



CHAPTER

4

SAS/GRAPH Output

<i>About SAS/GRAPH Output</i>	44
<i>What You Can Do With SAS/GRAPH Output</i>	44
<i>Displaying Graphics Output on Monitors or Terminals</i>	45
<i>Displaying Graphs with the GRAPH Window</i>	45
<i>Opening and Closing the GRAPH Window</i>	45
<i>Sizing the GRAPH Window</i>	46
<i>Displaying Graphs with Direct Display</i>	46
<i>Printing Graphics Output</i>	47
<i>Printing Directly to the Device</i>	47
<i>Saving and Printing a Graphics File</i>	47
<i>Printing From a Window</i>	48
<i>Previewing Output</i>	48
<i>Storing Graphics Output in SAS Catalogs</i>	49
<i>Accessing Catalogs from Different Versions of SAS</i>	49
<i>Creating and Specifying Catalogs</i>	50
<i>Names and Descriptions of Catalog Entries</i>	51
<i>Listing and Managing Catalog Entries</i>	51
<i>Modifying SAS/GRAPH Output</i>	51
<i>Transporting and Converting Graphics Output</i>	52
<i>Transporting Catalogs across Operating Environments</i>	52
<i>Example of Transporting GRSEGS</i>	53
<i>Example of Transporting Color Maps and Templates</i>	54
<i>Example of Transporting Fonts</i>	54
<i>Example of Transporting Device Attributes and Device Entries</i>	54
<i>Converting Catalogs to a Different Version of SAS</i>	55
<i>About Exporting SAS/GRAPH Output</i>	55
<i>About Graphics Stream Files</i>	56
<i>About Graphics File Formats</i>	57
<i>Ways to Export SAS/GRAPH Output</i>	57
<i>Exporting SAS/GRAPH Output Interactively</i>	58
<i>Exporting SAS/GRAPH Output with Program Statements</i>	58
<i>General Information</i>	59
<i>Common Requirements</i>	59
<i>Naming the Output</i>	59
<i>Using the NAME= option</i>	59
<i>Using the default output name</i>	59
<i>File extensions</i>	60
<i>Example</i>	60
<i>Saving One Graph to a File</i>	61
<i>Example</i>	61
<i>Saving Multiple Graphs to One File</i>	62

Example	62
Saving Multiple Graphs to Multiple Files	64
Example	65
Replacing Existing External Files	66
Example	66
Other Ways to Assign the Destination	68
Using GACCESS=GSASFILE	68
Using GACCESS= to Explicitly Specify a Destination	68
Exporting SAS/GRAPH Output Using Modified Device Entries	68
69	
The Metagraphics Driver Facility	69

About SAS/GRAPH Output

Most SAS/GRAPH procedures produce *graphics output*, which is distinct from *SAS output*. Whereas SAS output is made up of text, graphics output is made up of commands that tell a graphics device how to draw graphics.

This chapter discusses how to display, print, store, modify, and transport SAS/GRAPH output after you have created it. For information on SAS/GRAPH language elements and programs and on how procedure output is generated, see Chapter 2, “SAS/GRAPH Programs,” on page 21.

What You Can Do With SAS/GRAPH Output

By default, SAS/GRAPH procedures that produce graphics output display the output on your monitor or terminal, using either the display manager GRAPH window or the direct display method. For details, see “Displaying Graphics Output on Monitors or Terminals” on page 45. SAS/GRAPH also can direct graphics output to a variety of other destinations. Graphics output can be

- sent directly to a graphics hardcopy device, such as a printer, plotter, or slide camera. The way you send graphics output to a hardcopy device depends on your hardware, operating environment, and system configuration. For details, see “Printing Graphics Output” on page 47.
- saved in a temporary or permanent SAS catalog entry for later replay. For details, see “Storing Graphics Output in SAS Catalogs” on page 49.
- modified with the graphics editor. You can edit or create graphics output, and save the modified graph to a catalog. For details, see “Modifying SAS/GRAPH Output” on page 51.
- transported in catalogs from one operating environment to another. For details, see “Transporting and Converting Graphics Output” on page 52.
- converted for use with a different version of SAS by converting the catalog containing the graphics output. For details, see “Converting Catalogs to a Different Version of SAS” on page 55.
- exported to external files using different graphics file formats. For example, you can save SAS/GRAPH output in formats such as CGM or PostScript for use with other software packages. For details, see “About Exporting SAS/GRAPH Output” on page 55.

In addition, you can produce graphics output as GIF files and automatically generate HTML files so that you can display your graphics output with a Web browser. For details, see Chapter 5, “Bringing SAS/GRAPH Output to the Web,” on page 71.

Regardless of the other types of output generated, SAS/GRAPH procedures always generate a SAS catalog entry. The entry is stored in the WORK.GSEG catalog unless you specify a different catalog with the GOUT= option in a PROC statement. To generate only catalog entries and suppress all other forms of graphics output, use the NODISPLAY graphics option.

Displaying Graphics Output on Monitors or Terminals

If you want to see your graphics output immediately or preview it before generating a hardcopy, you can send it to your monitor or terminal.

Note: If you are using a terminal or PC that emulates a terminal, it must be a graphics terminal. Δ

In most environments, SAS/GRAPH automatically displays graphics output in the GRAPH window. If your environment does not support the GRAPH window, SAS/GRAPH displays your graphs with the direct display method. See “Displaying Graphs with Direct Display” on page 46.

You can suppress the display of graphics output with the NODISPLAY graphics option. Suppressing a display is useful when you want to create only a catalog entry.

Displaying Graphs with the GRAPH Window

The *GRAPH window*, which is available in the SAS windowing environment, displays catalog entries of type GRSEG. You can use this window to view either the graphics output that you are currently generating or graphics output that has been stored in a catalog. You can scroll backward and forward through the catalog entries.

Some devices allow you to use up to four graph windows: GRAPH1, GRAPH2, GRAPH3, and GRAPH4. By default, the GRAPH1 window displays the graphs in the default catalog, WORK.GSEG, or the catalog that you specified with the GOUT= option.

The GRAPH window acts like other SAS windows: you can resize it and move it, and you can submit global SAS window commands from it. For details on sizing windows, see the SAS documentation for your operating environment. For a description of SAS window commands, see *SAS Language: Reference*.

GRAPH window commands control how the graphs appear in the window. For a description of these commands, refer to the SAS Help facility for the GRAPH window.

Opening and Closing the GRAPH Window

The GRAPH Window opens automatically when you submit a procedure that produces graphics output, or when you select a catalog entry from the PROC GREPLAY window or from the SAS Explorer window. You can also open the GRAPH window with the GRAPH n command. The GRAPH n command opens the default catalog WORK.GSEG, or a catalog you specify, or a specific entry.

The GRAPH n command has the following form:

```
GRAPH $n$  << libref.> catalog-name <.entry-name <.GRSEG>>>
```

n

is a number from 1 to 4 that indicates which GRAPH window to open.

libref

points to the library where the catalog is or will be stored.

catalog-name

is the name of the catalog whose contents you want to view. The default is WORK.GSEG. If the specified catalog does not contain any graphics entries, the window opens but is empty. The catalog assignment is temporary and remains in effect only while the GRAPH window remains open. To change the catalog, resubmit the GRAPH*n* command.

entry-name

is the name of the catalog entry that you want to view. If you omit *entry-name* or the entry does not exist, the last graph in the catalog is displayed. If you specify *entry-name*, you must also supply the libref and catalog name.

GRSEG

is the type of catalog entry.

To close the GRAPH window, issue the END command.

Sizing the GRAPH Window

The default size of a GRAPH window depends on the display device.

CAUTION:

Resizing the GRAPH window after you have displayed a graph can distort the graphs. If you replay a graph in a GRAPH window that is a different size from the size at which you created the graph, the graph may be distorted. Distortion occurs if the new HSIZE and VSIZE values do not maintain the width-to-height ratio of the original window. \triangle

Doing any of the following may change the HSIZE and VSIZE values and consequently the size of the GRAPH window:

- changing the device driver
- specifying a target device
- specifying dimensions with the HSIZE= and VSIZE= graphics options.

In addition, resizing the GRAPH window can reduce the number of cells available for the output. In order to display some types of graphs in a reduced GRAPH window, you may need to increase the number of cells either by using the HPOS= and VPOS= graphics options or by modifying the device driver. To ensure that graphs have an adequate number of cells, use a full-size GRAPH window when creating the graphs. You can then reduce the window to replay the graphs. For more information on dimensions and cell size, see “About the Graphics Output Area” on page 29.

If you create most or all of your graphs at the same size or with the same aspect ratio, you can avoid distortion if you size the window before you draw the graphs and then use the WSAVE command to save the position and dimensions of the window. If you resize the GRAPH windows and do not use the WSAVE command, the new size is not saved, even in the same SAS session.

Displaying Graphs with Direct Display

Environments that do not support the GRAPH window use the direct display method. With this method, your display is cleared and the graph appears when you run a graphics procedure.

If the procedure produces more than one graph, you are prompted to press ENTER between each graph for the next one to be displayed. To return to your program, press END or ENTER after the last graph.

You can display the graphs automatically and control the amount of time between each graph with the GWAIT= and NOPROMPT graphics options. GWAIT= specifies the

number of seconds before the next graph is drawn. NOPROMPT suppresses the delay between the graphs displayed. In this case, SAS/GRAPH automatically returns to your program after the last graph displays. (See Chapter 9, “Graphics Options and Device Parameters Dictionary,” on page 301 for a complete description of these graphics options.)

With the direct display method, you cannot scroll to other graphs in the catalog. To redisplay graphs, you must rerun the procedure or replay the catalog entries. For information on replaying graphs, see Chapter 26, “The GREPLAY Procedure,” on page 919.

Printing Graphics Output

You can print your SAS/GRAPH output on hardcopy devices such as laser printers and plotters. In addition, you can send the output to cameras or film recorders. Regardless of the destination, there are several ways to produce hardcopy of your SAS/GRAPH output:

- Print SAS/GRAPH program output directly to a hardcopy device.
- Print SAS/GRAPH program output by creating an external file of graphics commands called a graphics stream file (GSF), saving it to disk, and printing the file with host commands.
- Print the displayed graph directly from the GRAPH window or the Graphics Editor window.

The following sections provide a general description of these methods.

Operating Environment Information: Whatever way you choose, the exact steps for printing graphics depend on the hardcopy device you are using and on the environment in which you are using it. For complete information on printing graphics output in your operating environment, see the SAS Help facility for SAS/GRAPH and the SAS companion for your operating environment. Δ

Printing Directly to the Device

You can send graphics output directly to a hardcopy device by sending the graphics commands directly to the device or to a device port.

On most systems you can use any of the following methods to print directly to a device:

- Use a FILENAME statement, a GOPTIONS statement, and a SAS/GRAPH native device driver. The FILENAME statement defines a fileref that points to the print commands. The GOPTIONS statement references the fileref, assigns the device, and specifies any additional parameters.
- Use the GDEVICE procedure to modify a SAS/GRAPH device entry to spool output directly to a printer. See Chapter 15, “The GDEVICE Procedure,” on page 651 for information on adding host commands to a device entry.
- Use the host printing interface. (Not all operating environments support this method.)

For detailed instructions on each of these methods, refer to the SAS Help facility for SAS/GRAPH.

Saving and Printing a Graphics File

There are two steps to printing graphics output from an external file:

- 1 Save your SAS/GRAPH output in an external file. For information on creating external files, see “About Exporting SAS/GRAPH Output” on page 55 and the SAS Help facility for SAS/GRAPH.
- 2 Print the file from your host environment. The host commands vary across operating environments and spooling utilities. See the SAS companion for your operating environment for more information on printing.

You can perform these two steps separately or combine them by incorporating the host printing commands into your program or device driver. In any case, you must choose a graphics file format that is compatible with your hardcopy device. For example, if you are using a PostScript printer, be sure to specify one of the PostScript device drivers supplied in SASHELP.DEVICES. This technique is frequently used on multi-user systems in which the output devices are shared.

You can use any of the following methods to create and print an external file:

- Use the FILENAME and GOPTIONS statements to create the graphics stream file. To route the output directly to the printer, include the print commands in the SAS/GRAPH statements. Otherwise, save the file to disk and use a host command to spool the file to a spooler for the device.
- Use the GDEVICE procedure to modify a SAS/GRAPH device driver to save the output to an external file and spool the output directly to a printer. See Chapter 15, “The GDEVICE Procedure,” on page 651 for information on modifying device entries.
- Use the host printing interface. (Not all operating environments support this method.)

For detailed instructions on each of these methods, refer to the SAS Help facility for SAS/GRAPH.

Printing From a Window

In some operating environments you can print directly from the GRAPH window or the Graphics Editor window by using the PRINT command in conjunction with the TARGETDEVICE= graphics option. To do this, use TARGETDEVICE= to specify the printer driver and use DEVICE= to specify the driver for the terminal or display on which you want to preview the output.

The driver specified by TARGETDEVICE= determines the characteristics of the printed output and sends the graphics output to either an output device or a graphics stream file. The driver specified by DEVICE= controls some characteristics of the output displayed in the window. In some cases, you may need to use a FILENAME and GOPTIONS statements to define the destination of the PRINT command.

Note: If you have not previously specified TARGETDEVICE= when you issue the PRINT command in the window, SAS/GRAPH prompts you for a device for the printed output. However, in this situation the output that is sent to the printer reflects the characteristics of the display device driver. To ensure that your printed output matches what you see on your display, always use TARGETDEVICE= with the PRINT command. Δ

For details on printing directly from the GRAPH or Graphics Editor window, refer to the SAS Help facility for SAS/GRAPH. See “TARGETDEVICE” on page 393 for a complete description of TARGETDEVICE=.

Previewing Output

If you want to preview how a graph is going to appear on another device before you send it to that device, you can use the TARGETDEVICE= graphics option. For example,

to preview output on your display as it would appear on a color PostScript printer, include TARGETDEVICE= in a GOPTIONS statement and specify the driver for the printer:

```
goptions targetdevice=pscolor;
```

The output is displayed on your screen using

- the orientation of the target device. As a result, the graph may not cover the entire display area of the preview device.
- the values of either the LCOLS and LROWS pair or the PROWS and PCOLS pair, depending on the orientation of the target device.
- the default colors list of the target device.
- the values of the HSIZE and VSIZE device parameters for the target device.
- the value of the CBACK device parameter for the target device.

All other device parameter values, including the destination of the output, come from the current device entry. Therefore, the output displayed by TARGETDEVICE= may not be an exact replication of the actual output, but it is as close as possible.

See "TARGETDEVICE" on page 393 for a complete description of TARGETDEVICE=.

Storing Graphics Output in SAS Catalogs

When you run a SAS/GRAPH procedure that produces graphics output, a copy of the graphics output is always stored in a SAS catalog. A *catalog* is a type of SAS file in which you can store different types of information, called *catalog entries*. Catalog entries of type GRSEG store graphics output. In addition, SAS/GRAPH procedures create other types of catalog entries such as DEV, CMAP, FONT, and TEMPLATE. You can store multiple catalog entries in one catalog.

You can use catalog entries to store graphics output that you want to display again without having to rerun the program. Use the GRAPH window, the GREPLAY procedure, or the SAS Explorer window to redisplay graphics output stored in catalog entries.

SAS catalogs can be either temporary or permanent. *Temporary catalogs* are erased at the end of each SAS session and cannot be retrieved. *Permanent catalogs* are retained until you delete them. Therefore, they exist after the end of a SAS session and can be used in later SAS sessions.

Unless you select another catalog, either temporary or permanent, SAS/GRAPH procedures always store a copy of your graph in the temporary catalog WORK.GSEG, regardless of the other forms of graphics output that you choose. By default, each graph is *appended* to the catalog. The graphs in WORK.GSEG are erased when you end the SAS session.

Accessing Catalogs from Different Versions of SAS

CAUTION:

You can read Version 6 catalogs from Version 7 and Version 8, but you cannot write to them unless you port them. Δ

In some cases you have to specify an engine to read a Version 6 catalog from the current SAS version:

- If you are running the current version and you assign a libref that points to a library that contains only Version 6 catalogs, the correct engine is automatically

applied and you can view the entries. However, you cannot write to the catalog without porting it.

- If you are running Version 7 or Version 8 and you assign a libref that points to a library that contains both Version 6 and the current version catalogs, the LIBNAME statement must specify the SASEB engine to view the entries in the Version 6 catalog. Even with the engine assigned, you cannot write to the catalog without porting it.

For more information, see “Transporting Catalogs across Operating Environments” on page 52.

All Version 7 and Version 8 applications that support catalog entries that also existed in Version 6 should be able to transparently read those Version 6 catalog entries. That is, if a Version 7 or Version 8 user specifies the SASEB (Version 6 compatibility) engine on the LIBNAME statement, the application should be able to read and process any Version 6 data that the SASEB engine may return.

The Version 7 and Version 8 compatibility engines also support write access to Version 6 catalogs.

Creating and Specifying Catalogs

To create or specify a catalog for graphics output, use the GOUT= option in a PROC statement. The general form is

```
PROC procedure-name GOUT=< libref.> catalog-name;
```

procedure-name

is the graphics procedure you are running.

<*libref.*>*catalog-name*

is the name of a catalog where you want the output stored. If the specified catalog already exists, the procedure simply adds a catalog entry to the catalog. If the catalog does not exist, it is created.

For a temporary catalog, specify the name of the catalog and omit the libref. (This is a *one-level name*.) The temporary catalog is stored in the WORK library and erased when you end the SAS session.

For a permanent catalog, specify both a libref and a catalog name. (This is a *two-level name*.) *Libref* must already be assigned to a SAS data library that contains or will contain the catalog. For more information on assigning a libref, see “LIBNAME Statement” on page 25.

When you run the procedure, the output is automatically appended to the end of the specified catalog. If the procedure produces multiple graphics, then multiple entries are added to the catalog.

If you want a catalog entry to replace *all* of the existing entries in the catalog, you can use the following statement:

```
goptions goutmode=replace;
```

CAUTION:

Be careful using GOUTMODE=REPLACE. GOUTMODE=REPLACE replaces *all* existing entries in the output catalog. You cannot retrieve previous entries once they have been replaced. See “GOUTMODE” on page 343 for more information. \triangle

Names and Descriptions of Catalog Entries

SAS/GRAPH software always assigns a name and a description to each catalog entry so you can identify it. By default, the names and descriptions are determined by the procedure. For example, a graph produced by the GCHART procedure would be assigned the name GCHART and a description like PIE CHART OF MONTH.

By default, SAS/GRAPH appends each new entry to the catalog. If you create more than one graph with a procedure during a SAS session and the graphs are stored in the same catalog, SAS/GRAPH software adds a number to the end of the name of subsequent catalog entries. This number makes the names unique within the catalog. For example, if you create three graphs with the GCHART procedure during the same SAS session, the catalog entries are named GCHART, GCHART1, and GCHART2. SAS/GRAPH software uses this naming convention whether catalog entries are being stored in a temporary or permanent catalog.

You can supply a name and description when you create the graph by using the NAME= and DESCRIPTION= options. If you create more than one graph of the same name, SAS/GRAPH software increments the specified name just as it does the default names.

Listing and Managing Catalog Entries

You can use the SAS Explorer window or the PROC GREPLAY window to display a list of all of the entries in a catalog. To open the Explorer window for a specific catalog, use the CAT command:

CAT *libref.catalog-name*

To open the PROC GREPLAY window, submit the PROC GREPLAY statement with or without a catalog specification. For details, see Chapter 26, “The GREPLAY Procedure,” on page 919.

Either method lets you view individual entries once you have displayed the list. For more information, see “Displaying Graphics Output on Monitors or Terminals” on page 45.

There are several ways to manage catalogs and catalog entries:

- The GREPLAY procedure can rename, delete, copy, or group graphics output that is stored in either temporary or permanent catalogs.
- The CATALOG procedure can copy or move an entire catalog or selected entries within a catalog, and can rename or delete catalog entries. For details, see *SAS Procedures Guide*.

Modifying SAS/GRAPH Output

The graphics editor is an interactive application that you can use from any GRAPH window to modify graphs produced with SAS/GRAPH software or imported from other graphics software.

You can invoke the graphics editor in several ways:

- To edit a graph that you are viewing in the GRAPH window, select Edit → Edit current graph.
- To open a GRAPH window and invoke the graphics editor without displaying an existing graph, select Tools → Graphics Editor from the SAS Explorer window. After the window appears, you can either open an existing graph or create a graphic image by drawing graphics elements in the window.

- To open a GRAPH window and edit a graph that is stored in a catalog, submit the GRAPH command from the command line of the SAS Explorer window and add EDIT to the end of the command. Separate the two commands with a semicolon (;). For example, the following statement opens the GRAPH3 window, displays the specified graph, and activates the graphics editor:

```
graph3 mylib.grafcat.slid1; edit
```

You can use the graphics editor to change graphics elements already displayed in the graph such as text, lines, and polygons, or you can add graphics elements to the graph. You can create, delete, or duplicate a graphics element and select, change, move, or resize it. You can also change an element's properties, such as its color or fill pattern, or its values, such as the coordinates of one of its points. For instance, you can move a bar from one side of the display area to the other, make a copy of it and place it in yet another place, and change its fill color and pattern. You can undo and redo changes, zoom in or out on the display, move an element to the foreground or background, and read in or link to another graph that has already been created.

After you have made changes, you can save your modifications to a catalog and send the modified graph to an output device, such as a printer.

For a complete description of the graphics editor, refer to the SAS Help facility for SAS/GRAPH.

Note: Modifying your graphics output with the graphics editor does not change the SAS programs that generated that output. Δ

Transporting and Converting Graphics Output

You can use the following methods to transport and convert graphics output within the SAS System:

- Use the CPORT and CIMPORT procedures in base SAS software to transport catalogs that contain graphics output to other operating environments that are running the same version of SAS/GRAPH software.
- Use a LIBNAME statement and the CATALOG procedure to convert catalogs from Version 6 to Version 7 or Version 8.

Transporting Catalogs across Operating Environments

Use the CPORT and CIMPORT procedures to transport catalogs and catalog entries from one machine to another machine running in a different operating environment. In addition to graphics output stored in GRSEG catalog entries, SAS/GRAPH software produces several other files that you can transport from host environment to host environment. These other files include

- colors maps
- templates
- fonts
- device descriptions.

To transport catalog entries that contain graphics output (catalog entries of type GRSEG), follow these steps:

- 1 Use the CPORT procedure to create a transport file from the catalog entries in the current host environment. A transport file is a sequential file that contains the catalog in SAS transport format. To create a transport file, you must specify a catalog to be converted and a fileref for the transport file.

To retain the original order of the GRSEG entries in the catalog, use `SELECT=` in the `PROC CPORT` statement to export individual graphs in the order they were created. Otherwise, when you use the `GREPLAY` procedure to list the graphics entries in the imported catalog, the procedure will list the entries in alphabetical order, rather than the order in which they were created. *Note:* Only the `GREPLAY` procedure can list catalog entries in the order they were created. All other procedures list entries in alphabetical order.

To export a catalog that contains groups of entries created using the `GREPLAY` procedure, you must use `SELECT=` in the `PROC CPORT` statement to select the names of the groups, rather than the names of individual graphs, to be included in the transport file. If you export the entire catalog without using `SELECT=`, the groups are not maintained in the catalog created when you import the transport file in the new host environment.

When you use the `CPORT` procedure, messages in the SAS log identify the catalog entries that have been placed in the transport file. If the catalog entry was created by replaying several graphs into a template, the log messages list the names of all of the entries that contributed to the templated graph.

- 2 Move the transport file to the target machine, if necessary. You must move the transport file in binary format. If you do not move the transport file in binary format, the `CIMPORT` procedure cannot read the file you create.

Operating Environment Information: Use communications software or tape to move the transport file. Refer to the documentation for your network or standard procedures for using tape files. Δ

- 3 Once you have moved the transport file to the target machine, import the transport file into a catalog in the new host environment using the `CIMPORT` procedure. The entries are imported in the order specified in `SELECT=` in the `PROC CPORT` statement used to create the transport file.

The `SELECT=` option in the `PROC CIMPORT` statement does not affect the order of the imported entries.

Note: You must use the `CIMPORT` procedure from the current version of the SAS System. The `CIMPORT` procedure in a previous release cannot read a transport file created by the `CPORT` procedure in the current version. For details on using the `CPORT` and `CIMPORT` procedures, see the *SAS Procedures Guide*. Δ

Example of Transporting GRSEGS

This example shows how to port three entries from the catalog `MYLIB.GRAPH.S`. First, the `CPORT` procedure writes selected graphs from `MYLIB.GRAPH.S` to the transport file `TRANFILE`. The `SELECT` option names the graphs to be ported.

```
libname mylib 'SAS-data-library';
filename tranfile 'external-file';

proc cport file=tranfile
  catalog=mylib.graphs
  select=(GPLOT.GRSEG, GPLOT1.GRSEG, GPLOT3.GRSEG);
run;
```

Once the transport file has been moved to the new host environment using communications software or tape, the `CIMPORT` procedure creates a new catalog called `MYLIB.GRAPH.S` on the new machine.

```

libname mylib 'SAS-data-library';
filename tranfile 'external-file';

proc cimport catalog=mylib.graphs
    infile=tranfile
    select=(GPLOT.GRSEG, GPLOT1.GRSEG, GPLOT3.GRSEG);
run;

```

Example of Transporting Color Maps and Templates

To transport color maps (catalog entries of type CMAP) and templates (catalog entries of type TEMPLATE) from one host environment to another, use the CPORT and CIMPORT procedures. For example, you could export a color map from the NEWLIB.CMAPS catalog using the following statements:

```

filename tranfile 'external-file';
libname newlib 'SAS-data-library';

proc cport file=tranfile catalog=newlib.cmaps select=(mymap.cmap);
run;

```

After moving the transport file to the new host environment, you can import the color map using the following statements:

```

filename tranfile 'external-file';
libname newlib 'SAS-data-library';

proc cimport infile=tranfile catalog=newlib.cmaps;
run;

```

Example of Transporting Fonts

To transport fonts (catalog entries of type FONT) from one operating system to another, use the CPORT and CIMPORT procedures. For example, you could export a font from the GFONT0.FONTS catalog using the following statements:

```

filename tranfile 'external-file';
libname gfont0 'SAS-data-library';

proc cport file=tranfile
    catalog=gfont0.fonts
    select=(figures.font);
run;

```

After moving the transport file to the new host environment, you can import the font using the following statements:

```

filename tranfile 'external-file';
libname gfont0 'SAS-data-library';

proc cimport infile=tranfile catalog=gfont0.fonts;
run;

```

Example of Transporting Device Attributes and Device Entries

To transport device entries (catalog entries of type DEV) from one operating environment to another, use the CPORT and CIMPORT procedures. For example, you

could export a device entry from the GDEVICE0.DEVICES catalog using the following statements:

```
filename tranfile 'external-file';
libname gdevice0 'SAS-data-library';

proc cport file=tranfile
  catalog=gdevice0.devices
  select=(cgm.dev);
run;
```

After moving the transport file to the new host environment, you can import the device entry using the following statements:

```
filename tranfile 'external-file';
libname gdevice0 'SAS-data-library';

proc cimport infile=tranfile catalog=gdevice0.devices;
run;
```

Converting Catalogs to a Different Version of SAS

To convert catalogs to a different version of SAS, for example from Version 6 to Version 8, use the LIBNAME statement and the CATALOG procedure.

Note: You will not be able to use your old catalogs without transporting them first. △

Before using PROC CATALOG, you must assign librefs to both catalogs and specify the Version 6 Compatibility Engine (saseb) on the input catalog libname. Then use PROC CATALOG with a COPY statement to convert a catalog from Version 6 to Version 7 or Version 8. For details on using the CATALOG procedure, see the *SAS Procedures Guide*.

For example, the following statements can be submitted from Version 8 to assign the Version 6 Compatibility Engine and convert a catalog from Version 6 to Version 8.

```
libname v6lib saseb 'SAS-data-library';
libname v8lib 'SAS-data-library';

proc catalog catalog=v6lib.v6cat;
  copy out=v8lib.v8cat;
run;
```

About Exporting SAS/GRAPH Output

By default, SAS/GRAPH output is stored in SAS catalogs as catalog entries of type GRSEG. These entries can be viewed and manipulated within the SAS System and, in some operating environments, can be printed directly as hardcopy. However, you may want to use your SAS/GRAPH output outside of the SAS System. For example, you may want to

- import your graphs into other software packages
- use host system commands or applications to print or manage your graphs
- run batch processes to create and print multiple copies of your graphs
- create graphics output and HTML files that enable you to display SAS/GRAPH output on the Web.

In order to do these kinds of things with your SAS/GRAPH output, you must export your graphs from SAS/GRAPH, using a different graphics file format, such as CGM, GIF, or TIFF, and store them in external files.

For information on creating SAS/GRAPH output for Web publishing, see Chapter 5, “Bringing SAS/GRAPH Output to the Web,” on page 71. For information on SAS/GRAPH language elements and programs and on how procedure output is generated, see Chapter 2, “SAS/GRAPH Programs,” on page 21. For information on using and managing SAS/GRAPH output, see Chapter 4, “SAS/GRAPH Output,” on page 43.

About Graphics Stream Files

When you export SAS/GRAPH output, you run the output through a device driver that creates a *graphics stream file* or GSF. A GSF is an external file that contains graphics commands. Typically, you select a device driver that produces the type of graphics file format that you want, such as CGM, PS or EPS, GIF, or TIFF, although you can select a driver that sends the output directly to a printer or other hardcopy device without creating an external file. You can specify the exact name and location of each file or assign a default location to which all files are sent.

Note: You can also use the Output Delivery System (ODS) or SAS/GRAPH device drivers to generate SAS/GRAPH output as HTML and GIF files that you can view with a Web browser. Details are discussed in “Using SAS/GRAPH Web Drivers to Create Web Output” on page 77. \triangle

Once you have created a GSF, you can

- print the file using host commands
- view the file with an appropriate viewer or browser
- edit the file with the appropriate editing software
- import the file into other software packages.

Note: A GSF is different from a SAS/GRAPH catalog entry. A GSF is an external file that is independent of SAS, and a catalog entry is a type of SAS file. Consequently, you use host commands to manipulate a GSF independent of the SAS System, whereas you must use the SAS System to manipulate SAS catalog entries. For example, to view graphics output stored in a catalog you must use the GREPLAY procedure or the GRAPH window. \triangle

About Graphics File Formats

You can export your SAS/GRAPH output in many different graphics file formats. These are some of the most common formats that SAS/GRAPH software supports:

BMP	Windows bitmap
CGM	Computer graphics metafile
EPS	Adobe's encapsulated PostScript language
GIF	GIF format
HP-GL	Hewlett Packard's Graphics Language (plotter control language)
JPEG	JPEG format
PBM	Portable bitmap
PDF	Portable Document Format
PNG	Portable Network Graphics format
PS	Adobe's PostScript language
PPM	Portable pixmap
TIFF	Tagged Image Format File

The type of graphics file format that you choose depends on how you are going to use the output. If you are planning to import the graph into other software products, such as Microsoft Excel or Word Perfect, you may prefer to create a CGM file. These vector-based files are usually smaller than bitmapped files, and they can be edited. In addition, they use hardware fonts and provide a clear image on high-resolution devices.

If you want to display the graph on a Web page, or import it into software that cannot accept vector graphics like CGM, you will need to create a bitmapped file using a format such as GIF or TIFF.

Note: The HTML and WEBFRAME drivers generate both HTML files and GIF files specifically for use with a Web browser. For details, see "Using SAS/GRAPH Web Drivers to Create Web Output" on page 77. Δ

Most software packages that process graphics input can accept one or more of these file formats. Check the documentation for the hardware or software product to which you want to send the graph to determine what file format or formats it can use.

For a complete list of graphics file formats available with SAS/GRAPH in your operating environment, refer to the Device Help for SAS/GRAPH in the SAS Help facility.

Ways to Export SAS/GRAPH Output

There are several ways to send SAS/GRAPH output to an external file in a different graphics file format. You can export graphics output in these ways:

- From the GRAPH window or the Graphics Editor window, use menu selections to select the type of file format and specify a destination for the output that is displayed in the window. When you export SAS/GRAPH output in this way, you are limited to the types of file formats that the Export as Image window supports in your operating environment. For details see "Exporting SAS/GRAPH Output Interactively" on page 58.

Operating Environment Information: In Windows operating environments, WMF formats cannot be exported using this method. Δ

- Use SAS/GRAPH program statements to direct the output to a graphics stream file. When you use program statements to create a GSF, you explicitly specify a SAS/GRAPH device driver. This driver can be one of the drivers supplied with SAS/GRAPH software and stored in SASHELP.DEVICES, or a custom driver that you have created. For details see “Exporting SAS/GRAPH Output with Program Statements” on page 58.
- Create a custom device driver that contains all of the commands for producing a GSF. For details see “Exporting SAS/GRAPH Output Using Modified Device Entries” on page 68.
- Use the SAS/GRAPH web drivers or the Output Delivery System to create HTML and GIF files. For details see Chapter 5, “Bringing SAS/GRAPH Output to the Web,” on page 71.

Exporting SAS/GRAPH Output Interactively

You can export SAS/GRAPH output interactively from either the Graph window or from the Graphics Editor window. To export a graph from one of these windows, follow these steps:

- 1 Open the window and display the graph.
- 2 From the **File** menu select **Export as Image**
- 3 In the Export as Image window, select a file type and specify a destination or file name.
- 4 Close the window by choosing Save or OK (button text depends on the operating environment).

Because you can export only one catalog entry at a time, this method is most useful for quickly exporting a few graphs.

Although you can use this method to create many types of graphics stream files, you have a much larger choice of device drivers when you use SAS/GRAPH program statements to create a GSF.

Exporting SAS/GRAPH Output with Program Statements

When you use program statements to create external files for your SAS/GRAPH output, you use one of these processes:

- create one file that contains one graph
- create one file that contains multiple graphs
- create multiple files that each contain one graph.

You can send the graphics output to external files either at the time you run the program that creates the graphs, or later when you replay them from the catalog in which they are stored. For this reason, these methods are most useful for processing large quantities of output. In addition, using program statements allows you to specify exactly the device driver you want and is therefore a more flexible and powerful way of exporting SAS/GRAPH output.

The following sections provide some information common to all the processes and then describes each process in detail.

General Information

Common Requirements

Regardless of the process you use to create a GSF from a SAS/GRAPH program, you must specify the following:

- a destination for the output. This can be an aggregate file storage location (for example, a directory or a partitioned data set) or a specific file. Typically you specify the destination with a FILENAME statement and one or more graphics options. For more information, see “FILENAME Statement” on page 24.
- a device driver that creates the type of graphics output that you want.
- whether SAS/GRAPH should replace an existing file or append new records to it. By default, SAS/GRAPH replaces an existing file with newly created output *of the same name*. For details, see “Replacing Existing External Files” on page 66.

Each requirement is explained in detail in the individual process descriptions.

Naming the Output

When you are working with both catalog entries and external files, you should understand how both types of output are named.

Using the NAME= option

You can use the NAME= option in the SAS/GRAPH procedure to specify a name for the catalog entry that the procedure generates. How this name is used depends on whether the FILENAME statement points to a specific external file or to an aggregate file storage location.

- If you specify a specific filename for the external file and also use the NAME= option, the external file is assigned the name specified in the FILENAME statement and NAME= controls only the name given to the created catalog entry. When you specify the filename, you should include the appropriate file extension, such as .CGM, .GIF, or .PS).
- If you specify an aggregate file storage location instead of a specific filename, and also use the NAME= option, the name of the external file is built from the name of the catalog entry, which is determined by the value of NAME=. In this case, SAS/GRAPH supplies the appropriate file extension.

See Table 4.1 on page 60 for examples.

Using the default output name

If you omit NAME=, SAS/GRAPH uses the default naming convention to name the catalog entry, and in some cases the external file. This convention uses up to eight characters of the procedure name as the base name for the catalog entry. If the name generated by the procedure duplicates an existing entry, the name is incremented, for example, GCHART, GCHART1, GCHART2, and so forth. For details, see the description of the NAME= option for a specific procedure.

- If you specify a specific filename for the external file and omit the NAME= option, the external file uses the name specified in the FILENAME statement and the catalog entry uses the default name. When you specify the filename, you should include the appropriate file extension, such as .CGM, .GIF, or .PS.

- If you specify an aggregate file storage location instead of a specific filename, and omit the NAME= option, both the catalog entry and the external file use the default name and SAS/GRAPH supplies the appropriate file extension.

See Table 4.1 on page 60 for examples.

File extensions

When you send SAS/GRAPH output to an aggregate file storage location, SAS/GRAPH generates the name of the external file by taking the catalog entry name and adding the appropriate file extension. Most drivers provide a default extension. If a driver does not generate an extension, SAS/GRAPH uses the default extension .GSF. To specify a different extension from the one SAS/GRAPH provides, use the EXTENSION=graphics option. (For details, see EXTENSION="EXTENSION" on page 329).

Example

illustrates how SAS/GRAPH generates names for catalog entries and external files, depending on 1) whether the NAME= option is used, and 2) on the fileref specification. This illustration assumes the GSLIDE procedure and DEV=GIF:

Table 4.1 How SAS/GRAPH Generates Entry Names and File Names

<i>If..</i>	<i>And..</i>	<i>Then</i>
NAME='FRED'	fileref points to a file named 'MYSLIDE.GIF'	catalog entry name: FRED external file name: MYSLIDE.GIF
NAME='FRED'	fileref points to a storage location (for example, a directory)	catalog entry name: FRED external file name: FRED.GIF
NAME= (not specified)	fileref points to a file named 'MYSLIDE.GIF'	catalog entry name: GSLIDE external file name: MYSLIDE.GIF
NAME= (not specified)	fileref points to a storage location (for example, a directory)	catalog entry name: GSLIDE external file name: GSLIDE.GIF

Note: When the fileref points to an aggregate file storage location, the name of the catalog entry *always* determines the name of the external file. It does not matter whether the catalog entry name is the default name or a name assigned by NAME=. \triangle

CAUTION:

If the graph created by the program already exists in the catalog, a new catalog entry with an incremented name will be created and a new external file may be created rather than updating the existing file. \triangle

You cannot replace individual entries in a catalog; therefore, to replace an entry you must first delete the entry and then re-create it. Therefore, even though the contents of the external file are replaced, the catalog entry is not. Each time you submit the program, a new entry is created and the catalog entry name is incremented.

Saving One Graph to a File

The simplest way to save one graph to a file is to use the FILENAME statement, the GSFNAME= graphics option, and the default setting GSFMODE=REPLACE to create one graphics stream file. These steps describe the general process:

- 1 Use a FILENAME statement to define a fileref for the external file where you want to send the output. The file name must be the complete physical name of the external file and should include a file extension that indicates what type of graphics file you are creating, for example .GIF for a GIF file.
- 2 Assign the fileref to the GSFNAME= graphics option.
- 3 Specify the device driver with the DEVICE= graphics option.
- 4 Use the default setting GSFMODE=REPLACE so that SAS/GRAPH produces only one graph per file (unless BY-group processing is in effect). Because REPLACE is the default setting, you can omit GSFMODE=.
- 5 Submit the SAS/GRAPH program.

Note: The GSF remains open while the SAS/GRAPH procedure is running. Be sure to end the procedure by submitting another procedure step, DATA step, or QUIT statement. To be really safe, you can submit a FILENAME *fileref* CLEAR; statement to explicitly close the GSF.

Operating Environment Information: On certain systems, other graphics options may be required. For more information on creating a graphics stream file, refer to the SAS Help facility for SAS/GRAPH Device Drivers for your operating environment. Δ

Example

This example creates one GSF that contains one text slide created by a group of TITLE and FOOTNOTE statements and the GSLIDE procedure.

Define the fileref. The FILENAME statement associates the fileref GRAFOUT with the external file that is the destination for the GSF. The file extension .PS indicates that the graphics output is PostScript.

```
filename grafout 'mygraph.ps';
```

Specify graphics options for the GSF. RESET=ALL resets all global statements and graphics options. DEVICE= specifies a PostScript device driver. GSFNAME= assigns the fileref GRAFOUT as the destination for the GSF. GSFMODE=REPLACE (the default) causes the contents of the external file to be replaced each time the graphics procedure is run.

```
goptions reset=all
         device=pscolor
         gsfname=grafout
         gsfmode=replace
         ftext=swissb;
```

Produce one text slide. NAME= specifies the name that is assigned to the catalog entry created by the procedure. If you omit NAME=, SAS/GRAPH uses the default naming convention to name the entry.

```
proc gslide border name='proposal';
  title1 h=4 'Proposed Design Improvements: ';
  title2 h=3 '* Increase Stability';
  title3 h=3 '* Increase Speed';
  title4 h=3 '* Reduce Weight';
```

```

        footnote h=2 j=1 'ABC Company';
run;
quit;

```

When you submit these statements, SAS/GRAPH does the following if no graphs of the same name exist in the catalog:

- Creates one catalog entry named PROPOSAL in WORK.GSEG.
- Creates one external file that contains the output from the GSLIDE procedure and sends a message to the LOG reporting the number of records and the name of the file to which they were written. The file name is the one specified in the FILENAME statement.

Because the destination is a specific file and because GSFMODE=REPLACE, each time you run the program it replaces the contents of the external file. Therefore, this method is particularly useful when you want to update an external file by resubmitting an existing program.

However, if there is more than one run of a graphics procedure in this program, the file would contain only the graphics output from the last procedure run because this program replaces the external file each time a graphics procedure is run.

Note: Even though the contents of the external file are replaced, the catalog entry is not. Unless you explicitly delete the existing entry named PROPOSAL, each time you submit the program, a new entry is created and the catalog entry name is incremented. This table illustrates what happens if you submit the above program three times without deleting the catalog entries: Δ

Pass	Catalog entries	File name
1	PROPOSAL	mygraph.ps
2	PROPOSA1	mygraph.ps
3	PROPOSA2	mygraph.ps

Note that each new catalog entry replaces the contents of the external file, in this case, mygraph.ps. For more information, see “Replacing Existing External Files” on page 66.

For a complete description of the graphics options used in this example, see Chapter 9, “Graphics Options and Device Parameters Dictionary,” on page 301.

Saving Multiple Graphs to One File

If your program creates multiple graphs that you want to store in one file, you follow the same steps as those for saving one graph to one file except you specify GSFMODE=APPEND to add each new graph to the end of the file instead of replacing the file.

Example

This example stores several text slides in one external file. The program is the same as the previous example except the GOPTIONS statement specifies GSFMODE=APPEND and the GSLIDE procedure uses RUN-group processing to create multiple slides. Each slide includes the current TITLE statement and all previously defined TITLE and FOOTNOTE statements.

Define the fileref.

```
filename grafout 'mygraph.ps';
```

Specify graphics options for the GSF. GSFNAME= assigns the fileref GRAFOUT as the destination for the GSF. GSFMODE=APPEND adds each new piece of output to the end of the external file.

```
goptions reset=all
         device=pscolor
         gsfname=grafout
         gsfmode=append
         ftext=swissb
         rotate=landscape;
```

Produce four text slides. Each RUN-group generates a new catalog entry. NAME= specifies the base name for each catalog entry created by the procedure.

```
proc gslide border name='proposal';
  footnote h=2 j=1 'ABC Company';
  title1 h=4 'Proposed Design Improvements: ';
run;
  title1 h=3 '* Increase Stability';
run;
  title1 h=3 '* Increase Speed';
run;
  title1 h=3 '* Reduce Weight';
run;
quit;
```

When you submit these statements, SAS/GRAPH does the following if no graphs of the same name exist in the catalog:

- Creates a new catalog entry for each graph, beginning with the name PROPOSAL. After the first graph is created, the entry name is incremented so that subsequent graphs are called PROPOSA1, PROPOSA2, and so forth.
- Creates one external file that contains all of the output from the GSLIDE procedure. The file name is the one specified in the FILENAME statement. Each time a graph is added to the file, SAS/GRAPH sends a message to the LOG reporting the number of records and the name of the file to which they were appended.

Note: Because the destination is a specific file and because the GSFMODE= setting is APPEND, each time you run the program SAS/GRAPH adds the new graphs to the external file. If you want the file to contain only the newly created graphs, delete it before resubmitting the program. Δ

In addition, if you resubmit the program without deleting the original catalog entries, SAS/GRAPH will create four new entries that will be added to the catalog entry and four new graphs appended to the external file, both of which will contain all eight graphs.

This table illustrates what happens if you submit this program twice without deleting the catalog entries or the external file:

Pass	Catalog entries	File name	File contents
1	PROPOSAL PROPOSA1 PROPOSA2 PROPOSA3	mygraph.ps	PROPOSAL, PROPOSA1, PROPOSA2, PROPOSA3
2	PROPOSAL PROPOSA1 PROPOSA2 PROPOSA3 PROPOSA4 PROPOSA5 PROPOSA6 PROPOSA7	mygraph.ps	PROPOSAL, PROPOSA1, PROPOSA2, PROPOSA3, PROPOSA4, PROPOSA5, PROPOSA6, PROPOSA7

For more information, see “Replacing Existing External Files” on page 66.

Saving Multiple Graphs to Multiple Files

When you want your SAS/GRAPH program to create multiple files that each contain one graph, you can either

- use a FILENAME statements for each PROC statement to explicitly specify a destination file for each graph. To do this, simply repeat as many times as necessary the process described in “Saving One Graph to a File” on page 61.
- use one FILENAME statement to specify an aggregate file storage location as the destination and let SAS/GRAPH automatically name and create the files for you.

These steps describe the general process:

- 1 Use a FILENAME statement to define a fileref for the aggregate file storage location, such as a directory or PDS, where you want to send the output. Do not point to a specific file.
- 2 Assign the fileref to the GSFNAME= graphics option.
- 3 Specify the device driver with the DEVICE= graphics option.
- 4 Use the default setting GSFMODE=REPLACE so that SAS/GRAPH produces only one graph per file. Because REPLACE is the default setting, you can omit GSFMODE=.
- 5 Submit the SAS/GRAPH program.

Although the general technique is the same, there are significant differences between directing your graphics output to a specific file and directing it to an aggregate file storage location. These differences are mostly concerned with how the file is named.

When the destination is an aggregate file storage location, SAS/GRAPH not only creates the external file, it also names it. When SAS/GRAPH names an external file, it always uses the name of the entry in the output catalog. This name is either

- the name you specify in the NAME= option in the procedure
- the default name supplied by SAS/GRAPH, such as GSLIDE.

In addition, SAS/GRAPH automatically appends the correct file extension to the external file name. For example, if the output is named Q1SALES and the external file is a GIF file, the file name is Q1SALES.GIF.

This technique of building the file name from the catalog entry name affects what you do when you want to replace the contents of a file created in this way. For details, see “Replacing Existing External Files” on page 66.

Example

This example creates four text slides and stores each one in a separate external file. The program is similar to the previous examples except the fileref points to an aggregate storage location instead of to a specific file.

Define the fileref. The FILENAME statement assigns an aggregate file storage location as the destination for the files.

```
filename grafout 'external-file-location'; /* such as a directory */
```

Specify graphics options for the GSF. GSFNAME= assigns the fileref GRAFOUT as the destination for the GSF. GSFMODE=REPLACE (the default) replaces the contents of the external files with catalog entries *of the same name*.

```
goptions reset=all
         device=pscolor
         gsfname=grafout
         gsfmode=replace
         ftext=swissb
         rotate=landscape;
```

Produce four text slides. Each RUN-group generates a new catalog entry. NAME= specifies the base name for each catalog entry that is generated by the procedure. This name is also the base name for the external files.

```
proc gslide border name='proposal';
  footnote h=2 j=1 'ABC Company';
  title1 h=4 'Proposed Design Improvements:';
run;
  title1 h=3 '* Increase Stability';
run;
  title1 h=3 '* Increase Speed';
run;
  title1 h=3 '* Reduce Weight';
run;
quit;
```

When you submit these statements, SAS/GRAPH does the following if no graphs of the same name exist in the catalog:

- Creates a new catalog entry for each graph, beginning with the name PROPOSAL. After the first graph is created, the entry name is incremented so that subsequent graphs are called PROPOSA1, PROPOSA2, and so forth.
- Creates one external file for each catalog entry. The name of the file is the same as the entry name plus the extension. In this case, the files are named PROPOSAL.PS, PROPOSA1.PS, and so forth. For each file created, SAS/GRAPH sends a message to the LOG reporting the number of records and the name of the file to which they were written.

Note: Because you cannot replace individual entries in a catalog, each time you run the program SAS/GRAPH creates new catalog entries and consequently new files. If you want to replace the files, you must delete the corresponding catalog entries before resubmitting the program. Δ

This table illustrates what happens if you submit this program twice without deleting the catalog entries:

Pass	Catalog entries	File name
1	PROPOSAL	proposal.ps
	PROPOSA1	proposa1.ps
	PROPOSA2	proposa2.ps
	PROPOSA3	proposa3.ps
2	PROPOSA4	proposa4.ps
	PROPOSA5	proposa5.ps
	PROPOSA6	proposa6.ps
	PROPOSA7	proposa7.ps

Because the catalog names increment, there is never a matching file that the new catalog can replace. Therefore, unless you delete the existing entries, the program continues to create new files. To delete the existing entries, first run a GREPLAY procedure with the DELETE option:

```
proc greplay igout=work.gseg nofs;
    delete proposal proposal1 proposa2 proposa3;
```

Replacing Existing External Files

When you are working with aggregate file storage locations and automatic file naming, it is important to remember that GSFMODE=REPLACE replaces a file *only* if the name of the catalog entry is the same as the name of the file and you are using the same type of driver. For example, to replace a file named Q1SALES.PS, your program must create a catalog entry named Q1SALES, and you must also be using a PostScript driver. If the entry named Q1SALES already exists in the catalog, SAS/GRAPH will increment the name to Q1SALES1, and either create a new file with the incremented name or replace an existing file whose name matches the new incremented name.

Therefore, to replace the contents of existing external files with a new set of graphs, you must be sure that the catalog you are using does not already contain entries of the same name. There are several ways to assure that a catalog does not contain entries with the same names as your files:

- Use a temporary catalog, such as the default WORK.GSEG, to store the output and start a new SAS session. Initially, the catalog is empty.
- Use a temporary or permanent catalog and use the GREPLAY procedure to do either of the following:
 - delete the entire contents of the catalog before you submit your program
 - delete specified entries before you submit your program.

One additional method for replacing catalog entries is rarely recommended because it is easy to accidentally delete catalog entries that you did not intend to delete. If you want to replace the entire contents of the catalog and if you are running only one procedure, you can use the graphics option GOUTMODE=REPLACE. Whenever a new procedure starts, GOUTMODE=REPLACE replaces the *entire contents* of the current catalog with the new entries; it does not replace individual entries.

Example

The following example uses the GREPLAY procedure to explicitly delete specified catalog entries so that you can re-create them and replace the corresponding external files. This example uses the permanent catalog MYLIB.GRAFCAT.

- The GREPLAY procedure explicitly deletes existing catalog entries that have the same name as the entries to be created. If no entries exist, PROC GREPLAY issues a message and the program continues.
- The GSLIDE procedure generates four text slides and stores them in the catalog specified by GOUT=. NAME= specifies PROPOSAL as the base name for the catalog entries created by the procedure.

Define the libref for the permanent catalog.

```
libname mylib 'SAS-data-library';
```

Define the fileref. The FILENAME statement assigns an aggregate file storage location as the destination for the files.

```
filename grafout 'external-file-location';
```

Specify graphics options for the GSF. GSFNAME= assigns the fileref GRAFOUT as the destination for the GSF. GSFMODE=REPLACE (the default) replaces the contents of the external files with catalog entries *of the same name*.

```
goptions reset=all
         device=pscolor
         gsfname=grafout
         gsfmode=replace
         ftext=swissb
         rotate=landscape;
```

Delete existing catalog entries of the same name. The GREPLAY procedure deletes the specified catalog entries. These are the catalog names generated by the NAME= option in the procedure. If the entries do not exist, PROC GREPLAY issues a message and the program continues.

```
proc greplay nofs igout=mylib.grafcat;
  delete proposal proposal1 proposa2 proposa3;
run;
```

Produce four text slides. Each RUN-group generates a new catalog entry. NAME= specifies the base name for each catalog entry generated by the procedure. This name is also the base name for the external files.

```
proc gslide border gout=mylib.grafcat name='proposal';
  footnote h=2 j=1 'ABC Company';
run;
  title1 h=3 '* Increase Strength';
run;
  title1 h=3 '* Reduce Drag';
run;
  title1 h=3 '* Increase Resistance to Sheer';
run;
quit;
```

When you submit these statements, SAS/GRAPH does the following:

- deletes the specified entries from the catalog MYLIB.GRAFCAT.
- creates a new catalog entry in MYLIB.GRAFCAT for each slide, and increments the entry names: PROPOSAL, PROPOSA1, PROPOSA2, and PROPOSA3.
- creates one external file for each catalog entry. The file name is built from the catalog entry name. If a file of the same name already exists, SAS/GRAPH replaces the contents of the file.

Other Ways to Assign the Destination

You can use the `GACCESS=` graphics option to assign the destination for a graphics stream file. There are two ways to do this.

Using `GACCESS=GSASFILE`

This method is similar to the `GSFNAME=` method described in the previous sections.

- Use a `FILENAME` statement and assign a destination to the fileref `GSASFILE`. When you use `GACCESS`, the fileref must be named `GSASFILE`. The destination can be either a specific file or an aggregate file storage location.
- Assign `GSASFILE` to the `GACCESS=` graphics option instead of to `GSFNAME=`.

For example, these statements define and assign the fileref for an aggregate file storage location:

```
/* define a fileref for the destination */
filename gsasfile 'external-file-location';

/* assign the fileref and specify a device */
goptions reset=all gaccess=gsasfile device=gif;
```

Using `GACCESS=` to Explicitly Specify a Destination

You can also use `GACCESS=` to assign the destination and omit the `FILENAME` statement. In this case, you must also include the `SASGASTD` output format and quote the entire value. The destination can be either a specific file or an aggregate file storage location.

For example, this statement assigns a specific file location as the destination for the graphics stream file:

```
/* assign the fileref and specify a device */
goptions reset=all
gaccess='sasgastd > my-graph-file.gif'
device=gif;
```

Exporting SAS/GRAPH Output Using Modified Device Entries

If you frequently send graphics output to the same file or device, you may want to create a custom device entry that automatically sends your output to that destination. This simplifies the process by eliminating several graphics options from the `GOPTIONS` statement and allowing you to create the `GSF` by simply specifying the custom device entry with the `DEVICE=` graphics option.

To modify a device entry, use the `GDEVICE` procedure using either code-based statements or the `GDEVICE` windows. In either case, copy the original entry from the `SASHELP.DEVICES` catalog to your personal catalog (typically, `GDEVICE0.DEVICES`). Then simply change the device parameters to create an entry that produces graphics stream files by default.

Often these device parameters correspond to the graphics options in your program. For example, if your program uses the fileref `GRAFOUT`, you can enter a value of **Grafout** in the **Gsfname:** field in the Host File Options window.

See Chapter 15, “The `GDEVICE` Procedure,” on page 651 for more information on modifying device entries. Refer to the Device Help for SAS/GRAPH in the SAS Help

facility for lists of available drivers and for specific information on modifying device entries in your operating environment.

The Metagraphics Driver Facility

You can use the SAS/GRAPH Metagraphics facility to write your own device drivers to create files in other formats for use by other graphics software. The Metagraphics driver facility can be used to develop driver support for virtually any device you choose. It is device-intelligent and can support most hardware features. A user-written program is required to translate the Metagraphics metafile into the appropriate graphics language recognized by the hardware product.

For assistance in creating a Metagraphics driver, contact Technical Support.

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

Copyright © 1999 by SAS Institute Inc., Cary, NC, USA.

ISBN 1-58025-525-6

All rights reserved. Printed in the United States of America.

U.S. Government Restricted Rights Notice. Use, duplication, or disclosure of the software by the government is subject to restrictions as set forth in FAR 52.227-19 Commercial Computer Software-Restricted Rights (June 1987).

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

1st printing, October 1999

SAS® and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

OS/2®, OS/390®, and IBM® are registered trademarks or trademarks of International Business Machines Corporation.

Other brand and product names are registered trademarks or trademarks of their respective companies.

The Institute is a private company devoted to the support and further development of its software and related services.