GPR_CONV

Version: 2.08.01.01
Last revision date: 8-1-2001
Description: GPR_CONV converts digital radar data from one storage format to another. It can also separate out one channel from multi-channel DZT files. The input to this program is a "CMD" file, an ASCII text file containing keywords (or commands) which are discussed in a section below. There is no graphic display of the data. To display the converted data, use programs such as GPR_DISP.EXE or FIELDVEW.EXE. If you need to select a subset of traces from a file or change the number of samples per trace then use GPR_SAMP.EXE.

THE KEYWORDS

Following is the list of keywords and their default values. The documentation format is:
"KEYWORD: **keyword** = default value".
Look at GPR_CONV.CMD as an example command file with correct usage and default keyword values. The file GPR_CONV.CMD has most comments stripped out, and GPR_CONV.CM_ has all comments removed.

*************************** PROGRAM CONTROL ********************************
KEYWORD: **batch** = "FALSE"
Place program in batch mode (no pauses) if "TRUE". If set to "FALSE", the program will pause after the keyword values are displayed and ask if you want to continue. After the data are processed, the program will end automatically.

***************** SPECIFICATION OF INPUT AND OUTPUT DATA ********************
Many data files can be read in, but at least one must be given. The data storage format is determined by inspecting the file. If the program cannot recognize one of the three storage formats below then that file is skipped. Data must be stored in a binary format. This program cannot read GPR data stored in text files.

Recognized storage formats are:
  DZT - GSSI DZT file
  DT1 - Sensors & Software pulseEKKO file with a matching HD text file
  SGY - SEG SEG-Y format

DT1 and HD files are assumed paired, i.e. both have the same filename with different extensions. So, if a data file with a ".DT1" extension is specified, the ".HD" filename will be assumed. Only DT1/HD files must have those filename extensions. See below for information on how to read other storage formats.

KEYWORD: **num_input_files** = 0
Replace the 0 with the number of files you wish to process. There is no set limit to the number of files. There must be the same number of output as input files. The first output file corresponds to the first input file.

KEYWORD: **input_filelist[]**
Add an equal sign, =, then list the filenames after the brackets. The list can extend across multiple lines. Input filenames can occur more than once in the input list but cannot appear in the output list (order-wise) after they appear in the input list. Any combination of recognized or user-defined storage formats may be used.

Extracted from Lucius, J.E. and Powers, M.H., 2002, GPR Data Processing Computer Software for the PC: U.S. Geological
Survey Open-File Report 02-166, 101 p.

2

KEYWORD: **output_filelist[]**
Add an equal sign, =, then list the filenames after the brackets. The list can extend across multiple lines.
Output filenames cannot occur more than once in the output list (that is, all names must be unique). The
output storage format is determined from the output filename extension. "DZT" files are stored in the
GSSI "DZT" format. "DT1" files are stored in the paired Sensors & Software binary- and text-file
formats. "SGY" files are stored as SEG SEG-Y files using Sensors & Software format for the reel header.
Any combination of recognized storage formats may be used.

Here are some examples; they would all be read in the same.

    input_filelist[] = file1.dzt file2.dzt file3.dzt file4.dzt file5.dzt

    input_filelist[] =
        file1.dzt file2.dzt file3.dzt file4.dzt file5.dzt

    input_filelist[] = file1.dzt
        file2.dzt file3.dzt
        file4.dzt file5.dzt

    output_filelist[] =
        file1c.dzt
        file2c.dzt
        file3c.dzt
        file4c.dzt
        file5c.dzt

KEYWORD: **channel** = 1
This is the channel to use in multi-channel data sets. GSSI data can have up to 4 channels (channel = 1, 2,
3, or 4). For pulseEKKO, RAMAC, and SEG-Y data, which contain only 1 channel, this keyword is
ignored. Note that output files are single-channel only.

############# NOTE #############
IF the GPR format DOES NOT CONFORM to any of the recognized formats then the next six parameters
(other_format, file_header_bytes, trace_header_bytes, samples_per_trace, total_time, and input_datatype)
MUST be specified. Otherwise, IGNORE THEM. You can use GPR_INFO.EXE to report the basic
information for recognized storage formats.
##################################

KEYWORD: **other_format** = "FALSE"
Replace with "TRUE" if you want to use the next five parameters to specify the input format.

KEYWORD: **file_header_bytes** = 0
Replace with number of bytes in the file header. pulseEKKO data files do not have a file header - the
information is held in another file with a .HD extension. GSSI files have either a 512-byte (old style) or
1024-byte (current style) header. However, DZT files can have up to 4 file headers - one for each channel.
SEG-Y files have a 3600-byte header. RAMAC data files have no file header.

KEYWORD: **trace_header_bytes** = 0
Replace with number of bytes in each trace header. For pulseEKKO files, a 128-byte header precedes
each GPR trace. For GSSI files, no header precedes each trace, but the first 2 samples (not necessarily
bytes) are reserved. SEG-Y files have a 240-byte trace header. RAMAC data files have no trace headers.

KEYWORD: **samples_per_trace** = 0
Replace with the number of samples per trace. For pulseEKKO data, the number of samples per trace is recorded in the HD file (NUMBER OF PTS/TRC). For GSSI data, the number of samples per trace is a power of 2, from 128 to 2048, typically 256, 512, or 1024. The information is recorded in the DZT file header in the rh_nsamp field. For RAMAC files, the RAD text file records the number of samples. For SEG-Y files, look in the comment area of the file header.

KEYWORD: **total_time** = 0
Replace with the total number of nanoseconds per trace. For pulseEKKO data, look at the "TOTAL TIME WINDOW" field in the .HD file. For GSSI data the value is recorded in the file header. For SEG-Y files, look in the comment area of the file header. For RAMAC files, the TIMEWINDOW parameter records the time per trace in microseconds (multiply by 1000 to get ns).

KEYWORD: **input_datatype** = 0
This defines the type of input data element. Replace with one of the following element types:

```
 1 for 1-byte signed characters
-1 for 1-byte unsigned characters          (GSSI)
 2 for 2-byte signed short integers        (pulseEKKO, RAMAC, SEG-Y)
-2 for 2-byte unsigned short integers      (GSSI)
-5 for 2-byte unsigned short integers, but only first 12-bits used
 3 for 4-byte signed long integers         (SEG-Y)
-3 for 4-byte unsigned long integers
-6 for 4-byte unsigned long integers, but only first 24-bits used
 4 for 4-byte floats                       (SEG-Y)
 8 for 8-byte doubles
```

For example: 8-bit GSSI data are unsigned characters (values from 0 to 255), use -1 for input_datatype. Use –2 for 16-bit GSSI data (values from 0 to 65535). pulseEKKO and RAMAC data are typically 16-bit signed integers (values from -32768 to 32767), use 2 for input_datatype. For SEG-Y data, the input_datatype can be 2 (signed short integers), 3 (signed long integers), or 4 (4-byte floating point reals). Data types are stored in the file header of DZT and SGY files. PulseEKKO and RAMAC do not record the data type.

############# NOTE ##############
Floating point input data are not supported at this time.
################################

****************** SPECIFYING OPTIONAL INPUT FILES *********************
These are additional input ASCII data file names that are optional.

KEYWORD: **use_mark_file** = "FALSE"
Set to "TRUE" to add marked traces to the output file. The "mark" file must have the same filename as the input file but with the "MRK" file extension. There must be a MRK file for every input file if this keyword is set to "TRUE".

Here is an example MRK file containing marked trace locations.
```
 3
 104
 256
 897
```

KEYWORD: **use_xyz_file** = "FALSE"
This keyword is not implemented at this time. It has no functionality.

Here is an example XYZ file containing X, Y, and Z locations of the marked traces:

```
 3
 10.0   10.0   293.456
 20.0   10.0   294.567
 30.0   10.0   295.678
```

******************* OPTIONAL FORCING OF OUTPUT DATATYPE ********************

KEYWORD: **output_datatype** = 0

This program will select the output data type (signed or unsigned integers) based on the chosen storage
format. PulseEKKO will always be 16-bit signed integers. GSSI data will default to 16-bit unsigned
integers. SEG-Y data will default to the input data type converted to signed if necessary. This parameter
will override these default data types if it is allowable by the storage format. Leave the value at 0 to use
the default.

```
      1 for 1-byte signed characters
     -1 for 1-byte unsigned characters          (GSSI)
      2 for 2-byte signed short integers         (pulseEKKO, RAMAC, SEG-Y)
     -2 for 2-byte unsigned short integers       (GSSI)
     -5 for 2-byte unsigned short integers, but only first 12-bits used
      3 for 4-byte signed long integers          (SEG-Y)
     -3 for 4-byte unsigned long integers
     -6 for 4-byte unsigned long integers, but only first 24-bits used
      4 for 4-byte floats                        (SEG-Y)
      8 for 8-byte doubles
```

Choices are limited as follows by the output storage format:

```
      GSSI DZT files:    -1 or -2        (default = -2)
      pulseEKKO DT1 files: 2             (only 2 allowed)
      SEG-Y SGY files:    2, 3, 4, or 8  (only 2 allowed)
      RAMAC RD3 files:    2              (only 2 allowed)
```

***************** OPTIONAL ADDITIONAL HEADER INFORMATION *******************

These keywords are optional - but supply useful information that may be stored with the output GPR data
depending on storage format. Many of these values can be determined from the GPR data information
header/file. Most of the time the values stored in the info header/file will override these values.
Exceptions are noted.

KEYWORD: **timezero_sample** = 0
The "time zero" is at this sample number.

KEYWORD: **traces_per_sec** = 0.0
This is the number of traces recorded per second.

KEYWORD: **number_of_stacks** = 1
This is the number of traces stacked to make one recorded trace.

KEYWORD: **nominal_frequency** = 0
This is the nominal frequency of antenna in MHz.

KEYWORD: **pulser_voltage** = 0.0
This is the transmitter voltage.

KEYWORD: **antenna_separation** = 0.0
This is the distance between the Tx and Rx antennas.

KEYWORD: **antenna_name** = ""
This is the antenna serial or model number, frequency, etc. Up to 15 characters are accepted.

KEYWORD: **traces_per_meter** = 0.0
This is the number of traces recorded per meter.

KEYWORD: **meters_per_mark** = 0.0
This is the number of meters between tick markers.

KEYWORD: **starting_position** = 0.0
This is the position of first trace in user units.

KEYWORD: **final_position** = 0.0
This is the position of last trace in user units.

KEYWORD: **position_step_size** = 0.0
This is the distance between traces in user units.

KEYWORD: **position_units** = ""
This is "feet", "inches", "meters", etc. Up to 15 characters are accepted.

KEYWORD: **year_created** = 0
This is the year the data were collected, example: 1995.

KEYWORD: **month_created** = 0
This is the month the data were collected. Use numbers from 1 to 12; 1=Jan, 2=Feb, etc.

KEYWORD: **day_created** = 0
This is the day the data were collected. Use numbers from 1 to 31.

KEYWORD: **hour_created** = 0
This is the hour the data were collected. Use numbers from 0 to 23.

KEYWORD: **minute_created** = 0
This is the minute the data were collected. Use numbers from 0 to 59.

KEYWORD: **num_gain_pts** = 0
This is the number of values that follow the gain_pts[] keyword below. It must be greater than or equal to two.

KEYWORD: **gain_pts[]** =
These are the gain values (in dB) separated by spaces.

KEYWORD: **text =** ""
Text or comment information can be added to the header. Input in the CMD file can be multi-line. A maximum of 640 characters is accepted. For GSSI DZT files, this field corresponds to the text area of the file header and will overwrite any text from an input DZT file. For S&S files, this field corresponds to the 1 or 2 records before the date at the start of the HD file. For SEG-Y files, this field corresponds to the 1 or 2 records before the date at the start of the ASCII section of the file header. For S&S and SEG-Y files, the maximum entry is 2 lines. For SEG-2 files, these strings will be assigned to the NOTE keyword in the File Descriptor Block String Sub-block.

KEYWORD: **proc_hist** = ""
Processing history information can be added to the header. Input in the CMD file can be multi-line. A maximum of 640 characters is accepted. For GSSI DZT files, the processing history is in coded binary, so this field does not apply. For S&S files, this field corresponds to the records after the "SURVEY MODE =" record in the HD file. For SEG-Y files, this field corresponds to the records after the "SURVEY MODE =" record in the ASCII section of the file header. If converting from DZT format to DT1, or SGY the coded DZT processing history will be decoded to text strings.

KEYWORD: **survey_mode** = ""
This is data collection mode such as "reflection", "transmission", "WARR", etc. Up to 31 characters are accepted.

KEYWORD: **nominal_RDP** = 0.0
This is the average relative dielectric permittivity. This value if greater than 0, will override input file value.

Usage: GPR_CONV cmd_filename
Required command line arguments:
      cmd_filename - The name of the keyword file.
Optional command line arguments (do not include brackets): none
Examples:
      `gpr_conv cfile1.cmd`