# TDAVGW Documentation
updated 10/July/2015 by Scott MacInnes

Tdavgw is a time-domain IP data averaging and quality-control utility program that reads TDIP field data from Zonge GDP raw, fld, cac or zdb files. The program includes a Review Data option for interactive quality control while viewing the data in pseudosection or transient curve plots. *Tdavgw* saves averaged data in an "avg" file. Unaveraged data are saved in a "zdb" ("Zonge data base") file, which *tdavgw* can reread to revisit data skipping decisions. {Zonge *zdb* and *avg* files have the same format, but use a different file-name extension to distinguish between unaveraged repeat data and averaged data for which all duplicate readings have been merged.} *Tdavgw* also includes an option to save averaged data in Geosoft IP format dat files. *Tdavgw* reads and writes survey configuration and processing control files from *mde* files, making it easier to duplicate a specific processing sequence when processing multiple GDP TDIP data files.

Start *tdavgw* by running it from the command line, the windows start menu or with a short cut icon. *Tdavgw* first reads default processing control parameters from tdavgw.ini. It will then open a full size window and show a file selection dialog. Select an appropriate *cac, raw*, *fld* or *zdb* file holding TDIP data and click on the **Open** button to start processing. Click on the **Cancel** button to abort *tdavgw*.

Open File Dialog
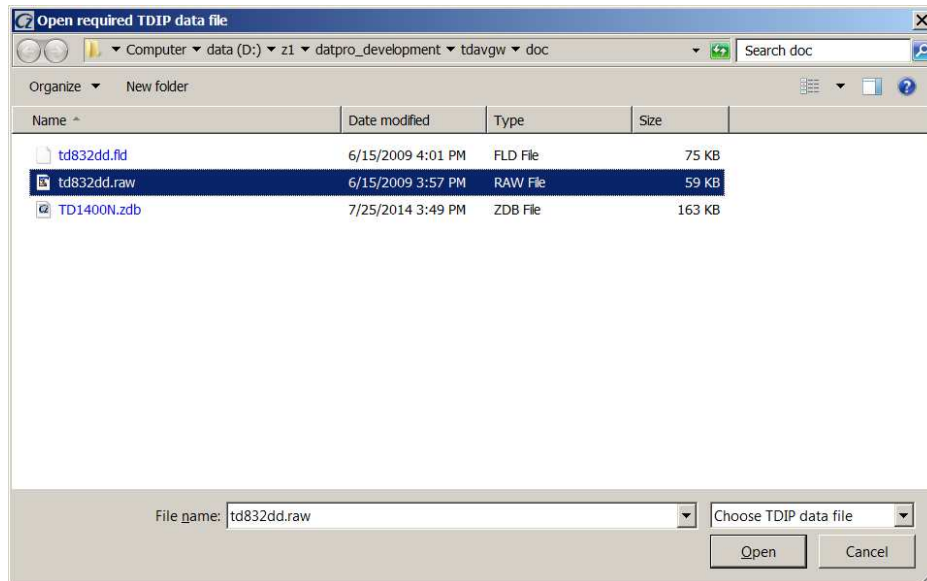
After getting an input data file name, *tdavgw* opens the input data file, counts the data and builds a list of line numbers. It shows a dialog to confirm processing control settings and, if the input data file includes multiple lines, to select the line number to be processed.

Processing Control Dialogs

TDIP *.cac processing control

Select Line to Process: D-D

cac File Data Source: Time-Series

Time-Series Stacking
Stack Type: Straight
Stack Shape: 5.0 (0 to 6)
Max # Cycles/Stack: 8192

Optional auxiliary input-data files
☐ Use *.Mde File — Select *.Mde
☐ Use *.Stn File — Select *.Stn

☐ Save Time-Series in csv => TDIP387TS##.csv files
☐ Save stacks in csv file => TDIP387Stack##.csv files

Calculate Apparent Resistivity with: X,Y,Z coordinates, recommended

Newmont chargeability = IPm
Integrate from 450.0 to 1100.0 msec* $\frac{0.125}{TxFreq}$
IPm Scale: 1.87 = published standard (Swift, 1973)

Averaging Repeats
Avg Type: Straight ☐ Average |Vp| ☐ Apply Auto Skip
☐ Apply Auto Synchronization

Continue   Cancel

TDIP *.raw processing control

Select Line to Process: 1400N D-D

Optional auxiliary input-data files
☐ Use *.Mde File td832dd.mde Select *.Mde
☐ Use *.Stn File Select *.Stn

Calculate Apparent Resistivity with: X,Y,Z coordinates, recommended

GDP Channel Number Meaning
E-field Channel: N-Spacing
H-field Channel: Stn Number

GDP Gain Mode => Anti-Alias Filter Effects
GDP32ii Unknown

Newmont chargeability = IPm
☐ Recalculate IP M from Vs(t)
Integrate from 451.4 to 1097.3 msec* $\frac{0.125}{TxFreq}$
IPm Scale: 1.87 = published standard (Swift, 1973)

Repeat Averaging: Straight ☑ Average |Vp|
☐ Apply Auto Skip

Continue   Cancel

TDIP *.fld processing control

Select Line to Process: 1400N D-D

Optional auxiliary input-data files
☐ Use *.Mde File td832dd.mde Select *.Mde
☐ Use *.Stn File Select *.Stn

Calculate Apparent Resistivity with: X,Y,Z coordinates, recommended

GDP Gain Mode => Anti-Alias Filter Effects
GDP32 Unknown

Newmont chargeability = IPm
☐ Recalculate IP M from Vs(t)
Integrate from 451.4 to 1097.3 msec* $\frac{0.125}{TxFreq}$
IPm Scale: 1.87 = published standard (Swift, 1973)
☑ Average |Vp|
Repeat Averaging: Straight
☐ Apply Auto Skip

Continue   Cancel

TDIP *.zdb processing

Select Line to Process: 1400N D-D

Optional auxiliary input-data files
☐ Use *.Mde File TD1400N.mde Select *.Mde
☐ Use *.Stn File Select *.Stn

Calculate Apparent Resistivity with: X,Y,Z coordinates, recommended

Newmont chargeability = IPm
☐ Recalculate IP M from Vs(t)
Integrate from 451.4 to 1097.3 msec* $\frac{0.125}{TxFreq}$
IPm Scale: 1.87 = published standard (Swift, 1973)

Repeat Averaging: Straight
☑ Average |Vp| ☐ Apply Auto Skip

Continue   Cancel

Tdavgw shows either a cac, raw, fld or zdb processing control dialog, depending upon the input file type. Cac, raw and fld processing control dialogs include a field showing a list of line numbers present in the input data set. The **Select Line to Process** field allows input from raw or cac files holding data for multiple lines. Dipole-dipole, pole-dipole, pole-pole, and Schlumberger data are processed one line at a time, so that the data can be reviewed as a pseudosection. Multiple lines can be processed for Gradient or LabRock input. Gradient data are shown in map view during data review, while LabRock data review is point by point.

Cac file input has the most options. TDIP cac files always include both stack data and the GDP calculated apparent resistivity and chargeability. Use the **cac File Data Source** field to select between recalculating apparent resistivity, Vs(t)/Vp and Newmont chargeability from stack data or using the GDP calculated values as is. If the cac file also includes time-series data, that is shown as a third alternative data source. If you elect to reprocess time-series data, the Stack Type and Stack Shape fields are activated. **Stack Type** can be a straight stack with all Tx cycles weighted equally, a tapered with a Gaussian-like taper with weights near zero at the beginning and end of the time series, or a robust stack which down-weights spikes light spherics from lightning. The **straight stack** is most efficient for environments with unpatterned random noise. **Tapered stacks** are effective against power-line noise, while **robust stacks** are best used to reduce impulse noise like lightening spherics. If present in the cac file, up to 99 time-series data may be exported to csv files, which use a *Ts##.csv name convention, where * is the source cac file name, and ## represents integer indexing that is incremented from 01 to a maximum of 99 for each consecutive time-series record that is exported. GDP-32^{24} TDIP cac file save x,y,z coordinates for electrode locations, where x is distance along line, y is distance across line and z depth below the surface. Apparent resistivity data are usually best calculated using x,y,z coordinates, but if the survey line includes a significant bend (more than 10 degrees), it can be helpful to import east, north coordinates from a stn file and use the **Calculate Apparent Resistivity with** east,north,Z option. Using the east,north coordinates adjusts apparent resistivity values for irregularities in electrode positions.

The processing control dialogs for all four input file formats, include a **Select *.Mde** option to interactively select a *.mde file to import survey configuration and annotation parameters. Tdavgw looks for a source_file_name.mde file by default, and displays it's name in the mde filename file if it's found, but it doesn't read the *.mde file unless the **Use *.Mde File** box is checked. Similarly, tdavgw looks for a default source_file_name.stn file and shows its name in the stn file name field if it's found. But ts2dip doesn't read the stn file unless the **Use *.Stn File** box is checked. **Select *.Stn** file can be selected to import electrode east, north and elevation coordinates. GDP along-line X electrode values correspond to stn file station numbers and GDP across-line Y values correspond to stn file line numbers. The GDP X,Y values are used as keys to interpolate tabulated stn file station, line, east, north, elevation data. For straight lines, it is better to use GDP X,Y,Z electrode coordinates for apparent resistivity calculations, because that option is aligned with the way that the two-dimensional Res/IP inversion program TS2DIP uses finite-element mesh distortion to account for surveying error and topographic effects rather than assuming that apparent resistivity calculations have already accounted for those irregularities.

Processing control for TDIP data input from raw or fld files is similar, but raw file input includes options for specifying how to interpret GDP channel numbers, while fld file input uses the interpretation set by the program shred.exe which creates fld files from raw files. Zdb file processing is even simpler, because fewer data parameters need to be calculated.

For raw files, **GDP Channel Number** options Stn Number, Stn # Offset, N-spacing or Antenna # set the interpretation of the GDP channel number field. N-spacing is the default option for electric-field components, while station-number is the default for magnetic-field components. The station number option uses the GDP channel number as the lower numbered end of a dipole or the station number at a roving pole electrode. For magnetic field components the station number is the coil or magnetic field antenna location. Default option **N-Spacing** uses values from each analog input channel's number as n-spacing value, mirroring GDP-32 array conventions. Option n-spacing offset, multiplies the GDP channel number by the GDP A-Space value to get a station number offset that is added to the GDP Rx value. For Stn # Offset, the GDP channel number is interpreted as an offset which is added to the GDP Rx field value to calculate a Rx loop station number. Option Antenna # is used for magnetic field components and uses the GDP channel number as an antenna ID number, so that antenna calibrate values can be recovered to correct data for the antenna's frequency-domain response. The antenna # option will be implemented in future versions of *tdavgw*, to allow the processing of B-field data acquired with specialized antennas.

GDP Gain Mode fields are used to recover anti-alias filter settings. The GDP model can be a GDP-16, GDP-32, or GDP-32^{II}. The GDP Gain mode can be **Noisy**, **Quiet** or **Unknown**. The default option **Unknown** does not make any time-window adjustment for anti-aliasing filter time delays. Selecting **Noisy** reduces time window values by 112 msec, accounting for the time-delay caused by the GDP's anti-aliasing filter. **Quiet** gain mode reduces time window values by 10.5 msec. The gain mode adjustments are most important when the Vs(t)/Vp transient data are to be later inverted to estimate Cole-Cole model parameters. Most GDP-32 TDIP data are acquired using Noisy gain mode for maximum

noise suppression. Default *tdavgw* processing option **Unknown** is sufficient when the Vs(t)/Vp transients are to be interpreted qualitatively.

Check box **Recalculate IP M from Vs(t)** controls *tdavgw*'s integration of Vs(t)/Vp transients to recalculate Newmont chargeability, M, in msec. If the field is unchecked, GDP raw file values are used as-is. Otherwise, *tdavgw* integrates Vs(t)/Vp transients to recalculate M estimates. If you plan to go through Data Review point by point skipping individual Vs transient outliers to clean up the transient shape, then recalculating Newmont chargeability values is worthwhile. *Tdavgw* uses linear interpolation between time-window centers for it's chargeability integration, so recalculated chargeability values account for integration endpoints that don't land on time-window centers. As Newmont chargeability doesn't have a single, universally accepted definition, *tdavgw* allows some adjustment of its chargeability calculation parameters. The chargeability time window is sometimes delayed to reduce inductive coupling effects. The integration time window start value can be adjusted to any value between 0 and 1354 msec. The integration window length has a fixed length of 646 msec for value so that some standardization is enforced. Newmont chargeability is defined for a transmitter frequency of 0.125 hertz. At higher frequencies, the transmitter current off time is shorter, so the Newmont integration window has to be scaled up and down in inverse proportion to transmitter frequency to keep the integration window within the current off segments of the transmitter waveform. Another variable is the arbitrary multiplier that is used to scale up the integration result to a "Newmont standard". Swift (1973, The L/M parameter of time-domain IP measurements – a computational analysis, Geophysics, v38, p61-67) published a definition of Newmont chargeability that uses a scaling factor of 1.87. Apparently, Newmont Mining uses an unpublished in-house scaling factor of 2.86, which can be selected in the **IPm Scale** field.

Repeat Averaging can be Straight or Robust. Straight averaging gives each repeat value the same weight, an averaging method which works well so long as the background noise level is fairly constant. Robust averaging helps when there are at least three repeat values for each reading (more than three repeats/reading is preferable). Robust averaging down-weights outlier data, which can be useful if there are lightning spherics present during data acquisition and if four or more repeats are acquired for each reading. Checking **Average |Vp|** takes the absolute value of Vp values before averaging, so that any hookup polarity errors are erased.

The **Apply Auto Skip** check box turns the Auto Skip function on and off. It is turned off by default, and can be applied during data input, by checking the Apply Auto Skip check box, or later via an option on the main menu.

After repeat data are read from the input file, *tdavgw* shows a TDIP Survey Configuration dialog to verify survey configuration parameters.

Survey Configuration Dialog

Fields in the upper left section of the dialog store descriptive survey annotation information. **Project** name, **Area**, **for**, **by** and **Job #** fields store arbitrary text strings that are stored in *zdb* and *avg* files for descriptive documentation. **Date** records the data acquisition date. **Length Units** can be m or feet.

IP Array is a list box showing abbreviations for common IP survey configurations.

- **D-D** indicates Dipole-Dipole, an in-line survey with uniform length transmitter and receiver dipoles. Both transmitter and receiver dipoles are aligned along a single survey line.
- **P-D** indicates Pole-Dipole, an in-line survey with one transmitter electrode fixed at a remote location and one roving Tx electrode moving along the survey line. Receiver dipoles are assumed to have a uniform length and be aligned along line.
- **P-P** means an in-line Pole-Pole array with Tx and Rx remote fixed electrodes and in-line roving Tx and Rx electrode poles.
- **Grd** array means a Gradient array with a large fixed transmitter bipole, and in-line arrays of uniform length receiver dipoles moving along lines parallel to the Tx bipole.
- **Sch** represents a Schlumberger resistivity sounding, with short receiver dipoles centered in large transmitter bipoles.
- **D-H**, or Down-Hole, is the most versatile array. The GDP Tx number is used as a key to a list of transmitter wire endpoint coordinates in a txc file (see file formats in Appendix A). Each transmitter can have multiple wire segments both on the surface and down hole. Similarly, receiver station values are used as a key to coordinates in a station file, and can represent either surface or down-hole measurement locations.
- **Lab** represents LabRock or Laboratory Rock sample measurements. Time domain IP properties are measured on hand-specimen sized rock samples. Tx and Rx values are used as sample identification numbers.
- **Gen**, for general array, is for field surveys with arbitrary electrode locations. It doesn't assume that the data are acquired along a line and does not show a pseudosection or map data overview. Quality control is based on point-by-point data review.

Line Name is a string of up to 16 characters that can be used to hold descriptive line labels like 200E or Drill Hole 001.

Line Number is a floating point line number, suitable for use in a Geosoft montaj *gdb* database or as a coordinate value in a curvilinear (line, station) coordinate system. Ideally, line numbers correspond to the across-line Y coordinate electrode locations with the same length scale as the along-line X electrode coordinates.

Line Azimuth is a line bearing indicating the direction of increasing station numbers (electrode X values) in degrees east of north.

Line Type can be Straight, Meander or Points. Most IP survey configurations use in-line arrays, so **Line Type** field is set to "Straight" for all arrays except D-H. The D-H (Down-Hole) array includes general configurations like vector IP, where stations can be point measurements at a set of randomly distributed locations, so all three line types are allowed for D-H arrays. For Straight and Meander lines, *tdravgw* can use along-line interpolation to locate electrodes, but for the Points line type, *tdavgw* uses dipole length and Ex azimuth to position the second receiver electrode.

It is not uncommon to decide that the station number system should be changed after data acquisition is complete. *Tdavgw* includes parameters for scaling and shifting the GDP station numbers used during data acquisition, to "client" station numbers suitable for use on plots and in deliverable data files. By default, there is no rescaling, but **Station Number Rescaling** values can be modified to rescale station numbers. **GDP Station Numbers** indicate the station number system used during data acquisition, while **Rescaled Station Numbers** are the possibly different station numbers needed for final data files and plots. Station number origin and increment values are specified for GDP and rescaled or client station numbers. Station number increments must be non-zero, but negative increment values can be entered to reverse station number order. Station numbers are expected to be proportional to distance along line. The m/stn # field is a scaling factor that scales station numbers to meters.

The **Unit-Dipole Length** field is used to specify the length of a "unit" electric field dipole. The unit dipole length is used to normalize the minimum radial separation between transmitter and receiver electrodes to get a self-consistent set of n-spacing values, even for general array setups than may use variable length dipoles for different GDP input channels or have across-line displacements between electrodes. The display only along-line X **Rx Component Azimuth** mirrors the value entered in the Line Azimuth field. The across-line Y azimuth will be 90 degrees counter-clockwise from X azimuth if you specify a right-handed x,y,z coordinate system with z positive up. If you specify z positive down, then Y azimuth is 90 degrees clockwise from X in map view. By convention Ex and Hx are along-line components, while Ey and Hy are perpendicular to line. Vertical magnetic field measurements indicated by Hz. Raw and fld file component "ON" labels are considered to be the same as "Ex". Down-hole electric field measurements are considered to be "along-line" and should be labeled as ON or Ex.

Tdavgw output files include x,y coordinate fields for all four electrodes, where x is along-line and y is across-line. (Z depth coordinates are include in *tdavgw* output files for down-hole surveys.) By default, pole-dipole and pole-pole remote electrode locations are put at numerical infinity. It is good practice to specify the actual remote electrode locations, which can be done by adding $Tx.XYZ2 and (for pole-pole) $Ch.XYZ2 keyword records to the input raw file (see raw file documentation in the File Format appendix). Remote electrode locations must be specified in the same along-line X, across-line Y and depth Z coordinates use for the survey line transmitter and receiver electrodes. There may be multiple remote locations, which are easily handled by adding multiple $Tx.XYZ2 and $Ch.XYZ2 records to the raw file. If a station file is to be imported to add electrode (east,north,elevation) coordinate information, the remote electrodes can be included by using a multi-line stn file format, with the remote's across-line coordinate as the stn file line variable.

Tdavgw saves averaged data to the Zonge avg file format by default. It can also write averaged data to the Geosoft IP *.dat format. Check boxes can be used to turn *.avg and *.dat format output on or off. A **Save data in legacy *o.avg** checkbox option is included if legacy-format TDIP files are required for backwards compatibility with old software. Checking **Save Unaveraged Data to Geosoft *raw.dat** will export unaveraged data to a Geosoft format *.dat file, an option which can be helpful if you are using Geosoft's IP module for data quality control and plotting.

After survey configuration values have been verified, click on the **Continue** button to go to the next processing step and review the data or select **Cancel** to abort data input.

Review Data Dialog
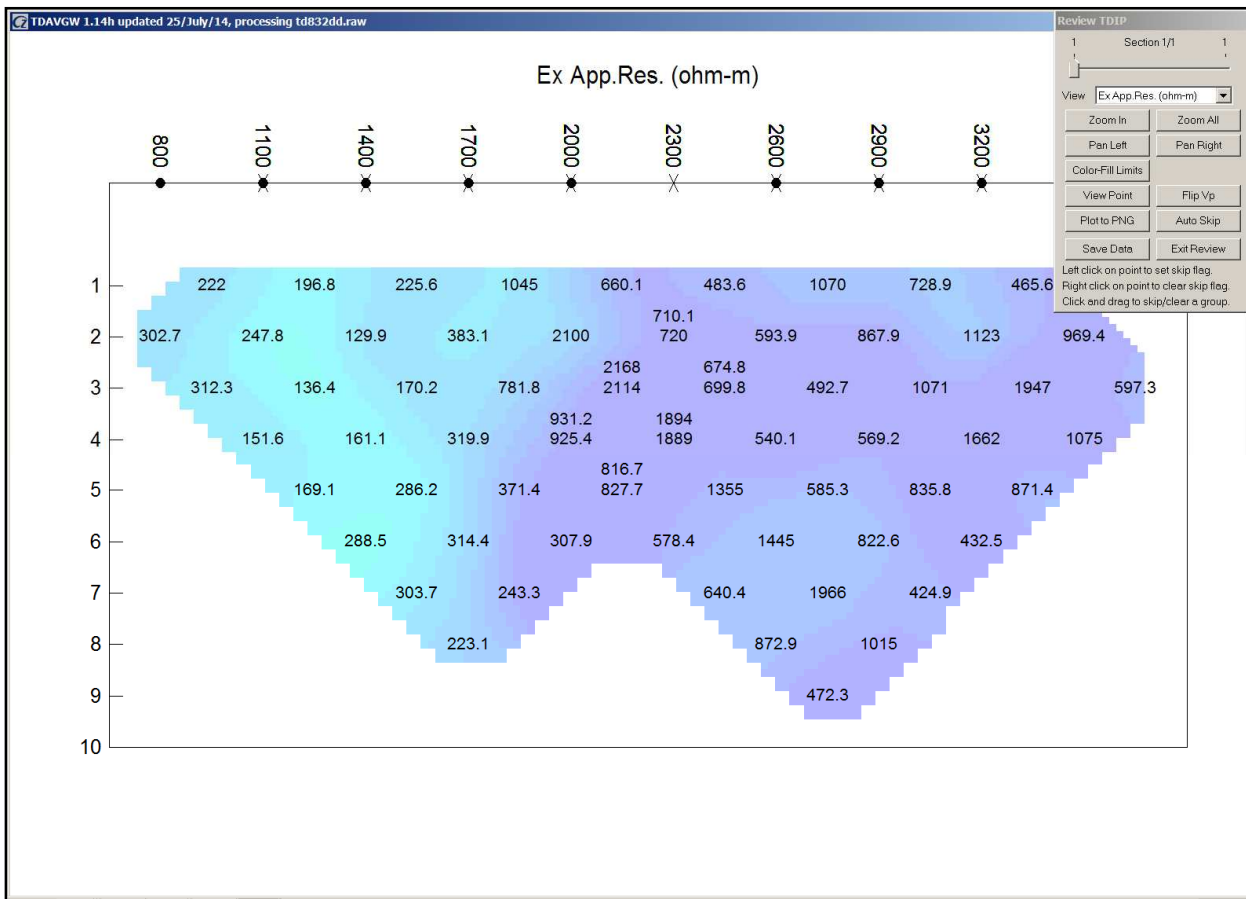
Review Data shows Dipole-Dipole, Pole-Dipole and Pole-Pole resistivity and IP data as a pseudosection or as individual data point transients. Gradient data are shown in map view or as individual points, while Down-Hole and Lab-Rock data are plotted point by point as Vs(t) transients. In the pseudosection and map plots the horizontal axis is scaled to client station numbers and the vertical axis is n-spacing for pseudosections or across-line distance for map views. Posted values indicate individual data values with color-filled contours showing the overall pattern. Black text shows data point values that have not been skipped, while gray pseudosection text indicates skipped data. A pseudosection or map data point can be skipped by pressing the left mouse button while the mouse cursor is over the bad data point. The skip flag can be cleared from a point with a right mouse button click while the mouse cursor is over the grayed-out data value.

A small pop-up dialog provides options for Data Review plot control. Each pseudosection plot shows data for one electromagnetic field component. If the data set includes more than one component, a trackbar along the top of the dialog controls which component is displayed in the pseudosection plot.
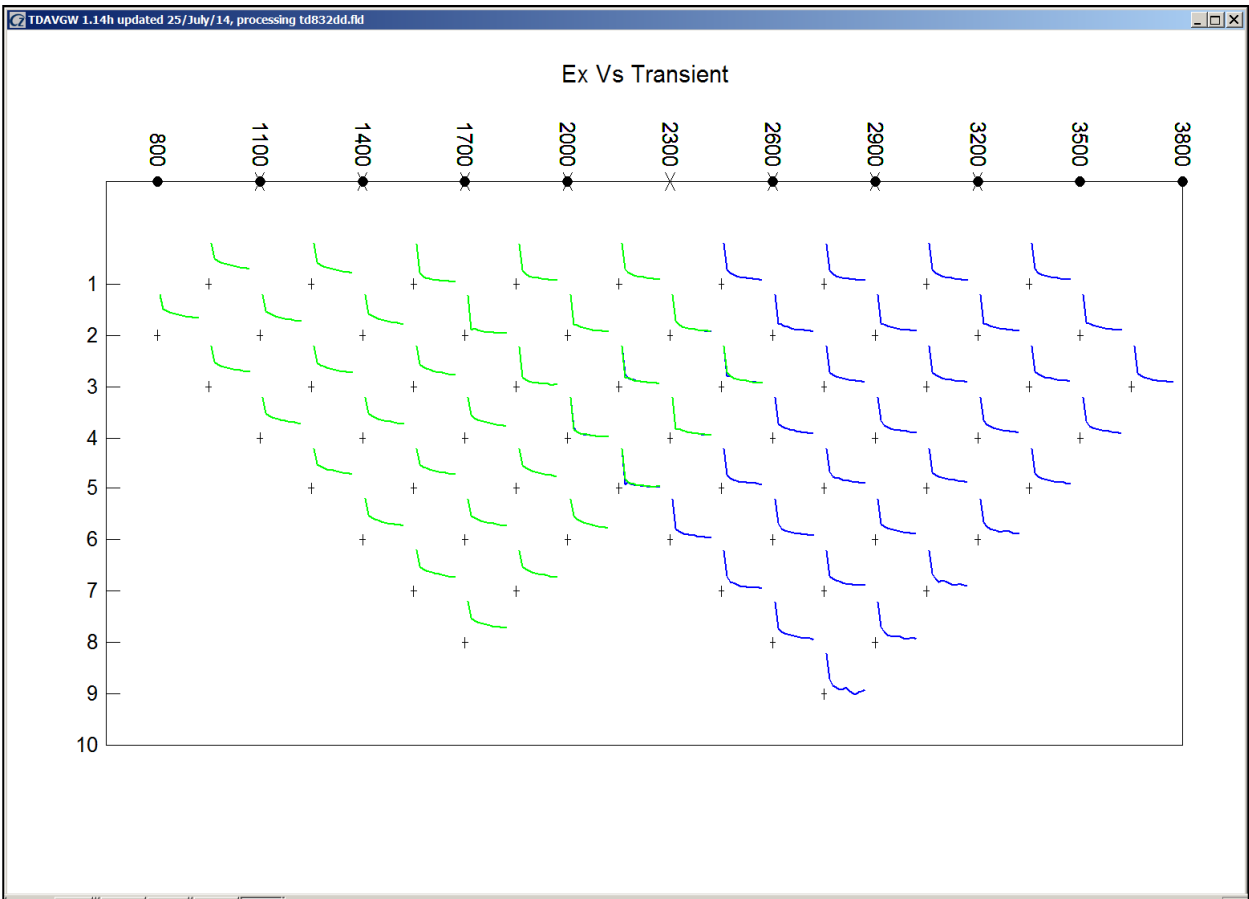
Clicking on the **View** field shows a list of data types available for display, Resistance (mV/A), apparent resistivity (ohm-m), Newmont chargeability (IPm in msec), IPm error (msec) or Vs(t) transients (mV/V). The Vs(t) transient plot (shown above) provides a quick data quality overview. Jagged transient curves indicate noisy data points with green curves for Tx X > Rx X and blue curves for Tx X < Rx X

To see more detail in a long line, **Zoom In** allows magnification of pseudosection details. *Tdavgw* will show a magnifying glass cursor when **Zoom In** has been clicked. Position the curser on one corner of the area of interest and "left click" (depress the left mouse button) to anchor a corner of the **Zoom In** selection rectangle. *Tdavgw* will then show a + shaped cursor and a green rubber-band box indicating the extent of the selection rectangle. Move the mouse cursor to adjust the selection box to include the area of interest and left click again to see a magnified pseudosection plot. The **Pan Left** and **Pan Right** buttons allow lateral movement when the pseudosection plot is magnified. Clicking on the Review Data dialog **Zoom All** button restores the pseudosection plot to a full scale view.

Data point polarity can be changed by clicking on the **Flip Vp** button, which will change the mouse cursor to a cross, +. A left click anchors one corner of a rubber band box, and *tdavgw* will then show a + shaped cursor and rubber band box outlining the extent of the selection area. A second left click anchors the second corner of the selection box, and the polarity of all data within the selection is reversed. Vp polarity flips are only apparent when the pseudosection is showing Resistance values.

The pseudosection plot can be exported to a png (portable network graphics) raster image file via the **Plot to PNG** button. The png plot file will have Ares, IPm, IPerr, Vp or Vs appended to its file name to indicate the pseudosection data type.

The **Color Fill Limits** button can be used to adjust the color fill limits, popping up a dialog with the color fill range for each pseudosection parameter type.
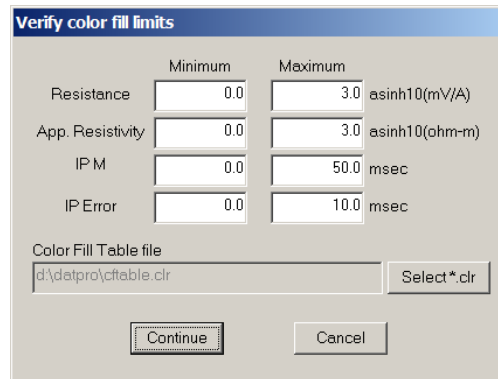
Resistance (mV/A = milliohms) and apparent resistivity are transformed with the asinh10 function for pseudosection color fill. Asinh10(x) looks like x for x = 0 to 1 and looks like log10(x) for x > 1, allowing representative color fill across at large range. The Color Fill Limits dialog also allows selection of an alternative color fill spectrum file (*.clr). Clr files use the Surfer "Color Spectrum File" format, an ASCII format that specifies a table of (percent, red, green, blue) values, where percent can vary from 0 at the color fill minimum to 100 at the color fill maximum, and red, green and blue are integer values between 0 and 255 representing color intensity.

An **Auto Skip** button provides a quality control short cut.
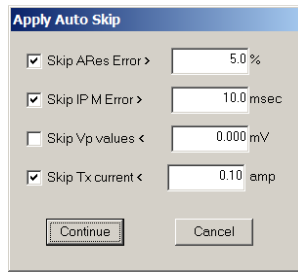
Auto Skip sets averaged data skip flags if selected criteria pass user selected thresholds. Check boxes turn Auto Skip criteria on or off. A check mark indicates that the criteria will be used to set skip flags. **Skip Ares Error >** skips averaged apparent resistivity values whose error exceeds the percent error threshold in the upper right numeric field. **Skip IP M Error >** skips averaged Newmont chargeability values whose error exceeds the specified msec threshold. **Skip Vp values <** is usually used to skip averaged apparent resistivity and IP values when Vp is less than 0, i.e. negative. **Skip Tx current <** can be used to skip averaged apparent resistivity and IP chargeability values when the transmitter current falls below the specified threshold in amps. The Skip Tx current < criteria should not be used on LabRock data, because LabRock currents are usually quite small, less than a milliamp. The Continue button applies the Auto Skip criteria to the averaged data, while Cancel aborts Auto Skip without setting any skip flags. Auto Skip is also available as an option on the main menu (which is described later in this documentation). The main menu also has a Clear Skip Flags option to undo Auto Skip and any manually set averaged data skip flags.

To review data on a point by point basis, click on the **View Transient** button. The mouse cursor changes to a green square. A left click with the square over a pseudosection data point selects the point to show as a transient plot.

TDAVGW 1.14h updated 25/July/14, processing td832dd.raw — Review TDIP Data Point

	along-line X	across-line Y	depth Z	ft
Tx 1	2600.00	1400.00	0.00	
Tx 2	2900.00	1400.00	0.00	
Rx 1	200.00	1400.00	0.00	
Rx 2	500.00	1400.00	0.00	

Gdp Block(s) 110-114 Chn 6 Gdp Time 11:56:20

PltSZXYZ = 1550, -1200, 1550, 1400, 0

Buttons: Skip All, Clear All, Skip IPm, Clear IPm, Zoom In, Zoom Default, Zoom Custom, Set Custom, View Section, Flip Vp, Plot to WMF, Plot all to WMF, Save Data, Exit Review

Tx Length 300.0 ft   Vp/I 2.097 mV/A
Rx Length 300. ft   ARes 303.7 ohm-m
n-space 7.000   IPm 20.52 msec

Left Click on point to set point skip flag. Ctrl+Left Click on point to skip transient.
Right Click on point to clear point skip flag. Ctrl+Right Click on point to clear transient skip.

Skip	ARes	IPm	Vp	Tx Amp
	303.4	20.8	4.1908E-03	2.000
	303.9	21.3	4.1977E-03	2.000
	303.6	18.8	4.1940E-03	2.000
	303.7	20.0	4.1954E-03	2.000
	303.8	21.7	4.1970E-03	2.000

D-D Line 1400 N
Tx 2900,2600, Rx 200,500, N=7
Tx freq 0.125, current 2
GDP blocks 110:114, chn 6, Ex
Vp/I=2.097 mV/A, ARes=303.7 ohm-m, CntRes=3.06 kohm
IPm=20.52 +/-0.461 msec

Ex (mV/V) vs Time (msec)

Transient plots show time-domain IP data one point at a time. The horizontal axis is time in msec while the vertical plot axis is asinh(Vs(t) in mV/V). Transient plots provide a more detailed view of the data and allow quality control skipping or clearing of individual Vs(t) transient data points. Unskipped repeat data points are indicated by small green circles, while skipped data are indicated by a red X. Averaged data points are indicated by a slightly larger black circle. If all of the repeat data for a particular time window are skipped, the corresponding averaged data point is skipped, and is plotted as a gray circle. Data point groups can be skipped by pressing the left mouse button to anchor one corner of a selection box, and then moving the mouse with the button held down to locate the second selection box corner. All data within the selection box is skipped when the left mouse button is released. Similarly, skip flags can be cleared from groups of data points by setting the corners of a selection box with the right mouse button.

A heavy black line segment indicates the extent of the Newmont chargeability integral. For 0.125 hertz data, the default Newmont chargeability integration extends from 451 to 1097 msec. Posted squares centered on the integration interval indicate Newmont chargeability amplitude. Individual repeat values are plotted as green squares and the averaged value is plotted as a black square. An entire transient can be skipped with a Ctrl + left_click while the cursor is over one of the transient's points. Conversely, Ctrl + right_click can be used to clear skip flags on an entire repeat measurement transient.

A plot control dialog for transient curve data review allows movement between stations via a trackbar. If the mouse cursor is in the plot window, the display can be advanced from station to station by pressing the right arrow key, the N (N for Next) key, or by rolling the mouse scroll wheel down. The transient curve plot can be moved to a previous station be pressing the left arrow key, the P (P for Previous) key, or by rolling the mouse scroll button up. The **Skip All** button skips the entire transient, apparent resistivity and Newmont chargeability, while **Clear All** clears skip flags from the entire transient, apparent resistivity and chargeability.

The Review TDIP Data Point control dialog includes electrode coordinate fields. Apparent resistivity values are recalculated if an electrode's coordinates are changed. Grayed out fields below the electrode coordinates report related parameters like dipole length, n-spacing, resistance in mV/A, apparent resistivity and Newmont chargeability.

A grid field on the right side of the Review TDIP Data Point dialog shows a list of individual repeat measurements. The **Skip** checkbox can be used to set or clear the skip flag for each measurement.

Plot to WMF exports the current Vs versus time transient plot to a wmf ("Windows MetaFile) vector graphics file, a file format that can be pasted into MS Word reports. A data point index number is appended to the wmf file name, to distinguish between data points. **Plot all to WMF** exports transient plots for all of the data points to a set of consecutively numbered wmf files.

The **View Section** button returns Data Review to the pseudosection (for D-D, P-D and P-P) or map display (for Grd). **Save Data** writes unaveraged data to a zdb file and saves averaged data values in an avg file. Zdb and avg files use the same ASCII file format, although they hold different data set types. **Exit Review** closes the Review Data option and shows a blank window with a menu strip along the top.

Main Menu Options

After the initial data review is finished *tdavgw* shows a strip of menu choices. **Open** brings up the Open File dialog to select a new input data file name. **Review Data** invokes the quality control pseudosection plot described in the previous section of this documentation. Auto Skip is a pop up menu category that includes the two menu options Apply Auto Skip and Clear Skip Flags. **Apply Auto Skip** menu option brings up the same Auto Skip control dialog as the Pseudosection Data Review, Auto Skip button described earlier in this documentation. **Clear Skip Flags** clears skip flags from the averaged apparent resistivity and Newmont IP chargeability values. It should be used with caution since it can undo skip flags set manually during Pseudosection Data Review.

Remedial Action\Import stn file can be used to import a stn file to add or update east, north, elevation values for each electrode location. If stn file coordinates are revised due to new information, open the zdb file with all of its saved quality control editing decisions and use Remedial Action\Import stn file to import the new geographic coordiantes. Menu option **Save** saves the current data to zdb, avg and optionally, Geosoft IP format dat files, while **Exit** closes *tdavgw*.

Drill-Hole Survey Configuration

The most general IP survey configuration is D-H, or Drill-Hole. To minimize restrictions on D-H array geometry, the GDP-32 does not attempt to calculate apparent resistivity values. The GDP-32 menu 4 Tx value is used as numeric label for a transmitter bipole. *Tdavgw* reads a twc file to recover transmitter wire coordinates by keying on the Tx ID number.

Partial Listing of a Drill-Hole Survey twc file with multiple transmitters:

```
Tx.GdpStn,   X1,      Y1,      Z1,      D1,      X2,      Y2,      Z2,      D2
0,           0,       0,       2140,    0,       1505,    -735,    1894,    0
300,         -62,     1,       1795,    300,     1505,    -735,    1894,    0
650,         1505,    -735,    1894,    0,       1505,    -735,    1894,    0
```
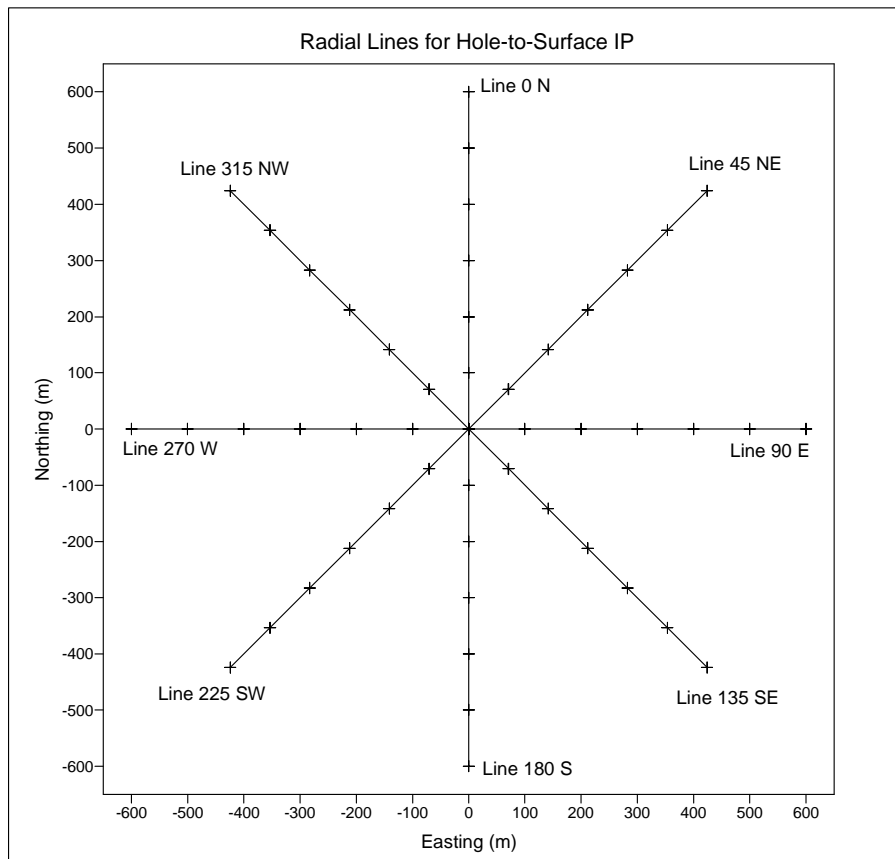
Numbers in the Tx.GdpStn column are used as Tx ID keys which identify each transmitter. There may be more than one file records for each transmitter if the transmitter wire has more than one straight-line segment. The first and last records for each group of transmitter wire coordinates are interpreted as electrode coordinates. Resistivity surveys only require coordinates for the two transmitter electrode locations, but MMR or MIP surveys need to have the actual transmitter wire trace for magnetic field calculations, which usually requires a twc file with multiple segments for each transmitter wire configuration. The example shown above specifies transmitter electrode coordinates for a down-hole array that used three different down-hole electrode positions all connected to a common remote electrode. X values represent either geographic eastings or grid east values. Y values are northings or grid north. Z values are elevations, while D values are depths relative to the surface.

To keep track of receiver electrodes, *tdavgw* uses line and station numbers as keys to east, north, elevation (and optionally depth) in a stn file. Stn file east, north and elevation values must be in the same coordinate system as twc file x, y and z values.

Partial Listing of a Drill-Hole Survey stn file with multiple lines:

```
Station,East,North,Elevation,Depth,  Line
  0,     0.0,    0.0,   2137,    0,     0
100,     0.0,  100.0,   2159,    0,     0
200,     0.0,  200.0,   2184,    0,     0
300,     0.0,  300.0,   2237,    0,     0
400,     0.0,  400.0,   2294,    0,     0
500,     0.0,  500.0,   2284,    0,     0
600,     0.0,  600.0,   2290,    0,     0
  0,     0.0,    0.0,   2137,    0,    45
100,    70.7,   70.7,   2142,    0,    45
200,   141.4,  141.4,   2178,    0,    45
  .       .       .       .       .
  .       .       .       .       .
  .       .       .       .       .
600,  -424.3,  424.3,   2309,    0,    315
```

A station file with multiple lines must include a `Line` column with numeric values corresponding to line numbers. The line column must not include non-numeric text line 315NW, the entries have to be numeric values line 315. Drill-hole surveys may have receiver electrodes down-hole, for which the corresponding `Depth` values should be non-zero. *Tdavgw* treats both positive and negative depth values as distance below the surface. Stations can be on the surface or below ground, but not suspended in the air above ground. Since (east,north,elevation) form a right-handed coordinate system with z=elevation positive up, *tdavgw* internally sets depths to negative values for subsurface station locations so that (east,north,depth) is also a right-handed coordinate system.

Radial Lines for Hole-to-Surface IP

Line 0 N
Line 315 NW
Line 45 NE
Line 270 W
Line 90 E
Line 225 SW
Line 135 SE
Line 180 S

Northing (m)
Easting (m)

A typical down-hole survey might have an array of receiver lines on the surface near the drill hole collar, with transmitter electrodes placed down hole. In the example shown above, a radial array of eight receiver lines is set up around a drill hole collar at (0,0,0). The station numbers indicate radial distance from the drill hole collar and line numbers indicate line azimuth. By default, *Tdavgw* calculates D-H receiver electrode station numbers with the same conventions used for GDP-32 Grd (gradient) arrays. Both D-H and Grd arrays must use station numbers that represent distance along line in the same units used to specify the A-SP field receiver dipole length. The GDP-32 channel numbers are used as dipole-length n-spacing offsets from the GDP-32 menu 4 Rx field value. If the n-spacing is greater than 0, then the first receiver electrode is at Rx + (n_space – 1)*a_space and the second receiver electrode for the same receiver dipole is at Rx + n_space*a_space. If the GDP input channel n-spacing values is less than or equal to 0, then the first receiver electrode is at Rx + n_space*a_space and the second receiver electrode is at Rx + n_space*(a_space+1). If you don't want to confuse yourself, use only positive integer n-spacings when acquiring Grd or D-H data. Other GDP channel number conventions can be used, entering the stn number or stn number offset in the GDP channel number fields, but be sure to specify the correct Channel Number Type in the processing control dialog.

Once receiver electrode station numbers have been calculated, *tdavgw* can use (line,station) coordinates to interpolate along-line to electrode positions in (east,north,depth) coordinates based on stn file values. Line may be either on the surface or down hole. Apparent resistivities are calculated based on the potentials expected in a uniform resistivity half-space with a flat surface. For D-H surveys *tdavgw* includes (x,y,z) columns for each transmitter and receiver electrode, keeping track of electrode positions that may be below the ground's surface.

Partial Listing of a Drill-Hole Survey avg file showing (x,y,z) coordinates for each Tx & Rx electrode:

```
Tx.GdpStn,Tx.X1,Tx.X2, Rx.X1,Rx.X2,  Tx.Y1,Tx.Y2,  Rx.Y1,Rx.Y2,  Tx.Z1,Tx.Z2,  Rx.Z1,Rx.Z2,...
0,        0,    1505,  70.7, 141.4,  0,    -735,    70.7, 141.4,  0,    0,      0,    0,     ...
300,      -62,  1505,  70.7, 141.4,  1,    -735,    70.7, 141.4,  -300, 0,      0,    0,     ...
650,      -112, 1505,  70.7, 141.4,  8,    -735,    70.7, 141.4,  -650, 0,      0,    0,     ...
1050,     -178, 1505,  70.7, 141.4,  21,   -735,    70.7, 141.4,  -1050,0,      0,    0,     ...
0,        0,    1505,  141.4,212.1,  0,    -735,    141.4,212.1,  0,    0,      0,    0,     ...
```

Lab Rocks Data Processing

The most unique IP survey configuration is Lab or Laboratory Rock sample measurements. LabRock surveys do not need to keep track of electrode locations, so GDP numeric sample numbers are used to identify rock samples. *Tdavgw* adds a Sample.Name column to Lab Rock zdb and avg files. By default Sample.Name is set to a text representations of the GDP Sample.Number value. *Tdavgw* can import alternative Sample.Name text (and optional sample east, north, elevation and depth) from a *.lab file. The processing control dialog includes fields for selecting a *.lab file when Lab Rock data are processed. **Tdavgw Data Review** Lab Rock plots show Sample.Number and Sample.Name as annotation, so rock sample data can be identified by both a sample number and a text name.

Partial Listing of a Lab Rock lab file:

```
GPS.Datum=WGS84
$GPS.UTMZone=16
Sample.Number, Sample.Name,  East,   North,   ELevation, Depth
51205,          "Shale 5a", 675442, 4346062, 2278,       0
51207,          "Chert 1",  675474, 4346155, 2273,       0
51208,          "SS 3NWa",  675506, 4346250, 2263,       0
51209,          "DH S4",    675537, 4346345, 2243,     500
```

Sample.Number values are used as a key to link *.lab file entries to the GDP LabRock IP data. Sample.Name values are arbitrary sample text labels up to 32 characters long.

Optional East, North, Elevation and Depth columns can be included to specify the sample's source location. If geographic coordinates are included $GPS.Datum and $GPS.UTMZone keyword records may be used to specify the datum and zone of UTM east, north, elevation coordinates. GPS.UTMZone values are positive for the northern hemisphere and negative for zones south of the equator.

Partial Listing of a Lab Rock zdb file after updating from a lab file

ample. Number is a numeric value used as a key to link GDP data and *.lab file input. text to represent a rock sample label. Sample.Name is an arbitrary text sample label. If sample location coordinate east, north, elevation, depth columns are in the *.lab file, their values are appended to the zdb and avg files in columns Plt.East, Plt.North, Plt.Elev and Plt.Depth

```
Sample.Number,Sample.Name, Sample.Length,Sample.Area,Plt.East, Plt.North,Plt.Elev, Plt.Depth
51205,         "Shale 5a", 1.7,           6.5,        675442,   4346062,  2278,     0
51205,         "Shale 5a", 1.7,           6.5,        675442,   4346062,  2278,     0
51207,         "Chert 1",  1.5,           17.3,       675474,   4346155,  2273,     0
51207,         "Chert 1",  1.5,           17.3,       675474,   4346155,  2273,     0
51208,         "SS 3NWa",  1.5,           8,          675506,   4346250,  2263,     0
51208,         "SS 3NWa",  1.5,           8,          675506,   4346250,  2263,     0
51209,         "DH S4",    1.5,           8.5,        675537,   4346345,  2243,     500
51209,         "DH S4",    1.5,           8.5,        675537,   4346345,  2243,     500
```
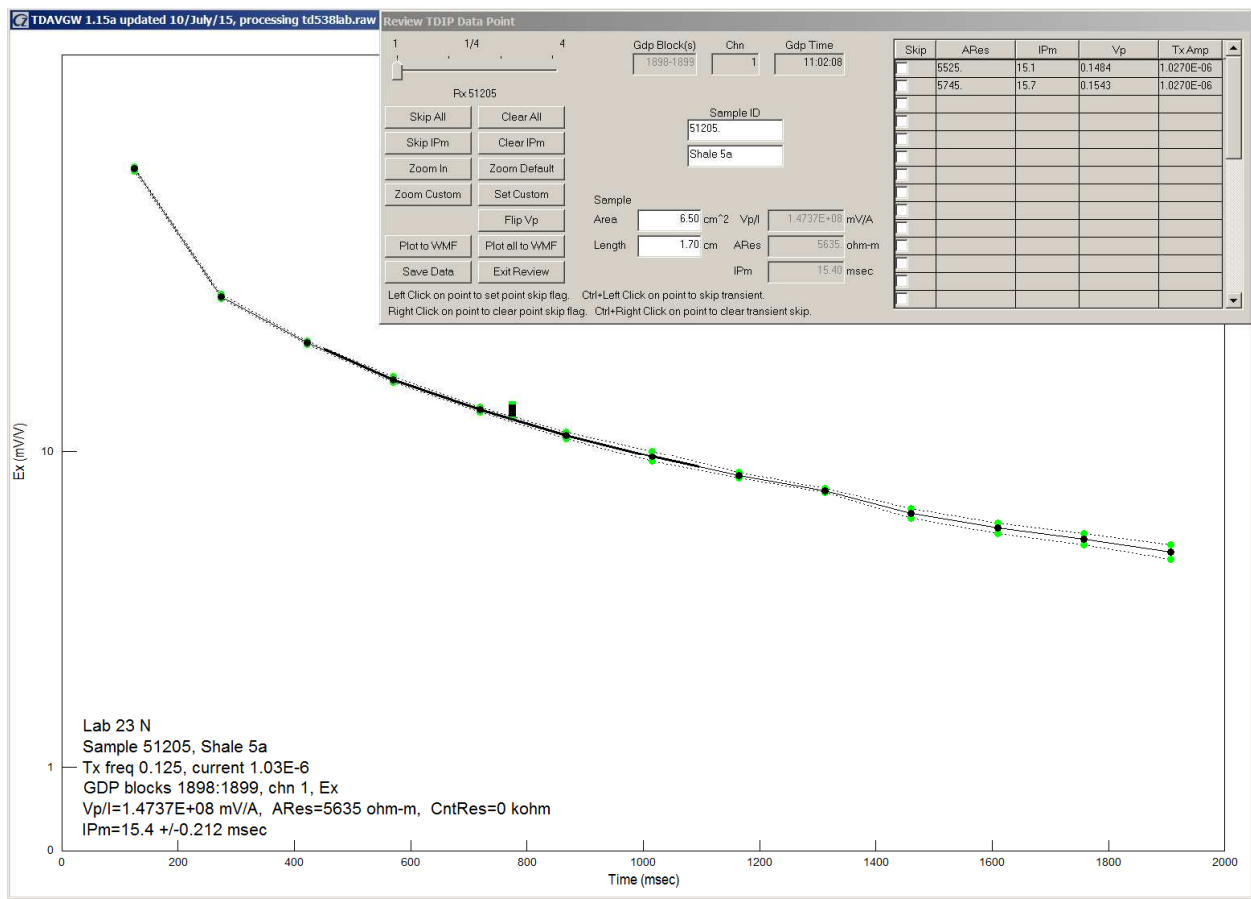

When sample labels are imported from a lab file, the Data Review plots show the Sample.Number and *.lab file Sample.Name text in the upper-left corner of the plot. For the first data point, the plot annotation identifies the sample as "Sample 51205, Shale 5a".

File Formats

tdavgw.ini: survey configuration and processing control default values

Tdavgw.ini is used to set default survey configuration and processing control values. It is an ASCII file that can be edited with a generic text editor. Each line is a keyword record with a `$program:keyword=value(s)` or `$keyword=value(s)` format. There may multiple comma-separated values after some keywords. For example `TBin.Begin` and `TBin.End` are followed by long list of comma-separated values. Comment lines beginning with the character ", !, / or \ can be anywhere within an ini file.   Blank lines and comment lines are skipped while *tdavgw* is reading *ini* files.

Tdavgw.ini is primarily used to store *tdavgw* processing control parameters, but generic survey configuration keyword records may be pasted into the ini file to control *tdavgw* default values. *Tdavgw* updates the $TDAVGW: records in *tdavgw.ini* with its current processing control values. Any additional keyword records in *tdavgw.ini* are saved as is.

Listing of sample *tdavgw.ini*

```
--------------------------------------------------------------------------
$TDAVGW:Version=1.10a applied 2009/06/16
$TDAVGW:Process.BINNED=Yes
$TDAVGW:Stack.Type=Straight
$TDAVGW:Stack.Taper=5
$TDAVGW:Stack.Trim=20
$TDAVGW:Stack.SNFOn=No
$TDAVGW:Stack.SNFreq=60
$TDAVGW:Avg.AbsVp=Yes
$TDAVGW:Avg.Type=Straight
$TDAVGW:IPM.Recalculate=No
$TDAVGW:IPM.Scale=1.87,"published Newmont M scale (Swift,1973)"
$TDAVGW:IPM.TwBeg=451.4 msec
$TDAVGW:IPM.TwEnd=1097.3 msec
$TDAVGW:Ch.NumberType=N-Spacing,Stn Number
$TDAVGW:GDP.Type=GDP32
$TDAVGW:GDP.GainMode=Unknown
$TBin.Name=GDP32ii TDIP
$TBin.NWaveform=1024
$TBin.Beg=58,8,27,46,65,84,103,122,141,160,179,198,217,236
$TBin.End=140,26,45,64,83,102,121,140,159,178,197,216,235,254
--------------------------------------------------------------------------
```

Version control
TDAVGW:Version   = string, tdavgw output only, saves *tdavgw* version number

Averaging
TDAVGW:Avg.AbsVp   = boolean, yes=average Vp absolute values, dropping any reverse polarity
TDAVGW:Avg.Type   = enumeration, {Straight,Robust}

Newmont Chargeability
TDAVGW:IPM.Recalculate   = boolean, yes=recalculate Newmont chargeability, M
TDAVGW:IPM.Scale   = enumeration, Newmont M "standard" scaling {1.87,2.86}
TDAVGW:IPM.TwBeg   = float, beginning of Newmont M integration (msec*Tx.Freq/0.125)
TDAVGW:IPM.TwEnd   = float, end of Newmont M integration (msec*Tx.Freq/0.125)

GDP
TDAVGW:Ch.NumberType   = enumeration {Stn # Offset,N-Spacing,N-Space Offset,Antenna #}
TDAVGW:GDP.Type=GDP32   = enumeration {GDP16,GDP32,GDP32ii,GDP3224}
TDAVGW:GDP.GainMode   = enumeration {Noisy,Quiet,Unknown}

Temagvw.ini (continued)

Auto skip control
```
$TDAVGW:Skip.Auto        = boolean, yes=apply Auto Skip on data input
$TDAVGW:Skip.OnARerr     = boolean, yes=skip if apparent resistivity error > ceiling
$TDAVGW:Skip.OnIPerr     = boolean, yes= skip if IP error > ceiling
$TDAVGW:Skip.OnVpMag     = boolean, yes=skip if Vp < floor
$TDAVGW:Skip.OnTxAmp     = boolean, yes=skip if Tx current < floor
$TDAVGW:Skip.ARerrCeiling = float, apparent resistivity error ceiling (%)
$TDAVGW:Skip.IPerrCeiling = float, IP error ceiling (msec)
$TDAVGW:Skip.VpMagFloor   = float, Vp floor (mV)
$TDAVGW:Skip.TxAmpFloor   = float, Tx current floor (amps)
```

Color fill limits
```
$TDAVGW:CF.RPdat = floats, resistance color fill minimum and maximum in asinh10(mV/A)
$TDAVGW:CF.ARdat = floats, app. resistivity color fill min and max in asinh10(ohm-m)
$TDAVGW:CF.IPdat = floats, IP chargeability color fill minimum and maximum in msec
$TDAVGW:CF.IPerr = floats, IP chargeability error color fill minimum and maximum in msec
```

24-bit cac file processing control (for future use)
```
TDAVGW:Process.TS      = boolean, yes=process cac file time series data
TDAVGW:Process.STACK   = boolean, yes=process cac file stack data
TDAVGW:Process.DECAY   = boolean, yes=process cac file decay data
TDAVGW:Process.BINNED  = boolean, yes=process cac file binned transient data
TDAVGW:Stack.Type      = enumeration, {Straight,Taper,Robust}
TDAVGW:Stack.Shape     = float, Kaiser window taper (0=rectangular to 5=Gaussian)
TDAVGW:Stack.SNFOn     = boolean, yes=apply synchronous noise filter to ts, stack or decay data
TDAVGW:Stack.SNFreq    = float, synchronous noise frequency (hertz)
TBin.Name              = string, label used to identify each time-window bin set (up to 32 char)
TBin.NWaveform         = integer, # of A/D samples/transmitter waveform
TBin.Beg               = integer list, 0-based offsets to beginning of time-window bins
TBin.End               = integer list, 0-based offsets to end of time-window bins
```

Color-fill table file (cftable.clr)

CfTable.clr is an ASCII file specifying the color spectrum to use for color fills. The installation version has a blue-to-red rainbow spectrum, but cftable.clr can be customized to suit individual tastes. CfTable.clr uses the Golden Software Surfer format, so sample Surfer clr files with different color spectrums can be copied from the ..\Surfer\Samples subdirectory.

Listing of **cftable.clr**:

```
ColorMap 1 1
        0      178      178      255
       25      153      255      255
       50      128      255      128
       75      255      255        0
      100      255        0        0
```

cftable.clr file variables:

Tdavgw ignores the header line "ColorMap 1 1", but it should be included for compatibility with Surfer.

The remaining numerical records specify range, red, green and blue values for a variable number of anchor points, where
Range   = percent between minimum and maximum contoured data (0 to 100),
Red   = amount of red color from 0=none to 255=maximum,
Green   = amount of green color from 0=none to 255=maximum,
Blue   = amount of blue color from 0=none to 255=maximum.

Colors are interpolated between anchor points to get a color-fill spectrum with continuous gradation. Range values should be ordered from low to high with no duplicate values, and should include records for the minimum and maximum range values, 0 and 100.

MDE file - line annotation, survey configuration and data processing control

Tdavgw looks for an optional *.mde file with the same file-name stem as the input data file to get project and line specific default values. MDE files have an ASCII format that is easy to edit and are usually used to set initial survey annotation, survey configuration, and processing control parameters for each survey line. A MDE file consists of one or more "mode" or keyword lines, each of which begins with a "$" in the first column, optionally followed by a program name and colon ":". The keyword is followed by an equal sign "=", then the value to assign to the associated variable. Spaces may be included between the elements of the mode line. Spaces in values defined as text will be included as part of the value. Some keywords, like Rx.HPR, take multiple comma-separated values. Annotation text strings should be enclosed in quotes, so that any enclosed commas are not parsed as value field separators. Lines that begin with a leading !, \, / or " character are interpreted as comment lines and skipped over during mde file input. If a keyword line includes a program name, only the named program will read the record. Including a specific program name identifies which program is to use the associated parameter value. The same $program:keyword=value(s) and $keyword=values(s) format is used in mde, ini, zdb and avg files. Programs ignore unknown keywords during input, so new keywords may be introduced without breaking old software.

Sample MDE file listing
```
--------------------------------------------------------------------------
$Job.Name = "TDIP D-D Data"
$Job.Area = "Zonge Datpro"
$Job.For = "Zonge AZ"
$Job.By = "Zonge AK"
$Job.Number = "110a"
$Job.Date = 09-06-15
$Survey.Type = TDIP
$Survey.Array = D-D
$Line.Azimuth = 90
$Stn.GdpBeg = 0
$Stn.GdpInc = 300
$Stn.GdpUnitScale = 0.3048 m/GdpStn#
$Stn.Beg = 0
$Stn.Inc = 300
$Stn.UnitScale = 0.3048 m/Stn#
$Rx.Length = 300,300 ft
$Rx.HPR = 90,0,0
$Unit.Length = ft
--------------------------------------------------------------------------
```

Keywords `Stn.GdpBeg`, `Stn.GdpInc`, `Stn.Beg` and `Stn.Inc` can be used to shift and scale the GDP station numbers used in *.raw or *.fld files to a different set of "client" station numbers suitable for report plots. Note that stn files use client station numbers, since station location information is often provided by the client. Client station numbers are calculated from GDP station numbers using Stn = (Stn.Gdp – Stn.GdbBeg)*Stn.Inc/Stn.GdpInc + Stn.Beg. Default values produce no station number shifting or scaling.

Stn.GdpBeg	= first GDP station number origin (legacy alias `StnLow`)
Stn.GdpInc	= GDP station number increment (legacy alias `StnDelt`)
Stn.Beg	= client station number origin (legacy alias `LblFrst`)
Stn.Inc	= client station number increment (legacy alias `LblDelt`)
Stn.UnitScale	= client station number to meter scale (meters/client_stn_number)
Stn.Left	= client station number on left side of pseudosection & section plots
Stn.Right	= client station number on right side of section plots
	`Stn.Left` may be greater than `Stn.Right` to reverse plot axis station number order

Survey Annotation

```
Job.Name        = string, project name (< 128 char) (legacy alias Project)
Job.Area        = string, project area (< 128 char)
Job.For         = string, client name  (< 128 char) (legacy alias Client)
Job.By          = string, contractor name (< 128 char) (legacy alias Company)
Job.Number      = string, identifying job label (< 16 char) (legacy alias JobNum, JobNumber)
Job.Date        = string, data acquisition date (< 16 char) (legacy alias JobDate)
Line.Name       = string, arbitrary line "number" (<16 char) (legacy alias Line)
Line.Number     = float, line number, used for coordinate interpolation
Line.Azimuth    = float, azimuth (deg E of N of increasing stn numbers) or string = NnnE bearing format
```

Survey Configuration

```
Survey.Type     = enumeration, always TDIP
Survey.Array    = enumeration, IP array type {D-D,P-D,P-P,Grd,Sch,D-H,Lab}
Unit.Length     = enumeration, length units {m,ft} (legacy alias Units)
Tx.Length       = float, Tx dipole length (length units) (legacy alias TxLength)
Rx.Length       = floats, X and Y Rx dipole lengths (length units)
Rx.HPR          = floats, Rx component orientation (heading,pitch,roll deg)
                  Roll = 0 => z positive up, roll = 180 => z positive down
```

STN file - station location and elevation

STN files hold information about station locations in a tabular format with space or comma separated values. A STN file should have at least two entries, corresponding to the first and last stations. Additional entries may be necessary to trace out topographic changes or curved lines. *Tdavgw* assumes that station numbers are related to distance along line. Station number values are used to linearly interpolate (easting,northing,elevation) coordinates for stations that do not have a matching entry in the STN file. If station numbers are scaled by entries in the MDE file, STN-file station numbers should be in the scaled and shifted client station numbers defined by Stn.Beg and Stn.Inc, not the unscaled and unshifted GDP station numbers defined by Stn.GdpBeg, Stn.GdpInc.

Records beginning with a !, ", \ or / character are classified as comment lines and are skipped.
Blank records are skipped.

Some grid coordinate systems, such as UTM coordinates, can generate very large coordinate values. Many programs store station locations as floating-point numbers with seven significant figures. It may be necessary to subtract a constant from large coordinate values in order to allow accurate representation with seven significant figures.

Station files may include optional Heading, Pitch and or Roll columns. A Heading column is often added if surveys include spot soundings where the EM component orientation varies from one station to the next, i.e. reconnaissance or tensor IP surveys. There may also be an optional Depth column, to locate down-hole survey receiver stations with respect to the surface of the ground. A station file may hold data for multiple lines if a Line column is present.

Listing of typical stn file:
```
-------------------------------------------------------------------------------
Station,    East,    North, Elevation,  Line
   84.0,    5595,    5887,     2030,    1400
   86.0,    5533,    6079,     2035,    1400
   88.0,    5471,    6271,     2040,    1400
   90.0,    5410,    6463,     2040,    1400
   92.0,    5348,    6654,     2045,    1400
     .       .        .         .        .
     .       .        .         .        .
     .       .        .         .        .
  120.0,    4487,    9340,     2120,    1400
  122.0,    4425,    9531,     2120,    1400
-------------------------------------------------------------------------------
```

STN file column definitions:
```
Station     = client station numbers
East        = grid east (m or ft, length units specified when STEMINV creates a new std file)
North       = grid north (m or ft)
Elevation   = elevation (m or ft)
Heading     = X component azimuth (deg E of N), optional column
Pitch       = X component pitch (deg up from horizontal), optional column
Roll        = Z component rotation about X axis (deg clockwise), optional column
Depth       = depth with respect to surface for D-H array (m or ft), optional column for D-H surveys
Line        = line number, optional column for stn files with multiple lines
```

TWC file – transmitter wire segment endpoints

TWC (Transmitter Wire Coordinate) files hold transmitter wire segment endpoint coordinates in a tabular format with space- or comma-separated values. TWC files are used for down-hole surveys, where a transmitter wire may have one end at a remote surface location and the other end down hole. The Tx.GdpStn column value is used as an identifier to match TWC file records to time-domain IP data points. A TWC file should have at least one entry for each transmitter, specifying the two transmitter wire endpoints. Additional records may be present to locate corners in the Tx wire layout for MMR and MIP magnetic field calculation.

TWC files must include nine columns, Tx.GdpStn, X1, Y1, Z1, X2, Y2, Z2 and D2. X is easting or grid east. Y is northing or grid north. Z is elevation or relative level, z is positive upward. D is depth relative to the surface, and is positive downward. TWC and STN files must use the same coordinate system. Most of the time, (x,y,z) = (east,north,elevation), but when D-H surveys are used to describe a complex survey with receiver dipoles in parallel lines, it may be more convenient to use a coordinate system with grid east aligned along line.

Listing of typical down-hole TWC file:

```
-------------------------------------------------------------------------
Tx.GdpStn,    X1,       Y1,    Z1,    D1,      X2,       Y2,    Z2,    D2
0,        403214, 4483381, 2140,    0, 404719, 4482646, 1894,    0
300,      403152, 4483382, 1795,  345, 404719, 4482646, 1894,    0
650,      403102, 4483389, 1502,  638, 404719, 4482646, 1894,    0
1050,     403036, 4483402, 1105, 1035, 404719, 4482646, 1894,    0
525,      403214, 4483381, 2140,    0, 403036, 4483402, 1105, 1035
-------------------------------------------------------------------------
```

TWC file column definitions:

Tx.GdpStn	= Tx identification number (numeric value)
X1	= positive east (m or ft, length units must be consistent throughout the data set)
Y1	= positive grid north (m or ft)
Z1	= positive elevation or relative level, z+ up (m or ft)
D1	= positive depth with respect to surface, depth is positive down
X2	= negative east (m or ft, length units must be consistent throughout the data set)
Y2	= negative grid north (m or ft)
Z2	= negative elevation or relative level, z+ up (m or ft)
D2	= negative depth with respect to surface, depth is positive down

LAB file – Lab Rock sample labels

Lab files hold Lab Rock sample label text in a tabular format with comma-separated values. LAB files use Sample.Number values as keys to Sample.Name and optional sample-location coordinate information. Lab files have several record types:

Records beginning with a !, ", \ or / character are classified as comment lines and are skipped.

Blank records are skipped.

Records beginning with a $ should have a keyword=value format. Keyword records may occur anywhere within a lab file. To record UTM coordinate datum and zone, *cravgw* recognizes the keywords $Gps.Datum and $Gps.UTMZone. Unrecognized keywords are ignored.

As the data columns may be in any order and some columns are optional and may be missing, the comma-separated numeric and text data block must be preceded by a column label line. Unrecognized column labels and the associated column data are ignored. Columns Sample.Number and Sample.Name are required. The geographic coordinate columns East, North, Elevation and Depth are optional (but all four should be included if geographic sample location coordinates are to be imported into *cravgw*).

Listing of typical Lab Rock LAB file:

```
--------------------------------------------------------------------------
Sample.Number, Sample.Name,  East,   North,  ELevation, Depth
51205,          "Shale 5a", 675442, 4346062, 2278,        0
51207,          "Chert 1",  675474, 4346155, 2273,        0
51208,          "SS 3NWa",  675506, 4346250, 2263,        0
51209,          "DH S4",    675537, 4346345, 2243,      500
$GPS.Datum=WGS84
$GPS.UTMZone=16
--------------------------------------------------------------------------
```

LAB file column definitions:

```
Sample.Number =  sample number to link with GDP LabRock measurements (numeric value)
Sample.Name   =  rock sample text label (up to 32 characters)
```
  Optional geographic coordinate columns
```
East            =sample location easting (numeric value, often UTM meters)
North           =sample location northing (numeric value)
Elevation       =sample location elevation (numeric value)
Depth           =sample location depth (numeric value)
```

LAB file $keyword=value records
```
$GPS.Datum    = geographic coordinate datum (string)
$GPS.UTMZone  = UTM coordinate zone, negative for southern hemisphere (integer)
```

TDIP *cac* file

The GDP-32[24] saves TDIP data in a binary cache file with a block-record format. Each record has a consistent external structure, so that programs can search through *cac* files and skip records that are not of immediate interest or are of an unknown type. The format anticipates the addition of new record types in the future, and new record types using the consistent external record structure will not break existing software.

Each *cac* record (of any type) in the binary cache is stored in the following manner.

```
 ___ ___ __ _____ ___  ___  ___ __ _____ ___
|___|___|__|___record content___|___||___||___|__|___next record__|___|
  \   \   \                  /    \      \  Record length of next record
   \   \   \Record Type     /      \ Record length of next record
    \   \                  /
     \   \Record Flag     /
      \                  /
       \-- Record Length -----/
```

The first and last element in every stored record is the length of the core data record in bytes (stored as a 4 byte integer). The second element in every record is a 4 byte code with all four bytes equal to FFFF hex. This in conjunction with the bracketing record length field allows processing of a damaged cache. Software can skip through the file without reading entire records. Record length values are used to adjust the file position pointer directly, without the overhead of reading the record's data content.

The first 2 byte element in every core data record tells which type of data record has been stored. Currently records can be either in an ASCII metadata format or binary records holding rectangular tables of numeric data (see below).

Cac file Navigate records (record type = 4)

```
------------------------------------------------------------------------
used for direct file access
binary record pointer and record size
------------------------------------------------------------------------
```

Cac file Survey Metadata records (record type = 514)

```
------------------------------------------------------------------------
```
Survey configuration data stored as ASCII text in the form of comma-separated-value sub-records with a `keyword,value1,value2,...,valueN` format. Sub-record endpoints are delineated by a carriage-return character (`\n`). Parameters are organized into groups, with most distinguished by a "GroupName." keyword prefix. Keywords are case insensitive and may include the letters `a` to `z`, numbers `0` to `9` and the underscore character. Keywords must be spelled exactly, no matching is done on substrings or aliases. Keyword records may have multiple comma- or colon-separated values.

Programs should skip sub-records with novel or unknown keywords, allowing future extension without breaking existing software.
```
------------------------------------------------------------------------
HEADER.TYPE         = enumeration, Survey

DATA.VERSION        = cac file format version number, string
DATA.BLOCK          = data  block number, integer
DATA.SKIP           = skip flag, 0=use data, 1= skip data
DATA.STACKCTDN      = count down of Tx cycles, is > 1 if GDP operator presses stop average, integer
```

(*cac* file Survey Metadata records, continued)

```
GDP.DATE           = string, data acquisition date (mm/dd/yyyy)
GDP.TIME           = string, data acquisition time of day (hh:mm:ss.sss)
SURVEY.TYPE        = TDIP, string
SURVEY.ACQMETHOD   = enumeration, {TimeSeries,Stack}
    TimeSeries     = full time series data saved in addition to stack and summary data
    Stack          = rectified stack and summary data saved
SURVEY.ARRAY       = enumeration, array type, CR: {D-D,P-D,P-P,Grd,Sch,D-H,Lab}
LINE.NAME          = string, line label
LINE.NUMBER        = float, line number, ideally an across-line coordinate
LINE.DIRECTION     = enumeration, {N,NE,E,SE,S,SW,W,NW,DH }
LINE.SPREAD        = string, spread label, not always interpretable as a number
JOB.NAME           = string, project name
JOB.FOR            = string, client name
JOB.BY             = string, contractor name


GDP.OPERATOR       = string, GDP-32 operator's name or initials
GDP.TYPE           = string, GDP model
GDP.PROGVER        = string, GDP acquisition program version
GDP.SIGSOURCE      = enumeration, signal source {Field,ISys,ESys}
GDP.CALVOLTS       = float, calibration signal level (volts)
GDP.SYNC           = enumeration, synchronization method {GPS,Manual,Asynchronous,non-Zero Tx }
GDP.GAINMETHOD     = enumeration, gain method {Done,Auto,Manual,SpOnly, Memory,MemSpOnly }
GDP.GAINMODE       = enumeration, gain EM noise tolerance mode {Noisy,Quiet }
GDP.FPSN           = string, GDP front panel serial number
GDP.SN             = string, GDP serial number
GDP.TCARDSN        = string, timing card serial number
GDP.NUMCARC        = integer, number of GDP A/D cards
GDP.ADCARDSN       = string list, a/d card hexidecimal serial numbers for all cards (active and inactive)
GDP.ADCARDSND      = integer list, a/d card decimal serial numbers for all cards
GDP.ADCARDVER      = string list, a/d card version
GDP.ADCARDFEAT     = string list, a/d card features
GDP.BAT            = float, GDP-32 battery voltage (volts)
GDP.TEMP           = float, GDP-32 temperature (deg C)
GDP.HUMID          = float, GDP-32 relative humidity (%)


GPS.LAT            = double, GPS latitude (deg N)
GPS.LON            = double, GPS longitude (deg E)
GPS.ALT            = float, GPS altitude (m)
GPS.PDOP           = float, position dilution of precision  (<5 good, <10 moderate, <20 fair)
GPS.NUMSAT         = integer, # gps satellites
GPS.TCDAC          = float, GPS Digital-Analog-Converter setting
GPS.STATUS         = integer, GPS status flag
GPS.SN             = integer, GPS serial number
GPS.FW             = integer, GPS firmware version
GPS.SW             = integer, GPS software version


GRID.XYZOFFSET     = float list, GDP grid coordinate offset (length units)
GRID.REF1          = float list, GDP grid reference point 1 (length units)
GRID.REF2          = float list, GDP grid reference point 2 (length units)
GRID.CALC          = float list, GDP grid calculated point (length units)
```

(*cac* file Survey Metadata records, continued)

```
TX.STN           = float, Tx dipole ID
TX.XYZ1          = floats, Tx electrode 1 along-line x: across-line y: depth z (Rx.SSpace units)
TX.XYZ2          = floats, Tx electrode 2 along-line x: across-line y: depth z (Rx.SSpace units)
TX.SN            = string, Tx serial number
TX.FREQ          = float, Tx repetition rate (hertz)
TX.DUTY          = float, Tx waveform duty cycle, usually 0.5 for TDIP or 1 for CR current on/off ratio)
TX.AMP           = float, Tx peak current in time domain
TX.SENSE         = float, current monitoring sensor resistance (ohms)
TX.DELAY         = float, transmitter turn-off ramp duration (μsec)

RX.STN           = float, Rx location
RX.AZIMUTH       = float, Rx x-component azimuh (deg east of north)
RX.ASPACE        = float, unit dipole length (length.units)
RX.SSPACE        = float, unit dipole length (Rx.XYZ units)
Rx.XYZ0          = float list, Rx location (Rx.SSpace units)
Rx.XYZ1          = float list, Rx electrode 1 along-line x: across-line y: depth z (Rx.SSpace units)
Rx.XYZ2          = float list, Rx electrode 2 along-line x: across-line y: depth z (Rx.SSpace units)
RX.LENGTH        = floats, dipole lengths for each channel (length.units)
RX.ADDELAY       = float, time offset from beginning of time-series to Tx current cut-off (μsec)
RX.ALIASDELAY    = float, anti-alias filter delay (μsec)
UNIT.LENGTH      = enumeration, {m,ft} used to specify dipole lengths

LAB.ID           = string, lab rock sample ID
```

LAB.AREA = float, lab rock sample end surface area (cm^2)

```
LAB.LENGTH       = float, lab rock sample length (cm)

TS.ADFREQ        = float, a/d sampling frequency (hertz)
TS.NCYCLE        = number cycles/waveform
TS.NWAVEFORM     = integer, number samples/waveform = TS.ADFreq/(TS.DecFac*Tx.Freq)
TS.NPNT          = integer, # time-series points  = TS.NWaveform*TS.NCycle
TS.DECFAC        = integer, decimation factor

CH.NUMON         = integer, # channels turned on
CH.ADCARDSM      = string list, a/d card hexidecimal serial numbers for active cards
CH.GDPSLOT       = integer list, GDP slots of active a/d cards
CH.FACTOR        = float list, volts per a/d counts excluding gain or attenuator settings (volts/ad_count)
CH.GAINFACTOR    = float list, channel gain (V/V ratio)
CH.PREAMP        = float list, pre-amplifier gain as a scaling multiplier
```

CH.GAIN = string list, gg-a format with g => 2^g gain factor and a => 2^{-a} attenuator

```
CH.NUMBER        = float list, component-specific meaning
    E-field chn, lower-numbered along-line electrode coordinate
    H-field chn, antenna number used to index AntCal table
CH.NOTCH         = enumeration list, notch filter setting
                   {NONE,60-359,60-3,60-59,50-359,50-3,50-59,50-60}
CH.HIGHPASS      = enumeration list, high-pass filter cut-off frequency {NONE,0.0106,0.319,192}
CH.LOWPASS       = enumeration list, anti-aliasing filter cut-off frequency {NONE,6,283,14000}
CH.CRES          = float list, contact resistance of each channel (ohms)
CH.CMP           = enumeration list, chn component label {Off,Ex,Ey,Ez,Hx,Hy,Hz,Ref}
CH.STATUS        = packed byte list, information about channel status
CH.SP            = float list, self-potential offset (mV)
```

Cac file Calibrate Metadata records (record type = 768)

Calibrate data stored as ASCII text in the form of comma-separated-value sub-records with a `keyword,value1,value2,...,valueN` format. Sub-record endpoints are delineated by a carriage-return character (\n).

`HEADER.TYPE = enumeration, Calibrate`
`CAL.VER`   = calibrate format version number
`CAL.BRD`   = system calibrate records with comma-separated calibrate value tuples
`CAL.BRD, E82A, 0.125:1.0:0.0,  0.375:1.0:0.0, 0.625:1.0:0.0, . . .`
The first system calibrate value is an A/D board's serial number, followed by multiple comma-separated calibrate-tuple fields, each tuple with a frequency: Real(Cal) Imag(Cal) format.   Each calibrate tuple has a frequency in hertz followed by a colon, then a calibrate magnitude in gdp_volt/ volt   followed by a colon, then a calibrate phase in milliradians.
`CAL.ANT`   = antenna calibrate records using the same format as `CAL.SYS`,
                but with an antenna number as the first value instead of a board serial number.
                antenna calibrate magnitudes are in V/uT, phase values are in milliradians
----

Cac file Data Metadata records (record type = 528)

Data Metadata records store a description of subsequent binary numeric data blocks.
Data metadata are stored as ASCII text in the form of comma-separated-value sub-records with a `keyword,value1,value2,...,valueN` format. Sub-record endpoints are delineated by a carriage-return character (\n).

`HEADER.TYPE` = enumeration, {`TimeSeries,Stack,Harmonics,SummaryData`}
`DATE0`   = data acquisition date (mm/dd/yyyy)
`TIME0`   = data acquisition time (hh:mm:ss.sss)
`DATA.SKIP`   = boolean, 0=use data, 1=skip data
`COL.LABEL`   = string list, data column labels
`COL.NUM`   = number of data columns in subsequent binary data records
`ROW.NUM`   = number data of rows in subsequent binary data records
`DATA.KIND`   = enumeration or enumeration list, numeric data kind {integer,float,double,complex}

Cac file binary Time-Series Data records (record type = 16)

A `COL.NUM` by `ROW.NUM` block of binary time-series data stored as four-byte integer values representing A/D counts . The numeric data are stored row by row with the column index varying first. `COL.NUM` is equal to the number of active GDP channels, while `ROW.NUM` is equal to `TS.NPNT`.   The A/D values should be scaled by `CH.FACTOR/(CH.GAIN*CH.EXGAIN)` to get gdp volts.

Cac file binary Stack Data records (record type = 1)

A `COL.NUM` by `ROW.NUM` block of binary time-series data stored as four-byte float values representing averaged A/D counts in a rectified stack . The numeric data are stored row by row with the column index varying first. `COL.NUM` is equal to the number of active GDP channels, while `ROW.NUM` is equal to `TS.NWAVEFORM/2`.

Cac file binary Result (TDIP) Data records (record type = 2048)

A `COL.NUM` by `ROW.NUM` block of binary time-series data stored as four-byte integer and float values The numeric data are stored row by row with the column index varying first. `COL.NUM` is equal to the number of variables, while `ROW.NUM` is equal to the number of active GDP channels.

TDIP result data have columns:
Chn   = channel index
Vp   = peak on-time voltage (volts)
Vs0   = Newmont chargeability (mV/V)
Vs1   = first Vs transient value (mV/V)
 . . .
Vs13   = last Vs transient value (mV/V)

Cac file binary Summary Data records (record type = 1024)

A `COL.NUM` by `ROW.NUM` block of binary time-series data stored as four-byte integer and float values representing derived geophysical parameters like apparent resistivity and IP chargeability. The numeric data are stored row by row with the column index varying first. `COL.NUM` is equal to the number of geophysical parameters, while `ROW.NUM` is equal to the number of active GDP channels.

TDIP summary data have the columns:
Chn   = channel index
IPSEM   = chargeability IP standard error of the mean (millseconds)
Vp   = peak on-time voltage (volts)
IPM   = Newmont chargeability (millseconds)
ARes   = apparent resistivity (ohm-m)

Cac file GDP Status (record_type = 640):

QCGdpTemp = float, GDP-32 temperature (deg C)
QCGdpTemp = float, GDP-32 relative humidity (%)

GDP-32 ASCII TDIP Data File (*.raw)

GDP-32 TEM and NTEM data are saved in ASCII files with a fixed-form block style. Blank lines separate blocks and there are two block types, header and data.

GDP-32 TDIP Header Block Samples

```
0097
TDIP0832 2007-05-02 11:10:43 12.9v D-D  20.5%  26.7 DegC
OPER    Curt C TX ID  temp A-SP    300 FT
JOB   745 LINE     1400 N  SPREAD  0 MAV Enabled
   1 DiffAmp Notch 60,3-5,9  S/N  465 Passed 1.00000
   .  .    .    .    .    .    .    .    .
   .  .    .    .    .    .    .    .    .


0012
TDIP0538 97-09-02 17:09:09 12.3v Grd
OPER    SCOTT TX ID  2009 A-SP    100 FT
JOB  9782   Ay    31600 N  SPREAD  1 Ax  -33800 Bx   -23100 MAV Enabled
   1 DiffAmp Notch+60,3-5,9  S/N  190 Passed 0.99519
   .  .    .    .    .    .    .    .    .
   .  .    .    .    .    .    .    .    .
```

GDP-32 TDIP Header block structure

```
-- FIELD_NAME ------- UNITS -- RANGE --- TYPE ------ WIDTH -- END --
Line 1
   Block Number           #   0 - 9999  Integer       4       4

Line 2
   Survey Type            -   ex:"TDIP" Character      4       4
   Version Number         #   0 - 9999  Integer       4       8
   Skip Flag              -   x or space Character     1       9
   Date               date   yyyy-mm-dd Date          10      19
   Time               time   hh:mm:ss   Time           8      28
   Battery Voltage       v   0 - 99.9   Scaled         5      34
   Array Type            -   ex:"INL"   Character      3      38
   Relative Humidity     %   0 – 100    Float          7      45
   Temperature       deg C   -90 – 200  Float          6      51

Line 3
   Operator              -   ex:"JOHN"  Character      9      14
   Tx ID                 -   ex:"ONE"   Character      5      26
   A-Spacing          m/ft   positive   Scaled         6      38
   Length Unit           -   M or FT    Character      2      41
   Ax (Grd or Sch Arrays) #  in m or ft Scaled         8      48
   Bx (Grd or Sch Arrays) #  in m or ft Scaled         8      60
   MAV telluric filter       Enabled or Disabled

Line 4
   Job Number            #   ex:"GOLD"  Character      5       9
   Line                  #   ex:"1+00"  Character      8      23
   Line Direction        -   ex:"N"     Character      2      25
   Spread                #   ex:"A"     Character      2      36


Line 5+ hardware inventory information, one line for each channel.
```

GDP-32 TDIP Data Block Sample

```
0108
TDIP0832 2007-05-02 11:50:12 12.8v D-D  20.5%  30.0 DegC
Tx    2300 Rx     1700 N 60, 5
.125 Hz     8 Cyc Tx Curr     4
1  ON   1 1.5320    4.3 660.1   000O   0.02 -19.36  3.68K
2  ON   2 1.2182    3.3  2100   000O   0.01   3.71  6.10K
3  ON   3 0.1814    1.6 781.8   002O   0.11  12.91  5.93K
4  ON   4 37.127m  15.7 320.0   004O   0.06  12.08  4.43K
5  ON   5 18.980m  20.1 286.2   005O   0.22   8.79  3.81K
6  ON   6 11.958m  20.1 288.5   006O   0.27  21.15  3.06K
Windows
     1        2        3        4        5        6        7        8
  4011     3972     3913     4296     4193     4375
    91       42       37      309      340      363
    56       43       25      194      246      248
    41       32       16      155      200      195
    35       27       13      129      168      169
    30       23       11      113      145      146
    27       20       10      101      129      128
    23       18       10       88      114      119
    21       16        7       78      101      101
    19       15       10       71       94       97
    18       14        4       65       88       90
    16       13        5       61       83       83
    15       12        8       58       73       76
```

GDP-32 TDIP data block structure

```
-- FIELD_NAME ------- UNITS -- RANGE --- TYPE ------ WIDTH -- END --
Line 1
   Block Number            #   0 - 9999   Integer        4       4

Line 2
   Survey Type             -   ex:"TDIP"  Character      4       4
   Version Number          #   0 - 9999   Integer        4       8
   Skip Flag               -   x or space Character      1       9
   Date                  date  yyyy-mm-dd Date          10      19
   Time                  time  hh:mm:ss   Time           8      28
   Battery Voltage       volts 0 – 13.0   Float          6      34
   Array Type              -   ex:"D-D"   Character       3      38
   Relative Humidity       %   0 – 100    Float          7      45
   Temperature           deg C -90 – 200  Float          6      51

Line 3
   Tx value                #   ±0-9999999 Scaled         8      11
   Rx value                #   ±0-9999999 Scaled         8      23
   Notch Filter            -   ex:"60, 5" Character      5      31

Line 4
   Frequency              Hz   ex:0.125   Scaled         4       4
   # Cycles/Stack          #   1 - 16384  Integer        5      14
   Tx Current            amps  0 - 99999  Scaled         6      32
```

GDP-32 TDIP data block structure (continued)

```
-- FIELD_NAME ------- UNITS -- RANGE --- TYPE ------ WIDTH -- END --

Line 5: repeated for each channel measured.
   Channel Number        #    1 - 16    Integer        1      2
   Skip Flag             -    sp,x,-,b  Character      1      3
   Channel Type          -    ex:"ON"   Character      3      6
   N-Spacing             #   ±0 - 99999 Scaled         6     12
   Primary Voltage       v              Scaled w/exp   8     20
   Chargeability         ms             Scaled w/exp   8     28
   Apparent Resistivity  Ωm             Scaled w/exp   7     36
   Gains/Attenuator      ###A ex:"006O" Character      4     42
   SEM                   ms             Scaled w/exp   6     49
   Self Potential        mv             Scaled         7     57
   Contact Resistance    Ω              Scaled w/exp   6     64
```

TDIP input data file window structure follows two header lines.
All 13 rows of the window data are the same:

```
   Channel 1      10*mV/V      Integer      7      7
   Channel 2      10*mV/V      Integer      7     15
   Channel 3      10*mV/V      Integer      7     23
   Channel 4      10*mV/V      Integer      7     31
   Channel 5      10*mV/V      Integer      7     39
   Channel 6      10*mV/V      Integer      7     47
      .              .            .         .      .
      .              .            .         .      .
      .              .            .         .      .
   Channel 16     10*mV/V      Integer      7    127
```

NOTE: Extremely large values will shift data in the columns that follow.

A data block may be skipped by putting an ″x″ just to the right of the GDP program name, i.e. change:
```
TDIP0832 2007-05-02 11:50:12 12.8v D-D  20.5%   30.0 DegC
  to
TDIP0832x2007-05-02 11:50:12 12.8v D-D  20.5%   30.0 DegC
```

Individual channels may be skipped by putting an ″x″ right after the channel number, i.e. change
```
2  ON     2 1.2182      3.3   2100    000O   0.01    3.71  6.10K
  to
2  xON    2 1.2182      3.3   2100    000O   0.01    3.71  6.10K
```

The polarity of individual channels can be flipped by putting a ″-″ after the channel number, i.e. change
```
2  ON     2 1.2182      3.3   2100    000O   0.01    3.71  6.10K
  to
2  -ON    2 1.2182      3.3   2100    000O   0.01    3.71  6.10K
```

Raw file survey configuration keywords

Starting with v1.12a **tdavgw** will read a limited set of IP survey configuration keywords from raw files, so that conventional IP array conventions can be extended to allow more general configurations. Raw file keyword records are similar to $keyword=values format used in mde, zdb and avg files. A leading $ flags the text as a keyword record. The keyword is separated from its values by the = character. Multiple values are separated by commas. (x,y) or (x,y,z) coordinate tuples use a hierarchy of separators, with commas separating tuples and colons separating individual values within each tuple, i.e. multiple (x,y) tuples look like 0:0, 100:0, 200:0, … and multiple (x,y,z) tuples look like 0:0:0, 100:0:0, 200:0:0. By default, raw file keyword variables are set to values consistent with standard GDP conventions. The standard conventions can be modified by inserting $keywords records into the raw file just before the data blocks that are to be affected.

Variable Rx dipole lengths

The first extension to conventional surveys is to allow multiple receiver dipole lengths in a single setup. **Tdavgw** uses A-SP, Tx, Rx and the lowest numbered n-spacing to calculate how station numbers scale to a unit dipole length in meters or feet. For dipole-dipole, the transmitter dipole is one A-SP long. The header block A-SP value in meters or feet is used as a unit dipole length to calculate both transmitter electrode station numbers and the lower numbered receiver electrode stations in combination with the n-spacing values specified for each GDP channel. Variable receiver dipole lengths specified by:

$Ch.Length=length1, length2, …, lengthN

can be used to relocate the higher number receiver electrodes. Length1, …, lengthN must be in the same length units as the A-SP value in the preceding raw file header block.

Longer receiver dipole lengths can be used to increase signal strength at large n-spacings.
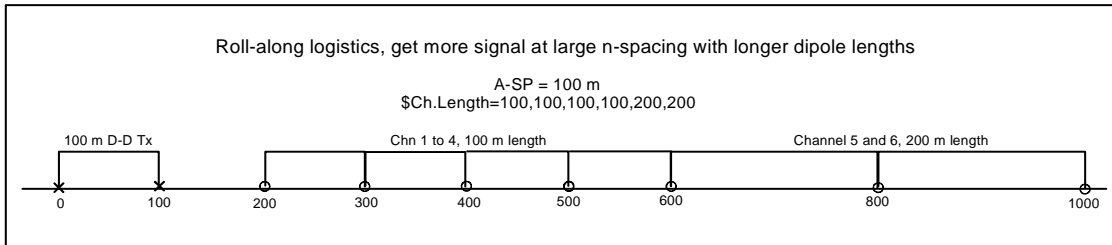
For a specific $Ch.Length example with raw file text, consider the following configuration:
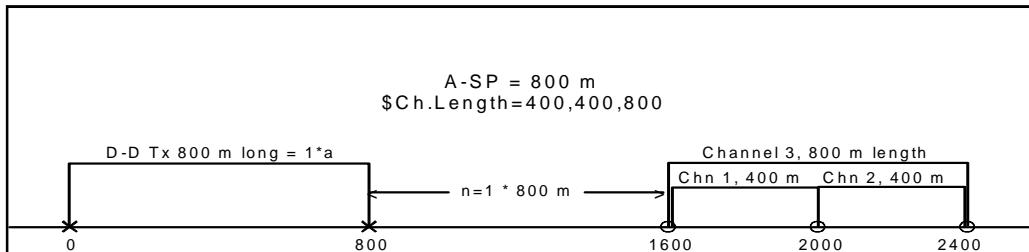
Example `$Ch.Length=400,400,800` raw file text
```
0271
TDIP0843 2009-08-23 11:25:01 12.6v D-D  47.3%  23.3 DegC
OPER     Caton TX ID    3 A-SP    800  M
JOB  9.86 LINE       M N  SPREAD  1 MAV Enabled
1  DiffAmp Notch 60,3-5,9  S/N  853 Passed 1.00000
2  DiffAmp Notch 60,3-5,9  S/N 1355 Passed 1.00000
3  DiffAmp Notch 60,3-5,9  S/N  134 Passed 1.00000
4  DiffAmp Notch 60,3-5,9  S/N  149 Passed 1.00000
5  DiffAmp Notch 60,3-5,9  S/N  162 Passed 1.00000
6  DiffAmp Notch 60,3-5,9  S/N  355 Passed 1.00000
Front Panel S/N   14, Cal S/N   22, Temp  23.3, Humidity  47.3, EPROM 030901BLD321

$Ch.Length=400,400,800
0272
TDIP0843 2009-08-23 11:33:29 12.5v D-D  47.3%  23.3 DegC
Tx       0 Rx     1600 N 60, 5
.125 Hz      8 Cyc Tx Curr   1.1
1  ON    1   43.348m   27.1   1300   004O   2.76   10.99  53.5K
2  ON   1.5   9.8708m   27.3  888.0   006O   5.23   -6.59  56.7K
3  ON    1   53.223m   27.0  729.6   004O   3.26   13.18  10.1K
```

`$Ch.Length=400,400,800` is inserted just after a header block specifying an A-SP = 800 meters and just before a D-D data block with three active channels, channels 1 and 2 each 400 m long and channel 3 spanning 800 m. *Tdavgw* depends upon the blank line separators between each raw file header and data block to mark the beginning and end of each block, so a blank line is left just above `$Ch.Length=400,400,800.`

Lateral offset between Tx and Rx dipoles

A second extension to conventional surveys is to offset the transmitter sideways from the receiver line, most commonly for Australian style pole-dipole where a long array of receiver dipoles is set up, and then the transmitter pole run is down the line with positions centered on each receiver dipole in turn.
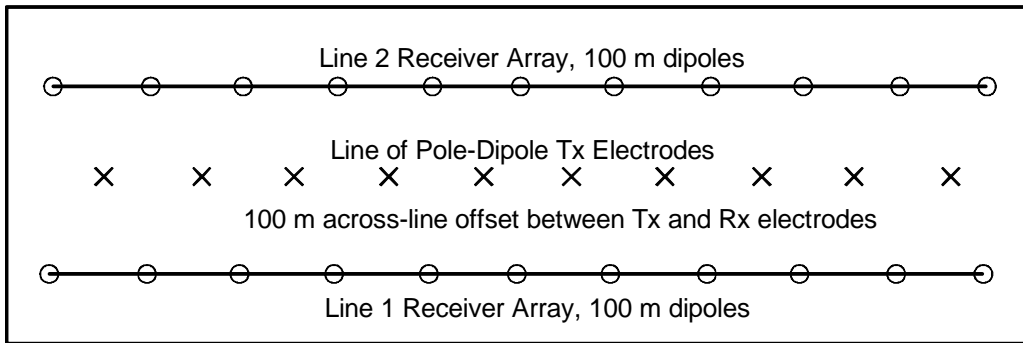
Across-line coordinates for line 1 receiver electrodes and the transmitter pole electrodes could be specified by inserting:
$Tx.Y1=100
$Tx.Y2=100
$Ch.Y1=0,0,0,0,0,0,0,0,0,0
$Ch.Y2=0,0,0,0,0,0,0,0,0,0
into the raw file after the line 1 header block(s). You can specify that GDP channel numbers be interpreted as receiver stations rather than n-spacings to avoid having to change the pole-dipole n-spacing every time the transmitter pole moved. (Apart from the fact that the n-spacing system breaks down anyway when the transmitter pole is centered in a receiver dipole.) For receiver line 2, the across-line coordinate would be 200 meters, so the raw file $keyword records would look like:
$Tx.Y1=100
$Tx.Y2=100
$Ch.Y1=200, 200, 200, 200, 200, 200, 200, 200, 200, 200
$Ch.Y2=200, 200, 200, 200, 200, 200, 200, 200, 200, 200

Specifying explicit Tx or Rx electrode (x,y) or (x,y,z)

To extend TDIP raw files to general surface array configurations, the keywords $Tx.XYZ1, $Tx.XYZ2, $Ch.XYZ1 and $Ch.XYZ2 can be used to explicitly specify electrode along-line and across-line coordinates. An example for a four-channel receiver array in something similar to a gradient array might look like:
$Tx.XYZ1=10200:500
$Tx.XYZ2=20200:500
$Ch.XYZ1=10300:100, 10400:100, 10500:100, 10600:100
$Ch.XYZ2=10400:100, 10500:100, 10600:100, 10700:100

Note that the colon character is used to separate x and y coordinates for each receiver channel, while commas are used to separate x:y coordinate pairs for each channel in turn.

The $Tx.XYZ2 keyword record is very helpful for specifying the actual location of one or more pole-dipole transmitter remote electrodes in (along-line,across-line,[depth]) coordinates.

When using the drill hole array, D-H, a Z coordinate representing electrode depth can be included:
$Ch.XYZ1=10300:100:000, 10400:100:100, 10500:100:200, 10600:100:300
$Ch.XYZ2=10400:100:100, 10500:100:200, 10600:100:300, 10700:100:400
Using the D-H array is the only way to get the Zdepth coordinates included in *tdavgw* zdb and avg files.

GDP-32 FLD Files: .raw reformatted by program SHRED or SHREDQW

The utility programs Shred or ShredQW flatten block oriented GDP raw files into a tabular spreadsheet format with space-separated values.

HEADER Record Structure

```
-- FIELD_NAME ------- UNITS -- RANGE --- TYPE ------ WIDTH -- END --
      Record Type            -    H , D       Character      1       1
      Skip/Average Flag      #    0 , 1       Integer        1       3
      Block Number           #    0 - 9999    Integer        4       8
      Header Label           #    0 - 999     Integer        3      12
      Sequence Number        #    0 - 9       Integer        1      14
      Survey Type, Version   -                Character      8      23
      Date               date    dd mmm yy    Date           8      32
      Time               time    hh:mm:ss     Time           8      41
      Voltage               v    0 - 99.9     Scaled         4      46
      Array Type             -    ex:"D-D"     Character      3      50
      Operator               -    ex:"JOHN"    Character      9      60
      Tx Identification      -    ex:"ONE"     Character      4      65
      A-Spacing              m    positive     Scaled         6      72
      Job Number             -    ex:"GOLD"    Character      5      78
      Line Number            -    ex:"1+00"    Character      8      87
      Line Direction         -    ex:"NE"      Character      2      90
      Spread                 -    ex:"A"       Character      2      93

For Grd (gradient) or Sch (Schlumberger) arrays:
      Ax       (Sch or Grd) m  ±0-9999999  Scaled         8     112
      Bx       (Sch or Grd) m  ±0-9999999  Scaled         8     121
```

TDIP Data Record Structure

```
-- FIELD_NAME ------- UNITS -- RANGE --- TYPE ------ WIDTH -- END --
     Record Type            -    H , D      Character     1      1
     Skip/Average Flag      #    0 , 1      Integer       1      3
     Block Number           #    0 - 9999   Integer       4      8
     Header Label           #    0 - 999    Integer       3     12
     Sequence Number        #    0 - 9      Integer       1     14
     Date                  date  dd mmm yy  Character     8     23
     Time                  time  hh:mm:ss   Character     8     32
     Voltage                v    0 - 99.9   Scaled        4     37
     Frequency             Hz    ex:"  16"  Scaled        5     43
     Cycles                 #    1 - 16384  Integer       5     49
     Tx Current           amps   0 - 99999  Scaled w/exp  8     58
     Tx                     #    ±0-9999999 Integer       8     67
     Rx Calculated          #    pos or neg Scaled w/exp 11     79

Secondary structure
     Channel Type           -    ex:"On"    Character     3     83
     Polarity Flip Flag     #    0 , 1      Integer       1     85
     N-Spacing              #    ±0-9999999 Scaled        8     94
     Primary Voltage        v               Scaled w/exp 10    105
     Chargeability         ms     ±9999.9   Scaled w/exp 10    116
     Resistivity           Ωm               Scaled w/exp 10    127
     Gains/Attenuator     ###A   ex:"026O"  Character     4    132
     SEM                   ms               Scaled        7    140
     Self Potential        mv               Scaled        7    151
     Contact Resistance    Ω                Scaled w/exp  9    161
     External Amp Gain      #    1-999999   Integer       6    167

Window Data structure (13 values, total 271 columns)
     Window Value      10*mV/V ±0-999999    Integer       7     +8

For array type Lab (Lab Rock)
     Tx Reference         volts -99.9 to 99.9 Scaled     11    282
     Tx Shunt              Ω               Scaled w/exp 10    292
     Sample Length         cm    positive   Scaled        8    302
     Sample Area           cm²   positive   Scaled        8    312
```

TDAVGW **zdb** and **avg** files: TDIP data

zdb *and* **avg** files have the same format, but **zdb** files are used to store data with unaveraged repeats, while the file name extension **avg** implies that repeat measurements have been averaged. **zdb** files keep track of the detailed skip-flag pattern set during interactive QC of unaveraged data. *Tdavgw* can re-read data from **zdb** files to revisit QC decisions without starting over from the beginning.

avg files may have comment lines preceded by a leading "\", "/", "!" or """anywhere within the file, although they are normally grouped at the top of the file.

Keyword records with a leading "$" character are also present and play a large role in recording information that does not fit into a tabular spreadsheet format. Keyword records have a `$program:group.variable=value(s)` format. The program name is optional, but if present the keyword record is ignored if the program name is not *tdavgw*. Version 2 keywords generally have a `group.variable` structure with the group name unifying keywords with a common theme. An equals "="sign separates the keyword from one or more comma-separated values. String values which may include commas as part of the value, instead of as a field separator, should be enclosed in quotes. To allow future file-format extension, programs should skip unrecognized keywords without generating an error.

Records with a leading letter are interpreted to contain column labels. At least one line holding column labels must precede numerical data. Column label matching is case insensitive, but does not allow substring matching. Column order is not fixed and all possible columns may not be present in a particular file. If an essential column is missing, the program will show a warning message and abort input.

Numeric data records begin with a leading number. Numerical values are free format with columns separated by either spaces or commas, although comma separated values are preferred. Missing numeric values are flagged by a "*" symbol or empty space bracketed by commas.

Tdavgw *.avg-file keywords:
Survey annotation

Job.Name	= project name	(128 character string)
Job.Area	= project area	(128 character string)
Job.For	= customer company name	(128 character string)
Job.By	= contractor company name	(128 character string)
Job.Number	= job label or number	(16 character string)
Job.Date	= data acquisition date	(16 character string)

Survey configuration

Survey.Type	= survey type (CSAMT, TEM, CR, TDIP)
Survey.Array	= array type (for IP: D-D, P-D, P-P, Grd, Sch, D-H, Lab)
Line.Name	= line label (16 character string) (alias = JobLine)
Line.Number	= line number across-line coordinate (float)
Line.Azimuth	= line azimuth in direction of increasing stn # (float, deg E of N) (alias = BrgLine)
Stn.GdpBeg	= first GDP station number (float) (alias = StnBeg)
Stn.GdpInc	= GDP station number increment (non-zero float) (alias = StnDelt)
Stn.GdpUnitScale	= GDP stn number scale (m/GdpStn#)
Stn.Beg	= possibly rescaled first station number (float) (alias = LblFrst)
Stn.Inc	= possibly rescaled station number increment (float) (alias = LblDelt)
Stn.UnitScale	= client station number scale (m/Stn#)
Stn.Left	= rescaled station number on left edge of pseudosection plot (float) (alias=LblFrst)
Stn.Right	= rescaled station number on right edge of pseudosection plot (float)
Rx.Area	= Rx magnetic field antenna effective area or rock sample area (m^2)
Rx.Length	= Rx dipole length (m)
Rx.HPR	= Rx x-component heading & pitch, z component roll (degrees)

Units

`Unit.Length`	= length units	(m,ft) (alias = Units)
`Unit.Time`	= time units	(usec,msec,sec)
`Unit.V`	= voltage units	(nV/A,uV/A,mV/A,V/A,nV,uV,mV,V)
`Unit.IP`	= IP units	(msec,mrad,PFE)

Time-window configuration parameter group

`TWCFG.NTWCfg`   = number of time-window configurations
`TWCFG(1).TxWaveform`   = transmitter waveform shape parameters
           transmitter frequency, % duty cycle, turn-off ramp (μsec), shape
`TWCFG(1).AliasFilter`   = anti-aliasing filter parameters
           filter type, -3dB frequency (hertz), roll-off slope (dB/decade),
           step-response delay (msec), step-response ramp width (msec)
`TWCFG(1).TBin.Name`   = GDP type (GDP-32^{II}, GDP-32^{24})
`TWCFG(1).TBin.NWaveform` = number of A/D points per waveform
`TWCFG(1).TBin.Beg`   = time-window bin start (A/D point wrt Tx turn-off)
`TWCFG(1).TBin.End`   = time-window bin end (A/D point wrt Tx turn-off)
`TWCFG(1).TWin.Beg`   = time-window begin (msec w.r.t. Tx turn-off)
`TWCFG(1).TWin.End`   = time-window end (msec w.r.t. Tx turn-off)

TDAVGW processing control parameters

`TDAVGW:Process.TS`   = yes: process cac file time series
`TDAVGW:Process.STACK`   = yes: process cac file stacks
`TDAVGW:Process.DECAY`   = yes: process cac file decays
`TDAVGW:Process.BINNED`   = yes: process binned data
`TDAVGW:Stack.Type`   = stack type, {Straight, Taper, Robust}
`TDAVGW:Stack.Taper`   = stack taper {0 to 6}
`TDAVGW:Stack.Trim`   = stack trim (alpha trimmed mean %)
`TDAVGW:Avg.Type`   = averaging method {Straight,Robust}
`TDAVGW:IPM.Recaculate`   = yes: recalculate Newmont chargeability
`TDAVGW:IPM.Scale`   = Newmont M scaling, 1.87, 2.86
`TDAVGW:IPM.TwBeg`   = Newmont M integration begin time (msec*0.125/Tx.Freq)
`TDAVGW:IPM.TwEnd`   = Newmont M integration end time (msec*0.125/Tx.Freq)
`TDAVGW:Ch.NumberType`   = channel number useage (stn #,stn # offset, n-spacing, n-offset, antenna #)
`TDAVGW:GDP.Type`   = GDP type (GDP16,GDP32,GDP32ii,GDP3224)
`TDAVGW:GDP.GainMode`   = GDP gain mode (Noisy,Quiet,Unknown)
`TDAVGW:CF.RPdat`   = resistance color fill min & max asinh10(mV/A)
`TDAVGW:CF.ARdat`   = apparent resistivity color min & max asinh10(ohm-m)
`TDAVGW:CF.IPdat`   = IP chargeability color fill min & max (msec)
`TDAVGW:CF.IPerr`   = IP error color fill min & max (msec)

TDAVGW *.avg-file columns:

```
Tx.GdpStn     = Tx GDP station number, used as Tx ID for D-H array key to TWC file
Rx.GdpStn     = Rx GDP station number
Tx.X1         = Tx electrode 1 along-line coordinate (in-line array stn #, D-H array easting)
Tx.X2         = Tx electrode 2 along-line coordinate (in-line array stn #, D-H array easting)
Rx.X1         = Rx electrode 1 along-line coordinate (in-line array stn #, D-H array easting)
Rx.X2         = Rx electrode 2 along-line coordinate (in-line array stn #, D-H array easting)
Tx.Y1         = Tx electrode 1 across-line coordinate (in-line array line #, D-H array northing)
Tx.Y2         = Tx electrode 2 across-line coordinate (in-line array line #, D-H array northing)
Rx.Y1         = Rx electrode 1 across-line coordinate (in-line array line #, D-H array northing)
Rx.Y2         = Rx electrode 2 across-line coordinate (in-line array line #, D-H array northing)
Tx.Z1         = Tx electrode 1 depth, down-hole array only (D-H array depth)
Tx.Z2         = Tx electrode 2 depth, down-hole array only (D-H array depth
Rx.Z1         = Rx electrode 1 depth, down-hole array only (D-H array depth)
Rx.Z2         = Rx electrode 2 depth, down-hole array only (D-H array depth)
Tx.Name       = Rock sample identifier label, LabRock array only (text string)
Rx.Name       = Rock sample identifier label, LabRock array only (text string)
Rx.Length     = Rock sample length, LabRock array only (cm)
Rx.Area       = Rock sample length, LabRock array only (cm)
Plt.X         = Pseudosection along-line plot point (stn #)
Plt.Z         = Pseudosection plot point depth (stn #)
Plt.Nsp       = n-spacing (when applicable)
Gdp.Blk       = GDP-32 data block number
Gdp.Chn       = GDP-32 analog channel number
Gdp.Time      = Data acquisition date
Gdp.Time      = Data acquisition time of day
Tx.Freq       = transmitter repetition rate (hertz)
Tx.Amp        = transmitter current (peak amps)
Tx.NCycle     = number of cycles per stack
Rx.Cmp        = EM component (should be Ex or Ey for TDIP)
Rx.CntRes     = Rx dipole contact resistance (ohms)
ARes.mag      = apparent resistivity magnitude (ohm-m)
ARes.%err     = apparent resistivity error (%)
ARes.wgt      = apparent resistivity weight (0=skip, otherwise use)
IP.mag        = chargeability magnitude (msec)
IP.err        = chargeability error (msec)
IP.wgt        = chargeability weight (0=skip, otherwise use)
MF.mag        = metal factor (usec/ohm-m)
SP.mag        = self-potential (mV)
SP.err        = self-potential error (mV)
Vp.mag        = primary voltage magnitude (mV)
Vp.err        = primary voltage error (mV)
TWCfg.index   = time-window configuration index (pointer to $TWCfg(:) parameters)
Vs.mag0       = chargeability time window secondary voltage (mV/V)
Vs.mag*       = secondary voltage transient (mV/V)
Vs.err*       = secondary voltage error (mV/V)
Vs.wgt*       = secondary voltage weight (0=skip, otherwise use)
```

TDAVGW *.avg-file columns (continued):

When geographic coordinates are read from a station file, *tdavgw* adds the following columns:

```
Plt.East    = Plot point easting (length units)
Plt.North   = Plot point northing (length units)
Plt.Elev    = Plot point elevation (length units)
Tx.East1    = transmitter electrode 1 easting (length units)
Tx.North1   = transmitter electrode 1 northing (length units)
Tx.Elev1    = transmitter electrode 1 elevation (length units)
Tx.East2    = transmitter electrode 2 easting (length units)
Tx.North2   = transmitter electrode 2 northing (length units)
Tx.Elev2    = transmitter electrode 2 elevation (length units)
Rx.East1    = receiver electrode 1 easting (length units)
Rx.North1   = receiver electrode 1 northing (length units)
Rx.Elev1    = receiver electrode 1 elevation (length units)
Rx.East2    = receiver electrode 2 easting (length units)
Rx.North2   = receiver electrode 2 northing (length units)
Rx.Elev2    = receiver electrode 2 elevation (length units)
```

D-H (down-hole) array
=> adds columns `Tx.Z1, Tx.Z2, Rx.Z1` and `Rx.Z2` to keep track of electrode depths.
Adds columns `Tx.GdpStn` and `Rx.GdpStn`
Tdavgw processing of down-hole array maps (station,line) tuples to (east,north,depth) coordinates via interpolation from twc and stn file tables, generating (x,y,z=depth) coordinates in length units.

Lab (Lab Rock) array
=> drops electrode coordinates and adds columns
`Tx.GdpStn, Rx.GdpStn, Tx.Name, Rx.Name, Rx.Length, Rx.Area`

LabRock processing uses `Tx.GdpStn` and `Rx.GdpStn` as sample number identifiers rather than station numbers. *Tdavgw* includes columns for text variables `Tx.Name` and `Rx.Name` in zdb & avg files so that rock samples can also be identified by arbitrary text labels. `Tx.Name` and `Rx.Name` columns should be to the right of `Tx.GdpStn` and `Rx.GdpStn` so that numeric data records do not start with a leading alphabet character or quote mark, which would flag the record as a column label or comment line rather than numeric data.

Geosoft IP *.dat format

Geosoft IP *.dat files are a generic format for IP data. Dat files are ASCII text that can be viewed with generic text editor. Record types depend upon position. The first line or record in the file is a title line which may hold descriptive text. The second record holds a set of space-separated keyword:value(s) fields describing the survey configuration. The third record must hold column labels to identify numeric data parameters. Subsequent records are space-separated numeric data in a tabular format. Missing or null data are indicated by a * character. Blank lines are ignored. The IP *.dat format is close enough to a spreadsheet format to make it a useful generic format.

For dipole-dipole, pole-dipole and pole-pole arrays *tdavgw* uses station numbers for along-line coordinates which may be in arbitrary units. Oasis montaj adds the expectation that station numbers are be the same as length units. *Tdavgw* can rescale the GDP station numbers used during data acquisition from arbitrary units to length units if necessary. Zonge GDP-32 gradient data acquisition requires that station numbers be equivalent to length units in m or ft. The *tdavgw* processing of down-hole array maps (station,line) tuples to (east,north,depth) coordinates via interpolation from twc and stn file tables, generating (x,y,z) coordinates in length units. LabRock data use station numbers as sample identifiers. Tx.Name and Rx.Name string fields are added to *tdavgw* Labrock data output files to allow sample identifiers with arbitrary text.

The Oasis montaj IP module dat import currently (June/2008) fails when the IP survey configuration is extended to include lines that are not oriented N-S or E-W, or for *tdavgw* Schlumberger, down-hole or LabRock data. However, the data can still be imported into Oasis montaj using generic text file input.

Partial listing of a *tdavgw* TDIP dat file

```
------------------------------------------------------------------------
TDAVGW v1.10a, TDIP data processed on 2009/06/18: TDIP D-D Data
LINE:1400N ARRAY:DPDP DIPOLE:300 UNITS:ft T=50.784,148.432,148.44,148.44,…
T1X T2X R1X R2X T1Y T2Y R1Y R2Y SPlt YPlt ZPlt NSp Line East North Elevation Vp TxAmp TxFreq … 1100
1400 200 500 1400 1400 1400 1400 800 1400 -450 2 1400N 800 1400 0 175.6 4 0.125 22.2 …
1100 1400 500 800 1400 1400 1400 1400 950 1400 -300 1 1400N 950 1400 0 515.13 4 0.125 9.477 … 1400
1700 200 500 1400 1400 1400 1400 950 1400 -600 3 1400N 950 1400 0 108.7 6 0.125 18.357 …
------------------------------------------------------------------------
```

The first dat file record is a title line holding descriptive text and is ignored by Oasis montaj.

The second keyword record holds a series of space-separated keyword:value fields. The first keyword should be `Line`. Oasis montaj expects that the line keyword value will be a numeric value and bearing letter {N,E,S,W}.

It uses the bearing letter to decide whether the along-line coordinate should be labeled X or Y. Oasis montaj IP dat file import fails for oblique lines with labels like 1400NE. Geosoft IP dat files can hold data for multiple lines, with the data block for each additional line after the first flagged by a keyword record.

Tdavgw dat file survey configuration keywords:

LINE = line name string, the Oasis montaj IP dat import restricts the line name format, see above paragraph.
ARRAY = array type, Oasis montaj restricts array to DPDP, PLDP or PLPL
Tdavgw may also write data for Grad, Sch, D-H or Lab arrays
DIPOLE = dipole length (length)
UNITS = length units {m or ft}
T = Vs(1) time window beginning, followed by 13 time window widths, comma-separated-values (msec)

Tdavgw dat file data columns:

T1X	= transmitter electrode 1 along-line coordinate (stn #)
T2X	= transmitter electrode 1 across-line coordinate (stn #)
R1X	= receiver electrode 1 along-line coordinate (stn #)
R2X	= receiver electrode 2 along-line coordinate (stn #)
T1Y	= transmitter electrode 1 across -line coordinate (line #)
T2Y	= transmitter electrode 2 across-line coordinate (line #)
R1Y	= receiver electrode 1 across -line coordinate (line #)
R2Y	= receiver electrode 2 across -line coordinate (line #)
T1Z	= transmitter electrode 1 depth, down-hole array (stn #)
T2Z	= transmitter electrode 2 depth, down-hole array (stn #)
R1Z	= receiver electrode 1 depth, down-hole array (stn #)
R2Z	= receiver electrode 2 depth, down-hole array (stn #)
SPlt	= plot point station number
YPlt	= plot point line coordinate
ZPlt	= pseudosection plot point depth
NSp	= n-spacing
TxGdpStn	= Tx gdp station number, used as D-H key to twc file
RxCmp	= EM component (Ex,Ey,Ez)
RxChnNumber	= GDP analog input channel number
Line	= line name
Vp	= peak voltage (mV)
TxAmp	= peak current (amps)
TxFreq	= transmitter frequency (hertz)
Sp	= self-potential offset (mV)
Ares	= apparent resistivity (ohm-m)
ARerr	= apparent resistivity error (percent)
IPm	= Newmont chargeability (msec)
IPerr	= Newmont chargeability error (msec)
MF	= metal factor (1000*msec/ohm-m)
IP1..IP13	= Vs(t) transient values (mV/V)

When geographic coordinates are read from a station file, *tdavgw* adds the following columns:

PltEast, PltNorth, PltElev = plot point east, north and elevation (length units)
TxEast1, TxNorth1, TxElev1 = transmitter electrode 1 east, north and elevation (length units)
TxEast2, TxNorth2, TxElev2 = transmitter electrode 2 east, north and elevation (length units)
RxEast1, RxNorth1, RxElev1 = receiver electrode 1 east, north and elevation (length units)
RxEast2, RxNorth2, RxElev2 = receiver electrode 2 east, north and elevation (length units)

TDAVGW cac time-series export to *TS##.csv files

When processing GDP-32[24] cac files time-series data, *Tdavgw*'s cac processing control dialog includes an option to export the time series data to csv files, one file for each time-series block up to a maximum of 99 files. The export csv files use the naming convention *TS##.csv, where * is replaced with the name of the input cac file, and ## represent time-series blocks indexed from 01 to 99.

The time series files include a time column with millisecond values followed by one column for each GDP channel with time-series values in volts.

Partial listing of time-series *TS##.csv file:
```
Time_msec,Ch01_Ref_volts,Ch02_Ex_volts,Ch03_Ex_volts,Ch04_Ex_volts,Ch05_Ex_volts, . . .
Line.Name=2000, Tx.Stn=350, Rx.Stn=550, Gdp.Blk=1, Gdp.Time=15:19:02.887, TS.NWaveform=8192, . . .
0.0000E+00,1.58548E-04,-1.06916E-05,-2.60212E-03,6.46122E-03,-3.53001E-03,3.12254E-03,-1.27573E-03
9.7656E-01,-8.03471E-04,-2.60398E-05,-2.58919E-03,6.48413E-03,-3.49984E-03,3.13982E-03,-1.26585E-03
1.9531E+00,-1.36566E-02,-6.01038E-04,-2.78324E-03,6.35225E-03,-3.59729E-03,3.11028E-03,-1.27394E-03
2.9297E+00,-4.74930E-02,-2.29761E-03,-3.43349E-03,5.87326E-03,-4.00990E-03,2.97185E-03,-1.33865E-03
3.9062E+00,-8.29947E-02,-4.32644E-03,-4.26102E-03,5.24972E-03,-4.56933E-03,2.78793E-03,-1.42477E-03
4.8828E+00,-1.11229E-01,-6.11596E-03,-5.02765E-03,4.63519E-03,-5.14261E-03,2.59034E-03,-1.52383E-03
5.8594E+00,-1.28371E-01,-7.36553E-03,-5.60328E-03,4.14476E-03,-5.62742E-03,2.41533E-03,-1.61540E-03
```

The first record holds column labels. Time values are in milliseconds relative to the beginning of the time series, followed by a column of time-series data for each channel in volts. The second record records metadata in a keyword=value format, followed by all of the numeric data.

Meta data keywords:
```
Line.Name       = line name (text string)
Tx.Stn          = transmitter station number (float)
Rx.Stn          = receiver location station number (float)
Gdp.Blk         = GDP block number (integer)
Gdp.Time        = GDP time (hh:mm:ss.sss)
TS.NWaveform    = number of points per Tx cycle (integer)
TS.NCycle       = number of Tx waveform cycles in time series (integer)
TS.NPnt         = number of time-series points (integer)
```

TDAVGW cac stacked-data export to *Stack.csv file

When processing GDP-32[24] cac files time-series or stack data, *Tdavgw's* cac processing control dialog includes an option to export the rectified stacks to a csv file. The default *Stack.csv export file name can be modified in the *.cac processing control dialog.

Partial listing of rectified stack *Stack.csv file:
```
Time_msec,Stack.Volts,Tx.X1,Rx.X1,Plt.X,Plt.Y,Plt.Z,Rx.Cmp,Gdp.Chn,Gdp.Blk,Gdp.Time,TS.NCycle
0.0000E+00,7.04080E-06,200,1,112.75,1551150,-1551150.004,Ref,1,1,15:19:02.887,8
9.7656E-01,-9.15118E-04,200,1,112.75,1551150,-1551150.004,Ref,1,1,15:19:02.887,8
1.9531E+00,-1.36212E-02,200,1,112.75,1551150,-1551150.004,Ref,1,1,15:19:02.887,8
2.9297E+00,-4.70314E-02,200,1,112.75,1551150,-1551150.004,Ref,1,1,15:19:02.887,8
3.9062E+00,-8.18045E-02,200,1,112.75,1551150,-1551150.004,Ref,1,1,15:19:02.887,8
4.8828E+00,-1.10018E-01,200,1,112.75,1551150,-1551150.004,Ref,1,1,15:19:02.887,8
5.8594E+00,-1.28292E-01,200,1,112.75,1551150,-1551150.004,Ref,1,1,15:19:02.887,8
```

The first record holds column labels followed by numeric data for all of the rectified stacks.

Rectified-stack-data csv file columns:

Time_msec	= time from beginning of rectified stack (msec)
Stack.Volts	= rectified stack data (volts)
Tx.X1	= along-line X coordinate of first Tx electrode (GDP stn #)
Rx.X1	= along-line X coordinate of first Rx electrode (GDP stn #)
Plt.X	= pseudosection/volume plot point along-line X (GDP stn #)
Plt.Y	= pseudosection/volume plot point across-line Y (GDP stn #)
Plt.Z	= pseudosection/volume plot point depth Z (GDP stn #)
Rx.Cmp	= EM component (Ex,Ey,Ez,Hx,Hy,Hz,Ref)
Gdp.Chn	= GDP channel index
Gdp.Blk	= GDP block number
Gdp.Time	= GDP time (hh:mm:ss.sss)
TS.NCycle	= number of Tx waveform cycles averaged in rectified stack

If *Tdavgw* has recalculated stacks from times series data, there is an additional column of
Stack.StdDev   = stacked data standard deviation (volts)