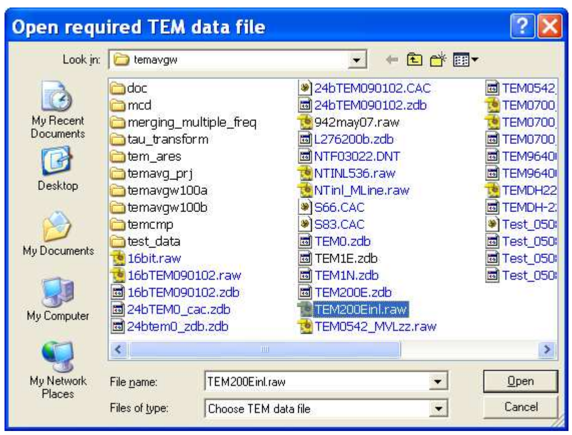
TEMAVGW Documentation

updated 28/Dec/10 by Scott MacInnes

Temavgw is a TEM data averaging and quality-control utility program that reads field data from Zonge GDP raw, dnt, cac, or zbl files. The program includes a Review Data option for interactive quality control while viewing the data in pseudosection or transient curve plots. ***Temavgw*** saves averaged data in an "avg" file. Unaveraged data are saved in a "zdb" ("Zonge data base") file, which ***temavgw*** can read 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.} ***Temavgw*** also includes an option to save averaged data in Australian AMIRA-format files, a format that is suitable for a broad range of TEM modeling software. ***Temavgw*** 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 TEM data files.

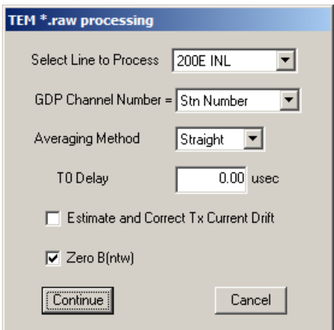
Start ***temavgw*** by running it from the command line, the windows start menu or with a short cut icon. ***Temavgw*** first reads default processing control parameters from temavgw.ini. It will then open a full size window and show a file selection dialog. Select an appropriate ***raw***, ***dnt***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

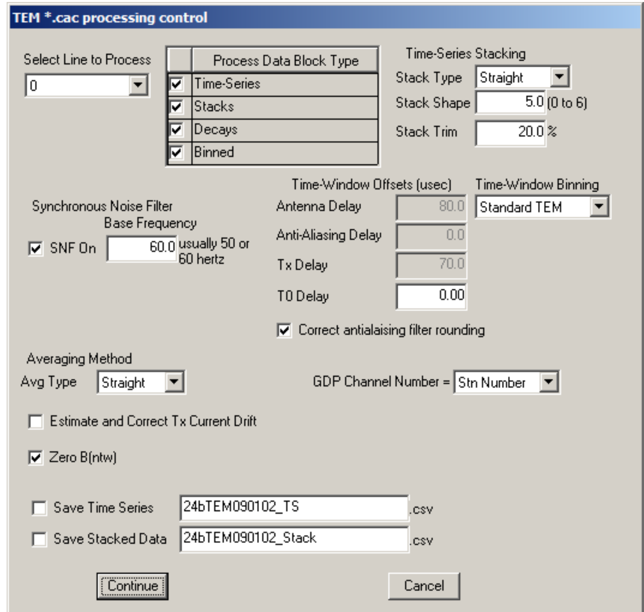
Open File Dialog



After getting an input data file name, ***temavgw*** searches for an optional ***mde*** file with the same file-name stem as the input TEM data file. If it finds a mde file with the right name, it reads it and updates default survey configuration and processing control values based on the mde file contents. It then 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





Temavgw shows either a raw or cac processing control dialog, depending upon the input file type. Processing control for TEM data input from raw, dnt or zdb files is similar because the data are always binned transients. Processing control for cac files is more complex, because the binary cac files can hold TEM data as time-series, stacks, decays or binned transients. All processing control dialogs include a field showing a list of line numbers present in the input data set. Only one line at a time can be processed, but the **Select Line to Process** field allows input from raw or cac files holding data for multiple lines.

GDP Channel Number options Stn Number, Stn # Offset, N-spacing or Antenna # set the interpretation of the GDP channel number field. With option Stn Number, ***temavgw*** interprets the GDP channel number as the station number at the location of the receiver coil or loop center. For Stn # offset, the GDP channel number is interpreted as an offset which is to be added to the GDP menu 4 Rx field value to calculate a Rx loop station number. Option n-spacing, multiplies the GDP channel number by the GDP dipole length value to get an offset that is added to the GDP menu 4 Rx value. Option Antenna # interprets 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 ***temavgw***, to allow the processing of B-field data acquired with specialized antennas.

Averaging Method 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.

T0 Delay may be used to shift the transient time origin from the time delay values (AntDelay and TxDelay) set during data acquisition. The flexibility of the GDP data acquisition system opens the door for operator error and estimating the correct receiver antenna delay or transmitter turn-off ramp delay can be tricky. The GDP uses AntDelay and TxDelay values to establish the transient time origin so that time=0 at the end of the transmitter turn-off ramp. If the time delay values used during data acquisition are inaccurate, T0 Delay can be used to shift the time origin. A positive T0 Delay value (in units of μsec) increases the amount of delay time between the Tx turn-off command and the transient time origin. Since the time origin is moving forward in time, while all of the time-window centers are fixed at the times used during data acquisition, a positive T0 Delay value decreases the time-window center values in ***temavgw*** output files.

Estimate and Correct Tx Current Drift check box triggers calculations that estimate how much the Tx current is drifting during data acquisition. It is an option than can be helpful when battery powered transmitters start to fade during TEM data acquisition.

When B-sensor (fluxgate) data in an input raw file are flagged by a $Ch.Sensor keyword record (see raw file documentation for more information on raw file keywords), check box **Zero B(ntw)** subtracts B magnitude for the last time window from every B value in the transient. Shifting the transient so that B(t) is 0 at the last time window removes any offsets due to fluxgate magnetometer drift. Zero B(ntw) has no effect on generic dB/dt data measured with a receiver loop or inductive coil.

The extra fields in the cac processing control dialog allow processing of time series, stack or decay data. Time series data are a continuous series of A/D samples for a time segment that extends across multiple transmitter waveform cycles. Stacked data hold a segment of A/D samples that have been stacked and rectified. Decays hold stacked and rectified time-series segments that start after the end of Tx turn-off and end just before the beginning of the next Tx current pulse. Transient data are decay segments that have been binned, so that groups of A/D samples are grouped into logarithmically spaced time windows. The **Process Data Block Type** check boxes allow selection of which cac data types should be processed. If a data type is checked it will be included in the ***temavgw*** data processing. Usually, all of the data block types are checked so that all available data gets processed.

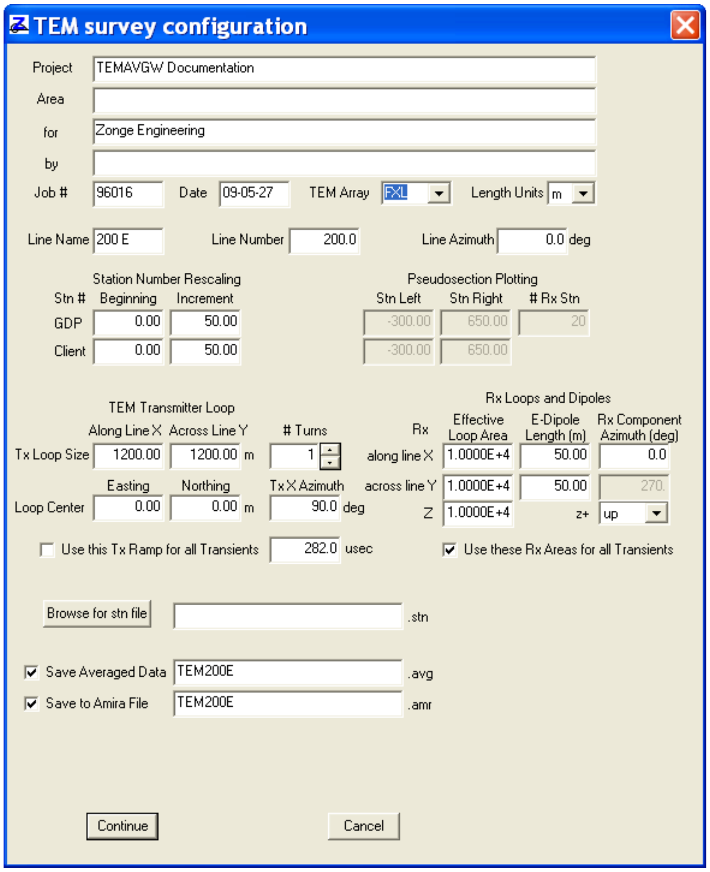
An advantage of acquiring time-series data is the ability to use more sophisticated (and computationally demanding) processing methods during post-acquisition data processing. **Time-series Stacking** has options for **Stack Type**, which can be Straight, Taper or Robust. Straight stacking is the most efficient method when background EM noise is completely random, so that the noise has no pattern. Tapered stacking is helpful in the presence of synchronous noise, like power line noise, that has a steady pattern. **Stack Shape** controls the taper shape. It controls the shape of a Kaiser taper and can vary from 0 (equal to a straight stack) to 5 (equal to a Gaussian shaped taper which is near optimal for 32-bit floating point values). Tapered stacking is more effective if it is applied over a long time series that spans a number of transmitter waveform cycles. **Stack Trim** is used with robust stacking and indicates the percentage of rejected outliers. It can vary from 0 to 49 percent. Stack Trim = 0 is equivalent to a straight stack, while 49% selects a median value. Intermediate stack trim values configure robust stacking so that it's an alpha-trimmed mean. Robust stack is most applicable when there are intermittent background noise bursts, like lighting spherics.

Time Window Offset parameters are used to calculate the time offset between the instant when the GDP signals the Tx to turn current off and end of Tx current turn-off in the digitized data stream. **Antenna Delay** indicates the time delay in μsec caused by Rx loop or coil pre-amps in TEM coils. **Anti-Aliasing Delay** is the time delay in μsec caused by the GDP's anti-aliasing filters. **Tx Ramp Delay** is the time required to turn Tx loop current off. It is generally a function of transmitter technology, Tx current, Tx loop size and number of Tx loop turns. Checkbox **Correct anti-alias filter rounding** controls an option to reduce rounding of the early-time transient due to the low-pass characteristics of 24-bit A/D chip's digital filter. Removing the early-time rounding caused by the 24-bit anti-aliasing filter improves STEMINV inversion results so the "Correct anti-aliasing filter rounding" checkbox should usually be left checked.

Time-Window Binning controls the grouping of uniformly spaced A/D sample values into the logarithmically spaced time windows of a "binned" TEM transient.

After the field data are read from the input file and after preliminary processing, ***temavgw*** shows a TEM 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.

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

- FXL indicates fixed-loop, with a single Tx loop at a fixed location and multiple receiver loop positions along line or down hole.
- Moving-Loop (MVL) indicates that both the Tx and Rx loops move along the survey line and that should interpret the GDP Tx field as a station number in the center of the Tx loop and the GDP Rx field as a station number in the center of the Rx loop or coil.
- In-Loop (INL) is a specialization of MVL, where the Rx coil is centered within a horizontal Tx loop.
- Coincident Loop (COL) means that the Tx and Rx loops had the same size and location. GDP Rx values are interpreted as Tx & Rx loop center for COL surveys.
- LOTEM (LOT) implies time-domain measurements of E and H fields excited by a grounded bipole Tx.
- The Continuous Recording (CNT) option is used to collect a stream of TEM transients without stacking, and is used with cart- or sled-mounted systems for environmental UXO mapping.

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.

Line Azimuth is a line bearing indicating the direction of increasing station numbers in degrees east of north.

It is not uncommon to decide that the station number system should be changed after data acquisition is complete. ***Temavgw*** 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 **Client 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 Client station numbers. Station number increments must be non-zero, but negative increment values can be entered to reverse station number order.

Transmitter Loop Size is used to specify the size of rectangular horizontal loops. **Loop Center** indicates the Tx loop center location for fixed-loop surveys. Fields are provided to specify the number of transmitter loop turns and the Tx X-azimuth in degrees east of north.

Fields in the center right of the survey configuration dialog allow control of receiver loop or dipole parameters. By convention Ex and Hx are the along-line EM components, Ey and Hy are across-line components and Hz is a vertical component. TEM receiver loops or coils are usually described sufficiently by their **Effective Surface Area** (in $meters^2$). A principal property of electric field dipoles is length. The EM X component azimuth is usually the same as line azimuth, but it may be rotated in some circumstances. The TEM convention is to use a right-handed (x,y,z) coordinate system with z positive up, but the survey configuration dialog allows specification of z positive down.

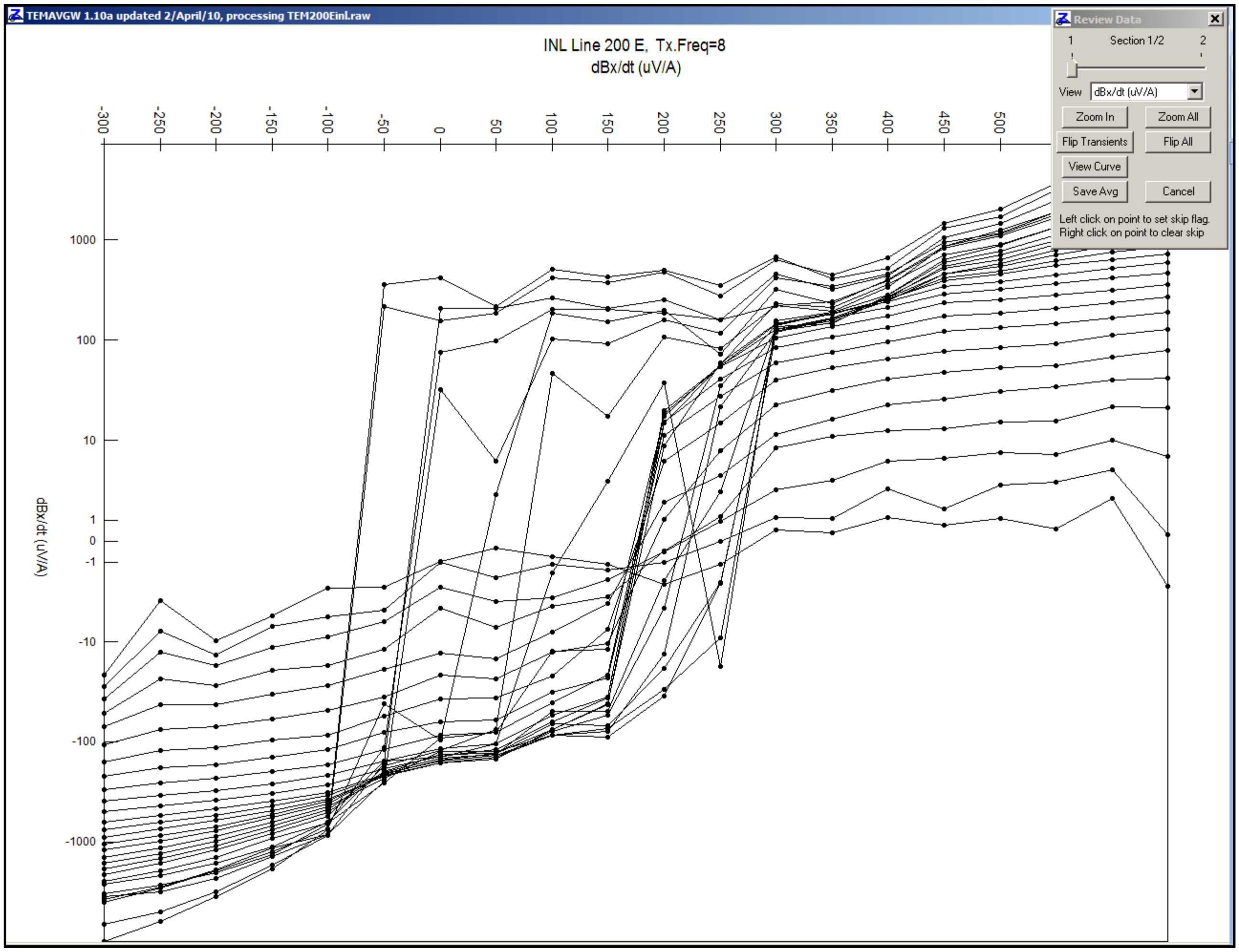
Check boxes can be toggled to override input data file transmitter turn-off ramp or receiver loop areas with the values specified in the TEM survey configuration dialog.

The **Browse for Stn File** button can be used to bring up an Open File dialog to select a stn file and import station coordinates. By default, ***temavgw*** looks for a stn file with the same file-name stem as the input data file, but a different file name can be selected for stn file input. Station files hold tabular data with columns of station, east, north and elevation. If a stn file is selected for input, the stn file tables are interpolated on station number to recover geographic coordinates for each TEM data station.

Temavgw saves averaged data to the Zonge avg file format by default. But it can also write averaged data to the Australian AMIRA format. Check boxes can be used to turn avg and amira format output on or off.

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

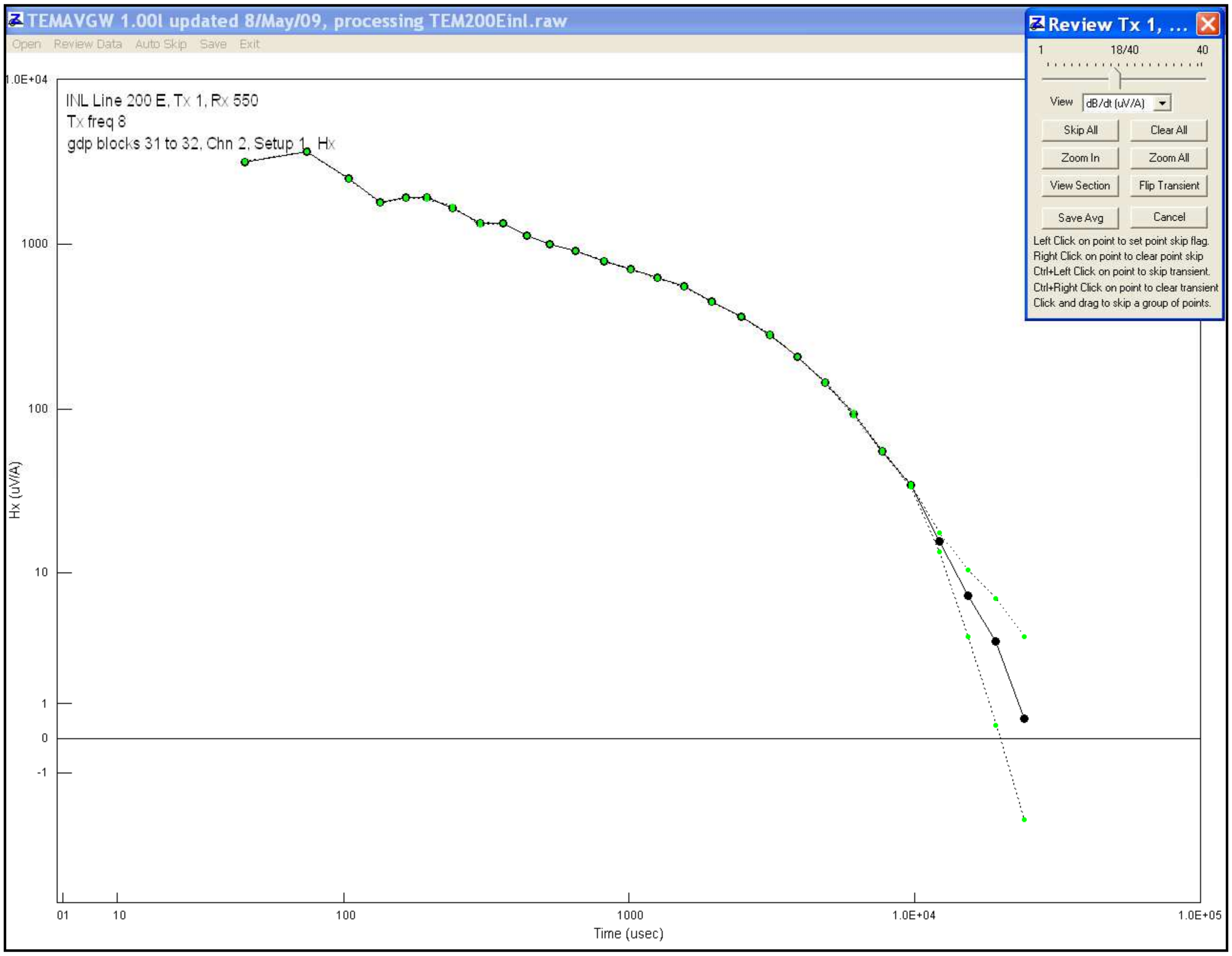


Review Data shows the TEM data as a pseudosection with profile lines tracing transient amplitudes for each time window along the length of the line. The horizontal axis is client station number and the vertical axis is asinh(dB/dt) or asinh(B). Small black circles indicate data points that have not been skipped, while data points skipped flagged as bad are indicated with a small gray circle. A data point can be skipped by pressing the left mouse button while the mouse cursor is over the bad data point, or unskipped with a right click.

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. The **View** field shows a list of data types available for display, usually either dB/dt or B(t). To see details in long lines, as for this sample data set, **Zoom In** allows magnification of pseudosection details. ***Temavgw*** 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. ***Temavgw*** 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. 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 Transients** button, which will change the mouse cursor to a double ended arrow, ↔. A left click anchors one corner of a rubber band box, and ***temavgw*** 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. Clicking on the **Flip All** button will reverse the polarity of every transient in the displayed transient.

Clicking on the **View Curve** button changes the mouse cursor to a green square. A left click with the square over a data selects a station to show as a transient plot.



Transient plots show TEM data for one component at one station. The horizontal axis is asinh(time) and the vertical axis is asinh(dB/dt or B). Transient plots provide a more detailed view of the data and allow quality control skipping or unskipping of individual repeat 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 plot control dialog for transient curve data review allows movement between stations via a track bar. If the mouse cursor is in the plot window, the display can be advanced from station to station by pressing the right arrow or the N (N for Next) key. The transient curve plot can be moved to a previous station be pressing the left arrow or the P (P for Previous) key. The **Skip All** button skips the entire transient, while **Clear All** clears the entire transient.

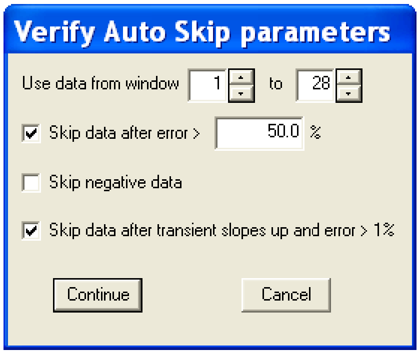
The polarity of the displayed transient can be reversed with the **Flip Transient** button. **View Section** returns Review Data to the pseudosection display.

Pressing **Save Avg** while in either the pseudosection or transient curve plot option saves the current data set to zdb and avg files, and then closes Review Data. Zdb files hold unaveraged transient data in ASCII files and preserve the skip flag pattern. Zdb files can be reopened to revisit data skipping decisions, or to complete an unfinished data review effort. Avg files hold averaged transient data suitable for modeling or final data plots. Zdb and avg files use the same ASCII file format.

Main Menu Options

After the initial data review is finished ***temavgw*** 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 an option for automatically setting skip flags. **Save** saves the current data to zdb or avg files, while **Exit** closes ***temavgw***.

Auto Skip Dialog



The Auto Skip menu option pops up a dialog showing auto skip control parameters. The **Use data from window** ?? **to** ?? fields are useful if a fixed range of time windows should be skipped. If for example, transmitter turn-off has saturated the first 2 windows, set the lower index to 3 and time windows 1 and 2 will be skipped on all transients. If the **Skip data after error is > ??%** check box is checked, then all transient values after the first point with an error level greater than the error-level floor will be skipped. Checking **Skip negative data** will skip all data with transient values less than or equal to 0. It is useful for dropping negative early-time out-of-loop data before inversion with ***steminv***. **Skip data after transient slopes up and error > 1%** is a combination filter that skips all data after the first up-sloping transient segment. It also checks for data error > 1%, so that up-sloping early-time data (as when the Rx is outside the Tx loop) will not set skip flags over the entire transient. Click on the **Continue** button to apply the Auto Skip filters or click on **Cancel** to return to the main menu without modifying the data. Note that using Auto Skip clears all previous skip flag settings, including any that have been set manually in Review Data.

File Formats

Temagvw.ini: survey configuration and processing control default values

Temavgw.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 `keyword=values(s)` format. Keywords usually have the form `group.variable`, with a parameter-group name separated from a variable name by a period. There may multiple comma-separated values after some keywords. `TWin.BinOffset` and `TWin.BinWidth` 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 ***temavgw*** is reading ***ini*** files.

Listing of sample ***temavgw.ini***

```
--------------------------------------------------------------------------------
$Tx.Ramp=282 usec
$Rx.Area=1.0000E+4,1.0000E+4,1.0000E+4 m^2
$Rx.Length=50,50 m
$Unit.Length=m
$TEMAVGW:Process.TS=Yes
$TEMAVGW:Process.STACK=Yes
$TEMAVGW:Process.DECAY=Yes
$TEMAVGW:Process.BINNED=Yes
$TEMAVGW:Stack.Type=Straight
$TEMAVGW:Stack.Taper=5
$TEMAVGW:Stack.Trim=20
$TEMAVGW:Stack.SNFOn=Yes
$TEMAVGW:Stack.SNFreq=60
$TEMAVGW:Ch.NumberType=Stn Number,Stn Number
$TEMAVGW:AD24Rounding = Yes
$TEMAVGW:TxDrift = No
$TEMAVGW:ZeroBntw = Yes
$TEMAVGW:Avg.Type=Straight
$File.StackToCsv=No
$File.TemToAvg=No
$File.TemToAmr=Yes
$TWin.Name=Standard TEM
$TWin.Num=49
$TWin.BinOffset=0,1,2,3,4,5,6,7,9,11,14,17, . . .
$TWin.BinWidth=1,1,1,1,1,1,1,2,2,3,3,4,6,7, . . .
--------------------------------------------------------------------------------
```

Survey configuration

`Tx.Ramp`	= float, transmitter turn-off ramp duration (usec)
`Rx.Area`	= float, x,y,z receiver antenna effective areaS (m^2). m^2 is used for loop areas since, for example, the TEM3 coil always has Rx.Area=10^4 m^2 even if user wants to flag coordinate units or Tx lengths as feet by setting `Survey.LengthUnits=feet`
`Rx.Length`	= float, along-line and across-line electric field dipole length (m)
`Unit.Length`	= enumeration, {m,ft} used to specify dipole lengths and loop sizes

Processing control

`TEMAVGW:Process.TS`	= boolean, yes=process cac file time series data
`TEMAVGW:Process.STACK`	= boolean, yes=process cac file stack data
`TEMAVGW:Process.DECAY`	= boolean, yes=process cac file decay data
`TEMAVGW:Process.BINNED`	= boolean, yes=process cac file binned transient data
`TEMAVGW:Stack.Type`	= enumeration, {Straight,Taper,Robust}
`TEMAVGW:Stack.Shape`	= float, Kaiser window taper (0=rectangular to 5=Gaussian)

```
TEMAVGW:Stack.SNFOn      = boolean, yes=apply synchronous noise filter to ts, stack or decay data
TEMAVGW:Stack.SNFreq     = float, synchronous noise frequency (hertz)
TEMAVGW:Ch.NumberType    =  enumeration, {Stn Number,   Stn # Offset, N-Spacing , Antenna #}
TEMAVGW:AD24Rounding     =  boolean, yes=correct early-time rounding due to 24-bit A/D digital low-pass filter
TEMAVGW:TxDrift          =  boolean, yes=estimate battery-powered transmitter current drift
TEMAVGW:ZeroBntw         =  boolean, yes=offset B(t) transient so last time-window's B value is zero
TEMAVGW:Avg.Type         = enumeration, repeat averaging type {Straight,Robust}
```

File output

```
File.StackToCSV  = boolean, yes=save cac stack data to csv file
File.TemToAvg    = boolean, yes=save averaged transients to Zonge avg format
File.TemToAmr    = boolean, yes=save averaged transients to Amira format
```

Time-window binning

```
TWin.Name        = string, label used to identify each time-window bin set (up to 32 char)
TWin.Num         = integer, # of time-windows bins (1 to 49)
TWin.BinOffset   = integer list, 0-based offsets to beginning of time-window bins
TWin.BinWidth    = integer list, time-window A/D bin sizes
```

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

Temavgw looks for a 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 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, zdb and avg files. Programs ignore unknown keywords during input, so new keywords may be introduced without breaking old software.

TEM MDE file listing

```
-------------------------------------------------------------------------------
$ Job.Name = "STEMINV Demo"
$ Job.Area = ""
$ Job.For = "Zonge Engineering"
$ Job.By = ""
$ Job.Number = "9200"
$ Job.Date = Sep 93
$ Survey.Array = INL
$ Line.Name = "0E"
$ Line.Number = 0
$ Line.Azimuth = -18
$ Rx.HPR = -18,0,0
$ Unit.Length = ft
$ Unit.Time = msec
$ Unit.dBdt = uV/A
$ Stn.GdpBeg = 84
$ Stn.GdpInc = 2
$ Stn.Beg = 84
```

```
$ Stn.Inc = 2
$ Tx.Center = 0,0,0 ft
$ Tx.Length = 400,400 ft
$ Tx.Area = 1.4884E+04 m^2
$ Stn.Left=84.0
$ Stn.Right=122.0
--------------------------------------------------------------------------------
```

Keywords Stn.GdpBeg, Stn.GdpInc, Stn.Beg and Stn.Inc can be used to shift and scale the GDP station numbers used in RAW, AVG and Z 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.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

`Unit.Length` = enumeration, length units {`m,ft`} (legacy alias `Units`)
`Tx.Length` = float, Tx loop side length (length units) (legacy alias `TxDX, TxDY, TxLength`)
`Tx.Area` = float, Tx loop area (m^2) (legacy alias `TxArea`)
`Tx.Stn` = float, MVL Tx center stn # or FXL Tx ID
`Tx.Center` = floats, Tx center east,north,elevation (length units) (legacy alias `TxCX,TxCY`)
`Tx.Ramp` = float, Tx turn-off ramp extent (μsec)
`Rx.HPR` = floats, Rx cmp orientation (heading,pitch,roll deg)
`Rx.Area` = float, Rx coil effective area (m^2)

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. STEMINV 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.

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. Heading, Pitch and Roll columns are sometimes used when TEM systems are mounted on a cart for continuous NanoTEM (CNT) surveys.

Listing of typical stn file:

```
-------------------------------------------------------------------------------
Station,   Easting, Northing,Elevation
   84.0,    5595,    5887,    2030
   86.0,    5533,    6079,    2035
   88.0,    5471,    6271,    2040
   90.0,    5410,    6463,    2040
   92.0,    5348,    6654,    2045
   94.0,    5287,    6846,    2050
   96.0,    5225,    7038,    2055
   98.0,    5164,    7230,    2060
  100.0,    5102,    7422,    2060
  102.0,    5041,    7613,    2070
  104.0,    4979,    7805,    2070
  106.0,    4918,    7997,    2080
  108.0,    4856,    8189,    2080
  110.0,    4795,    8381,    2090
  112.0,    4733,    8572,    2090
  114.0,    4671,    8764,    2100
  116.0,    4610,    8956,    2100
  118.0,    4548,    9148,    2110
  120.0,    4487,    9340,    2120
  122.0,    4425,    9531,    2120
-------------------------------------------------------------------------------
```

STN file column definitions:

`Station` - client station numbers
`Easting` - grid east (m or ft, length units specified when STEMINV creates a new std file)
`Northing` - grid north (m or ft)
`Elevation` - elevation (m or ft)
`Heading` - X component azimuth (deg E of N)
`Pitch` - X component pitch (deg up from horizontal)
`Roll` - Z component rotation about X axis (deg clockwise)

GDP-32 ASCII TEM 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 TEM Header Block Sample

```
0059
TEM 0843 2008-12-29 13:43:48 12.6v INL  36.3%  25.0 DegC
OPER       JBF TX ID  ZT59 A-SP     61  M
JOB  test LINE        1 N  SPREAD  1
 50% RxM  10000 TxX   61 TxY   61 #T  1
Tx Delay   70 Antenna Delay   80 Alias OUT
Robust None
1  DiffAmp Notch 60,3-5,9  S/N  466 Passed 1.00000
.     .     .     .     .     .     .     .     .
```

GDP-32 TEM Header block structure

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

Line 2
   Survey Type             -   ex:"TEM "  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
   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

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
   Duty Cycle              %  50 or 100%  Character      3       3
   Rx Effective Area      m2   1 - 999999  Integer       6      15
   Tx Length (X)          m    1 - 9999   Integer        4      24
   Tx Length (Y)          m    1 - 9999   Integer        4      33
   Tx Loop turns          #    1 - 99     Integer        2      39

Line 6
   Tx Turnoff Delay      µs    1 - 9999   Integer        4      13
   Rx Antenna Delay      µs    1 - 9999   Integer        4      32
   Alias Filter           -   ex:"OUT"    Character      3      42


Line 8
   Robust Stacking            On/Off      Character      4      11
```

Line 8+ hardware inventory information, one line for each channel.

GDP-32 TEM Data Block Sample

```
0060
TEM 0843 2008-12-29 13:45:02 12.5v INL  36.3%  25.0 DegC
Tx        1 Rx        2 N OUT
  32 Hz    256 Cyc Tx Curr   1.5 183.1u   26u  30.52u
1  Hz      1  40.114u  1.551m  28.92   0600 3.491u   -9.59     0
2  Hz      1  48.014u  1.551m  25.66   0600 1.294u    5.86     0
3  Hz      1  38.257u  1.551m  29.85   0600 4.675u    1.30     0
  Wn     Mag 1    Mag 2    Mag 3    Rho 1    Rho 2    Rho 3
 33.11u  20.679m  19.815m  20.341m  274.01   281.92   277.04
 63.62u  12.368m  11.860m  12.083m  129.94   133.62   131.97
 94.14u  8.0912m  7.7930m  7.9932m  89.739   92.014   90.471
 124.7u  5.6425m  5.5030m  5.5315m  71.467   72.669   72.420
 155.2u  4.1481m  4.0074m  4.0454m  60.908   62.327   61.935
 185.7u  3.1180m  3.0481m  3.1051m  54.622   55.455   54.774
 230.9u  2.1626m  2.1464m  2.1396m  48.490   48.733   48.836
```

Line 5 is repeated for each channel measured.
Line 7 is repeated for each window measured (only 7 are shown here).

GDP-32 TEM data block structure

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

Line 2
   Survey Type            -   ex:"NTEM"  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
   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
   Tx value               #  ±0-9999999  Scaled         8      11
   Rx value               #  ±0-9999999  Scaled         8      23
   Notch Filter           -   ex:"OUT  " Character      5      31

Line 4
   Frequency             Hz   ex:"   1"  Scaled         4       4
   # Cycles/Stack         #   1 - 16384  Integer        5      14
   Tx Current          amps   0 - 99999  Scaled         6      32
   Sampling Delay         s              Scaled w/exp   7      39
   Alias Filter Delay     s              Scaled w/exp   5      45
   Sampling Interval      s              Scaled w/exp   7      53
```

GDP-32 TEM 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:"Hz"    Character     3       6
   Station Number         #  ±0 – 99.9   Scaled        6      11
   Magnitude            v/a              Scaled w/exp  8      20
   Reference Window Time  s              Scaled w/exp  7      29
   Resistivity @ RefWin  Ωm              Scaled w/exp  7      37
   Gains/Attenuator    ###A   ex:"060O"  Character     4      42
   SEM                  v/a              Scaled w/exp  6      49
   Self Potential        mv              Scaled        7      57
   Contact Resistance     Ω              Scaled w/exp  6      64
```

The magnitude and resistivity table will have up to 31 lines of data. A column of Time is followed by a column of Magnitude (and Resistivity for particular loop types and components) for each channel.

The structure for the three types of columns are:

```
   Time                   s              Scaled w/exp  7
   Magnitude            v/a              Scaled w/exp  8
   Resistivity           Ωm              Scaled w/exp  8
```

A data block may be skipped by putting an ″x″ just to the right of the GDP program name, i.e. change:

```
NANO0618 2001-05-02 12:51:35 11.9v INL  20.5%  52.2 DegC
```

to

```
NANO0618x2001-05-02 12:51:35 11.9v INL  20.5%  52.2 DegC
```

Individual channels may be skipped by putting an ″x″ right after the channel number, i.e. change

```
1  Hz    128  1.4052m  6.751u      0   040O 4.742m    0.00     0
```

to

```
1 xHz    128  1.4052m  6.751u      0   040O 4.742m    0.00     0
```

The polarity of individual channels can be flipped by putting a ″-″ after the channel number, i.e. change

```
1  Hz    128  1.4052m  6.751u      0   040O 4.742m    0.00     0
```

to

```
1 -Hz    128  1.4052m  6.751u      0   040O 4.742m    0.00     0
```

Raw file survey configuration keywords

Starting with v1.00a ***temavgw*** will read a limited set of survey configuration keywords directly from raw files, so that conventional TEM array conventions can be extended to 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 from each other by commas. 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.

TEM Sensor Type

By default, ***temavgw*** assumes that TEM measurements are made with a loop or coil, so that transient values are in dB/dt units of V/A. To accommodate B-field TEM measurements, typically made with a fluxgate magnetometer, a $Ch.Sensor keyword record can be inserted into the raw file after the first header block. For example the keyword record
$Ch.Sensor=B,dBdt,dBdt
specifies a B-field measurement on GDP32 channel 1 (in uT/A), and conventional dB/dt measurements on GDP32 analog input channels 2 and 3. ***Temavgw*** expects raw file dB/dt values in V/A and B-field values scaled to µT/A.
Given dB/dt in V/A, ***temavgw*** multiplies the data by 1000 to get uV/A and calculates B(t) from dB/dt by integrating dBdt /RxArea with respect to time in microseconds to get B in pT/A. Give a channel of B-sensor values in µT/A, ***temavgw*** multiplies the data by 1000 to get pT/A, and differentiates dBdt*RxArea to get dB/dt in uV/A. So each channel will produce both dB/dt and B(t) transients starting from either a dB/dt or a B(t) sensor.

TEM Data Scaling

A second raw file keyword implemented in ***temavgw*** v1.10a is $Ch.Scale, which enables arbitrary rescaling of transient values. The keyword record
$Ch.Scale=10,1,-1
will increase GDP32 channel 1 transient values by a factor of 10, sometimes needed to scale transient values from a fluxgate sensor to µT/A. The $Ch.Scale example leaves channel 2 values unchanged and reverses the polarity of GDP32 channel 3 transient values.

TEM Transmitter Current

GDP32 raw files transient values are normalized by transmitter current, dB/dt transient values are in volts/amp.
The GDP32 operator enters the transmitter current values by hand, and occasionally enters the wrong value. To correct a transmitter current error, use the raw file keyword $Tx.Amp. The keyword record
$Tx.Amp=10.1
will correct subsequent data to use a transmitter current value of 10.1 amps, even if some other transmitter current value was originally entered in the GDP32. ***temavgw*** will multiply transient values by the original incorrect transmitter current and then divide by the $Tx.Amp value to calculate corrected transient values in V/A or for B-field sensors µT/A. The transmitter current correction feature can be turned off with at $Tx.Amp keyword record using a blank value, i.e.
$Tx.Amp=
turns off the transmitter current corrections and ***temavgw*** returns to using the original raw file values.

cac **Files: binary 24-bit TEM data**

The GDP-24bit saves 24-bit time-series, stack, decay and or binned transient 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 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 equal to FFFF hex. This in conjunction with the bracketing record length field allows processing of a damaged cache. The record length fields are used to verify cache integrity and to manage cache positioning in the face of arbitrary data record contents.

The first 2 byte element in every core data record tells which type of data record has been stored (1=top, 4=navigation record, 16=metadata, 514=time series data). The contents of individual records vary by record type.

cac file metadata records, record type =16

The internal body of metadata records is ascii text holding comma separated value `keyword,value` sub-records. Subrecord endpoints are delineated by a carriage-return character (\n). 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. Array parameter values are saved as a string of comma-separated fields.

Programs should skip sub-records with novel or unknown keywords, allowing future extension without breaking existing software.

Parameters are organized into groups, distinguished by a "`GroupName.`" prefix on the keyword. Each header type has a distinct collection of groups. In addition the groups associated with `Header.Type=Data` metadata records vary depending upon `Data.Kind` As a consequence, metadata record output for each of the `Header.Type` metadata records can be organized by a list of associated group names. (Having distinct group names also makes metadata keyword parsing more efficient, although that is a small point.)

Each metadata record starts with `Header.Type`, enumeration, {`Survey,Data,Calibrate`}
Group names associated with `Header.Type=Survey` are `Job, Survey, GDP, TX, RX, TS, CH`
Group names associated with `Header.Type=Calibrate` are to be implemented
Group names associated with `Header.Type=Data`
Data, Row, Col are present for all Data.Kind
  if `Data.Kind=Stack:      Stack`
  if `Data.Kind=Decay:      Stack, Decay`
  if `Data.Kind=Transient: Stack, TWin`
  if `Data.Kind=FFT: Stack, FFT`
  if `Data.Kind=FD:  Stack, FFT, FD`

```
------------------------------------------------------------------------
Header.Type = Survey
```

`Job.Name` = string, project name, added post acquisition
`Job.Area` = string, project name, added post acquisition

Job.By = string, client name, added post acquisition
Job.For = string, contractor name, added post acquisition
Job.Number = string, job number, currently included in GDP-32 menus

Survey.Type = enumeration, {LOTEM,TEM,CSAMT,NSAMT,CR,RPIP,TDIP}
Survey.AcqMethod = enumeration, {TimeSeries,Stack,Decay,Transient,FFT,FD}
- Stack = represents a stacked and rectified half-waveform with NWaveform/2 floats
- Decay = represents a segment of A/D points from a stack, made a distinct type because the GDP-32 can be gained up if on-time or turn-off waveform can be saturated.
- Transients = binned time-window decays: binned time-domain "Transients" for both TEM and TDIP
- FTT = FFT of a waveform with NWaveform/2 (real,imag) pairs = NWaveform floats
- FD = requency domain (real,imag) data extracted from an FFT, usually the 1,3,5,7,9th harmonics

Survey.Array = Survey.Type specific enumeration, array type labels
TEM => the GDP-32 is currently using {INL,FXL,MVL,COL,LOT,CNT}
Survey.Spread = string, spread label, not always interpretable as a number
Survey.sLine = string, line label, not always interpretable as a number although that can cause problems later (as in Oasis montaj)
Survey.fLine = float, line number, incremented by 1 when Continue pressed during DNT survey
Survey.LengthUnits = enumeration, {m,ft} used to specify dipole lengths and loop size.
Note that invariants like TEM Rx coil effective area (Rx.Area) are always in m^2, and GUI fields for editing those values can be labeled with a trailing m^2.
Job.Number = string, job number, serves as a key to Zonge record keeping.

GDP.SN = string, GDP serial number
GDP.TCardSN = string, timing card serial number
GDP.ADCardSN = string array, a/d card serial numbers
GDP.Operator = string, GDP-32 operator's name or initials

Tx.Freq = float, Tx repetition rate (hertz)
Tx.Duty = float, Tx waveform duty cycle, usually 50 or 100 (%)
Tx.Amp = float, Tx peak current in time domain
Tx.Stn = float, survey specific meaning
- FXL = Tx loop ID
- INL & MVL = often use Tx.Stn as stn at Tx loop center
- DNT/SNT = Loop_Array integer label
- CSAMT = Tx dipole ID
- CR/TDIP = lower electrode of Tx dipole or roving Tx electrode stn for PLDP

Tx.Type = enumeration, type of Tx in use {GGT, NT-20, ZT-20}
Tx.Turns = integer, # turns in Tx loop
Tx.LengthX = float, along line dimensions of Tx loop/dipole (LengthUnits)
Tx.LengthY = float, across line dimensions of Tx loop/dipole (LengthUnits)
- Tx.Area = Tx.LengthX* Tx.LengthY* Tx.Turns*(m/LengthUnit)^2 used in TEM apparent resistivity calculation (m^2)
- Tx.CenterX = float, along line location of Tx loop/dipole (LengthUnits)
- Tx.CenterY = float, across line location of Tx loop/dipole (LengthUnits)
- Tx.LengthX,[Tx.LengthY],Tx.CenterX,Tx.CenterY used for CSAMT & CR/TDIP grid survey description

Tx.Delay = float, transmitter turn-off ramp duration (μsec)
Tx.Shunt = float, current monitoring shunt resistance (ohms)

TS.ADFreq = float, a/d sampling frequency (hertz)
TS.DecFac = integer, decimation factor
TS.NCycle = number cycles/stack

```
TS.NWaveForm          = integer, number samples/waveform = TS.ADFreq/(TS.DecFac*Tx.Freq)
TS.NPnt               = integer, # points stored in subsequent data record

Rx.Stn                = float, Rx station number, survey-specific interpretation
   FXL,INL,MVL,COL    = GDP-32 location
   CSAMT              = GDP-32 or H-field coils location
   CR                 = lower electrode of Rx dipole with lowest n-spacing
   SNT                = offset table integer index
Rx.ASpace             = float, default dipole length (LengthUnits)
Rx.Area               = float, receiver antenna effective area (m^2)   TEM3 coil always has Rx.Area=10^4 m^2
                        even if user wants to flag coordinate units or Tx lengths as feet by setting
                        Survey.LengthUnits=feet
Rx.AntDelay           = float, antenna preamp delay (μsec)
Rx.AliasDelay         = float, anti-alias filter delay (μsec)

Ch.Factor             = float list, scale a/d counts to volts excluding gain or attenuator settings
                        (volts/ad_count)
Ch.Gain               = string list, gain stage settings as binary exponents
Ch.Number             = float list, cmp and survey-specific meaning
   E-field chn, lower electrode stn, dipole center location, n-spacing, n-spacing offset from Rx.Stn
   TEM H-field chn, Rx loop stn number, Rx loop offset wrt Rx.Stn
   FD  H-field chn, antenna number used to index AntCal table
Ch.ASpace             = float list, E-field dipole length for variable Aspace option,  defaults to Rx.ASpace
Ch.Area               = float list, TEM loop effective area (m^2)
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 (volts)
Ch.PreAmp             = float list, pre-amplifier gain as a scaling multiplier
Ch.CRes               = float list, contact resistance of each channel (ohms)

GDP.numCh             = integer, number of active channels in GDP-32
GDP.ch                = integer list, card slots of active channels
GDP.bat               = float, GDP-32 battery voltage (volts)
GDP.temp              = float, GDP-32 temperature (deg C)
GDP.humid             = float, GDP-32 relative humidity (%)
Stack.Type            = enumeration, {Straight,Tapered,Robust}
Stack.Taper           = float, 0=straight stack, 5=Gaussian, (0 to 6)
Stack.Trim            = float, % trimmed from robust stack (0 to 49)

-----------------------------------------------------------------------------------
```

TEM/LOTEM will have a data-header record in the generic metadata record format before each data record or data record series.

```
Header.Type           = Data

Data.Date0            = string, date when acquisition started (YY/MM/DD)
Data.Time0            = string, time when acquisition started (HH:MM:SS.SSSS)
Data.Kind             = enumeration {TimeSeries, Stack, Decay,Transient,FFT,FD} =
                        Survey.AcqMethod
Data.NumBlocks        = integer, # of data blocks contained in the following data record, allows for continuous
                        mode acquisition
Data.Skip             = enumeration, block skip flag, {yes,no}
```

`Row.num` = integer, number of rows in subsequent data block(s)

- `TimeSeries` => TS.NPnt
- `Stack & FFT` => TS.NWaveForm/2
- `Transient` => Window.Num
- `FD` => FD.num = number of harmonic frequencies saved from FFT

`Row.BlockFreq` = float, rate at which data blocks were collected (hertz).

`Col.num` = integer, GDP.numCh
`Col.Chn` = integer list, list of active GDP-32 channels = GDP.ch list
`Col.Label` = string list, formatted as `cmp.label`, i.e. `Ex.re,Ex.im,Ex.re,Ex.im,Ex.re,Ex.im`
`Col.Units` = enumeration list, data value units

`ADmag,ADreal,ADimag` for a/d counts before or after FFT, but without scaling by Ch.Factor, Ch.Gain or any calibrate values
= Phase phase values (radians)
= Vmag,Vreal,Vimag for TD or FD volts with Ch.Factor, Ch.gain, board calibrates applied (but not antcals)
= Emag, Ereal,Eimag for E-field values scaled to V/m after `/Survey.Aspace`
= Hmag, Hreal,Himag for H-field values scaled to nT by application of antcals
= AmpMag, AmpReal,AmpImag for Ref channel values scaled by Tx.Shunt

`Col.Skip` = boolean enumeration list, {`yes,no`} 1 = don't flip, added post-acquisition
`Col.Flip` = integer list, 1 = don't flip, -1 = flip polarity, added post-acquisition

Proposed keyword (if data record units are stored as a/d counts).

`Col.Scale` = float list, scaling factors for A/D counts to EM units taking into account Ch.Factor, Ch.Gain (& attenuator), notch filters. For frequency domain Col.Scale values come in real,imag pairs requiring 2*GDP.numCh*FFT.num or 2*GDP.numCh*FD.num values. Frequency-domain Col.Scale values include Board.cal and Antenna.cal scaling.

- `For GDP.cmp` = Ex,Ey,Ez; final units are V/m
- `For GDP.cmp` = Hx,Hy,Hz; final units are nT
- `For GDP.cmp` = Ref; final units are amps

`TWin.Name` = string, label used to identify each window set
i.e. shown in a GUI list field for user selection (<32 characters)
`TWin.num` = integer, # of time-windows bins... only present when Data.Kind = `Transient`
`TWin.BinWidth` = integer list, window bin sizes
Assumes contiguous windows to calculate each bin’s start-point offset.
`TWin.BinOffset` = integer list, 0-based offsets to beginning of time-window bins,
needed only if non-contiguous windows are allowed.
`TWin.Center` = float list, Geometric time-window centers with time = 0 at end of TxDelay (μsec)
`TWin.Width` = float list, time-window widths between first and last a/d sample used (μsec)

A bin based on one a/d sample has a TWin.Width of 0 μsec by legacy GDP-32 convention, so that TWin.Center = a/d sample time.

`Data.Kind = FFT`
`FFT.num` = integer, number of (real,imag) pairs produced by an FFT = NWaveform/2
`FTT.Width` = float, value representing FFT bin width (hertz)
`FFT.Width =` 2/(NWaveform*TxFreq)

`Data.Kind = FD` for frequency-domain
`FD.num` = integer, number of harmonic frequencies saved from the FFT
`FD.BinOffset` = integer list, 0-based offsets to FFT harmonic values

Will be 0, 2, 4, 6, 8 to recover 1,3,5,7&9^{th} harmonics from HACSAMT or CR data.
MT may use weighted average bins of FFT values to improve statistics versus log frequency.

`FD.BinWidth` = float list, for MT, number of FFT values averaged in MT frequency-domain "binning", assume FD.BinWidth[]=1 for controlled-source applications {`CSAMT,CR,RDIP`}.
`FD.Freq` = float list, arithmetic center frequencies (hertz) corresponding to FD.BinOffset,
`FD.Width` = float list, frequency-domain FFT bin widths (hertz)

Not needed except for NSAMT using weighted averages of FFT (real,imag) pairs (would be 0 width for single point samples if we used GDP-32 TEM conventions, otherwise, more rationally, would be `FFT.Width` for single point samples).

`Data.Kind = Stack`
`Stack.NCycle` = integer, number of stacks performed = TS.NCycle
`Stack.Type` = enumeration {`Straight,Taper,Robust`}

Tapered stacks generally overlap, typically by 50%
Use Bessel-Kaiser window for taper with equal data-point weighting.
Robust (alpha-trimmed mean) stacking requires sorting of NCycle*2 points NWaveform/2 times.

`Stack.Shape` = float, Kaiser window taper (0=rectangular to 5=Gaussian)
`Stack.Trim` = float, outlier rejection for robust stacking (0 to 49 % rejection)

`Decay.Points` = integer, number of sample points in decay
`Decay.Num` = integer, number of sample points in decay
`Decay.Offset` = integer, number of sample points dropped before start of Decay to skip over Rx.AntDelay + Rx.AliaDelay + Tx.Delay

cac **file data records, record type =514**

Survey.AcqMethod=TimeSeries

Time-series data are held in rectangular blocks of 4-byte binary integer values representing A/D levels which must be scaled to volts using the ChFactor and ChGain information from the proceeding metadata record. The time-series data block is equivalent to a NChn by NPnt array with the channel index varying the fastest.

Survey.AcqMethod=Stack

Stacked data are held in rectangular blocks of 4-byte floats representing stacked A/D levels. Values must be scaled to volts using the ChFactor and ChGain information from the proceeding metadata record. The time-series data block is equivalent to a NChn by NWaveform array with the channel index varying the fastest.

Survey.AcqMethod=Decay

"Decay" data are held in rectangular blocks of 4-byte floats representing stacked and rectified A/D levels over length of the off-time transient. Values must be scaled to volts using the ChFactor and ChGain information from the proceeding metadata record. The time-series data block is equivalent to a NChn by NPnt array with the channel index varying the fastest. Each segment is offset beyond the end of the Tx turn-off ramp based on AntDelay + AliasDelay + Tx.Delay.

Survey.AcqMethod=Transient

"Transient" data are held in rectangular blocks of 4-byte floats representing stacked, rectified and binned A/D levels. Values must be scaled to volts using the ChFactor and ChGain information from the proceeding metadata record. The time-series data block is equivalent to a NChn by NPnt array with the channel index varying the fastest. Each segment is offset beyond the end of the Tx turn-off ramp based on AntDelay + AliasDelay + Tx.Delay and time-window values indicate averages of groups of A/D samples.

DNT Files: Continuous or Dynamic NanoTEM binary transient data

A dnt extension on a file name indicates a CNT or DNT binary cache file. DNT files contain a mixture of 3 different record types, header records, pre/post data records, and data records. 1 header record precedes 1 pre-data record, which precedes 1 or more data records (typically 1000's more), followed by a single post data record. Thus a pre/post data records bracket a data record sequence. Data records are collected at the acquisition frequencies determined by user configuration of the continuous NanoTEM (CNT) program on the GDP-32.

Each record (no matter which record type) in the binary cache is stored in the following manner.

```
 ___ ___ _______       ________ ___      __ ____       ______
|___|___|_______ o o o _______|___|     |__|____  o o o  ______|  Any record
 \   \   \                        /        \
  \   \   \_______ Any record   /          \ record type
   \   \                       /
     \   \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 2 byte integer). The second element in every record is a 4-byte code equal to FFFF hex. This, in conjunction with the bracketing record length field, let’s us process a damaged cache. The record length fields are used to verify cache integrity (yes there is a 1 in 65535^2 chance of a miss-hit), and manage cache positioning in the face of arbitrary data record contents.

The first element in every core data record tells which type of data record has been stored (1=top, 2=header, 4=prep, 8=data, 16=pre, 32=post, 64=bottom). The contents of the individual record types are given by the data structures defined in this file.

A cache index can be built dynamically by skipping through the cache header records. A compact header list is then held in memory and can be used for direct random access into the binary cache.

HEADER RECORD (Record_Type = 2):

Variable	Storage Type	Description
Record_Length	2 byte short	record length (bytes)
Record_Flag	4 byte string	flag record beginning; each byte = FF hex or 255 decimal
Record_Type	2 byte short	2=header
nPanelSN	2 byte short	GDP panel serial number
nADcardSN[MAX_CHN]	2 byte short	A/D card serial numbers
nCardType[MAX_CHN]	2 byte short	14 or 16 bit A/D card indicator
nTMcardSN	2 byte short	timing card serial number
nTXcardSN	2 byte short	transmitter card serial number
nHdwCfg	2 byte short	hardware configuration index (link to cfg file)
nWinCfg	2 byte short	time-window A/D-point-table index (link to cfg file)
nQualRate	2 byte short	data records/quality record ratio
Rx_Location	4 byte float	Rx location from GDP menu 4
Tx_Location	4 byte float	Tx location from GDP menu 4
Tx_DX	2 byte short	Tx loop x width (m)
Tx_DY	2 byte short	Tx loop y width (m)
NTxTurn	2 byte short	# Tx loop turns
GDP_Operator	12 byte string	GDP operator initials
Job_Number	12 byte string	job number
Ch_Slot	3*2 byte short	channel's GDP slot number, -1=channel_off
Ch_Type[MAX_CHN]	3*2 byte short	0=channel_off, Hx=4,Hy=5,Hz=6,Ref=7,Rx=8,Ry=9
RxArea[MAX_CHN]	3*4 byte float	Rx loop effective area (m^2)
Ch_Gain[MAX_CHN]	3*4 byte float	channel gain (unitless ratio)

fCardFactor[MAX_CHN]	4 byte float	14 (1638.4) or 16 bit (4*1638.4) A/D scaling
TxDelay	4 byte float	Tx ramp time (# sample periods)
SampleRate	4 byte float	A/D sample rate (usec)
AntDelay	4 byte float	antenna delay (# sample periods)
AliasDelay	4 byte float	anti-alias filter delay (# sample periods)
DutyCycle	4 byte float	duty cycle % = 50 or 100
TxFreq	4 byte float	Tx frequency (hertz)
NCycles	2 byte short	# of stacked waveform pairs per data block
LineNumb	4 byte float	line number
LineDir	12 byte string	line direction (N,NE,E,SE,S,SW,W,NW)
Numb_Blk	4 byte long	# of data blocks governed by this header
StartBlk	4 byte long	block # of first data block in this data set
FPtoDataBlks	4 byte long	File pointer to first data block governed by this header
Repeat_Record_Length	2 byte short	Record length repeated to support processing of corrupted cache

PRE and POST RECORDS (Record_Type = 16 and 32):

Variable	**Storage Type**	**Description**
Record_Length	2 byte short	record length (bytes)
Record_Flag	4 byte string	flag record beginning; each byte = FF hex or 255 decimal
Record_Type	2 byte short	16=pre, 32=post
Tx_Current	4 byte float	Tx current (peak amplitude amps)
Tx_Battery_Voltage	4 byte float	Tx battery voltage (volts)
GDP_Temperature	4 byte float	GDP temperature (deg C)
GDP_Humidity	4 byte float	GDP humidity (%)
GDP_Battery_Voltage	4 byte float	GDP battery voltage (volts)
Repeat_Record_Length	2 byte short	Record length repeated

DATA RECORD (Record_Type = 8):

Variable	**Storage Type**	**Description**
Record_Length	2 byte short	record length (bytes)
Record_Flag	4 byte string	flag record beginning; each byte = FF hex or 255 decimal
Record_Type	2 byte short	8=data
Block_Number	4 byte long	block number
Sync	2 byte short	synchronization information
Time_Stamp	4 byte string	GDP time stamp (hours minutes seconds tenths)
Packed_Volts	2 byte short	companded transient magnitude values
Packed_Shift	4 byte short	companded magnitude shift
Repeat_Record_Length	2 byte short	Record length repeated

Structure of companded transient values

DNT dB/dt transient values are packed into data records with a custom compression format. Raw 32 bit (actually 24 bits at most) transient values are stored in a 16 bit field and a 4 bit shift field allowing us to keep the 16 most significant bits of any window. This introduces an error whose maximum value is 1/65535 or 0.003% of any stored value. This yields 4.7 significant figures (base 10) of precision. Each 16-bit window value has bit 15 as the sign bit and bits 0-14 as a positive magnitude. Note: 2's complement form has been removed so that 2 and -2 have identical bit patterns.

The bit shift fields are stored separately from the 16 bit windows, and are packed into 32 bit long unsigned integers, 8 per long integer. The bit shift field order and long array index correspondence are shown below:

```
                0                                  1
---------------------------------   ---------------------------------
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |   | 15| 14| 13| 12| 11| 10| 9 | 8 |
---------------------------------   ---------------------------------
```

Each box is 4 bits wide, each group of boxes is 32 bits wide.

dB/dt transient values in volts/amp =

unpacked_value/(Ch_Gain[jch]*Twp_Width[itw]*2*NCycles*ADScale*Tx_Current), where jch = Rx loop channel index and itw = transient time window index, Twp_Width[] is stored in a cfg file [TWP_TABLE_*] parameter block and other parameters are described below in the header record definition.

NCycle can be 1 - 1024, but is usually 1. NCycle = 1 means two half-cycles rectified and summed together.

zdb **and** ***avg*** **Files: ASCII TEM transient 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.

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

Metadata records with a leading "`$`" character are also present and play a large role in recording information that does not fit into a tabular spreadsheet format. Metadata records have a `$program:group.variable=value(s)` format. The program name is optional, but if present the metadata record is ignored if the program name is not ***temtrim*** or ***tem***. Version 2 metadata 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 text, instead of as a separator, should be enclosed in quotes.

Records with a leading letter are interpreted to column label lines. A line holding column labels must precede numerical data. Column label matching is not case sensitive, 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 (Twin.Center or dBdt.mag), ***temtrim*** 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 a missing value bracketed by commas.

Partial listing of TEM average file.:

```
\\ TEM data from TEMAVGW v1.00n on 01/04/10
$ Job.Name = "In-Loop TEM"
$ Job.Area = "Zonge Tucson"
$ Job.For = "TEMAVGW Documentation"
$ Survey.Type = TEM
$ Survey.Array = INL
$ Line.Name = "200 E"
$ Line.Number = 200
$ Line.Azimuth = 0
$ Stn.GdpBeg = -300
$ Stn.GdpInc = 50
$ Stn.Beg = -300
$ Stn.Inc = 50
$ Stn.Left = -300
$ Stn.Right = 650
$ Tx.Ramp = 282 usec
$ Tx.Turns = 1
$ Tx.Length = 1200,1200 m
$ Tx.Area = 1.4400E+06 m^2
$ Rx.Area = 1.0000E+4,1.0000E+4,1.0000E+4 m^2
$ Rx.Length = 50,50 m
$ Rx.HPR = 0,0,0
$ Unit.Length = m
$ Unit.Time = msec
$ Unit.E = nV/Am
$ Unit.B = pT/A
$ Unit.dBdt = uV/A
$ TEMAVGW:Version = 1.00n applied 2010/04/01
$ TEMAVGW:Avg.Type = Straight
$ TEMAVGW:Ch.NumberType = Stn Number,Stn Number
$ TEMAVGW:Rx.T0Delay = 0 usec

$Gdp.Blk=35.01,36.01
$Gdp.Setup=1
$Gdp.Date=96-03-25
$Gdp.Time=10:36:47.0
$Gdp.AcqMethod=Transient
$Tx.GdpStn=1
$Tx.Stn=1
$Tx.Freq=8 hertz
$Tx.nCycle=240
$Tx.Amp=3.6 amp
$Tx.Ramp=282 usec
$Rx.GdpStn=650
$Rx.Stn=650
$Rx.HPR=90,0,0
$Rx.AreaZ=1.0000E+04
$Rx.Cmp=Hz
$Rx.AntDelay  = 15 usec
$Rx.AliasDelay= 26 usec
$Tx.Delay= 282 usec
TWin.Index,TWin.Center,TWin.Beg,TWin.End,dBdt.Mag,dBdt.Err,dBdt.%Err,dBdt.Wgt,B.Mag,B.Err,ARes.Mag,ARes.Depth_m
 1, 4.0426E-2, 2.7950E-2, 5.8470E-2, 1.04105E+04, 1.95E+01,    0.2,   1.00, 4.41370E+02, 4.70E+00,  1970.5, 232.05
 2, 7.2134E-2, 5.8470E-2, 8.8990E-2, 1.13860E+03, 5.10E+00,    0.4,   1.00, 4.23746E+02, 4.66E+00,  7110.2, 588.82
 3, 1.0311E-1, 8.8990E-2, 1.1946E-1, 3.11110E+03, 1.49E+01,    0.5,   1.00, 4.17272E+02, 4.68E+00,  2230.5, 394.29
 4, 1.3389E-1, 1.1946E-1, 1.5006E-1, 6.56780E+03, 4.00E+00,    0.1,   1.00, 4.02463E+02, 4.70E+00,  816.21, 271.8
 5, 1.6461E-1, 1.5006E-1, 1.8056E-1, 5.45745E+03, 8.55E+00,    0.2,   1.00, 3.84125E+02, 4.70E+00,  753.42, 289.54
 6, 1.9522E-1, 1.8056E-1, 2.1106E-1, 2.98480E+03, 2.20E+00,    0.1,   1.00, 3.71250E+02, 4.69E+00,  1056.7, 373.43
 7, 2.4100E-1, 2.1106E-1, 2.7519E-1, 2.74595E+03, 1.10E+01,    0.4,   1.00, 3.58299E+02, 4.72E+00,  859.81, 374.27
 8, 3.0220E-1, 2.7519E-1, 3.3186E-1, 2.82650E+03, 6.50E+00,    0.2,   1.00, 3.41247E+02, 4.77E+00,  609.45, 352.85
 9, 3.6330E-1, 3.3186E-1, 3.9771E-1, 2.16870E+03, 3.00E-01,    0.1,   1.00, 3.25987E+02, 4.79E+00,  590.82, 380.92
10, 4.3880E-1, 3.9771E-1, 4.8413E-1, 2.00925E+03, 2.95E+00,    0.1,   1.00, 3.10215E+02, 4.80E+00,  476.73, 376.05
11, 5.3050E-1, 4.8413E-1, 5.8131E-1, 1.69185E+03, 7.25E+00,    0.4,   1.00, 2.93245E+02, 4.78E+00,  414.76, 385.67
12, 6.5050E-1, 5.8131E-1, 7.2793E-1, 1.41580E+03, 6.00E-01,    0.1,   1.00, 2.74600E+02, 4.74E+00,  351.93, 393.39
13, 8.1770E-1, 7.2793E-1, 9.1855E-1, 1.15740E+03, 2.00E+00,    0.2,   1.00, 2.53088E+02, 4.76E+00,  290.27, 400.57
14, 1.0150E+0, 9.1855E-1, 1.1216E+0, 9.45220E+02, 5.16E+00,    0.5,   1.00, 2.32345E+02, 4.83E+00,  242.45, 407.86
15, 1.2580E+0, 1.1216E+0, 1.4110E+0, 7.53430E+02, 2.27E+00,    0.3,   1.00, 2.11707E+02, 4.92E+00,  205.81, 418.36
16, 1.5610E+0, 1.4110E+0, 1.7269E+0, 5.91430E+02, 2.08E+00,    0.4,   1.00, 1.91332E+02, 4.99E+00,  175.49, 430.33
17, 1.9540E+0, 1.7269E+0, 2.2109E+0, 4.51450E+02, 4.36E+00,    1.0,   1.00, 1.70839E+02, 5.12E+00,  150.02, 445.16
18, 2.4690E+0, 2.2109E+0, 2.7572E+0, 3.45895E+02, 1.75E-01,    0.1,   1.00, 1.50308E+02, 5.23E+00,  125.04, 456.84
19, 3.1070E+0, 2.7572E+0, 3.5012E+0, 2.67680E+02, 2.24E+00,    0.8,   1.00, 1.30735E+02, 5.31E+00,  103.59, 466.45
20, 3.8950E+0, 3.5012E+0, 4.3331E+0, 2.08530E+02, 7.10E-01,    0.3,   1.00, 1.11972E+02, 5.43E+00,  85.588, 474.72
```

TEM avg file metadata keywords:

`Job.Name` = string, project name (< 128 char)
`Job.Area` = string, project area (< 128 char)
`Job.For` = string, client name (< 128 char)
`Job.By` = string, contractor name (< 128 char)
`Job.Number` = string, identifying job label (< 16 char)
`Job.Date` = string, data acquisition date (< 16 char)
`Survey.Type` = enumeration, {`LOTEM,TEM,CSAMT,NSAMT,CR,RPIP,TDIP`}
`Survey.Array` = enumeration, TEM array {`FXL,MVL,INL,COL,LOT,CNT`}
`Line.Name` = string, arbitrary line "number" (<16 char)
`Line.Number` = float, line number, used for coordinate interpolation
`Line.Azimuth` = float, azimuth (deg E of N) or string = NnnE bearing format
`Stn.GdpBeg` = float, minimum gdp stn number, used for gdp to final stn # `scaling`
`Stn.GdpInc` = float, GDP stn number increment
`Stn.Beg` = float, final stn number
`Stn.Inc` = float, final stn number increment
`Stn.Left` = float, stn number on left side of pseudosection plots
`Stn.Right` = float, stn number on right side of pseudosection plots
`Unit.Length` = enumeration, length units {`m,ft`}
`Unit.Time` = enumeration, transient-delay time units {`usec,msec,sec`}
`Unit.dBdt` = enumeration, dB/dt unit {`uV/A,nV/Am2`}
`Unit.B` = enumeration, B unit {`pT/A,uT/A,nT`}
`TEMAVG:Version` = temavgw version number
`TEMAVG:Avg.Type` = enumeration, averaging method {straight,robust}
`TEMAVG:Ch.NumberType` = enumeration gdp channel number interpretation
{stn number, stn # offset, n-spacing,n-spacing offset}
`TEMAVG:Rx.T0Delay` = real, time window origin offset (usec)
`Gdp.Blk` = floats, first and last GDP raw file block.channel numbers
`Gdp.Date` = string, GDP data acquisition date (`yy:mm:dd`) (<16 char)
`Gdp.Setup` = float, GDP setup number
`Gdp.Time` = string, GDP data acquisition time (`hh:mm:ss.sss`) (<16 char)
`Tx.GdpStn` = float, original GDP Tx value
`Tx.Stn` = float, Tx center stn #
`Tx.Center` = floats, Tx center east,north,elevation (length units)
`Tx.Length` = float, Tx loop side length (length units)
`Tx.Area` = float, Tx loop area (m^2)
`Tx.Freq` = float, Tx repetition rate (hertz)
`Tx.nCycle` = integer, number of Tx waveform cycles/stack
`Tx.Turns` = integer, number Tx loop turns
`Tx.Amp` = float, Tx peak current (amps)
`Tx.Ramp` = float, Tx turn-off ramp duration (μsec)
`Rx.GdpStn` = float, original GDP Rx station value
`Rx.Stn` = float, Rx coil stn number
`Rx.Center` = floats, Rx coil east,north,elevation coordinates
`Rx.HPR` = floats, Rx component orientation (heading,pitch,roll deg)
`Rx.Area` = float, Rx coil effective area (m^2)
`Rx.Cmp` = enumeration, Rx component {`Hx,Hy,Hz`}
`Rx.AntDelay` = float, Rx antenna preamp delay (usec)
`Rx.AliasDelay` = float, anti-aliasing filter delay (usec)
`Tx.AliasDelay` = float, time delay to get past Tx turn-off ramp (usec)

TEM avg file column labels:

`TWin.Index`	= time window index
`TWin.Center`	= time window geometric center (msec)
`TWin.Beg`	= time window beginning (msec)
`TWin.End`	= time window end (msec)
`dBdt.Mag`	= dB/dt magnitude (μV/A)
`dBdt.Err`	= dB/dt error (μV/A)
`dBdt.%Err`	= dB/dt relative error (percent)
`dBdt.Wgt`	= dB/dt data skip-flag weight (0=skip,1=use)
`B.Mag`	= B(t) magnitude (pT/A)
`B.Err`	= B(t) error (pT/A)
`ARes.Mag`	= apparent resistivity (ohm-m)
`ARes.Depth`	= image section plot-point depth (length units)