CHAPTER

17

The GIMPORT Procedure

Overview

The GIMPORT procedure enables you to import into SAS/GRAPH software graphics output that is produced with other software applications, graphics output that is produced by SAS/GRAPH software, or graphics output that is produced on other machines. The GIMPORT procedure takes as its input a computer graphics metafile (CGM) and produces graphics output that can be displayed in your SAS/GRAPH session and stored in a SAS catalog. This graphics output can be reviewed and played like any other SAS/GRAPH output using the GREPLAY procedure. The GIMPORT procedure may also write any of the following information to the log:

- any elements used in the CGM that the procedure cannot process
- color mapping information when a color in a CGM is not available on the destination device
- a list of fonts that are used by the application that produced the CGM.

Note: In addition to the GIMPORT procedure, you can use commands in the File pull-down menu in the Image Editor, Graph Editor, and Graph window to import other graphic formats such as GIF, TIFF, and WMF. △

Concepts

About Importing Graphics

A computer graphics metafile (CGM) is a graphics output file that is created according to a standard (ANSI X3.122). Since many graphics applications, including SAS/GRAPH software, can generate and import CGMs, these files can be read by different applications programs or used on different machines.

The GIMPORT procedure imports a CGM with which a fileref has been associated. Using the CGM as input, the procedure displays the graphics output and creates a catalog entry. The following sections address how to assign the fileref to the external file (CGM) and how to import the file.

Specifying a Fileref

You must assign a fileref to the external file that contains the CGM that you want to use as input so that the GIMPORT procedure can locate it. You can do this with a FILENAME statement that has the following form:

```
FILENAMEcgm-fileref'external-file';
```

Replace *cgm-fileref* with any fileref name that you want. Replace '*external-file*' with the complete file name of the CGM. You can omit the FILENAME statement if you have already defined the fileref. You can also specify a fileref with a host command in some operating environments. See "FILENAME Statement" on page 24 for additional information.

Importing the File

The PROC GIMPORT statement reads the input CGM and displays the graphics output. When the CGM is displayed using only the PROC GIMPORT statement, the resulting graphics output may not be sized or positioned correctly for the device on which it is displayed. In these cases, you can use the SCALE and TRANSLATE statements to adjust the size and location of the new graphics output.

In addition, if the CGM contains the FONT LIST element, the procedure lists in the log the fonts used in the CGM. You can change these fonts to SAS software fonts using the MAP statement. If you do not change these fonts to SAS software fonts, the GIMPORT procedure uses a default font.

Because it is easier to determine what adjustments the graphics output needs after it has been displayed, you may want to follow these steps:

1. Import the CGM and display the graphics output using only the PROC GIMPORT statement.
2. Decide what adjustments you want to make to the size and position of the graphics output.
3. If the procedure lists the fonts that are used by the CGM, decide what font substitutions you want to make.
4. Run the procedure again with the appropriate MAP, SCALE, or TRANSLATE statements.

Note: Once you have determined the correct values for the SCALE and TRANSLATE statements for the graphics output produced by a particular CGM, you

can use the same values for all other graphics output that is generated by the same software application. △

CGM Elements Not Supported

The GIMPORT procedure does not support certain CGM elements. If the input CGM contains any of the following elements, the GIMPORT procedure writes a message to the log noting that the procedure cannot process them:

- the CELL ARRAY primitive element (a bitmap CGM file)
- the CHARACTER SPACING attribute element
- the APPLICATION DATA element
- the ESCAPE element.

These elements are rarely used and their absence should not affect the graphics output produced by the GIMPORT procedure.

About Color Mapping

If the CGM specifies colors for the graphics elements that it generates, you may or may not be able to map them to the color that you want in your SAS/GRAPH output, depending on the way these colors are specified in the CGM.

You cannot change the color mapping if, in the CGM, the COLOUR SELECTION MODE element is set to DIRECT. In this case, the colors are explicitly defined by the CGM and you cannot change them. However, if the CGM was created with a SAS/GRAPH CGM device driver, you can control the colors by specifying the appropriate colors when you create the graphics output or by changing the colors in the CGM device entry and re-creating the CGM. See Chapter 15, "The GDEVICE Procedure," on page 651 for details. In addition, you can use a color map with the GREPLAY procedure to remap the colors. In the color map, the FROM color must be specified in RGB format, but the TO color can be any valid color name. See Chapter 26, "The GREPLAY Procedure," on page 919 for details on color maps.

You can change the color mapping if the COLOUR SELECTION MODE element is set to INDEXED and there is no color table defined in the CGM file. In this case, you can map the colors from the CGM to the colors of your choice by using the COLORS= graphics option when you run the GIMPORT procedure. The CGM colors are mapped to match the order of the colors in the colors list. If the procedure cannot reproduce the colors specified in the CGM, the following message is written to the log:

```
WARNING: Invalid color index n encountered.
         It has been mapped to color-name.
```

Note: The color name from the CGM is converted to the RGB format for SAS/GRAPH color names; that is, WHITE is converted to CXFFFFFF, and so on. See Chapter 7, "SAS/GRAPH Colors," on page 139 for details. △

About Pattern Mapping

If the CGM contains pattern specifications, you may be able to map them to patterns of your choice using SAS/GRAPH PATTERN definitions.

If the CGM defines a PATTERN TABLE, then the patterns defined by this table are the patterns that are used and you cannot change them.

If a PATTERN TABLE is not defined in the CGM, under certain conditions you may be able to use SAS/GRAPH PATTERN definitions to control the patterns that are used. If INTERIOR STYLE is set to PATTERN and if a PATTERN TABLE INDEX has been specified, then the GIMPORT procedure uses the PATTERN TABLE INDEX to look up SAS/GRAPH PATTERN definitions. If patterns are defined, the procedure uses the first available pattern. For example, if the PATTERN TABLE INDEX *n* has been defined, the procedure uses SAS/GRAPH PATTERN definition *n*. If the SAS/GRAPH PATTERN definition is not the correct pattern type, the procedure modifies the pattern as necessary. If no PATTERN definitions are currently in effect, an INVALID PATTERN TABLE INDEX warning is issued and no pattern is used.

About Font Mapping

By default, the GIMPORT procedure maps all of the fonts in the CGM to the font that is specified by the FTEXT= graphics option. If the FTEXT= graphics option is not used, the default is the hardware font NONE. However, you may be able to specify a different font either by mapping the fonts or by using a graphics option.

When the CGM is imported, a numbered list of the fonts that are used in the CGM may be displayed in the LOG window. These are the fonts that were available to the application that originally generated the CGM. Depending on how the fonts are represented in the CGM, you may be able to map these fonts to fonts of your choice.

If the font and text in the imported graphics output are produced with move and draw commands that are included in the CGM, then no font name appears in the LOG window and the font cannot be mapped to a different one.

If the fonts used in the imported graphics output are represented in the CGM as a font name accompanied by a text string, they can be mapped to SAS/GRAPH fonts using the MAP statement. You can use the MAP statement if the message "WARNING: Invalid font index *n*. Font has been mapped to *font-name*" appears in the LOG window after the list of fonts. This means that font *n* in the list could not be reproduced and was mapped to the font specified in the FTEXT= graphics option or to the hardware font. You can map this font to a SAS/GRAPH software font of your choice using the MAP statement. See “MAP Statement” on page 710 for more information on mapping fonts.

You can also specify a font with the FTEXT= or CHARTYPE= graphics options if both of the following conditions are true:

- The font has not been mapped with a MAP statement.
- The CGM font contains a font name and text rather than the move and draw commands that draw the text in the specified font. In the latter case, the font name is not included in FONT LIST.

However, using a graphics option causes all fonts to be mapped to the one that is specified. See Chapter 6, “SAS/GRAPH Fonts,” on page 125 for details of font specification and Example 2 on page 715.

Procedure Syntax

Supports: Output Delivery System (ODS)

PROC GIMPORT FILEREF=*cgm-fileref* | *'external-file'*
FILETYPE=CGM
FORMAT=BINARY | CHARACTER | CLEARTEXT
<GOUT=<*libref.*>*output-catalog*>;

MAP *'cgm-font'* TO *font* ;
SCALE X=*factor* | Y=*factor* | X=*factor* Y=*factor*;
TRANSLATE X=*offset* | Y=*offset* | X=*offset* Y=*offset*;

PROC GIMPORT Statement

Identifies the input file to be processed, and specifies its file type and format. Optionally specifies an output catalog.

Syntax

PROC GIMPORT FILEREF=*cgm-fileref* | *'external-file'*
FILETYPE=CGM
FORMAT=BINARY | CHARACTER | CLEARTEXT
<GOUT=<*libref.*>*output-catalog*>;

Required Arguments

FILEREF=*cgm-fileref* | *'external-file'*
specifies the computer graphics metafile (CGM) that is input for PROC GIMPORT. Following are the possible values for FILEREF=:

cgm-fileref
a fileref that is associated with the CGM and that has been previously defined using a FILENAME statement or host command.

'external-file'
the complete file name of the CGM that you want to import. See the operating system companion for your system for valid values for *external-file*.

Featured in: Example 2 on page 715

FILETYPE=CGM
specifies the type of the input file, that is, the graphics standard to which the file conforms. CGM is the only valid value for the FILETYPE= argument. If the FILETYPE= argument is omitted, an error is issued and the procedure stops.

Featured in: Example 2 on page 715

FORMAT=BINARY | CHARACTER | CLEARTEXT
specifies the format of the input file. CGMs can be encoded in one of the following three formats:

BINARY
specifies binary encoding. It is not printable.

CHARACTER
specifies an encoding suitable for transfer through networks that cannot support binary transfers. It is printable but not readable.

CLEARTEXT
specifies a text format that can be read using a standard text editor.

Most graphics packages use BINARY format. If you specify the wrong format, an "ERROR: Unable to interpret the CGM file" message is issued and the procedure stops. If this occurs, try a different format.

Featured in: Example 2 on page 715

Options

GOUT=<*libref.*>*output-catalog*
specifies the SAS catalog in which to save the graphics output produced by the GIMPORT procedure. If you omit the libref, SAS/GRAPH looks for the catalog in the temporary library called WORK and creates the catalog if it does not exist.

See also: "Storing Graphics Output in SAS Catalogs" on page 49

MAP Statement

Substitutes a SAS/GRAPH software font for a font in the CGM.

Requirements: Submit a separate MAP statement for each CGM font that you want to map.

Tip: You can submit multiple MAP statements with the procedure.

Featured in: Example 2 on page 715

Syntax

MAP *'cgm-font'* TO *font*;

Required Arguments

'cgm-font'
identifies a font in the CGM. The name of the font must be enclosed in single quotation marks and written exactly as it appears in the font list; *cgm-font* is case sensitive. Do not include the font list number in *cgm-font*.

font
specifies the SAS/GRAPH font to which the CGM font is mapped. You can specify software fonts or hardware fonts for the destination device. You can also use fonts that are created by the GFONT procedure.

Note: Remember to specify the libref GFONT0 with a LIBNAME statement if *font* is a user-generated font. △

By default, the GIMPORT procedure maps all of the CGM fonts to the font specified by the FTEXT= graphics option or, if the FTEXT= graphics option is not used, to the default hardware font, NONE.

Details

If the CGM includes the FONT LIST element, the GIMPORT procedure automatically lists the CGM font names in the log. Use this list to select the fonts for mapping. For example, suppose the font list includes the following entry:

3. Times Roman

If the LOG window displays the message "WARNING: Invalid font index *n*," you can map the Times Roman font to the SAS/GRAPH font CENTX with the following statement:

```
map 'Times Roman' to centx;
```

SCALE Statement

Enlarges or reduces the graphics output by increasing or decreasing the values of the *x* and *y* coordinates.

Requirements: You can submit only one SCALE statement.

Tip: You can submit the SCALE statement alone or in conjunction with the TRANSLATE statement, but the SCALE statement is always processed first.

Featured in: Example 2 on page 715

Syntax

SCALE X=*factor* | Y=*factor* | X=*factor* Y=*factor*;

Required Arguments

At least one of the following arguments is required; both may be used and can be listed in either order:

X=*factor*
specifies the enlargement or reduction of the values of the *x* coordinates. *Factor* is the number by which these values are multiplied and cannot be less than or equal to 0. By default, X=1. Values less than 1 reduce the size of the graphics output while values greater than 1 increase the size of the graphics output. There is no limit on the size of *factor*.

Y=*factor*
specifies the enlargement or reduction of the values of the *y* coordinates. *Factor* is the number by which these values are multiplied and cannot be less than or equal to 0. By default, Y=1. Values less than 1 reduce the size of the graphics output while values greater than 1 increase the size of the graphics output. There is no limit on the size of *factor*.

Details

If the shapes in the imported graphics output are too narrow, you can make them wider by increasing the values of the *x* coordinate. To make the elements in the graphics output twice as wide, specify X=2. To make them half as high, specify Y=.5.

For example, if the values of the *x* coordinates range from 5 to 50 and if in the SCALE statement the factor for X= is specified as 2, then the values of all of the *x* coordinates are multiplied by 2 and the range of these values increases. The new range is 10 to 100. And if the values of the *y* coordinates range from 0 to 25 and if in the SCALE statement the factor for Y= is specified as .5, then the values of all of the *y* coordinates are multiplied by .5 and the range of these values decreases. The new range is 0 to 12.5.

If you specify a factor that causes the graphics output to exceed the size of the graphics output area, the procedure draws as much of the graphics output as will fit in the available space.

TRANSLATE Statement

Adjusts the location on the display of the graphics output imported by the procedure. Graphics output can be shifted left or right by offsetting the *x* values or shifted up or down by offsetting the *y* values.

Requirements: You can submit only one TRANSLATE statement.

Tip: You can submit the TRANSLATE statement alone or in conjunction with the SCALE statement but the SCALE statement is always processed first.

Featured in: Example 2 on page 715

Syntax

TRANSLATE X=*offset* | Y=*offset* | X=*offset* Y=*offset* ;

Required Arguments

At least one of the following arguments is required; both may be used and can be listed in either order:

X=*offset*
specifies the number of units in percent of the display area to move the graphics output right (positive numbers) or left (negative numbers). The value of *offset* is added to the value of the *x* coordinate. By default, X=0.

Y=*offset*
specifies the number of units in percent of the display area to move the graphics output up (positive numbers) or down (negative numbers). The value of *offset* is added to the value of the *y* coordinate. By default, Y=0.

Details

The TRANSLATE statement adjusts the position of the graphics output without changing its size. The amount of the *offset* that is specified for X= or Y= in the TRANSLATE statement is the amount that the graphics output is moved.

For example, suppose your imported graphics output is positioned in the upper-left corner of the display. To move it right 10% and down 5%, use the following statement:

```
translate x=10 y=-5;
```

Examples

The following examples illustrate major features of the GIMPORT procedure. For illustration purposes, these examples create a CGM using SAS/GRAPH software and import the resulting CGM by using the GIMPORT procedure. Ordinarily, you would use the GIMPORT procedure to import graphics output that is generated by another software package.

Note: Because this example uses a CGM device driver to produce a graphics stream file, you may need to respecify a device driver for your output device. In addition, these examples use the HSIZE= and VSIZE= graphics options to set a specific size for the graphics output area for the CGM so that the second example can illustrate the use of

the SCALE and TRANSLATE statements. Depending on the output device that you are using, you may need to adjust the HSIZE= and VSIZE= values in this example and the values in the SCALE and TRANSLATE statements in the second example. △

Example 1: Creating and Importing a CGM

Procedure features:

GIMPORT statement options:

FILEREF=
FILETYPE=
FORMAT=

Other features:

FILENAME statement
GOPTIONS statement

Sample library member: GR17N01

This example creates a CGM in binary format by directing SAS/GRAPH output to a graphics stream file (GSF) and using a CGM device driver. It uses the GIMPORT procedure to import the resulting CGM into SAS/GRAPH where it can be viewed and stored in a catalog. (See Chapter 2, "SAS/GRAPH Programs," on page 21 for additional information on catalog entries and graphics stream files.) The output shows the imported version of the graphic. Note that the output uses the default font because the specified fonts are unavailable. Example 2 on page 715 shows how to map these fonts to get the output that you want. Also see "About Font Mapping" on page 708 for additional information.

Assign the fileref for a graphics stream file and set the graphics environment. Set graphics stream file characteristics, and select CGMCRT device drive for binary CGM.

```
filename gsasfile 'external-file';
goptions reset=global gunit=pct border cback=white
```

```
        colors=(black)
        gaccess=gsasfile gsfmode=replace
        noprompt device=cgmcrt
        hsize=5 in vsize=5 in;
```

Define titles and footnote for slide.

```
title1 f=script h=7 'Title One is SCRIPT Font';
title2 f=centb h=5 'Title Two is CENTB Font';
title3 f=zapf h=5 'Title Three is ZAPF Font';
footnote h=3 f=swiss j=r 'GR17N01 ';
```

Generate a slide. The graphics output is stored in the GSF file that was specified with the fileref and in the GOPTIONS statement.

```
proc gslide;
run;
quit;
```

Reset the graphics environment.

```
goptions reset=goptions border cback=white
        colors=(black);
```

Import the GSF file created by the CGMCRT device driver. FILEREF= specifies the fileref where the CGM is located. FILETYPE= specifies the type of file to be imported. FORMAT= specifies the format of the CGM being imported.

```
proc gimport fileref=gsasfile
             filetype=cgm
             format=binary;
run;
```

Output 17.1 on page 714 shows the font list that is displayed in the log file. The font list contains all of the fonts that are used by the CGM. The warning messages following the font list indicate which fonts can be remapped using the MAP statement.

Output 17.1 Font List

```
 .
 .
 .

 NOTE: These fonts are used in this CGM file. You may use the MAP statement
       to map these fonts to  SAS/GRAPH
fonts.
 1. SIMPLEX
 2. BRUSH
 3. CENTB
 4. CENTBE
 5. CENTBI
 6. CENTBIE
 7. CENTX
 8. CENTXE
 9. CENTXI
 10. CENTXIE
 11. GERMAN
 12. GITALIC
 13. DUPLEX
 14. COMPLEX
 15. TRIPLEX
 16. TITALIC
 17. ITALIC
 18. OLDENG
 19. SCRIPT
 20. CSCRIPT
 21. SWISS
 22. SWISSE
 23. SWISSB
 24. SWISSBE
 25. SWISSBI
 26. SWISSBIE
 27. SWISSX
 28. SWISSXE
 29. SWISSXB
 30. SWISSXB
 31. SWISSXBE
 32. SWISSI
 33. SWISSIE
 34. SWISSL
 35. SWISSLE
 36. ZAPF
 37. ZAPFE
 38. ZAPFB
 39. ZAPFBE
 40. ZAPFBI
 41. ZAPFBIE
 42. ZAPFI
 43. ZAPFIE
 WARNING: Unspecified font index 19. Font has been mapped to the default font.
 WARNING: Unspecified font index 3. Font has been mapped to the default font.
 WARNING: Unspecified font index 36. Font has been mapped to the default font.
 WARNING: Unspecified font index 21. Font has been mapped to the default font.
 .
 .
 .
```

Example 2: Adjusting the Graphics Output

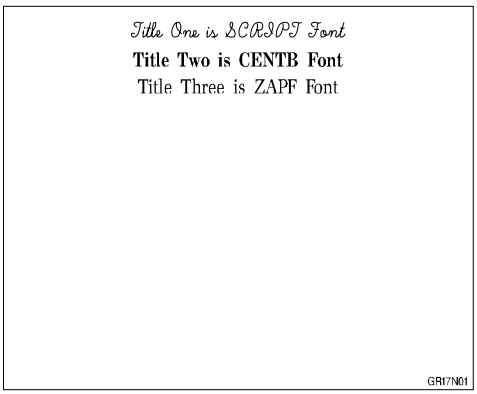
Procedure features:

SCALE statement

TRANSLATE statement

MAP statement

Sample library member: GR17N02



This example imports the CGM file that was created in the earlier example and modifies the output. This example uses the SCALE and TRANSLATE statements to correct the size and position of the imported CGM. The MAP statement is also used to substitute a SAS/GRAPH software font for a font in the CGM.

Assign the fileref for a GSF file and set the graphics environment.

```
filename gsasfile 'external-file';
goptions reset=goptions gunit=pct border cback=white
         colors=(black) htitle=6 htext=3;
```

Import the GSF file created by the CGMCRT device driver. The SCALE statement specifies the scale factor for the values of the x and y coordinates. The TRANSLATE statement specifies the amount that the imported graphics output should be moved horizontally and vertically. The MAP statements remap the fonts shown in the first example.

```
proc gimport fileref=gsasfile filetype=cgm format=binary;
   scale x=.7 y=.8;
   translate x=3.5 y=10;
   map 'SCRIPT' to script;
   map 'CENTB' to centb;
   map 'ZAPF' to zapf;
   map 'SWISS' to swiss;
run;
```

Output 17.2 on page 717 shows the font list that is displayed in the log file. Note that no warning messages follow the font list because all of the fonts that are used in the CGM have been remapped.

Output 17.2 Font List

```
 .
 .
 .

 NOTE: These fonts are used in this CGM file. You may use the MAP statement
       to map these fonts to  SAS/GRAPH
fonts.
 1. SIMPLEX
 2. BRUSH
 3. CENTB
 4. CENTBE
 5. CENTBI
 6. CENTBIE
 7. CENTX
 8. CENTXE
 9. CENTXI
 10. CENTXIE
 11. GERMAN
 12. GITALIC
 13. DUPLEX
 14. COMPLEX
 15. TRIPLEX
 16. TITALIC
 17. ITALIC
 18. OLDENG
 19. SCRIPT
 20. CSCRIPT
 21. SWISS
 22. SWISSE
 23. SWISSB
 24. SWISSBE
 25. SWISSBI
 26. SWISSBIE
 27. SWISSX
 28. SWISSXE
 29. SWISSXB
 30. SWISSXB
 31. SWISSXBE
 32. SWISSI
 33. SWISSIE
 34. SWISSL
 35. SWISSLE
 36. ZAPF
 37. ZAPFE
 38. ZAPFB
 39. ZAPFBE
 40. ZAPFBI
 41. ZAPFBIE
 42. ZAPFI
 43. ZAPFIE
 .
 .
 .
```

References

ANSI X3.122–1986, *Computer Graphics Metafile for the Storage and Transfer of Picture Description Information*.

Arnold, D.B. and Bono, P.R. (1988), *CGM and CGI: Metafile Interface Standards for Computer Graphics*, New York: Springer-Verlag.

The correct bibliographic citation for this manual is as follows: SAS Institute Inc., *SAS/GRAPH® Software: Reference, Version 8*, Cary, NC: SAS Institute Inc., 1999.

SAS/GRAPH® Software: Reference, Version 8

ISBN 1-58025-525-6

SAS Institute Inc., SAS Campus Drive, Cary, North Carolina 27513.

1st printing, October 1999