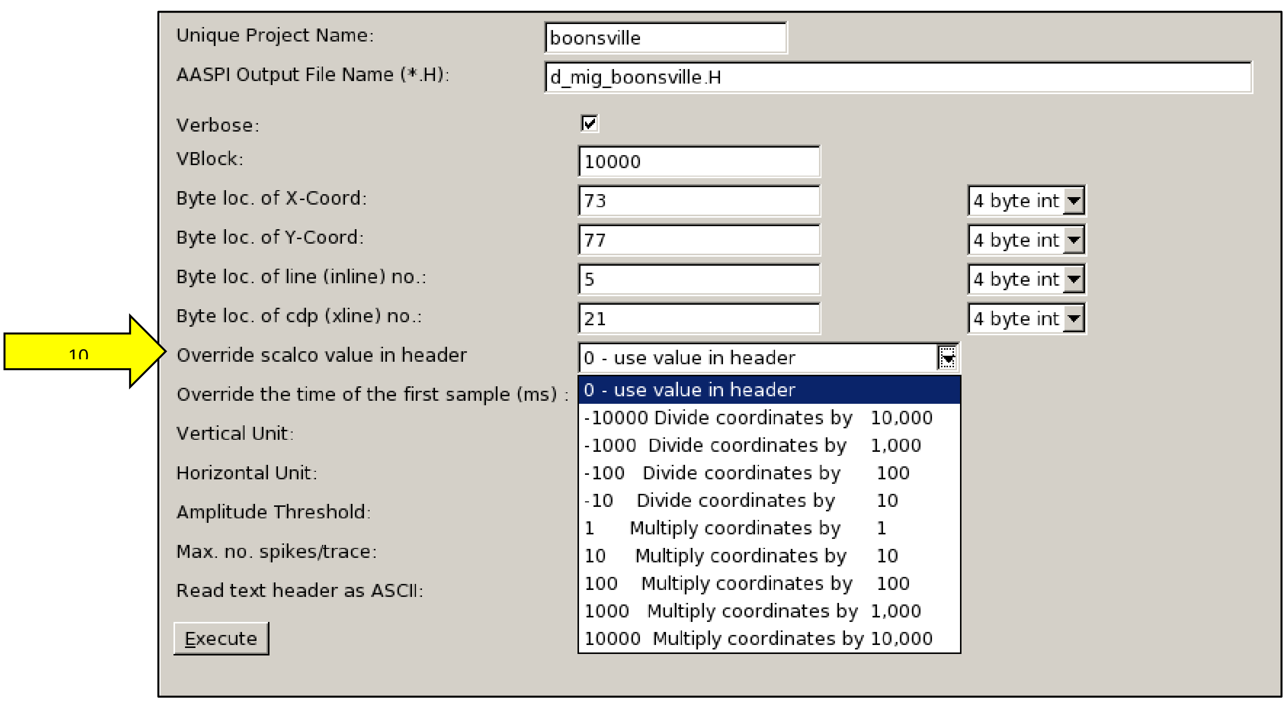
Interchange Code (EBCDIC) used by IBM in the 1970s has been almost entirely supplanted by the American Standard Code for Information Interchange (ASCII) for storing text information. However, EBCDIC is well preserved in the SEG-Y standard. If the EBCDIC header appears to be garbage (e.g. exports from seismic inversion software packages) it was probably saved in ASCII format. If so, place a check mark in the (3b) ASCII header button.

Almost all commercial software requires inline and crossline numbers. Most packages (Petrel, Charisma, etc.) require accurate x and y coordinates for each trace as well, while others (e.g. Kingdom Suite) may allow you to define the coordinates of the four survey corners. The 2002 SEG-Y rev1 standard states that the (in)line number should be in bytes 189-192, the crossline number (CDP no.) in bytes 193-196, the CDP x (Easting) coordinate bytes 181-184, and the CDP y (Northing) coordinate in bytes 185-188. Since this standard postdates most of our software interpretation packages, the software vendors had to define their own software-specific defaults long before 2002, thus cementing in the confusion. Be forewarned that at least one of the big four seismic acquisition companies commonly provides inline and crossline numbers in the (later adopted) SEGY x and y coordinate locations and puts the coordinates in the SEGY inline and crossline number locations.

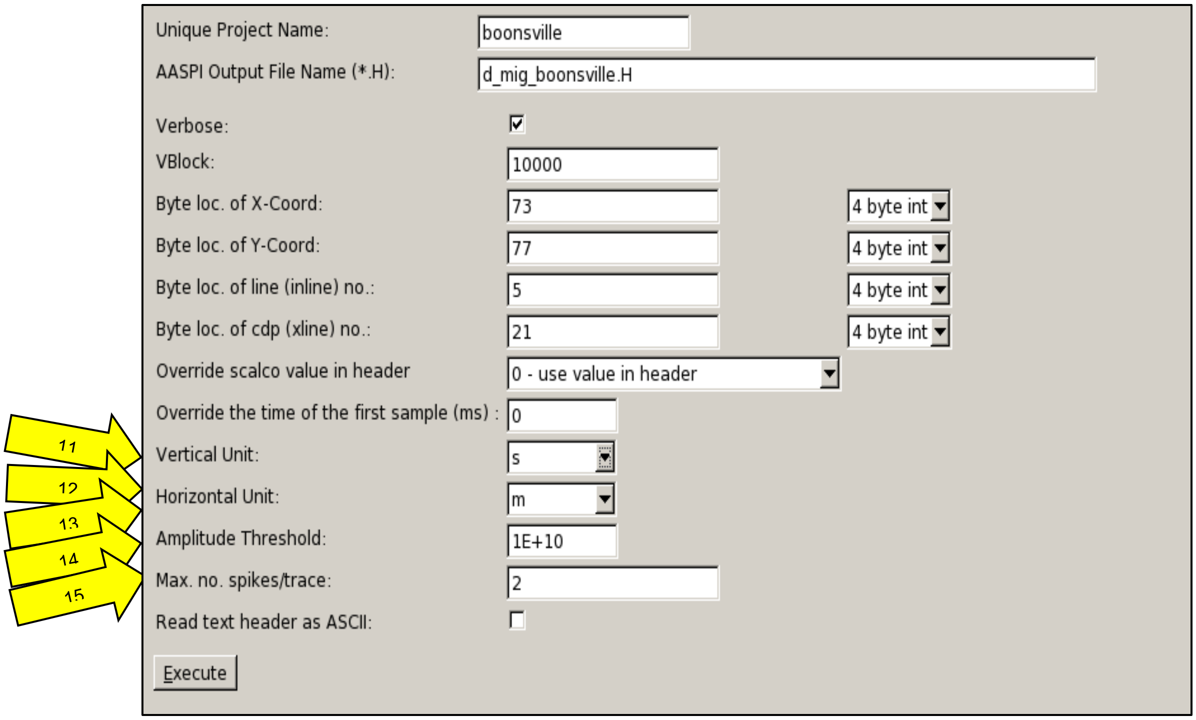
In this data volume, which was exported from Petrel, (4) the line no. starts in byte 5, the (5) crossline (cdp) no. in byte 21, the (6) cdp x coordinate in 73, and the (7) cdp y coordinate in 77. Type the value ‘5’ to indicate the beginning byte number into the **Byte loc. of line (inline) no**. box, followed by values of 21, 73, and 77 for the next three entries. Finally, (8) close the EBCDIC window.

Data Conversion: The **AASPI_util** SEGY to AASPI format conversion tab





After we convert to AASPI format in a later step and inspect the inline and crossline spacing (reasonable numbers might be 110 ft, 30 m, and so forth) we may find values that are orders of magnitude too low or too high (e.g. 1100 ft or 3 m). Such errors imply that the SEG-Y standard 'scale coordinate' value (10) is inconsistent with the *x* and *y* values stored in the data. Such errors can arise when exporting data in and out of some processing and interpretation packages. Override this value if necessary.



The time of the first sample can also be corrupted in data export and transfer: type in the correct first sample time/depth (11) if it is wrong.

<u>We will always wish to check our vertical and horizontal measurement units</u>. In (12) use the pull-down arrow to select the vertical units. The default for SEG-Y time data will be 's' (for seconds), with a 2 ms sample increment stored as 2000 μs (microseconds). Depth data are more challenging. If the physical sample increment is 5 m (or 15 ft), and if we store them as 5000 mm, (or 15000 mft) then you should enter the unit 'km' (or 'kft') as the vertical unit. You would type 'm' ('ft') only if the physical 5 m (or 15 ft) sample increment is stored as 5000000 μm (15000000μft).

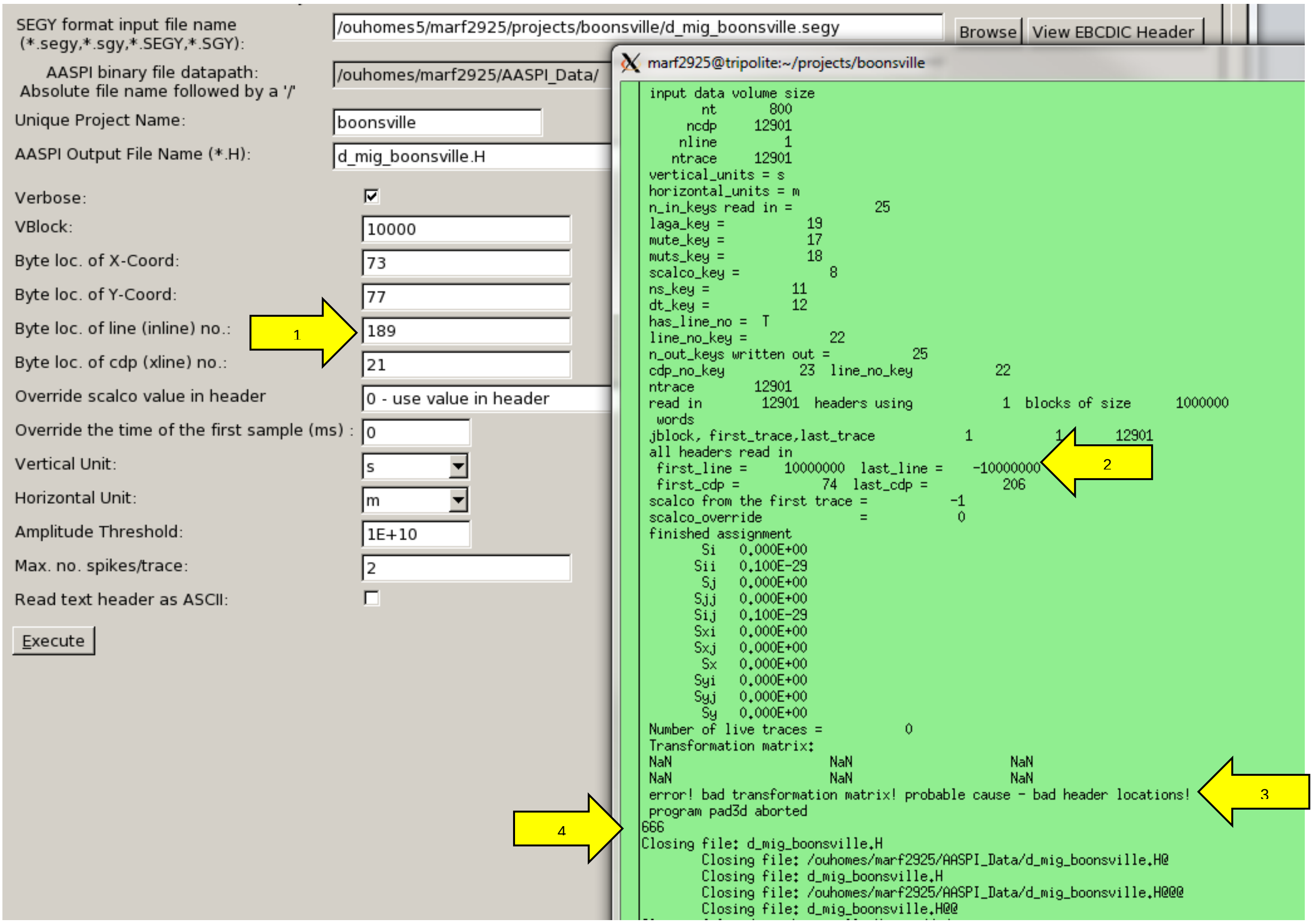
In the United States, almost all surveys define horizontal distances in ft. In almost every other place in the world, we will use m. However, be forewarned that you may obtain trade data from a European operator working the Gulf of Mexico with survey coordinates converted to m. There is a SEG-Y flag for this (1 for m, 2 for ft) but it is often set to 0 during processing and import/export from various packages. For this reason, the AASPI software allows you to (13) explicitly choose the 'Horizontal units' using the dropdown arrow selector.

Finally, from time to time you may obtain a data set with 'glitches'. Ideally, such numbers are flagged somewhere along the way as 'NaN's, which stands for 'Not a Number', but it can be a cold, cruel world out there. Tape transcription, faulty disk drive controllers, or errors in seismic processing with insufficient internal error checking can introduce bad numbers. If you have NaNs, you will need to clip your data by (14) typing in an appropriate number (say 10 standard deviations away from 0.0 for data that have had the outliers removed). The AASPI software will detect the outliers and attempt to interpolate their values from adjacent time or depth samples.

In general, if you work as an interpreter, it is good practice to QC your data using your workstation interpretation software of choice and then to export the (potentially cropped) seismic amplitude data volume in 32-bit IBM floating point format using what your workstation defines as the SEG-Y standard. Your workstation will always store the inline no., crossline no., x, and y locations in the exact same place, such that once you have figured it out, you will have few further problems. In addition, most interpretation software will generate a histogram. You can clip the 32-bit data to remove any glitches before exporting as SEG-Y. If you are working for a processing company, all of these features described above are normal operating procedure and fade into the background of more significant data format problems that you deal with daily.

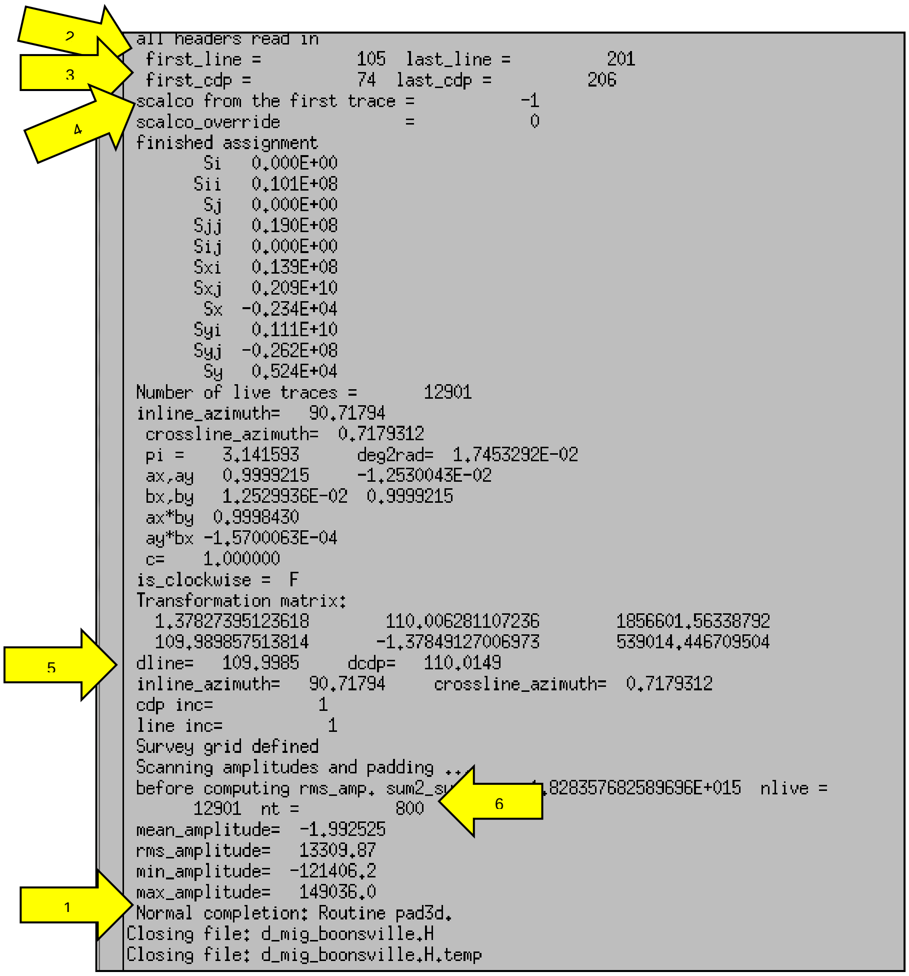
Once you have defined all these parameters, double-click (15) ‘*Execute*’.

Let us first look at an example of a conversion gone badly. In this example below, we have (1) left in the SEG-Y 2002 value of 189 for **Byte loc. of line (inline) no.** and clicked *Execute.*



The output of the job will be displayed in the original (in this case, green) Linux *xterm* from which you launched **aaspi_util**. Note that the first and last cdp numbers are reasonable with values of 74 and 206, but that (2) the first line and last line numbers are both -10000000. These incorrect values result in (3) an error message. For those of you who read the Book of Revelations every night before going to bed, you will recognize that (4) a stop code of 666 is not a good thing to have, since it is the 'sign of the beast'. Whenever you see a stop code of 666, you need to look through the output to determine if an error message has been printed out. In this case the error message reads 'probable trace header error!'.

Let us go back, type in the correct byte location ('5') for the **Byte loc. of line (inline) no.**, and click *Execute* again. The **aaspi_util** GUI sends output to the original window from which it was executed (displayed in the lower right). The GUI runs a shell script, which can be found (and modified if you wish) under the AASPI directory as **${AASPIHOME}/pyscripts/aaspi_segy2aaspi.py.** For large data volumes (or for data sitting over a slow NFS connection), this program may take a while to write the results to your local disk.



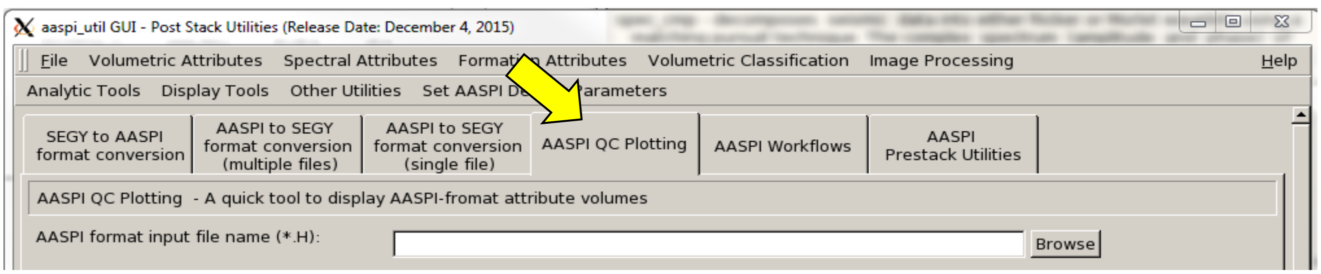
The above image is a screen capture of the bottom of the dialogue that pops up in your original xterm window. Be sure you have your IT folks set up your environment so that you can monitor the progress of your jobs. (Marfurt has visited one location where this is not the case, which makes error tracking somewhat more tedious).

First look for (1) the normal completion of AASPI program **pad3d**…so far so good. Next, quality control (2) the range of lines (3) and the range of cdps. Note that (4) the data had a scalco value of -1, which using the SEGY standard definition, indicates that the data were stored in tenths of a foot. The (5) line increment (*dline*) and cdp increment (*dcdp*) are very close to the nominal 110 ft by 110 ft bin definition, so everything is as it should be. The inline and crossline azimuths (directions of increasing indices) will allow us to define dip azimuth and curvature strike**. If any of these values are incorrect, you will be wasting your time to do anything further since you probably will not be able to load any results into your workstation for interpretation.** The number of samples (6) is 800. Some summary statistics of the range of amplitudes follows. Finally, some simple statistics of the seismic data volume itself (min, max, and RMS amplitudes) will be used later to avoid truncation errors in our add-drop semblance and covariance matrix calculations. If any of these numbers is a NaN (Not a Number), your conversion did not occur properly and you should check the input file formats.

The data conversion is a two-step process. The first step converts the 2D SEG-Y data format to a 2D AASPI data format using program **aaspi_segy_read**. Then program **pad3d** reads in the data and headers of the 2D AASPI data format and determines where the data lie on a grid. After considerable work by Tim Kwiatkowski, we feel we can now read in data that have been stored with increasing or decreasing line and cdp increments, and data that are either padded or unpadded with dead traces to generate a rectangular grid.

Program **pad3d** also searches for dead traces. We have found that many data volumes do not follow the 2002 SEG-Y standard for the trace-id, which should read 1 for a live trace, 2 for a dead trace, and 3 for a padded trace. Many times we will find the number 0. Program **pad3d** scans each trace to determine if it is live, dead, or has been padded by the program, assigning appropriate x, y locations for padded data. The program also scans each trace from the top and bottom and detects the location of the first non-zero value, allowing subsequent programs to preserve mutes. (Certain Canadian data volumes required muting data below a given time level for resource ownership reasons).

Once the data are converted, you should try to plot using the AASPI QC Plotting tab at the top of the **aaspi_util** toolbar:



If you are not yet familiar with the plotting tab, please point your browser to that section of the AASPI manual, called *QC_plotting-AASPI_display_poststack.pdf* .