Release notes for GageFileConverter

Table of contents

****

1. What is GageFileConverter?

GageFileConverter is a utility that will convert GageScope signal (.SIG) files into ASCII data files and ASCII data files into GageScope SIG files.

The application is a tabbed dialog box with 2 pages. The first page converts ASCII files into SIG files (see section 2 for details). The second page converts SIG files into ASCII files (see section 3 for details). The actual conversion is done when the Convert button on either page is pressed.

The application may also be run in command line mode. In this mode, no GUI is displayed and parameters are passed via command line arguments. This mode is designed for use with scripts and batch files (see section 4 for details).

****

2. The ASCII File page

This page converts an ASCII data file, in either 1- or 2-column format, to a GageScope SIG file. If the file is in a 1-column format, the values are assumed to be the data samples. The sample values can be voltages, hex values or decimal values. If the file is in a 2-column format, it's assumed that the first column contains the time at which the sample was acquired (in seconds) and the second column contains sample values. The program automatically determines if the ASCII data are in voltage, hexadecimal or decimal format. No other validation is done on the data.

Pressing the Input File and Output File buttons allow the user to choose the input (ASCII) and output (.SIG) file names. Once these are chosen, the Convert button is enabled.

Conversion parameters will determine how ASCII data are converted into SIG file data. The user may select whether to obtain these parameters from the header of a ASCII DAT data file that was stored by a C/C# or MATLAB SDK (version 4.00

and higher) or to customize the parameters by entering them directly into the GageFileConverter utility. The available conversion parameter controls are Sample Size, Sample Resolution, Sample Offset, Sample Rate, Input Range, DC Offset, and Trigger Address.

If the ASCII DAT file has two columns, the sample rate is determined based on the time spacing between the first two samples and the sample rate control will be disabled. If the ASCII DAT file has a header containing enough information about the file, this information will be used by default to generate the SIG file data. The Customize option should be used if the DAT file header is absent or does not contain enough information. The converted file may then be loaded by GageScope or any other program capable of reading SIG files.

****
Batch Mode
****

Batch mode allows the conversion of several ASCII files into GageScope (.SIG) format at one time. This feature is useful when converting results of a multiple segment capture or several pages of a deep acquisition.

To enable batch mode operation inside GageFileConverter, you need to check the batch mode check box. GageFileConverter will query a specific folder using a supplied file mask to retrieve the number of files that are available for conversion. Clicking on the Convert button will initiate a conversion process for each of the located input ASCII files. The output files will be stored in a directory with same name as the search mask, using a naming format that is compatible with the Split File playback feature of GageScope. Consequently, converted files may be viewed together using GageScope’s Split File playback feature.

The file mask is a standard wild character mask, but it must have an extension (e.g. *.dat)

****

3. The SIG File page

This page converts a GageScope SIG file into an ASCII DAT data file.
The user chooses the input SIG file and the output ASCII DAT file. Once the file names are chosen, the Convert button is enabled. Pressing this button will do the conversion.

If the Time Base box is checked, the output ASCII DAT file will contain two columns of data – sample time and sample values. Otherwise, only a single-column of sample values is stored. The data format can be chosen in the File Type control by clicking the Volts, Decimal or Hex buttons. The default is Volts. All other values are obtained from the SIG file header.

The header is verified when the program starts the conversion process. If a valid GageScope SIG file header cannot be read, the conversion process is halted. The output ASCII DAT file has a header with the same format as the DAT files created by the C/C# and MATLAB SDKs.

****

4. Command line operation

GageFileConverter may operate in command line mode without invoking of the GUI. This option is useful for converting multiple files from a batch file.

Possible command line options are:

For ASCII to SIG conversion:

```
GageFileConverter /nogui [/batch] /if:(file|mask) /of:OutputFileName
[/re:SampleResolution] [/so:SampleOffset] [/ss:SampleSizeInBytes]
[/sr:SampleRate] [/ir:InputRange] [/dc:Dcoffset] [/ta:TriggerAddress]
```

For SIG to ASCII conversion:

```
GageFileConverter /nogui  /if:file /of:path [/time] [/ft:(hex|dec|volts)]
```

To display help:

```
GageFileConverter /(?|help)
```

[] - optional parameter
(x|y) - x or y

****

5. Contact information

If you would like to contact us, please use the following e-mail address: prodinfo@gage-applied.com

To phone or fax us, dial:

Toll free phone in North America: 1-800-567-GAGE
Toll free fax in North America: 1-800-780-8411

From outside North America call: +1-514-633-7447
From outside North America fax: +1-514-633-0770

****

I. Appendix 1: Gage ASCII file format

The ASCII DAT files saved by the CompuScope C/C# and MATLAB SDKs each contain a header. This header is necessary in order for the correct interpretation of the acquired data. The header format is common among all the SDKs and is described in this document.

The amount of information in the header varies and can include any or all of the fields mentioned below. Among other things, these field values may be used to convert raw ADC values to voltages or vice versa. Below is a sample file ASCII DAT header.

```
----------------------
Start address  =  0
Data length    =  8192
Sample size    =  2
Sample res     =  -2048
Sample offset  =  -1
Input range    =  2000
Dc offset      =  0
Segment count  =  1
Segment number =  1
----------------------
```

To make it easier to parse the header information, the header begins and ends with a line of dashes (at least 5 consecutive '-' characters). There are currently 10 possible fields in the header (when and if more fields are added, they will be added to this list):

- Start address: The position of the first data sample in the file relative to the trigger point.
- Data length: The number of valid data samples in the file.
- Sample size: The size of the raw data in bytes:
  - 8-bit data has a sample size of 1;
  - 12, 14 and 16-bit data have a sample size of 2;
  - 32-bit data has a sample size of 4.

Note that for files saved in floating point format, the Sample size is the size of the raw ADC data.

The Sample resolution, Sample offset, Input range and Dc offset are used to convert raw ADC data samples to Volts. The conversion formula is:

$$\text{Volts} = \frac{\text{Sample offset} - \text{ADC data sample}}{\text{Sample res}} * \text{Input range} + \text{Dc offset}$$

Segment count: Segment count denotes how many segments (or records) were captured during the acquisition.

The Segment count field is useful for Multiple Record captures, since it shows how many records were captured in total. In files stored from a non-Multiple Record acquisition, the Segment count, if available, will be 1.

Segment number: Segment number denotes this segment's (or record's) order number within a Multiple Record acquisition. For instance, if Segment number is 5, then it was the 5th segment or record number acquired in a Multiple Record acquisition.

Time stamp: Time stamp is the time of occurrence of the trigger event associated with the segment from which the data is saved.

****

II. Appendix 2: GageScope SIG file format

GageScope reads and stores captured signals in Gage's SIG file format. This format has evolved and changes over time. While GageScope and GageFileCoverter are capable of reading SIG files of all versions, they save them only in the newest form.

GageScope SIG files are binary files which contain a 512 byte header, followed by any number of data points.

The first 512 bytes of a GageScope SIG file contain the header, whose exact format of which is listed below[1]:

File Index	Field Type	Field Size	Field Variable	Field Description
0	char.	14	file_version	GS V. 3.00[2]
14	int16	2	crlf1	A carriage return line feed pair.
16	char	9	name	The channel name when stored.
25	int16	2	crlf2	A carriage return line feed pair.
27	char	256	comment	The channel comment when stored.
283	int16	2	crlf3	A carriage return line feed pair.
285	int16	2	control_z	A control Z, artificial end of file.
287	int16	2	sample_rate_index	Index to the sample rate table[3].
289	int16	2	**operation_mode**	1 = not-interleaved, 2 = interleaved.
291	int32	4	**trigger_depth**	Number of samples after the trigger point.
295	int16	2	trigger_slope	1 = positive slope, 2 = negative slope.
297	int16	2	trigger_source	1 = chan A, 2 = chan B, 3 = external,

[1] Fields in bold are essential to data interpretation and are always stored by GageFileConverter. The other fields may or may not be stored by GageFileConverter and so users should not rely on the presence of these fields.
[2] Legacy SIG files have a different version string
[3] In the current SIG file format, the external_clock_rate field always contains the sample rate, even if the signal was captured using internal clocking

299	int16	2	trigger_level	Stored as an int16, actually a byte with the same format as the data.
301	int32	4	**sample_depth**	Number of samples stored in the signal section of the file.
305	int16	2	**captured_gain**	Index to the input range table.[4].
307	int16	2	captured_coupling	1 = DC, 2 = AC.
309	int32	4	current_mem_ptr	Where display started when signal was stored.
313	int32	4	**starting_address**	The first point in the data.
317	int32	4	**trigger_address**	The point in the data where trigger occurred.
321	int32	4	**ending_address**	The last point of the captured data.
325	uInt16	2	trigger_time	The time when the trigger event occurred.
327	uInt16	2	trigger_date	The date on which the trigger event occurred.
329	int16	2	trigger_coupling	1 = DC, 2 = AC. For the external trigger input.
331	int16	2	trigger_gain	Index to the input range table.
333	int16	2	probe	Index to the probe table.
335	int16	2	inverted_data	0 = normal data, 1 = inverted data (CS220), 2 = inverted and flipped data (CS220).
337	uInt16	2	board_type	The CompuScope board type on which the saved data was captured.
339	int16	2	**resolution_12_bits**	Sample size in bytes minus 1.
341	int16	2	**multiple_record**	0 = normal mode, 1 = Hardware multiple record, 2 = Software multiple record. **Note**: mode = 1 is obsolete.
343	int16	2	trigger_probe	Index to the probe table.
345	int16	2	**sample_offset**	Used to offset the data for display and conversion to real voltages. Normally 127 for 8-bit CompuScopes and -1 for 12-bit CompuScopes. File versions before GS V. 2.95 will have sample offset=0 for 12-bit CompuScopes and 128 for 8-bit CompuScopes, if sample_resolution is zero use sample_offset_32 field.
347	int16	2	**sample_resolution**	Used to scale the data for display and conversion to real voltages. Normally 128 for 8-bit CompuScopes and 2048 for 12-bit CompuScopes, 0 means use sample_resolution_32 field.
349	int16	2	sample_bits	Number of bits in the sampled data. Normally 8 for 8-bit CompuScopes and 12 for 12-bit CompuScopes.
351	uInt32	4	extended_trigger_ time	The time when trigger event occurred.

[4] Captured gain may be stored in two formats. If the most significant bit is 0, then the value refers to the index of the Input range within the Input range table. If the most significant bit is 1, then the value refers to the input gain in a fixed point format: FRRppNNNNnnnnddd, where F is the format bit, R is reserved, pp determines decimal point position pp = 0 Gain = .Nnd V; p = 1 Gain = N.nd V p = 2; Gain = Nn.d V p = 3 Gain = Nnd V; NNNN, nnnn are the first decimal digits of the gain ddd is the third one divided by 2.

355	int16	2	imped_a	Impedance for Channel A. 0 = 1 MΩ; 0x10 = 50 Ω.
357	int16	2	imped_b	Impedance for Channel B. 0x0 = 1 MΩ; 0x10 = 50 Ω.
359	float	4	**external_tbs**	Time between samples in nanoseconds when using external clock.
363	float	4	**external_clock_rate**	Sample rate when using external clock.
367	int32	4	file_options	01 = GAGE_SINGLE_CHAN 02 = GAGE_DUAL_CHAN 04 = GAGE_MODE_EXT_CLK_ADJ. Can be OR'ed with either GAGE_DUAL_CHAN or GAGE_SINGLE_CHAN. 08 = GAGE_MODE_MR_JITTER_ADJ. Can be OR'ed with either GAGE_DUAL_CHAN or GAGE_SINGLE_CHAN. 10 = GAGE_FAST_RAM_ADJ. Can be OR'ed with either GAGE_DUAL_CHAN or GAGE_SINGLE_CHAN. 20 = GAGE_X012X_VERS_ADJUST. Can be OR'ed with either GAGE_DUAL_CHAN or GAGE_SINGLE_CHAN.
371	uInt16	2	version	Version of the CompuScope hardware, 4 BCD digits.
373	uInt32	4	eeprom_options	Hardware options stored in the CompuScope board's on-board EEPROM.
377	uInt32	4	trigger_hardware	Auxiliary trigger hardware in use. 0 = none; 1 = Trigger Marker Board (TMB).
381	uInt32	4	record_depth	The size of each Multiple Record record when operating in Pre-trigger Multiple Record (PTM) mode.
385	int32	4	**sample_offset_32**	32 bit version of sample_offset
389	int32	4	**sample_resolution_32**	32 bit version of sample_resolution
393	uInt32	4	multiple_record_count	multiple_record_count
397	int16	2	**dc_offset**	Dc offset stored in 1000^{th} of input range
401	uInt8	111	padding	0 filled section to complete the 512 byte header.