A Flexible Electronic Graph Paper
Version 4.08
File Format Description

James Tappin
james.tappin@stfc.ac.uk

May 2016

Contents

1 Introduction.

Graffer files are tagged datasets which contain all the information needed to describe the plot, its layout on the page, how to print it etc. A major objective in designing the format was that there should be no need for any extra files to be transferred along with the graffer file (thus all the data are incorporated into the file).

There are two distinct graffer file formats, binary and ASCII. It is intended that binary files should be used as the normal format, and ASCII files should be used to transfer between computers which have a different representation of numbers. Historically binary files were not portable as they use native number formats, not XDR format. However since most machines use IEEE number representations and endianness can be detected this is not normally an issue with current systems. They are also smaller and quicker to read and write, but cannot (unless you're really hot with your hex editor) be manually repaired.

2 Generic Issues.

All Graffer files start with a general header, that allows Graffer to recognise that the file is a graffer file and whether it is binary or ASCII. Fields within the file are identified by tags which precede the field. Tags are from 1 to 3 uppercase letters and numbers. The order of tags is not significant except that tags referring to a specific object (dataset or free text annotation) are grouped together.

As is normal in IDL:

Int is a short (2 byte) integer

Long is a long (4 byte) integer

Byte is a single byte

Float is a short (4 byte) floating point number

Double is a long (8 byte) floating point number

String is a string of characters of variable length.

Null refers to a tag with no values associated with it.

This document could hardly be described as a formal definition of the Graffer file format; but rather it tries to give enough information to allow you to figure out what's going on. It should probably be read alongside the source code of the reading and writing routines (`gr_get_asc.pro`, `gr_get_bin.pro`, `gr_asc_save.pro`, `gr_bin_save.pro` and the routines which they call). It is also useful to have at least a working knowledge of IDL's plotting keywords and variables.

Field	Length	Type	"Name"	Content
1	7	String		"GRAFFER"
2	2	Int	Vmaj	Major version number.
3	2	Int	Vmin	Minor version number.
4	4	Long	Ldir	Length of the directory name.
5	Ldir	String	Dirname	+The directory name.
6	4	Long	Lfile	Length of the file name.
7	Lfile	String	Fname	The file name.
8	4	Long	Ldate	Length of the date string.
9	Ldate	String	Date	The date of writing.

Table 1: The GRAFFER header layout for a binary file.

Name	Type	Number	Description
TAG	Str*3	1	The field identification tag
TYPE	Long	1	The type code as returned by IDL's `SIZE` function.
NDIMS	Long	1	How many dimensions, 0 for a scalar.
DIMS	Long	NDIMS	The size of the dimensions (not present when NDIMS=0).
LENS	Long	NVALS	The lengths of strings. NVALS = PRODUCT(DIMS) or 1 for a scalar. Not present if TYPE is not 7 (string).
DATA	TYPE	NVALS	The actual values.

Table 2: The subfields of a Graffer V4 record.

2.1 Binary files.

The header of a binary file is described in Table 1.

Since we may reasonably assume that the major version number will not exceed 256 (at the present rate of progress this will take over a millenium), the magnitude of the version number is used to determine whether the file needs to be byteswapped.

For binary files, tags are always padded with spaces to a length of 3.

2.1.1 Version 4

For version 4 and above the format has been improved such that the file is much more robust against bad tags, this will mean that in principle files will be readable by older versions of Graffer than that which wrote the file (this was not possible in earlier versions as any unrecognized tag or change of variable size [e.g. int to long] would confuse the binary reader completely). Each tag is now followed by information which defines how much data and of what type follows, the record structure is described in Table 2. Version 4 files are always written in little-endian byte order (although the read routines will handle either ordering).

2.1.2 Version 2 and 3

For versions 2 and 3, the format was much less robust. The size of the data field that follows is determined by the type and number of data required by the particular tag. For string quantities, the string is preceded by its length (a long).

2.1.3 Version 1

Version 1 files were a fixed format file with no tags, and relatively limited capabilities. Version 4.x will still read V1 files. They are very rare nowadays.

2.2 ASCII files.

The ASCII file header is:

> `Graffer V` *<major version>*`.`*<minor version>*`:`*<directory>*`<`*filename*`>:  @` *<time of writing>*

as a single line. (The case of the word Graffer is used to decide whether to try to read the file as binary or ASCII). For ASCII files, tags and data are delimited by colons. Apart from the header each record of the file must begin with a tag (which is not preceded by a colon); multiple fields may occur in one record, in which case the tags are preceded by colons. For obvious reasons, a string field has special rules: a string field is terminated by end of line, so a string quantity must be the last item on the line.

The formats used for the different variable types are not fixed as some quantities have ranges which are much more limited than the type that they are stored as. The format needs to leave spaces between quantities as free-form reads are used after the line has been split at the colons.

3 General Tags

The general tags are those which introduce quantities which apply to the whole file.

Tag	Type	Quantity	Meaning
GT	String	1	Plot title (IDL's TITLE key)
GS	String	1	Plot subtitle (SUBTITLE key)
GC	Float	1	Character size for annotations, relative to default.
GA	Float	1	Thickness of axes on plot
GP	Float	4	Position of plot in Normalized coordinates (Format identical to the IDL `!P.Position` variable).
GR	Float	2	Aspect ratio of plot and smaller margin as a fraction of the page size. (Only one of GP and GR has non-zero values)
GI	Byte	1	Whether the plot has isotropic axes.
GHA	Byte	1	Whether the screen plot should use the hard copy aspect ratio.

Tag	Type	Quantity	Meaning
The next group of tags are responsible for defining the X-axis properties			
XR	Double	2	The range of the X-axis
XL	Byte	1	Whether the X-axis has logarithmic scaling.
XSI	Int	1	X-axis style setting (The IDL `XSTYLE` keyword)
XSE	Int	1	X-axis extra style items (1⇒Omit minor ticks (only in old files, this is now replaced by the `XMN` tag); 2⇒Place an axis at the origin; 4⇒Suppress annotations on the axis; 8⇒Annotate an origin axis; Other bits unused at present)
XSG	Int	1	X-axis grid lines setting (0⇒No grid lines; other values ⇒ Grid line in IDL linestyle (value -1)
XST	Int	1	X-axis time labelling options (lsb: time labelling on, next 2 storage unit, next 2 display unit (0=s, 1=m, 2=h, 3=d)
XSZ	Float	1	X-axis value for stored zero in time label. (Display units).
XMJ	Int	1	The number of major intervals to use in the X-axis.
XMN	Int	1	The number of minor intervals to use on the X-axis.
XFM	String	1	The format specifier to use (Overridden by time labelling options)
XNV	Int	1	The number of tick values used (ascii files only).
XVL	Double	<XNV>	A list of explicit major tick locations. If present, this overrides the value of XMJ.
XT	String	1	X axis label.
All the X tags have corresponding tags starting with Y for the Y axis, and R for the secondary Y axis			
YIR	Byte	1	Whether the secondary Y axis is to be displayed.
ZT	Int	1	Colour table number for 2-D datasets displayed as colour plots (IDL colour table indices), as of Version 3.08 this is a default used to initialize the table for the dataset or if the display is 8-bit.
ZG	Float	1	Gamma setting for the colour table (since version 3.08 this is a default setting).
DN	Int	1	The total number of datasets in the plot (Must precede any dataset definitions).
DC	Int	1	The currently selected dataset (NB Zero-based, unlike the values on the GUI which are 1 based)
TN	Int	1	The total number of Text annotation strings in the plot. (Not including standard plot & axis titles).
REM	Strings	n	A comment attached to the dataset, not plotted anywhere.
	Long+Strings	1+n	For V3 and ASCII files.
The following tags define how hardcopy will be generated.			
HC	Byte	1	Whether to use colour PostScript
HE	Byte	1	Whether to generate Encapsulated PostScript
HO	Byte	1	Whether to plot in landscape (0) or portrait (1) mode
HY	Byte	1	Whether to use RGB (0) or CMYK (1) colour representation.
HP	Byte	1	Whether to use A4 (0) or US Letter (1) paper
HT	Byte	1	Whether to put a timestamp on the plot.

Tag	Type	Quantity	Meaning
HS	Float	2	The size of the plot (in cm)
HD	Float	2	The offset of the bottom left of the plot from the bottom left of the paper (as you would normally view the plot)
HAB	String	1	The part of the spooling command before the filename
HAA	String	1	The part of the spooling command after the filename.
HVB	String	1	The part of the view command before the filename
HVA	String	1	The part of the view command after the filename.
HFN	String	1	The filename for the hardcopy output.
HF	Int	1	The font family (for hardware fonts) 0=courier, 1=helvetica, 2=helvetica narrow, 3=schoolbook, 4=palatino, 5=times, 6=avantgarde book, 7=avantgarde demi, 8=bookman demi, 9=bookman light, 10=zapfchancery, 11=zapfdingbats, 12=symbol. These are the fonts used for titles, axis annotations and keys in hard-copy output, manual annotations have their own font settings.
HWS	Int	1	The weight (0 normal, 1 bold) and slant (0 roman, 2 italic).
The next group of tags control the display of a key or legend on the plot.			
KU	Byte	1	Whether to display a key.
KX	Double	2	The x range in which to place the key
KY	Double	2	The y range in which to place the key
KN	Int	1	The coordinate system for the placement (0=data [always the primary Y-axis], 1=NDC[1], 2=frame [Normalized coordinates relative to the axes]).
KC	Int	1	How many columns to use in the layout.
KF	Byte	1	Whether to put a box around the key
KP	Byte	1	Whether to plot 2 (0) or 1 (1) points on the example
KS	Float	1	Character size for the key.
KT	String	1	Title for the key
KL	Int	n	How many datasets in the key and which datasets to include.
	Int	1+n	For V3 and ASCII.

4 Dataset Tags

These tags define a specific dataset within the plot.

Tag	Type	Quantity	Meaning
DS	Int	1	Start a dataset, must come after the DN tag. Value is the dataset number.
J	Int	1	Joining option, 0=none, 1=line, 2=histogram
P	Int	1	Symbol (IDL `PSYM` value + some extras)

[1] In practice, NDC coordinates in GRAFFER are relative to the plotting region, rather than the whole drawing surface

Tag	Type	Quantity	Meaning
S	Float	1	Symbol size (Relative to default)
L	Int	1	Line style (IDL linestyle values)
C	Int	1	Colour (Table as in PGPLOT). -1 denotes omit this dataset, -2 denotes a custom colour.
CV	Byte	3	A colour triple (R,G,B), only present is the colour index is -2.
W	Float	1	Line thickness
O	Byte	1	Whether to sort the X-axis values before plotting
K	Byte	1	Whether to clip at the plot limits (0=clip, 1=noclip)
E	Byte	1	Whether to allow editing of the dataset with the mouse
D	String	1	Description of the dataset (only displayed if a key is plotted)
N	Long	1	How many points in the dataset (on the X-axis for a 2-D dataset)
N2	Long	1	How many points on the Y axis of a 2-D dataset.
T	Int	1	Dataset type (+ve values specify observational sets, -ve functions) 0: Plain X Y dataset, 1: X Y and Y-errors, 2: X, Y and X errors, 3: X, Y and +Y,-Y errors, 4: X, Y and +X,-X errors, 5: X, Y andX & Y errors, 6: X, Y, and X, -Y, +Y errors, 7: X, Y and -X,+X,Y errors, 8: X, Y, and -X,+X, -Y,+Y errors. 9: 2-D data. -1: $y = F(X)$, -2: $x = F(Y)$, -3: $x = F(T), y = G(T)$; -4: $z = F(X, Y)$
M	Int	1	Coordinate system (0: rectangular, 1: Polar radians, 2: Polar degrees).
Y	Int	1	Which Y-axis to use (0: main, 1: secondary).
ZF	Int	1	2-D dataset format (Contours or Gray/Colour)
ZNL	Int	1	Number of levels for contour plot. In version 4, this is only present for automatic levels.
ZL	Double	<ZNL>	The contour levels to use
ZLL	Null	0	Indicates that the contour levels follow on multiple lines. Only used in V4 ASCII files when ZNL is more than 5.
ZNC	Int	1	Number of colours to use for coloured contouring (Not present in V4).
ZC	Int	<ZNC>	The colour indices to use.
ZCL	Null	0	Indicates that the colours follow on multiple lines. Only used in V4 ASCII files when ZNC is more than 20.
ZNT	Int	1	The number of line thicknesses to use in contouring (Not present in V4).
ZT	Float	<ZNT>	The line thicknesses to use.
ZTL	Null	0	Indicates that the thicknesses follow on multiple lines. Only used in V4 ASCII files when ZNT is more than 20.
ZNS	Int	1	The number of linestyles to use in contouring (Not present in V4).
ZS	Int	<ZNS>	The linestyles to use.

Tag	Type	Quantity	Meaning
ZSL	Null	0	Indicates that the linstyles follow on multiple lines. Only used in V4 ASCII files when ZNS is more than 20.
ZCF	Int	1	Whether to fill contours (1) or add downhill ticks (2).
ZLI	Int	1	Labelling of contours (Label every $<$ ZLI $>^{\mathrm{th}}$ contour
ZCS	Float	1	The character size (relative to default) used for contour labels.
ZR	Double	2	The range of Z values to display in gray/colour display
ZP	Float	1	The pixel size to use (in mm) for hardcopy.
ZIL	Byte	1	Whether the colour scaling is logarithmic.
ZIN	Byte	1	Whether the colour map is inverted.
ZM	Double	1	A value to use for regions of an image display that do not map from the data.
ZCT	Int	1	The colour table for this dataset (set to 1+colour table number, 0 is use default).
ZCG	Float	1	The gamma value for the colour mapping.
R	Float	2 or 4	The range over which to plot a function dataset. (4 values for 2-D function datasets).
F	String	1	Function (for type -1 and type -2 datasets)
FX	String	1	X-function for type -3 function datasets
FY	String	1	Y-function for type -3.
VS	Double	<N>×Cols	The actual data. Note that this tag and the other start data tags are not quite like other valued tags in that in an ASCII dataset the values must start on a new line and be in <N> rows with the number of columns determined by the type setting.
VE	Null	0	End of data. For binary files this is just a verification. Not present in V4
ZX2	Byte	1	Flag for 2-d X array (must precede the ZXS tag, not present in V4).
ZY2	Byte	1	Flag for 2-d Y array (must precede the ZYS tag, not present in V4).
ZXS	Double	<N>	X values for a 2-D dataset (will be <N>×<N2> if ZX2 flag is set)
ZXE	Null	0	End of X-values. Not present in V4.
ZYS	Double	<N2>	Y values for a 2-D dataset (will be <N>×<N2> if ZY2 flag is set)
ZYE	Null	0	End of Y values. Not present in V4.
ZZS	Double	<N>×<N2>	Z values in 2-D dataset
ZZE	Null	0	End of Z values. Not present in V4.
DE	Null	0	End of Dataset.

5 Annotations.

An annotation is a text string which can be freely placed on the plot, it is not associated with a specific dataset, nor is it a standard title or part of a key.

Tag	Type	Quantity	Meaning
TS	Int	1	Start text string, value is string index.
TTS	Null	0	Start text template (not all the following keys are used in a template).
C	Int	1	Colour of text string. -2 denotes a custom colour.
CV	Byte	3	Colour triple for custom colours.
S	Float	1	Size of characters (relative to default)
O	Float	1	Orientation (degrees anticlockwise for L->R).
A	Float	1	Alignment relative to anchor, 0 - starts at anchor,1 - ends at anchor, 0.5 centred on anchor.
FF	Int	1	The font family (corresponds to the values of IDL's `!p.font` system variable.
F	Int	1	IDL font number. In V3, -1 is hardware font (using the same font as plot annotations).
W	Float	1	Line thickness to use for drawing.
X	Double	1	X position of anchor. (Not in template)
Y	Double	1	Y position of anchor. (Not in template)
N	Byte	1	Coordinate system (0 = data coords, 1 = NDC, 2 = Frame coordinates [Normalized coordinates in terms of the axis box]).
AX	Int	1	Which Y-axis to use for Data coordinates. (Not in template)
T	String	1	The actual text string.
TID	String	1	An identifying ID to help programmatic access.
TE	Null	0	End the text description
TTE	Null	0	End the template.