CHAPTER 2

SAS/GRAPH Programs

Overview

In general, SAS/GRAPH programs work like other SAS programs: the SAS/GRAPH procedures use data from SAS data sets to produce output – in this case, graphics output. In addition, SAS/GRAPH programs define the output environment and control the format and destination of the graphics output. This chapter discusses SAS/GRAPH programs and explains how they produce graphics output. It describes

□ the language elements used by SAS/GRAPH programs
□ ways of running SAS/GRAPH programs

- □ where and how procedure output is produced
- □ how to control certain attributes of the procedure output.

For information on using and managing SAS/GRAPH output, see Chapter 4, "SAS/GRAPH Output," on page 43; for information on using SAS/GRAPH programs to create other kinds of graphics output, see "About Exporting SAS/GRAPH Output" on page 55. For information on bringing SAS/GRAPH output to the Web, see Chapter 5, "Bringing SAS/GRAPH Output to the Web," on page 71.

Language Elements

The language elements used by SAS/GRAPH programs include SAS/GRAPH procedures, SAS/GRAPH statements, and Annotate data sets. In addition to SAS/GRAPH language elements, your SAS/GRAPH program may include Base SAS statements and procedures that you use to process your data or control the destination or format of your program output.

SAS/GRAPH Procedures

SAS/GRAPH procedures create graphics output, process data for other SAS/GRAPH procedures to use, or manage graphics output that has been stored in a catalog. A SAS/GRAPH procedure step typically contains these statements:

PROC statement
: starts the procedure. Typically it identifies input and output data sets, and assigns a destination for graphics output. For information on data sets and data requirements, see "SAS Data Sets" on page 25. For information on assigning graphics catalogs, see "Storing Graphics Output in SAS Catalogs" on page 49.

Subordinate statements
: perform the work of the procedure; subordinate statements that generate graphs are called *action statements*. For example, the HBAR statement in the GCHART procedure is an action statement.

RUN statement
: executes the statements in the procedure step. Use the QUIT statement to end the procedure. See also "RUN-Group Processing" on page 28.

In addition, many SAS/GRAPH procedures can use the following statements:

BY statement
: causes the procedure to produce multiple graphs, each corresponding to a BY variable value. Each graph that is produced for a value of a BY variable is stored as a separate catalog entry in either the default catalog, WORK.GSEG, or in the catalog you specify with a GOUT= option in the PROC statement. See "BY Statement" on page 177 for a complete description.

NOTE statement
: adds text to the graphics output. See "TITLE, FOOTNOTE, and NOTE Statements" on page 251 for a complete description.

You can also use other SAS language statements with SAS/GRAPH procedures. See "Other SAS Language Statements" on page 23.

SAS/GRAPH Global Statements

SAS/GRAPH has its own set of statements that affect only graphics output that is generated by the SAS/GRAPH procedures and the graphics facilities Annotate and DSGI.

SAS/GRAPH global statements define or modify the titles, footnotes, legends, axes, symbols, and patterns that appear on your graphs, as well as controlling the appearance of the graph, the graphics environment, the destination of the output, and device characteristics.

You can specify these statements anywhere in your program, and they remain in effect until explicitly changed or canceled. These are the SAS/GRAPH global statements:

AXIS
: modifies the appearance, position, and range of values of axes in charts and plots.

GOPTIONS
: specifies graphics options that control the appearance of graphics elements by specifying characteristics such as default colors, fill patterns, fonts, or text height. Graphics options can also temporarily change device settings.

LEGEND
: modifies the appearance and position of legends generated by procedures that produce charts, plots, and maps.

PATTERN
: controls the color and fill of patterns that are assigned to areas in charts, maps, and plots.

SYMBOL
: specifies the shape and color of plot symbols as well the interpolation method for plot data. It also controls the appearance of lines in contour plots.

TITLE and FOOTNOTE
: add titles and footnotes to graphics output.

See Chapter 8, “SAS/GRAPH Statements,” on page 159 for complete descriptions of these statements.

Annotate DATA Step

An Annotate DATA step generates a data set of graphics commands that can be applied to SAS/GRAPH procedure output. See Chapter 10, “The Annotate Data Set,” on page 403 for information on building and using Annotate data sets. See Chapter 11, “Annotate Dictionary,” on page 427 for a complete description of all Annotate functions and variables.

Other SAS Language Statements

These SAS language statements can also be used within SAS/GRAPH procedures:

FILENAME statements
: identify external files or aggregate file storage locations that you want to use for input or output. See “FILENAME Statement” on page 24 for more information.

FORMAT statement
: assigns a format to a variable. SAS/GRAPH procedures use formatted values to determine such aspects of the graph as midpoints, axis labels, tick-mark values, and legend entries.

LABEL statement
: assigns a descriptive text string to a variable. Unless other text is specified in the SAS/GRAPH program, the label appears in place of the variable name.

LIBNAME statements
: identify SAS libraries that contain SAS data sets or catalogs that you want to use with your SAS/GRAPH programs. See “LIBNAME Statement” on page 25 for more information.

ODS statements
: direct the output from certain SAS/GRAPH procedures to the Output Delivery System.

The ODS LISTING statement directs PROC GDEVICE output to the SAS listing file.

The ODS HTML statement is used with the GIF driver to direct graphics output to one or more GIF files and create a variety of HTML files that can display the GIF files in a Web browser. See “ODS HTML Statement” on page 200 for information on using the ODS HTML statement with SAS/GRAPH procedures.

OPTIONS statement
: changes the value of one or more SAS system options.

QUIT statement
: executes any statements that have not executed and ends the procedure.

WHERE statement
: specifies observations from SAS data sets that meet a particular condition. Using a WHERE statement provides an easy way to graph a subset of your data.

For a complete description of these statements, see *SAS Language Reference: Dictionary*.

FILENAME Statement

The FILENAME statement associates a SAS fileref with an external text file or output device. With SAS/GRAPH software, you can use a FILENAME statement to

- point to a text file that you want to use for data input or output.
- assign the destination of a graphics stream file (GSF). This destination can be either a single, specific file or an aggregate file storage location, such as directory or PDS. See “About Exporting SAS/GRAPH Output” on page 55 for information on creating graphics stream files.

You can also use the FILENAME statement to route input to and from other devices. For details, see the SAS documentation for your operating environment.

A FILENAME statement that points to an external file has this general form:

FILENAME *fileref 'external-file'*;

fileref
: is any SAS name.

external-file
: is the physical name of the external file or aggregate file storage location you want to reference. For details on specifying the physical names of external files, see the SAS documentation for your operating environment.

For a complete description of the FILENAME statement, see *SAS Language Reference: Dictionary*.

LIBNAME Statement

The LIBNAME statement associates a libref with a SAS data library. A SAS data library can be either temporary or permanent. Typically, SAS data libraries used with SAS/GRAPH software contain

- □ SAS files for data input and output.
- □ SAS catalogs that contain maps, fonts, or device entries.
- □ SAS catalogs that contain graphics output. These catalogs are often stored in permanent libraries. See "Storing Graphics Output in SAS Catalogs" on page 49 for information on storing graphics output in a permanent catalog.

The LIBNAME statement has this general form:

LIBNAME *libref* '*SAS-data-library*';

libref
is any SAS name.

SAS-data-library
is the physical name for the SAS data library on your host system. For details on specifying *SAS-data-library*, see the SAS documentation for your operating environment.

The libref WORK is reserved; it always points to an area where temporary data sets and catalogs are kept. The contents of WORK are deleted when you exit a SAS session.

For a complete description of the LIBNAME statement, see *SAS Language Reference: Dictionary*.

SAS Data Sets

Many SAS/GRAPH procedures use SAS data sets as input or output. When a SAS/GRAPH procedure requires an input SAS data set, you usually specify the data set with the DATA= option in the procedure statement, as shown in this example:

```
proc gplot data=reflib.stocks;
```

If you omit the DATA= option, the procedure uses the value of the SAS system option _LAST_=. The default for _LAST_= is the most recently created SAS data set (either permanent or temporary) in the current SAS job or session.

If you do not specify a data set and no data set has been created in the current SAS session, an error occurs and the procedure stops.

Most of the procedures that read data sets or create output data sets accept data set options. SAS data set options appear in parentheses after the data set specification, as shown in this example:

```
proc gplot data=reflib.stocks(where=(year=1997));
```

For more information on SAS data sets and other data processing details, see *SAS Language Reference: Concepts*. For a complete discussion of SAS data set options and SAS system options, see *SAS Language Reference: Dictionary*.

Temporary and Permanent SAS Data Sets

SAS data sets are stored in SAS libraries and can be temporary or permanent. You can specify a data set in either of two methods: using a library reference, or using a file specification. A library reference is specified without quotation marks in the form *libref.SAS-data-set-name*. A file specification must be enclosed in single quotation marks and uses the file naming conventions of your operating environment.

Using a Library Reference to Specify a Data Set

Typically, temporary SAS data sets are stored in the WORK data library and are referenced with a one-level name. The WORK library is defined automatically at the beginning of the SAS session and is automatically deleted at the end of the SAS session. Procedures assume that SAS data sets that are specified with a one-level name are to be read from, or written to, the WORK data library, unless you specify a USER data library. For example, this statement specifies a temporary data set from the WORK library:

```
proc gplot data=stocks;
```

Typically, permanent SAS data sets have a two-level name of the form *libref.SAS-data-set-name* in which *libref* identifies a storage location on your host system. A LIBNAME statement associates a libref with the storage location. See also "LIBNAME Statement" on page 25. For example, these statements specify a permanent data set:

```
libname reflib 'my-SAS-library';
proc gplot data=reflib.stocks;
```

You can use a one-level name for permanent SAS data sets if you specify a USER data library. In this case, the procedure assumes that data sets with one-level names are in the USER data library instead of in the WORK data library. You can assign a USER data library with a LIBNAME statement or the USER= SAS system option. For example, these statements use a single-level name to specify a permanent data set that is stored in the library identified as the USER library:

```
options user='my-SAS-library';
proc gplot data=stocks;
```

Using a File Specification to Specify a Data Set

To use a file specification for specifying a data set, enclose the file specification in single quotation marks. The specification can be a filename, or a path and filename. The specification must follow the file naming conventions of your operating environment.

For example, the following code creates a file named *mydata* in the default storage location, which is the location where the SAS session was started:

```
data 'mydata';
```

The quotes are required for a file specification; if omitted, SAS treats the specification as a library reference. In the above example, if the quotes are omitted, SAS creates the data set in the temporary WORK catalog and identifies it by the name WORK.MYDATA.

To create the file in a location other than the default location, the quoted file specification must include the full path to the desired location.

You cannot use quoted file specifications for

- SAS catalog names
- MDDB and FDB references
- PROC SQL
- the _LAST_= system option.

Data Set Requirements

SAS/GRAPH procedures often have certain requirements for the input data sets they use. Some procedures may expect the input data set to be sorted in a certain way while others may require the data set to contain certain variables or types of information. If necessary, you can use DATA steps and base SAS procedures in your program to manipulate the data appropriately. For specific requirements, see "About the Input Data Set" in the "Concepts" section of the procedure chapter.

Automatic Data Set Locking

All SAS/GRAPH procedures that produce graphics output automatically lock the input data sets during processing. By locking a data set, SAS/GRAPH software prevents another user from updating the data at the same time you are using it to produce a graph. If data in a data set changes while you are using it to draw a graph, unpredictable results can occur in the graph or your program may end with errors.

Using Engines with SAS/GRAPH Software

In the SAS System, procedures use *engines* to access data. Characteristics of these engines vary; generally, they allow SAS procedures to access a data library in a particular way – the expected format for the SAS data file, the type of read/write activity that can occur in SAS data files, and so on. In most cases, you use the default engine for the current SAS version and do not specify an engine. If you are using an engine other than the default, the engine must

□ support nonsequential access

□ equate observation numbers with internal record IDs (required for the GREDUCE procedure only)

□ disallow shared update or spin a copy of the data set for input processing when a procedure requires multiple passes over the data.

Note: The default engine for Versions 7 and 8 do not work with the GREDUCE procedure if the input data set is compressed. △

For more information about SAS engines, see *SAS Language Reference: Concepts*.

Running SAS/GRAPH Programs

Modes of Operation

There are several ways to run a SAS program. You can use

□ SAS Display Manager System, a windowing system that gives you a text editor from which to submit programs, windows for the SAS log and SAS output, and many other facilities

□ interactive line mode, in which you submit programs one line at a time

- □ noninteractive mode, which executes a SAS program (stored in a file) in your current terminal session
- □ batch mode, which executes a SAS program (stored in a file) in a separate session.

The mode you use determines whether the graphics output displays on your monitor. If you use display manager, interactive line mode, or noninteractive mode, the SAS/GRAPH program can display graphics output on your monitor as well as store the output in a catalog.

If you use batch mode, the graphics output is not displayed on your monitor. In this case, your program must send the graphics output to a hardcopy device, permanent catalog, or a graphics stream file. See Chapter 4, "SAS/GRAPH Output," on page 43 for more information on the destination of graphics output.

Regardless of how you run your programs, SAS/GRAPH software uses the values stored in the device entry or specified by graphics options in a GOPTIONS statement to determine how to handle the graphics output.

For more information on modes of operation, see *SAS Language Reference: Concepts* and the SAS documentation for your operating environment.

RUN-Group Processing

You can use RUN-group processing with the GCHART, GMAP, GPLOT, GREPLAY, and GSLIDE procedures to produce multiple graphs without restarting the procedure every time.

To use RUN-group processing, you start the procedure and then submit multiple RUN-groups. A *RUN-group* is a group of statements that contains at least one action statement and ends with a RUN statement. It can contain other SAS statements such as AXIS, BY, GOPTIONS, LEGEND, TITLE, or WHERE. As long as you do not end the procedure, it remains active and you do not need to resubmit the PROC statement.

To end RUN-group processing, submit a QUIT or RUN CANCEL statement, or start a new procedure.

Note: When using SAS/GRAPH with the ODS HTML statement, it is best to use the QUIT statement after each procedure that uses RUN-group processing, rather than relying on a new procedure to end the processing. △

With global and local statements

Global statements and NOTE statements that are submitted in a RUN-group affect all subsequent RUN-groups until you cancel the statements or exit the procedure. For example, each of these two RUN-groups produces a plot and both plots display the title defined in the first RUN-group:

```
/* first run group*/
proc gplot data=sales;
   title1 'Sales Summary';
   plot sales*model_a;
run;

      /* second run group */
   plot sales*model_b;
run;
quit;
```

With BY statements

BY statements persist in exactly the same way. Therefore, if you submit a BY statement within a RUN-group, the BY-group processing produces a separate graph for each value of the BY variable for the RUN-group in which you submit it and for all subsequent RUN-groups until you cancel the BY statement or exit the procedure. Thus, as you submit subsequent action statements, you continue to get multiple graphs (one for each value of the BY variable). For more information, see “BY Statement” on page 177.

With the WHERE Statement

The WHERE statement enables you to graph only a subset of the data in the input data set. If you submit a WHERE statement with a RUN-group, the WHERE definition remains in effect for all subsequent RUN-groups until you exit the procedure or reset the WHERE definition.

Using a WHERE statement with RUN-group processing follows most of the same rules as using the WHERE statement outside of RUN-group processing with these exceptions:

- With the GMAP procedure, the WHERE variable must be in the input data set.
- With a procedure that is using an Annotate data set, the following requirements must be met:
  - The ANNOTATE= option must be included in the action statement.
  - The WHERE statement must be executed before the action statement.
  - The WHERE variable must occur in the Annotate data set.

Procedure Output and the Graphics Output Area

The result of most SAS/GRAPH procedures is the graphic display of data in the form of graphics output. *Graphics output* is made up of commands that tell a graphics device how to draw graphic elements. A *graphics element* is a visual element of graphics output – for example, a plot line, a bar, a footnote, the outline of a map area, or a border.

To generate graphics output, your program uses a device driver that directs the graphics output to a display device (a graphics monitor or terminal), a hardcopy device, or a file. Despite the fact that all graphics devices do not understand the same commands, SAS/GRAPH software can produce graphics output on many types of graphics devices. It does so by producing output in two steps:

1. It creates a catalog entry made up of graphics commands in a generic, device-independent format.
2. It uses a device driver to translate the commands from the generic format to commands that a particular graphics device understands. This is called device-dependent output.

Your program controls this process as well as the graphics environment in which the graphics appear. This section describes this graphics environment and how you can modify it, how SAS/GRAPH uses it, and how you can make your programs work for different output devices.

About the Graphics Output Area

When SAS/GRAPH software produces graphics output, it draws the graphic elements inside of an area called the *graphics output area*. Characteristics of the graphics output area are determined by the values of certain device parameters within the device entry. You can modify some of these characteristics for a single graph or an entire SAS session by using graphics options to change the values of the device parameters. This section describes changes you can make to the external dimensions, the resolution, the cell size, and the type of units. For a description of the graphics options and device parameters referred to in this section, see Chapter 9, “Graphics Options and Device Parameters Dictionary,” on page 301.

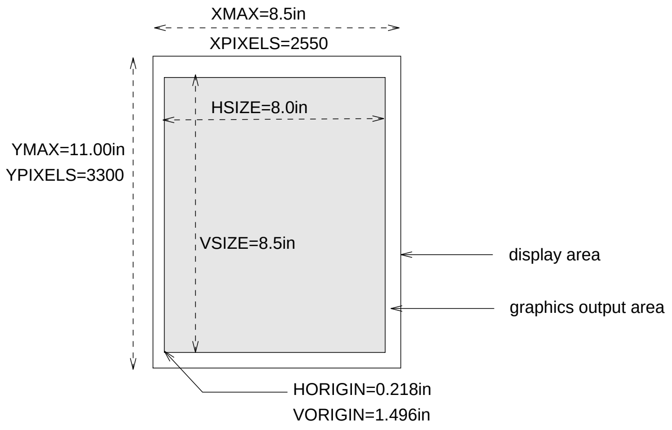
External Dimensions

The graphics output area is contained within the device’s display area. The external dimensions of the device’s display area are controlled by the values of the XMAX and YMAX device parameters. XMAX sets the maximum horizontal dimension; YMAX sets the maximum vertical dimension. The *orientation* of the graphics output area, that is, whether it is landscape or portrait, is determined by whether the larger value is XMAX (orientation is landscape) or YMAX (orientation is portrait).

The external dimensions of the graphics output area are controlled by the values of the HSIZE and VSIZE device parameters.

Typically, the default dimensions of the graphics output area are the same as the dimensions of the device. This is usually true for display devices. For those devices, the default value of HSIZE and VSIZE is 0. However, for hardcopy devices, the XMAX, YMAX values represent the external boundaries of the output medium (such as a sheet of paper). If these devices need a margin, HSIZE, VSIZE and HORIGIN, VORIGIN are assigned default values and the default graphics output area is somewhat smaller than the device’s display area. Figure 2.1 on page 30 illustrates such a device.

Figure 2.1 Default Dimensions of the PSCOLOR Device



Note that HORIGIN and VORIGIN define the left margin and bottom margin, respectively. The right margin and top margin are calculated by the device driver as follows:

right-margin = XMAX – (HSIZE + HORIGIN)

top-margin = YMAX –(VSIZE + VORIGIN)

You cannot specify values for *right-margin* and *top-margin*.

You can change the dimensions of the graphics output area for a SAS session or for a single graph with the HSIZE= and VSIZE= graphics options. Changing the size of the graphics output area does not change the dimensions of the device’s display area or affect the resolution. The values of HSIZE= and VSIZE= cannot exceed the maximum dimensions for the device as specified by XMAX and YMAX. Furthermore, you cannot specify values for graphics options HSIZE= and VSIZE= that exceed the HSIZE and VSIZE values in the device entry.

Device Resolution

The resolution of a device is the number of pixels per inch. It is determined by the values of the device parameters XMAX, YMAX, and XPIXELS, YPIXELS, and is calculated by dividing the number of pixels by the corresponding external dimension. For example,

x-resolution = XPIXELS / XMAX

Therefore, the X resolution of the PSCOLOR device illustrated in Figure 2.1 on page 30 is 300dpi (dots per inch).

Ordinarily, you do not want to change the device resolution because changing it may distort your image. However, you may want to change the size of the display area. To do so without changing the resolution, use the GOPTIONS statement to change the values of only XPIXELS= and YPIXELS=, or the values of only XMAX= and YMAX=. Then SAS/GRAPH will automatically calculate the correct value for the unspecified parameters so that the device retains the default resolution.

If you do want to change the device resolution (usually for image files or graphs that are displayed online), specify values for both XMAX= and XPIXELS= (horizontal resolution) or both YMAX= and YPIXELS= (vertical resolution), or all four. In these cases, SAS/GRAPH changes the dimensions and recalculates the device resolution. Table 2.1 on page 31 summarizes the interaction of these options.

Table 2.1 Interactions of Graphics Options That Affect Resolution

If you specify values for	*and...*	*then SAS/GRAPH...*
XPIXELS= and YPIXELS=	not XMAX= and YMAX=	changes the dimensions and recalculates the value of XMAX= and YMAX= in order to retain the resolution
XMAX= and YMAX=	not XPIXELS= and YPIXELS=	changes the dimensions and recalculates the value of XPIXELS= and YPIXELS= in order to retain the resolution
XMAX=	XPIXELS=	changes the horizontal dimension and recalculates the resolution
YMAX=	YPIXELS=	changes the vertical dimension and recalculates the resolution

Cells

Within the graphics output area, SAS/GRAPH software defines an invisible grid of rows and columns. This grid is made up of *character cells* as shown in Figure 2.2 on page 32.

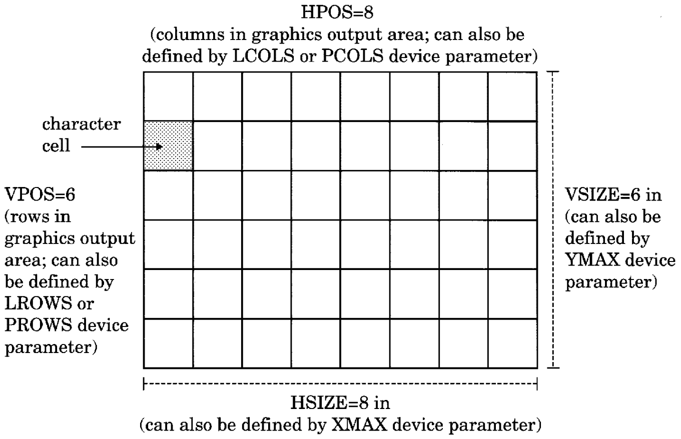
The size and proportion of these cells affects the size and appearance of graphic elements that are drawn using units of CELLS. The attributes of the cells are determined by both the external dimensions of the graphics output area (controlled by

HSIZE and VSIZE) and the number of rows and columns. The number of rows is controlled by the LROWS (if orientation is landscape) or PROWS (if orientation is portrait) device parameter. Similarly, the number of columns is controlled by the LCOLS (landscape) or PCOLS (portrait) device parameter.

You can change the number of rows and columns in the grid with the HPOS= and VPOS= graphics options. HPOS= overrides the value of LCOLS or PCOLS and sets the number of columns in the graphics output area. VPOS= overrides the value of LROWS or PROWS and sets the number of rows in the graphics output area.

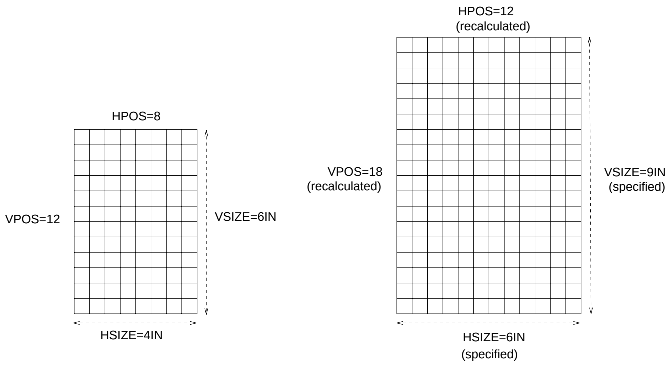
Figure 2.2 on page 32 illustrates the relationship between the graphics options or device parameters that determine the dimensions of the graphics output area and those that determine the number of character cells within the graphics output area.

Figure 2.2 Rows, Columns, and Cells in the Graphics Output Area


HPOS=8
(columns in graphics output area; can also be defined by LCOLS or PCOLS device parameter)
character cell
VPOS=6 (rows in graphics output area; can also be defined by LROWS or PROWS device parameter)
VSIZE=6 in (can also be defined by YMAX device parameter)
HSIZE=8 in
(can also be defined by XMAX device parameter)

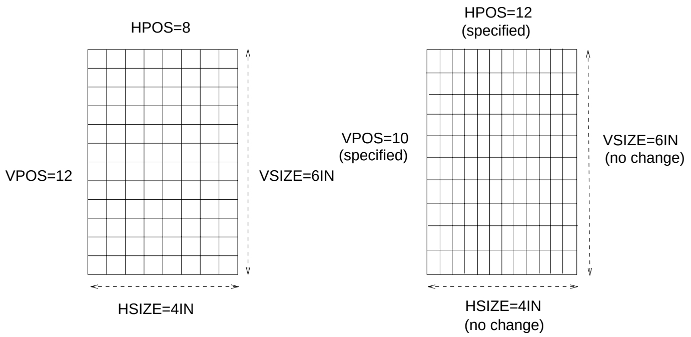
Changing only the external dimensions of the graphics output area (HSIZE= and VSIZE=) retains the cell size but causes SAS/GRAPH to automatically recalculate the number of rows and columns, as illustrated in Figure 2.3 on page 32.

Figure 2.3 Changing HSIZE=, VSIZE= Changes Dimensions and Recalculates Number of Rows and Columns



Changing only the number of rows and columns (HPOS and VPOS) changes the size of the cells without altering the overall size of the output. Figure 2.4 on page 33 shows how increasing the number of rows and columns reduces the size of the individual cells.

Figure 2.4 Changing HPOS= and VPOS= Changes Cell Size



Usually, you should not change the number of rows and columns from the default for your device. However, if you must change them to make a graph fit, note that the size of text in the graphics output will change *if* you specified text size using units of CELLS. If the cells are large (that is, HPOS= and VPOS= have small values), the text may not fit. If the cells are too small, the text may be too small to read. In this case, you can adjust the size of the text with the HEIGHT= statement option or the HTEXT= graphics option.

To change all the attributes of the graphics output area, specify values for all four options, as shown in Figure 2.5 on page 33.

Figure 2.5 Changing HSIZE=, VSIZE= and HPOS=, VPOS= Changes Dimensions and Number and Size of Cells

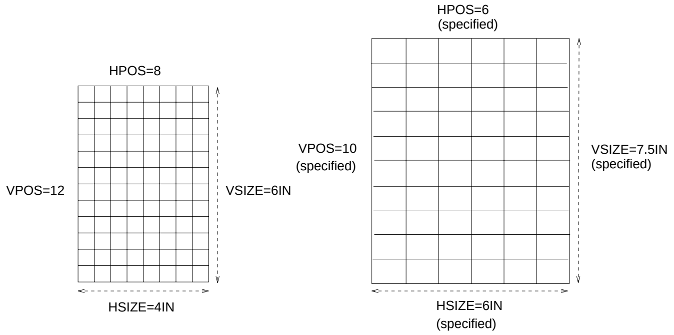


Table 2.2 on page 34 summarizes the interaction of the HSIZE=, VSIZE=, HPOS=, and VPOS= graphics options.

Table 2.2 Interaction of Graphics Options Affecting Cells

If you specify values for ...	*and ...*	*then SAS/GRAPH ...*
HSIZE= and VSIZE=	not HPOS= and VPOS= (or specify HPOS=0 and VPOS=0)	changes the external dimensions of the graphics output area and recalculates the number of rows and columns in order to retain cell size and proportions.
HPOS= and VPOS=	not HSIZE= and VSIZE=	keeps the external dimensions but changes the cell size according to the number of rows and columns.
HSIZE= and VSIZE=	HPOS= and VPOS=	changes the dimensions of the graphics output area, the number of rows and columns, and recalculates the cell size.

Units

By default, most graphic elements are drawn using units of CELLS to determine their size. For example, the default character height for the TITLE1 definition is two cells; for all other text the default height is one cell.

Changing the cell size to control the size of one element, such as text, may distort other parts of your graph. Instead, you may want to change the type of units that SAS/GRAPH uses to control the size of the graphic elements. In addition to CELLS and other absolute units such as inches (IN), centimeters (CM), and points (PT), you can often use units of percent of the graphics output area (PCT). This unit specification allows the height of the graphic elements to change in proportion to the size of the graphics output area.

You can specify the type of unit for individual graphic elements or you can use the GUNIT= graphics option to set the default units that will be used for most height specifications.

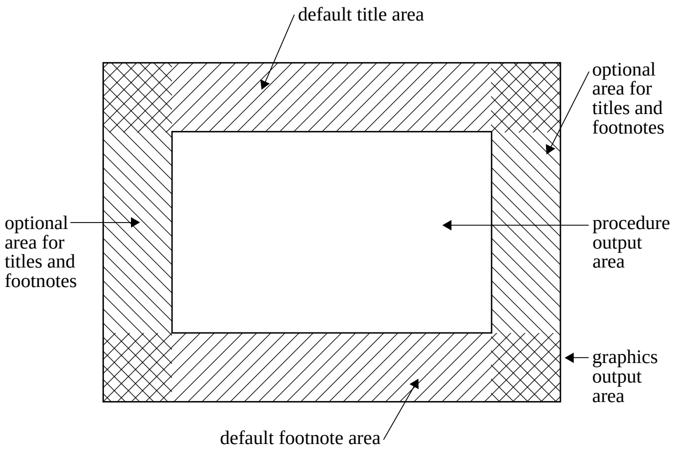
Placement of Graphic Elements in the Graphics Output Area

By default, SAS/GRAPH software positions certain graphics elements in predefined locations in the graphics output area. Figure 2.6 on page 35 shows the graphics output area and the areas within it that are used by the following graphic elements:

- □ Titles are placed in the title area at the top of the graphics output area.
- □ Footnotes are placed in the footnote area at the bottom of the graphics output area.
- □ The graph itself uses the *procedure output area*, which is the area left after the titles and footnotes have been drawn.
- □ Legends also use the procedure output area and may affect the amount of space available for the graph. By default, space is reserved for the legend below the axis area of a graph and above the footnote area. However, you can position the legend in the part of the procedure output area that is reserved for the graph. For details, see “LEGEND Statement” on page 187.

Note: Titles and footnotes can be positioned elsewhere on the graph as well, with different effects on space allocation. See “TITLE, FOOTNOTE, and NOTE Statements” on page 251 for details. △

Figure 2.6 Default Locations for Graphic Elements in the Graphics Output Area



Note: If the titles, footnotes, and legend are very large, they may make the procedure output area too small for the graph. You can control the size of title and footnote text and of most legend elements with statement options. For details, see Chapter 8, "SAS/GRAPH Statements," on page 159 for a description of the appropriate statement. In addition, "GOPTIONS Statement" on page 182 lists the graphics options that control the size of various graphic elements. See also "Making Programs Portable" on page 35. △

How Errors in Sizing Are Handled

Sometimes SAS/GRAPH cannot fit one or more graphic elements on the graph. This can happen if an element is too big for the available space (for example, the title is too long), or if you have too many elements to fit in a given space (for example, a bar chart has too many bars). In these cases, SAS/GRAPH either

- resizes the graphics element and issues a warning explaining what it did
- issues an error message and does not attempt to produce the graph.

For example, it adjusts the size of titles to make them fit but it does not drop bars in order to produce a readable bar chart. If you get unexpected results or no graph, check the SAS log for notes, warnings, and errors.

Making Programs Portable

When you want to write a program that will produce the same graphics output on two different devices, you can use features in SAS/GRAPH software to simplify the process:

- Use percent of the graphics output area (PCT) as the unit of measure when specifying sizes of text and other graphics to make sure that text is proportional in size across devices. A one-inch-high title may be appropriate on a standard piece of paper, but it is almost all of the display area of a slide. To make units of percentage the default for size specifications, use the GUNIT= graphics option:

```
goptions gunit=pct;
```

You can also specify PCT anywhere you specify a size:

```
axis1 label=(height=3 pct 'Year');
```

See “GUNIT” on page 350 for a complete description of the GUNIT= graphics option.

□ Compare colors and patterns for the devices you will use and use the colors of the device that has the fewest colors, With a slide camera, for example, you can choose from over 16 million colors, but some graphics monitors display only four or eight colors at a time.

If you want to generate a graph on both a color device and a black-and-white device, such as a laser printer or a monochrome graphics monitor, all colors are remapped to black, white, or scales of gray, depending on the device. However, with the PATTERN statement, you can substitute line-patterns for colors.

□ Preview the appearance of the output on a different device with the TARGETDEVICE= graphics option. For example, to see how the output will look on a color PostScript printer, specify

```
goptions targetdevice=pscolor;
```

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

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