

CHAPTER

9

Graphics Options and Device Parameters Dictionary

<i>Overview</i>	304
<i>Dictionary of Graphics Options and Device Parameters</i>	304
<i>ADMGDF</i>	305
<i>ASPECT</i>	305
<i>AUTOCOPY</i>	306
<i>AUTOFEED</i>	306
<i>AUTOSIZE</i>	307
<i>BINDING</i>	308
<i>BORDER</i>	308
<i>CBACK</i>	309
<i>CBY</i>	309
<i>CELL</i>	310
<i>CHARACTERS</i>	311
<i>CHARREC</i>	311
<i>CHARTYPE</i>	312
<i>CIRCLEARC</i>	312
<i>CMAP</i>	313
<i>COLLATE</i>	314
<i>COLORS</i>	314
<i>COLORTYPE</i>	315
<i>COLS</i>	316
<i>CPATTERN</i>	316
<i>CSYMBOL</i>	317
<i>CTEXT</i>	317
<i>CTITLE</i>	318
<i>DASH</i>	318
<i>DASHLINE</i>	319
<i>DASHSCALE</i>	320
<i>DELAY</i>	320
<i>DESCRIPTION</i>	320
<i>DEVADDR</i>	321
<i>DEVICE</i>	321
<i>DEVMAP</i>	322
<i>DEVOPTS</i>	322
<i>DEVTYPE</i>	325
<i>DISPLAY</i>	326
<i>DISPOSAL</i>	326
<i>DRVINIT</i>	327
<i>DRVQRY</i>	327
<i>DRVTERM</i>	328
<i>DUPLEX</i>	328

ERASE 329
EXTENSION 329
FASTTEXT 330
FBY 330
FCACHE 331
FILECLOSE 331
FILEONLY 332
FILL 333
FILLINC 333
FONT NAME 334
FONTRES 334
FORMAT 335
FTEXT 335
FTITLE 336
FTRACK 336
GACCESS 337
GCLASS 338
GCOPIES 339
GDDMCOPY 339
GDDMNICKNAME 340
GDDMTOKEN 340
GDEST 341
GEND 341
GPILOG 342
GFORMS 343
GOUTMODE 343
GPROLOG 343
GPROTOCOL 344
GRAPHRC 346
GSFLEN 346
GSFMODE 347
GSFNAME 348
GSFPROMPT 349
GSIZE 349
GSTART 350
GUNIT 350
GWAIT 351
GWRITER 352
HANDSHAKE 352
HBY 353
HEADER 354
HEADERFILE 354
HORIGIN 354
HOSTSPEC 355
HPOS 355
HSIZE 356
HTEXT 357
HTITLE 357
ID 358
INTERACTIVE 358
INTERLACED 359
INTERPOL 359
ITERATION 360
KEYMAP 360

LCOLS 361
LFACTOR 361
LROWS 362
MAXCOLORS 362
MAXPOLY 362
MODEL 363
MODULE 363
NAK 364
OFFSHADOW 364
PAPERDEST 365
PAPERFEED 365
PAPERLIMIT 366
PAPERSIZE 366
PAPERSOURCE 367
PAPERTYPE 368
PATH 369
PCLIP 369
PCOLS 371
PENMOUNTS 371
PENSORT 372
PIEFILL 372
POLYGONCLIP 373
POLYGONFILL 374
POSTGPILOG 374
POSTGPROLOG 375
POSTGRAPH 375
PPDFILE 376
PREGPILOG 376
PREGPROLOG 377
PREGRAPH 377
PROCESS 378
PROCESSINPUT 378
PROCESSOUTPUT 379
PROMPT 379
PROMPTCHARS 380
PROWS 382
QMSG 382
RECTFILL 383
RENDER 384
RENDERLIB 385
REPAINT 385
RESET 386
REVERSE 386
ROTATE 387
ROTATION 388
ROWS 388
SCALABLE 389
SIMFONT 389
SPEED 390
SWAP 390
SYMBOL 391
SYMBOLS 391
TARGETDEVICE 393
TRAILER 393

<i>TRAILERFILE</i>	394
<i>TRANSPARENCY</i>	394
<i>TRANTAB</i>	395
<i>TYPE</i>	395
<i>UCC</i>	396
<i>VORIGIN</i>	396
<i>VPOS</i>	397
<i>VSIZE</i>	398
<i>V6COMP</i>	398
<i>XMAX</i>	399
<i>XPIXELS</i>	400
<i>YMAX</i>	401
<i>YPIXELS</i>	402

Overview

This chapter provides a detailed description of all of the graphics options and device parameters used with SAS/GRAPH software. These include

- all graphics options used by the GOPTIONS statement
- all device parameters that can be specified as options in the ADD and MODIFY statements in the GDEVICE procedure
- all device parameters that appear as fields in the GDEVICE windows.

The descriptions provide the syntax, defaults, and required information for each option and parameter.

The graphics options and device parameters are intermixed and listed alphabetically. When the graphics option and device parameter have the same name, they are discussed in the same dictionary entry and the description uses only that name and does not distinguish between the option and the parameter except where the distinction is necessary.

For a list of all the graphics options, see “GOPTIONS Statement” on page 182. For a list of all the device parameters, see “ADD Statement” on page 657.

If the syntax for the graphics option and the device parameter is different, both forms are shown. If the syntax is the same, one form is shown.

Use a GOPTIONS statement to specify the graphics options. Some graphics options can also be specified in an OPTIONS statement. Use the GDEVICE procedure to specify the device parameters. (See “GOPTIONS Statement” on page 182 and Chapter 15, “The GDEVICE Procedure,” on page 651 for details.)

Note: The syntax for device parameters is the syntax for specifying parameters when using the GDEVICE procedure statements. With the GDEVICE windows, simply enter values into fields in the windows. \triangle

Note: The values that you specify for any option or parameter must be valid for the device. If you specify a value that exceeds the device’s capabilities, SAS/GRAPH software reverts to values that can be used with the device. \triangle

Dictionary of Graphics Options and Device Parameters

When the syntax of an option includes *units*, use one of these unless the syntax specifies otherwise:

CELLS	character cells
CM	centimeters
IN	inches
PCT	percentage of the graphics output area
PT	points (there are approximately 72 points in an inch)

If you omit *units*, a unit specification is searched for in this order:

- 1 the value of GUNIT= in a GOPTIONS statement
- 2 the default unit, CELLS.

ADMGDF

Specifies whether to write an ADMGDF or GDF file when the GSFNAME= and GSFMODE= graphics options are used with a GDDM device driver.

Used in: GOPTIONS statement

Default: NOADMGDF

Restriction: GDDM device drivers on IBM mainframe systems only

Syntax

ADMGDF | NOADMGDF

ADMGDF

instructs the GDDM device driver to write out an ADMGDF file.

NOADMGDF

instructs the GDDM device driver to write out a GDF file.

ASPECT

Sets the aspect ratio for graphics elements.

Used in: GOPTIONS statement GDEVICE procedure GDEVICE Detail window

Default: device dependent

Syntax

ASPECT=*scaling-factor*

scaling-factor

is a nonnegative integer or real number that determines the ratio of width to height for graphics elements. If you specify ASPECT=1, each graphics element has equal

horizontal and vertical scaling factors; ASPECT=2 scales the graphics element twice as wide as its height; and so on. If ASPECT= is not specified or is set to 0 or null, SAS/GRAPH uses the aspect ratio of the hardware device.

Details

The aspect ratio affects many graphics characteristics, such as the shape of software characters and the roundness of pie charts. Some graphics drivers do not produce correct output if the aspect ratio is anything other than the default. When you use a device that uses local scaling (that is, the device itself can scale the output, for example, some plotters), use ASPECT= to tell SAS/GRAPH the scaling factor. If you change ASPECT, you can use the GTESTIT procedure to run a sample graphics output to inspect the effects of the changes.

Note: You may get more reliable results if you use the default aspect ratio and use the HSIZE= and VSIZE= graphics options to set the dimensions. Δ

AUTOCOPY

Specifies whether to generate hardcopy automatically.

Used in: GOPTIONS statement; GDEVICE procedure; GDEVICE Parameters window

Defaults: GOPTIONS: NOAUTOCOPY; GDEVICE: AUTOCOPY=N

Restriction: device dependent

Syntax

GOPTIONS: AUTOCOPY | NOAUTOCOPY

GDEVICE: AUTOCOPY=Y | N

AUTOCOPY

AUTOCOPY=Y

prints a copy of the graph automatically.

NOAUTOCOPY

AUTOCOPY=N

suppresses printing a copy of the graph. A blank **Autocopy** field in the Parameters window is the same as AUTOCOPY=N.

Details

AUTOCOPY is used only for older terminals that have printers attached directly to the device.

AUTOFEED

Specifies whether devices with continuous paper or automatic paper feed should roll or feed the paper automatically for the next graph.

Used in: GOPTIONS statement; GDEVICE procedure; GDEVICE Parameters window

Defaults: GOPTIONS: AUTOFEED (if a device is specified); GDEVICE: AUTOFEED=Y

Restriction: device dependent

See also: PPDFILE

Syntax

GOPTIONS: AUTOFEED | NOAUTOFEED

GDEVICE: AUTOFEED=Y | N

AUTOFEED

AUTOFEED=Y

causes the device to feed new paper automatically for the next graph. A blank **Autofeed** field in the Parameters window is the same as AUTOFEED=Y.

NOAUTOFEED

AUTOFEED=N

suppresses the automatic paper feed.

Details

For PostScript devices, if AUTOFEED is unaltered, the PostScript file is unchanged. If you specify NOAUTOFEED and do not select a PPD file with the PPDFILE option, a PostScript Level 1 manualfeed command is added to the driver output. If you specify NOAUTOFEED and select a PPD that contains a manualfeed option, the procedure code for that manualfeed option is sent. If there is no manualfeed option in the PPD, no manualfeed code is sent. See “PPDFILE” on page 376.

AUTOSIZE

Controls whether to change the size of the character cells in order to preserve the number of rows and columns specified in the device entry.

Used in: GOPTIONS statement

Default: device dependent

See also: DEVOPTS

Syntax

AUTOSIZE=ON | OFF | DEFAULT

ON

changes the cell size in order to preserve the number of rows and columns.

OFF

preserves the device’s original cell size and temporarily changes the number of rows and columns.

DEFAULT

uses the default setting (ON or OFF) that is controlled by DEVOPTS bit 50 (see “DEVOPTS” on page 322).

Details

AUTOSIZE is useful when you change the size of the graphics display area using one or more of the options PAPERSIZE, XPIXELS, YPIXELS, XMAX, or YMAX. It lets you control image text size without using PROC GDEVICE. Typically, AUTOSIZE is on for most image drivers and off for all other types of drivers.

Note: If you use HSIZE or VSIZE, the character cell size changes regardless of the AUTOSIZE setting. Δ

BINDING

Specifies which edge of the document is the binding edge.

Used in: GOPTIONS statement OPTIONS statement

Default: DEFAULTEDGE

Restriction: PostScript printers require a PPD file

See also: DUPLEX, PPDFILE

Syntax

BINDING=DEFAULTEDGE | LONGEDGE | SHORTEGE

Details

BINDING controls how the page is flipped when DUPLEX is in effect. It does not change the orientation of the graph. DEFAULTEDGE refers to the hardware’s factory-default setting. LONGEDGE and SHORTEGE refer to the paper’s long and short edges.

For PostScript printers, a PPD file must also be specified, using the PPDFILE= option. The PPD file contains the command that SAS/GRAPH needs to request the appropriate binding method on the printer being used. If a PPD file is not specified, BINDING= is ignored because SAS/GRAPH will lack the command needed to request the binding method.

BORDER

Specifies whether to draw a border around the graphics output area.

Used in: GOPTIONS statement

Default: NOBORDER

Syntax

BORDER | NOBORDER

Featured in: “Example 8. Creating a Simple Web Page with the ODS HTML Statement” on page 284 and “Example 9. Combining Graphs and Reports in a Web Page” on page 287

Details

The placement of the border on the display is defined by the HSIZE= and VSIZE= graphics options, if used. Otherwise the placement is defined by the XMAX and YMAX device parameters.

CBACK

Specifies the background color of the graphics output.

Used in: GOPTIONS statement; GDEVICE procedure; GDEVICE Gcolors window

Default: as specified in the Gcolors window

Syntax

CBACK=*background-color*

background-color

specifies any SAS/GRAPH color name. See Chapter 7, “SAS/GRAPH Colors,” on page 139 for information about specifying colors.

Details

CBACK is valid on all devices but may be ignored by some (for example, plotters). Specify the default in the Gcolors window of the device entry.

If you explicitly specify a background color with CBACK, the background color you select should contrast with the foreground colors.

CBY

Selects the color of the BY lines that appear in the graphics output.

Used in: GOPTIONS statement

Default: (1) CTEXT= graphics option, if used; (2) first color in current colors list

Syntax

CBY=BY-line-color

BY-line-color

specifies any SAS/GRAPH color name. See Chapter 7, “SAS/GRAPH Colors,” on page 139 for information about specifying colors.

Details

When you use a BY statement with a SAS/GRAPH procedure to process a data set in subgroups, each graph produced by that procedure is headed by a BY line that displays the BY variables and their values that define the current subgroup.

CELL

Controls whether to use cell alignment.

Used in: GOPTIONS statement; GDEVICE procedure; GDEVICE Parameters window

Default: device dependent

Syntax

GOPTIONS: CELL | NOCELL

GDEVICE: CELL=Y | N

CELL

CELL=Y

causes the device to use cell alignment, in which case SAS/GRAPH attempts to place hardware (or simulated hardware) characters inside character cells. This restriction on the location of characters means that in some cases the SAS/GRAPH procedure may generate axes that do not occupy the entire procedure output area or may be unable to create the requested graph. A blank **Cell** field in the Parameters window is the same as CELL=Y.

NOCELL

CELL=N

suppresses cell alignment, causing the procedure to use the entire procedure output area and place axis and tick mark labels without regard to cell alignment.

Details

Specify N in the device entry or use NOCELL in a GOPTIONS statement if you want to preview a graph on a cell-aligned display but intend to produce the final graph on a device that is not cell-aligned, such as a pen plotter.

CHARACTERS

Specifies whether the device's hardware font is used when no font or `FONT=NONE` is specified in a SAS statement.

Used in: GOPTIONS statement; GDEVICE procedure; GDEVICE Parameters window

Defaults: GOPTIONS: CHARACTERS; GDEVICE: CHARACTERS=Y

Syntax

GOPTIONS: CHARACTERS | NOCHARACTERS

GDEVICE: CHARACTERS=Y | N

CHARACTERS

CHARACTERS=Y

causes SAS/GRAPH to use the device's default hardware font when you do not specify a font in a SAS program. A blank **Characters** field in the Parameters window is the same as CHARACTERS=Y.

NOCHARACTERS

CHARACTERS=N

causes SAS/GRAPH to draw the characters using the SIMULATE font and suppresses the use of *all* hardware fonts, regardless of values you specify in other SAS statements.

Details

The hardware font is not used if you changed the HPOS= and VPOS= graphics options from the default, or if you used the HEIGHT= option in a SAS statement *and* the device does not have scalable characters.

CHARREC

Specifies a hardware font for a device by associating a CHARTYPE number with a hardware font. Also defines a default size to use with that font.

Used in: GDEVICE procedure

Default: device dependent

Syntax

CHARREC=(*charrec-list(s)*)

charrec-list

a list of values that correspond to the fields in the Chartype window. *Charrec-list* has this form:

<i>type, rows, cols, 'font', 'Y' 'N'</i>	
<i>type</i>	is the CHARTYPE number and can be an integer from 0 to 9999. (See “CHARTYPE” on page 312 for more information.)
<i>rows</i>	is the number of rows of text in the font that will fit on the display. (See “ROWS” on page 388 for more information.)
<i>cols</i>	is the number of columns of text in the font that will fit on the display. (See “COLS” on page 316 for more information.)
<i>font</i>	is a character string enclosed in quotation marks that contains the name of the corresponding hardware font on the device. (See “FONT NAME” on page 334 for more information.)
Y	represents a scalable font. A scalable font can be displayed at any size. (See “SCALABLE” on page 389 for more information.)
N	represents a nonscalable font. A nonscalable font can be displayed only at a fixed size. (See “SCALABLE” on page 389 for more information.)

For example, these values assign the device's Helvetica font to be the first hardware font in the CHARTYPE window of the driver entry:

```
charrec=(1, 100, 75, 'helvetica', 'y')
```

CHARTYPE

Selects the number of the default hardware character set.

Used in: GOPTIONS statement; GDEVICE procedure; GDEVICE Parameters window

Default: device dependent

Syntax

CHARTYPE=*hardware-font-chartype*

hardware-font-chartype

is a nonnegative integer from 0 to 999. *Hardware-font-chartype* refers to the actual number for the hardware font you want to use as listed in the Chartype window of the device entry for the selected device driver. By default, CHARTYPE is 0, which is the default hardware font for the device.

CIRCLEARC

Specifies whether SAS/GRAPH should use the device's hardware circle-drawing capability, if available.

Used in: GOPTIONS statement; GDEVICE procedure; GDEVICE Parameters window

Default: device dependent

Syntax

GOPTIONS: CIRCLEARC | NOCIRCLEARC

GDEVICE: CIRCLEARC=Y | N

CIRCLEARC

CIRCLEARC=Y

causes SAS/GRAPH to use the built-in hardware circle- and arc-drawing capability of the device. A blank **Circlearc** field in the Parameters window is the same as CIRCLEARC=Y.

Hardware drawing is faster, but not all devices have the capability. SAS/GRAPH device drivers do not try to use the capability if the device does not have it.

NOCIRCLEARC

CIRCLEARC=N

causes SAS/GRAPH to use software move and draw commands to draw circles and arcs.

CMAP

Specifies a color map for the device.

Used in: GDEVICE procedure; GDEVICE Colormap window

Syntax

CMAP=(*from-color* : *to-color* <...>, *from-color-n* : *to-color-n*>)

from-color

specifies the name you want to assign to the color designated by the *color* value. In the Colormap window, enter this value in the **From** field.

to-color

specifies any SAS/GRAPH color name up to eight characters long. In the Colormap window, enter this value in the **To** field. See Chapter 7, "SAS/GRAPH Colors," on page 139 for information on specifying colors.

Details

Once you have defined the color mapping, you use the new color name in any color option. For example, if your device entry maps the color name DAFFODIL to the SAS color value PAOY, you can specify the following:

```
pattern1 color=daffodil;
```

and the driver will map this to the color value PAOY.

COLLATE

Specifies whether to collate the output, if collation is supported by the device.

Used in: GOPTIONS statement; OPTIONS statement

Default: NOCOLLATE

Restriction: hardware dependent, PostScript printers require a PPD file

See also: GPROLOG, PPDFILE

Syntax

COLLATE | NOCOLLATE

Details

A limited number of printers can *collate* output, which means to separate each copy of printed output when you print multiple copies of output.

For PostScript printers, if a device's PPD file has Collate defined as "True", the COLLATE option is supported.

For PCL printers that support collation, use the GPROLOG= option to specify a Printer Job Language (PJM) command to enable the collation. For information on the appropriate PJM command, consult the Printer Commands section of your printer's user manual.

COLORS

Specifies the foreground colors used to produce your graphics output if you do not specify colors explicitly in program statements.

Used in: GOPTIONS statement; GDEVICE procedure; GDEVICE Gcolors window

Default: device dependent

Syntax

GOPTIONS: COLORS=<(colors-list | NONE)>

GDEVICE: COLORS=(<colors-list>)

colors-list

specifies one or more SAS color names. If you specify more than one color, separate each name with a blank. See Chapter 7, "SAS/GRAPH Colors," on page 139 for information on specifying colors and using a colors list.

To change some of the colors in the colors list and retain others, you can use a null value for colors you do not want to change. For example, to change COLORS=(RED GREEN BLUE) to COLORS=(WHITE GREEN BROWN), you can specify COLORS=(WHITE,,BROWN).

NONE

tells SAS/GRAPH to use only the colors that you explicitly specify in program statements and to ignore the device's default colors list.

Note: If you specify COLORS=(NONE) and omit a color specification for a graphics element, such as patterns, SAS/GRAPH selects at random one of the colors already specified in your program. △

Featured in: “Example 3. Rotating Plot Symbols through the Colors List” on page 271

Details

- The order of the colors in the list is important when you use default colors. For example, the colors used for titles, axes, and surfaces in the G3D procedure are assigned by default according to their position in the colors list.
- If you omit or reset COLORS=, SAS/GRAPH uses the default colors list for the current device. To explicitly reset the colors list to the device default, specify either

```
goptions colors=;
goptions colors=();
```

- If you use default patterns with a colors list specified by COLORS=, the patterns rotate through every color in the list. If the colors list contains only one color, for example COLORS=(BLUE), the solid pattern is skipped and the patterns rotate through only the appropriate default hatch patterns for the graph.

Note: By default, if black is the first color in a device's colors list, default pattern rotation skips black as a pattern color, but uses black as the area-outline color. Thus, the outline color is never the same as an area's fill color. Using COLORS= to change the colors list changes this default pattern behavior. When COLORS= is used, all colors in the specified colors list are used in color rotation, and the outline color is the first color in the specified colors list. Thus, the outline color will match any area using the first color as its fill. △

See “PATTERN Statement” on page 211 for more information on pattern rotation.

COLORTYPE

Specifies the color space used by the user-written part of the Metagraphics device driver.

Used in: GDEVICE procedure; GDEVICE Metagraphics window

Default: NAME

Syntax

COLORTYPE=NAME | RGB | HLS | GRAY | CMY | CMYK | HSV | HSB

NAME SAS predefined color names.

RGB	red-green-blue (RGB) color specifications.
HLS	hue-lightness-saturation (HLS) color specifications.
GRAY	gray-scale level.
CMY	cyan-magenta-yellow color specifications.
CMYK	cyan-magenta-yellow-black color specifications.
HSV HSB	hue-saturation-value color specifications. These specifications are also referred to as hue-saturation-brightness (HSB).

See Chapter 7, “SAS/GRAPH Colors,” on page 139 for a description of these color types.

Details

Use the COLORTYPE device parameter also to specify the color-naming scheme that is used for devices that support more than one color-naming scheme.

For information about Metagraphics drivers, contact Technical Support.

COLS

Sets the number of columns that the hardware font uses.

Used in: GDEVICE Chartype window; GDEVICE procedure; CHARREC= option

Default: 0

See also: CHARREC

Syntax

See “CHARREC” on page 311 for syntax.

Details

If you are using a device driver from SASHELP.DEVICES, this parameter is already set for hardware fonts that have been defined for your installation. If you are adding to or modifying the hardware fonts available for a particular device driver, specify a positive value for the COLS device parameter. If COLS is greater than 0, it overrides the values of the LCOLS and PCOLS device parameters. For scalable fonts, you can specify 1 for COLS, and the actual number of columns will be computed based on the current text width.

CPATTERN

Selects the default color for PATTERN definitions when a color has not been specified.

Used in: GOPTIONS statement

Default: first color in current colors list

Syntax

CPATTERN=*pattern-color*

pattern-color

specifies any SAS/GRAPH color name. See Chapter 7, “SAS/GRAPH Colors,” on page 139 for information about specifying colors.

Details

CPATTERN= is overridden by any color specification in a PATTERN statement. For details on how CPATTERN= affects the PATTERN statement, see “The Effect of the CPATTERN= Graphics Option” on page 223.

If you specify CPATTERN=, the solid pattern is skipped and the patterns rotate through only the appropriate default hatch patterns for the graph. See “PATTERN Statement” on page 211 for more information on pattern rotation.

CSYMBOL

Specifies the default color for SYMBOL definitions when a color has not been specified.

Used in: GOPTIONS statement

Default: first color in current colors list

Syntax

CSYMBOL=*symbol-color*

symbol-color

specifies any SAS/GRAPH color name. See Chapter 7, “SAS/GRAPH Colors,” on page 139 for information about specifying colors.

Details

CSYMBOL= is overridden by any color specification in a SYMBOL statement. See “SYMBOL Statement” on page 226.

CTEXT

Selects the default color for all text and the border.

Used in: GOPTIONS statement

Default: first color in current colors list

See also: CTITLE

Syntax

CTEXT=*text-color*

text-color

specifies any SAS/GRAPH color name. See Chapter 7, “SAS/GRAPH Colors,” on page 139 for information about specifying colors.

Details

The CTITLE= graphics option overrides CTEXT= for all titles, notes, and footnotes, as well as the border. Any other color specifications for text in SAS statements also override the value of CTEXT=.

CTITLE

Selects the default color for all titles, footnotes, and notes, and the border.

Used in: GOPTIONS statement

Default: (1) color specified by CTEXT=, if used; (2) first color in current colors list

See also: CTEXT

Syntax

CTITLE=*title-color*

title-color

specifies any SAS/GRAPH color name. See Chapter 7, “SAS/GRAPH Colors,” on page 139 for information about specifying colors.

Details

Any color specification in a TITLE, FOOTNOTE, or NOTE statement overrides the value of CTITLE= for the text. The border, however, still uses the color specified in CTITLE=.

DASH

Specifies whether to use the device's hardware dashed-line capability, if available.

Used in: GOPTIONS statement; GDEVICE procedure; GDEVICE Parameters window

Default: device dependent

See also: DASHLINE

Syntax

GOPTIONS: DASH | NODASH

GDEVICE: DASH=Y | N

DASH

DASH=Y

causes SAS/GRAPH to use the built-in hardware dashed-line drawing capability of the device when generating graphics output. A blank **Dash** field in the Parameters window is the same as DASH=Y.

Hardware drawing is faster, but not all devices have the capability. SAS/GRAPH device drivers do not try to use the capability if the device does not have it.

NODASH

DASH=N

causes SAS/GRAPH to draw the dashed lines.

DASHLINE

Specifies which dashed lines should be generated by hardware means if possible.

Used in: GDEVICE procedure; GDEVICE Parameters window

Default: device dependent

See also: DASH

Syntax

DASHLINE=*'dashed-line-hex-string'*X

dashed-line-hex-string

is a hexadecimal string 16 characters long that must be completely filled. Each bit in the string corresponds to a line type. See Figure 8.22 on page 249 for line types that correspond to each bit.

To use line type 1, turn on bit 1; to use line type 2, turn on bit 2; and so on. For example, in the following option the first byte is '1000'; only bit 1 is on and only line type 1 is selected:

```
dashline='8000000000000000'x
```

To turn on both bits 1 and 2, specify

```
dashline='c000000000000000'x
```

Bit 1 should always be on because it corresponds to a solid line.

Details

If the DASH device parameter is N in the device entry or if NODASH is used in a GOPTIONS statement, SAS/GRAPH ignores the hexadecimal string in the DASHLINE device parameter.

DASHSCALE

Scales the lengths of the dashes in a dashed line.

Used in: GOPTIONS statement

Default: DASHSCALE=1

Syntax

DASHSCALE=*scaling-factor*

scaling-factor

can be any number greater than 0. For example, GOPTIONS DASHSCALE=.5 reduces any existing dash length by one-half.

Details

Only dashes or spaces with lengths greater than one pixel are scaled. Dots are not scaled because their length is effectively zero. DASHSCALE= always uses software line styles instead of the device's hardware dashed line capabilities.

DELAY

Controls the amount of time between graphs in the animation sequence.

Used in: GOPTIONS statement

Default: 0

Restriction: browser dependent, GIFANIM driver only

Syntax

DELAY=*delay-time*

delay-time

specifies the length of time between graphs in units of 0.01 seconds. For example, to specify a delay of .03 seconds, specify DELAY=3.

Details

SAS/GRAPH puts the DELAY= value into the image file. Based on this value, the browser determines how to display the series of graphs.

DESCRIPTION

Provides a description of the device entry.

Alias: DES

Used in: GDEVICE procedure GDEVICE Detail window

Default: none

Syntax

DESCRIPTION=*'text-string'*

DEVADDR

Specifies the location of the device to which the output of device drivers is sent.

Used in: GOPTIONS statement

Default: host dependent

Restriction: IBM mainframe systems only

Syntax

DEVADDR=*device-address*

DEVICE

Specifies the device driver to which SAS/GRAPH sends the procedure output. The device driver controls both the form and destination of the output.

Alias: DEV

Used in: GOPTIONS statement OPTIONS statement

Default: device dependent

Syntax

DEVICE=*device-entry*

device-entry

specifies the name of a device entry that is stored in a device catalog.

Details

A device driver can direct graphics output to a hardware device, such as a terminal or a printer, or can create an external file in another graphics file format, such as TIF, GIF,

or PostScript. Some device drivers also generate both graphics files and HTML files that can be viewed with a Web browser.

Usually a device driver is assigned by default. If a default driver is not assigned or if you specify RESET=ALL in a GOPTIONS statement, and you do not specify a device driver, SAS/GRAPH prompts you to enter a driver name when you execute a procedure that produces graphics output. If you are producing a graph to the screen and the Graph window is active, SAS/GRAPH selects the display driver for you automatically.

For a description of device drivers and for more information on selecting a device entry and changing device parameters, see Chapter 3, “Device Drivers,” on page 37.

For information on using device drivers to display and print graphics output, see Chapter 4, “SAS/GRAPH Output,” on page 43.

For information on using device drivers to export graphics output to external files, see “About Exporting SAS/GRAPH Output” on page 55. For information on using device drivers to create output for the Web, see Chapter 5, “Bringing SAS/GRAPH Output to the Web,” on page 71.

DEVMAP

Specifies the device map to be used when hardware fonts are used.

Used in: GOPTIONS statement; GDEVICE procedure; GDEVICE Host File Options window

Default: device dependent

Syntax

DEVMAP=*device-map-name* | NONE

device-map-name

is a string up to eight characters long that is the name of the device map entry. See Chapter 18, “The GKEYMAP Procedure,” on page 719 for details.

NONE

specifies that you do not want to use a device map. This may cause text to be displayed incorrectly or not at all.

Details

Device maps usually are used only when national characters appear in the text and you want them to display properly.

DEVOPTS

Specifies the hardware capabilities of the device.

Used in: GDEVICE procedure; GDEVICE Parameters window

Default: device dependent

Syntax

DEVOPTS=*'hardware-capabilities-hex-string'*X

hardware-capabilities-hex-string

is a hexadecimal string 16 characters long that must be completely filled. The following table lists the hardware capabilities of each bit:

Table 9.1 Device Capabilities Represented in the DEVOPTS String

Bit On	Capability
0	hardware circle generation
1	hardware pie fill supported
2	scalable hardware characters
3	device is a CRT-type (See TYPE device parameter)
4	translate table needed for non-ASCII hosts
5	hardware polygon fill available
6	hardware characters cell-aligned
7	user-definable colors supported
8	hardware polygons with multiple boundaries supported
9	not used
10	not used
11	adjustable hardware line width
12	double-byte font (non-US) supported
13	hardware repaint supported
14	hardware characters supported
15	no hard limit on x coordinate
16	no hard limit on y coordinate
17	not used
18	ability to justify proportional text
19	driver can produce dependent catalog entries
20	device cannot draw in default background color
21	flush device buffer when filled
22	colors defined using HLS
23	colors defined using RGB
24	not used
25	polyline supported
26	polymarker supported
27	graphics clipping supported

Bit On	Capability
28	not used
29	linkable device driver
30	pick CHARTYPE by name in CHARREC entries
31	device dependent pattern support
32	treat SCALABLE=Y CHARREC as metric
33	size CHARTYPE as HW from CHARREC entries
34	device supports rotated arcs
35	device supports target fonts
36	device supports drawing images
37	device supports multiple color maps
38	image rotation direction
39	device requires sublib for image rotation
40	device is a 24 bit truecolor machine
41	device supports setting font attributes
42	use scan line font rendering
43	device can scale images
44	text clipping supported
45	static color device
46	driver does prolog processing
47	driver does epilog processing
48	driver output only uses a file
49	driver output requires a directory or PDS
50	autosize text to fit rows and columns
51	default binding is SHORTEEDGE
52	driver supports duplex printing
53	device does right edge binding
54	ActiveX device
55	Java device

Details

Each capability in the table corresponds to a bit in the value of the DEVOPTS device parameter. For example, if your device can generate hardware pie fills, the second bit in the first byte of the DEVOPTS string should be turned on if you want the driver to use that capability. If your device is capable of generating only hardware circles and pie fills, specify a value of 'C000000000000000'X as your DEVOPTS value (the first byte is '1100' so the first 2 bits of the first byte are set to 1). Many of the hardware capabilities specified in the DEVOPTS string are overridden by graphics options or other device parameters.

CAUTION:

Do not modify the DEVOPTS device parameter unless you are building a Metagraphics driver. If you want to prevent an Institute-supplied driver from using certain hardware capabilities, change the specific device parameter or use the corresponding graphics option. △

If the DEVOPTS string indicates that a capability is available, the driver uses it unless it is explicitly disabled by another device parameter or graphics option. If the DEVOPTS string indicates that the capability is not available, it is not used by the driver, even if the corresponding device parameter or graphics option indicates that it should be used. For example, if the DEVOPTS value indicates that the device can do a hardware pie fill, the driver uses the hardware pie fill capability unless the PIEFILL device parameter is set to N or NOPIEFILL has been specified in a GOPTIONS statement. However, if the DEVOPTS device parameter indicates that the device cannot do a hardware pie fill, the driver does not attempt to use one, even if the PIEFILL device parameter is set to Y or PIEFILL is used in a GOPTIONS statement.

DEVTYPE

Specifies the information required by SAS/GRAPH routines to determine the nature of the output device.

Used in: GDEVICE procedure; GDEVICE Host File Options window

Default: device dependent

Syntax

DEVTYPE=*device-type*

device-type

is a string eight characters long containing either blanks or some token name that is interpreted by the host. *Device-type* can be:

GTERM

indicates that the output device is a graphics device that will be receiving graphics data; most device drivers use this value.

G3270

indicates that the output device is an IBM 3270 graphics data stream. If your device is an IBM 3270 type of device, DEVTYPE= must be G3270.

Note: GTERM and G3270 are SAS/GRAPH device types. Other valid values depend on your operating environment. DEVTYPE supports any of the device-type values supported on the FILENAME statement. Refer to the SAS Help facility for the device types the FILENAME statement supports in your operating environment. In most cases, this field should not be changed. △

DISPLAY

Specifies whether output is displayed on the graphics device but does not affect whether a graph is placed in a catalog.

Used in: GOPTIONS statement

Default: DISPLAY

Syntax

DISPLAY | NODISPLAY

Details

In most cases, NODISPLAY suppresses *all* output except the catalog entry written to the catalog selected in the GOUT= option. Therefore, you usually specify NODISPLAY when you want to generate a graph in a catalog but do not want to display the graph on your monitor or terminal while the catalog entry is being produced.

Be careful how you use NODISPLAY when you are creating a graphics stream file (GSF). If you use GACCESS= to create the GSF, NODISPLAY prevents SAS/GRAPH from writing to the file. However, if you use GSFNAME= to create the GSF, NODISPLAY does not affect the creation of the GSF.

DISPOSAL

Specifies what happens to the graphic after it is displayed.

Used in: GOPTIONS statement

Default: NONE

Restriction: GIFANIM driver only

Syntax

DISPOSAL=NONE | BACKGROUND | PREVIOUS | UNSPECIFIED

NONE

causes the graphic to be left in place after displaying. This is the default.

BACKGROUND

causes the background color to be returned and the graph erased after displaying.

PREVIOUS

causes the graphic area to be restored with what was displayed in the area previously.

UNSPECIFIED

indicates that no action is necessary.

Details

In Version 6, the ERASE | NOERASE graphics option performed this function for the GIFANIM driver.

DRVINIT

Specifies host commands to be executed before driver initialization.

Used in: GOPTIONS statement; GDEVICE procedure; GDEVICE Host Commands window

Syntax

DRVINIT1='system-command(s)'

DRVINIT2='system-command(s)'

system-command(s)

specifies a character string that is a valid system command and can be in upper- or lowercase letters. You can include more than one command in the string if you separate the commands with a command delimiter, which is host-specific; for example, some operating environments use a semicolon. The length of the entire string cannot exceed 72 characters.

Details

The DRVINIT command is executed before the driver is initialized. DRVINIT is typically used with FILECLOSE=DRIVERTERM to allocate a host file needed by the device driver.

DRVQRY

Specifies whether the device can be queried for information about the current device configuration.

Used in: GDEVICE procedure GDEVICE Detail window

Default: device dependent

Syntax

DRVQRY | NODRVQRY

Details

Generally, this setting is device dependent and you should not change it.

DRVTERM

Specifies host commands to be executed after the driver terminates.

Used in: GOPTIONS statement; GDEVICE procedure; GDEVICE Host Commands window

Syntax

DRVTERM1='system-command(s)'

DRVTERM2='system-command(s)'

system-command(s)

specifies a character string that is a valid system command and can be in upper- or lowercase letters. You can include more than one command in the string if you separate the commands with a command delimiter, which is host-specific; for example, some operating environments use a semicolon. The length of the entire string cannot exceed 72 characters.

Details

The DRVTERM command is executed after the driver terminates. DRVTERM is typically used with FILECLOSE=DRIVERTERM to de-allocate a host file and execute utility programs that send the data to the graphics device. For example, DRVTERM might specify commands to send the file to a host print queue.

DUPLEX

Specifies whether to use duplex printing if available on the device.

Used in: GOPTIONS statement; OPTIONS statement

Default: NODUPLEX

Restriction: duplex printers only

See also: BINDING, GSFMODE, PPDFILE

Syntax

DUPLEX | NODUPLEX

Details

When DUPLEX is on, the driver sets up the printer for duplex operation. Before producing the first graph, set GSFMODE=REPLACE on the GOPTIONS statement, and DUPLEX on an OPTIONS or GOPTIONS statement. You can also use the BINDING= option in conjunction with DUPLEX. Before producing the second graph, set GSFMODE=APPEND on the GOPTIONS statement so that the driver knows to place succeeding graphs on the next available side of paper.

If DUPLEX is in effect, the page's inside (binding) margin is set equal to the current HORIGIN setting, and the outside margin is set equal to

$$\text{XMAX} - \text{HSIZE} - \text{HORIGIN}$$

In terms of even- and odd-numbered pages, this means the following:

odd-numbered pages	HORIGIN determines the left margin, and XMAX-HSIZE-HORIGIN determines the right margin
even-numbered pages	XMAX-HSIZE-HORIGIN determines the left margin, and HORIGIN determines the right margin

For PostScript printers, if you do not use the PPDFILE= option to specify a PPD (PostScript Printer Description) file, a generic PostScript Level 1 duplex command is added to the driver output. If PPDFILE= is used, the duplex command is obtained from the PPD file.

ERASE

Specifies whether to erase graph after display.

Used in: GOPTIONS statement; GDEVICE procedure; GDEVICE Parameters window

Defaults: GOPTIONS: NOERASE; GDEVICE: ERASE=N

Syntax

GOPTIONS: ERASE | NOERASE

GDEVICE: ERASE=Y | N

ERASE

ERASE=Y

causes the graph to be erased when you press RETURN after the graph has been displayed.

NOERASE

ERASE=N

causes the graph to remain on the display when you press RETURN after the graph has been displayed. A blank **Er**ase field in the Parameters window is the same as ERASE=N.

Details

ERASE is useful for those devices that overlay the graphics area and the message area – that is, those devices that have separate dialog and graphics areas. On other devices, the graph is erased.

EXTENSION

Specifies the file extension for an external graphics file.

Used in: GOPTIONS statement
Default: device dependent
See also: GACCESS, GSFNAME

Syntax

EXTENSION=*file-type*'

file-type

a string up to eight characters long that is a file extension, such as GIF or CGM, that you want to append to an external file.

Details

The extension specified on EXTENSION= is used when the output destination is a storage location. The extension is ignored when the output destination is a file. To specify the output destination, you can use a FILENAME statement, or the graphics options GACCESS= or GSFNAME=.

Assuming the output destination is a storage location,

- if EXTENSION='.', no extension is added to the file name
- if EXTENSION=' ' or EXTENSION= is not used, the driver's default extension is added to the file name
- if the driver has no default extension, SAS/GRAPH uses the default extension .GSF.

FASTTEXT

Specifies whether to use integer-based font processing for faster font rendering.

Used in: GOPTIONS statement
Default: FASTTEXT

Syntax

FASTTEXT | NOFASTTEXT

FBY

Selects the font for BY lines.

Used in: GOPTIONS statement
Default: (1) font specified by FTEXT=, if used; (2) default hardware font
See also: "BY Statement" on page 177

Syntax

FBY=*BY-line-font*

BY-line-font

specifies the font for all BY-lines on the graphics output. See Chapter 6, “SAS/GRAPH Fonts,” on page 125 for information about specifying fonts.

Details

When you use a BY statement with a SAS/GRAPH procedure to process a data set in subgroups, each graph produced by that procedure is headed by a BY line that displays the BY variables and their values that define the current subgroup.

FCACHE

Specifies the number of software fonts to keep open at one time.

Used in: GOPTIONS statement

Default: FCACHE=3

Syntax

FCACHE=*number-fonts-open*

number-fonts-open

specifies the number of software fonts to keep open. *Number-fonts-open* must be greater than or equal to zero.

Details

Each font requires from 4K to 10K memory. Graphs that use many fonts may run faster if you set the value of *number-fonts-open* to a higher number. However, graphs that use multiple fonts may require too much memory on some computer systems if all the fonts are kept open. In such cases, set the value of *number-fonts-open* to a lower number to conserve memory.

FILECLOSE

Controls when the graphics stream file (GSF) is closed when you are using the device driver to send graphics output to a hardcopy device.

Used in: GOPTIONS statement; GDEVICE procedure; GDEVICE Host File Options window

Default: DRIVERTERM (if a device is specified)

See also: “About Exporting SAS/GRAPH Output” on page 55

Syntax

FILECLOSE=DRIVERTERM | GRAPHEND

DRIVERTERM DRIVER

closes the GSF and makes it available to the device after all graphs have been produced and the procedure or driver terminates. A host command may be needed to actually send the GSF to the device. Host commands may be specified with the DRVINIT or DRVTERM parameters or entered in the Host File Options window of the device entry.

If multiple graphs are produced by a procedure, this specification creates one large file. Specifying DRIVERTERM is appropriate for batch processing because it is slightly more efficient to allocate the file only once.

GRAPHEND GRAPH

closes the GSF after each separate graph is produced and releases it to the device before sending another. This method creates smaller files if multiple graphs are produced by a procedure. You can specify a command that sends the graph to the device with the POSTGRAPH parameter or use the Host File Options window.

Specifying GRAPHEND is appropriate for drivers that are used interactively, or for devices that require only one graph per physical file.

FILEONLY

Specifies whether a file or a storage location is the default destination for graphics output.

Used in: GOPTIONS statement

Default: device dependent

Restriction: FILEONLY ignored if the device requires the output destination to be a storage location

See also: DEVOPTS, GSFNAME

Syntax

FILEONLY | NOFILEONLY

FILEONLY

specifies that a file rather than a storage location is the default destination for graphics output.

NOFILEONLY

specifies that a storage location is the default destination for graphics output, unless a file of the same name exists.

Details

Most devices use FILEONLY as the default. However, devices that require the output destination to be a storage location use NOFILEONLY as the default. For example, the HTML and WEBFRAME devices require a storage location because they produce two types of output (HTML files and GIF image files) that cannot be written to the same file.

To determine what the default is for a particular device, look at the settings for DEVOPTS bits 48 and 49.

For more information, see “Exporting SAS/GRAPH Output with Program Statements” on page 58.

FILL

Specifies whether to use the device’s hardware rectangle-fill capability.

Used in: GOPTIONS statement; GDEVICE procedure; GDEVICE Parameters window

Default: device dependent

Syntax

GOPTIONS: FILL | NOFILL

GDEVICE: FILL=Y | N

FILL

FILL=Y

causes SAS/GRAPH to use the built-in hardware rectangle-filling capability of the device. A blank **Fi11** field in the Parameters window is the same as FILL=Y.

Hardware drawing is faster, but not all devices have the capability. SAS/GRAPH does not try to use the capability if your device does not support it.

NOFILL

FILL=N

causes SAS/GRAPH to use software fills to fill rectangles.

FILLINC

Specifies the number of pixels to move before drawing the next line in a software fill of a solid area.

Used in: GOPTIONS statement; GDEVICE procedure; GDEVICE Parameters window

Default: device dependent

See also: FILL, PIEFILL, POLYGONFILL

Syntax

FILLINC= 0...9999

Details

In order for FILLINC to have any effect, a software fill must be used. To force a software fill, use the options NOFILL, NOPIEFILL, and NOPOLYGONFILL in a GOPTIONS statement.

If FILLINC is set to 0 or 1, adjacent lines are used (solid fill with no gaps). If FILLINC is set to 2, a pixel-width line is skipped before drawing the next line of a fill.

This option can be useful for keeping plotters from over saturating a solid area and for speeding the plotting. Some inks spread on paper. The type of paper used also can affect ink spread.

FONT NAME

Specifies the hardware font associated with CHARTYPE.

Used in: GDEVICE Chartype window; GDEVICE procedure; CHARREC= option

Required if adding or modifying a CHARREC

See also: CHARREC

Syntax

See “CHARREC” on page 311 for syntax.

Details

Use FONT NAME if you are adding to or modifying the hardware fonts available for a particular device driver. The fonts that you specify must be valid for the output device. If you are using an Institute-supplied device entry, this parameter already is set for most available hardware fonts.

FONTRES

Controls the resolution of software fonts.

Used in: GOPTIONS statement

Default: NORMAL

See also: FASTTEXT, FCACHE, RENDER, RENDERLIB

Syntax

FONTRES=NORMAL | PRESENTATION

NORMAL

renders fonts in memory using integer rendering routines, which improves character drawing speed for most host systems. NORMAL has the same effect as specifying the default values for these graphics options.

```
render=memory
renderlib=saswork
fasttext
fcache=0
```

PRESENTATION

disables the storage or use of rendered versions of Bitstream fonts, but produces the fonts at their highest resolution. FONTRES=PRESENTATION has the same effect as specifying these graphics options:

```
render=none
renderlib=saswork
nofasttext
fcache=3
```

FORMAT

Sets the file format of the metacode file produced by the Institute-supplied part of the Metagraphics device driver.

Used in: GDEVICE procedure; GDEVICE Metagraphics window

Default: CHARACTER

Restriction: Used only with user-supplied Metagraphics drivers.

Syntax

FORMAT=CHARACTER | BINARY

Details

A blank field defaults to CHARACTER. For information about Metagraphics drivers, contact Technical Support.

FTEXT

Selects the default font for all text.

Used in: GOPTIONS statement

Default: Default hardware font (except the first title)

See also: FTITLE

Syntax

FTEXT=*text-font*

text-font

specifies the font for all text on the graphics output. See Chapter 6, “SAS/GRAPH Fonts,” on page 125 for information about specifying fonts.

Featured in: “Example 6. Enhancing Titles” on page 278

Details

The FTITLE= graphics option overrides FTEXT= for the *first* title.

FTITLE

Selects the default font for the first TITLE line.

Used in: GOPTIONS statement

Default: (1) font specified by FTEXT=, if used; (2) SWISS font

See also: FTEXT

Syntax

FTITLE=*title-font*

title-font

specifies the font for the TITLE1 statement. See Chapter 6, “SAS/GRAPH Fonts,” on page 125 for information about specifying fonts.

Featured in: “Example 1. Ordering Axis Tick Marks with SAS Datetime Values” on page 266 and “Example 6. Enhancing Titles” on page 278

FTRACK

Controls the amount of space between letters in the Institute-supplied Bitstream fonts (Brush, Century, Swiss, and Zapf).

Used in: GOPTIONS statement

Default: TIGHT

Syntax

FTRACK=LOOSE | NONE | NORMAL | TIGHT | TOUCH | V5

LOOSE

leaves the most visible space between characters and produces a longer string.

NONE

spacing depends on the size of the font. NONE might produce a shorter or longer string than LOOSE for the same font at different point sizes, because some sizes add space between the characters while others remove it.

NORMAL

is the recommended setting.

TIGHT

reduces the space between characters.

TOUCH

leaves the least visible space between characters.

V5

places a fixed amount of space between the characters and does not adjust for the shape of the character; that is, it does not support kerning. This spacing is compatible with Version 5 Bitstream fonts.

Details

The spacing you specify with FTRACK= affects all Bitstream text in a graph. For example, you cannot produce TIGHT Century type and LOOSE Zapf type simultaneously. This option has no effect on other font types.

Because the value of FTRACK= is stored with the graph, the spacing that you specify when the graph is created is always used when the graph is replayed.

GACCESS

Specifies the format or the destination or both of graphics data written to a device or graphics stream file (GSF).

Used in: GOPTIONS statement; GDEVICE procedure; GDEVICE Host File Options window

Default: device dependent

Syntax

GACCESS=*output-format* | '*output-format destination*'

output-format

specifies the format or the destination (the SAS log or a fileref) of the graphics data. *Output-format* varies according to the operating environment. These values can be specified in all operating environments:

SASGASTD

specifies that a continuous stream of data is written. SASGASTD is the default for most devices and is typically appropriate when the output file will be sent directly

to a device. If you specify GACCESS=SASGASTD, use the GSFNAME= and GSFMODE= graphics options or device parameters to direct your graphics output to a GSF.

SASGAEDT

specifies that the file be host-specific editable format. Some hosts allow editing by inserting characters at the end of each record. SASGAEDT is typically used when the output file is to be edited later. If you specify GACCESS=SASGAEDT, use the GSFNAME= and GSFMODE= graphics options or device parameters to direct your graphics output to a GSF.

SASGAFIX

specifies that fixed-length records be written. (The record length is controlled by the value of the GSFLLEN= graphics option or device parameter or the sixth byte of the PROMPTCHARS value.) The records are padded with blanks where necessary. SASGAFIX is typically used when the output file will be transferred to a computer that requires fixed-length records. If you specify GACCESS=SASGAFIX, use the GSFNAME= and GSFMODE= graphics options or device parameters to direct your graphics output to a GSF.

Note: The value of the GPROTOCOL= graphics option or device parameter can greatly affect the length of the records; for example, if GPROTOCOL=SASGPLCL, the length of the records is doubled. △

SASGALOG

specifies that records are to be written to the SAS log.

GSASFILE

specifies that the records are to be written to the destination whose fileref is GSASFILE. The fileref can point to a specific external file or to an aggregate file location. See “FILENAME Statement” on page 24 for more information on specifying a fileref.

'output-format destination'

specifies the destination in addition to one of these output format values: SASGASTD, SASGAEDT, or SASGAFIX. *Destination* is the physical name of an external file or aggregate file location, or of a device. For details on specifying the physical name of a destination, see the SAS documentation for your operating environment.

This form is not available in all operating environments. See “About Graphics Stream Files” on page 56 for more information on creating graphics stream files.

Note: In the **Gaccess** field of the Host File Options window, you can specify a destination without an output format, in which case the format defaults to SASGASTD. When you specify a value in the **Gaccess** field, you do not need to quote it. △

Operating Environment Information: Depending on your operating environment, you may be able to specify other values for GACCESS=. See the SAS companion for your operating environment for additional values. △

GCLASS

Specifies the output class for IBM printers

Used in: GOPTIONS statement

Default: GCLASS=G

Restriction: used only with IBM3287 and IBM3268 device drivers on OS/390 systems only

Syntax

GCLASS=*SYSOUT-class*

Details

Specifies the SYSOUT class to which the IBM3287 and IBM3268 device driver output is written.

GCOPIES

Sets the current and maximum number of copies to print.

Used in: GOPTIONS statements; GDEVICE Parameters window; GDEVICE procedure; OPTIONS statement

Defaults: GOPTIONS: GCOPIES=(0,20) GDEVICE: GCOPIES=0

Syntax

GOPTIONS: GCOPIES=(*<current-copies>*,*<max-copies>*)

GDEVICE: GCOPIES=*current-copies*

current-copies

is a nonnegative integer ranging from 0 through 255, but it cannot exceed the *max-copies* value specified. A value of 0 or 1 produces a single copy.

max-copies

is a nonnegative integer ranging from 1 through 255.

If you do not specify GCOPIES, a default number of copies is searched for in this order:

- 1 the number of copies specified on an OPTIONS COPIES setting
- 2 0 current copies, and 20 maximum copies.

Details

Not all devices have the capability to print multiple copies. See the **Gcopies** field in the Parameters window for your device to determine its capabilities.

GDDMCOPY

Instructs the driver to issue either an FSCOPY or GSCOPY call to GDDM when AUTOCOPY is in effect.

Used in: GOPTIONS statement

Default: FSCOPY

Restriction: GDDM device drivers on IBM mainframe systems only

See also: AUTOCOPY

Syntax

GDDMCOPY=FSCOPY | GSCOPY

FSCOPY

used when sending output to an IEEE attached plotter.

GSCOPY

used when creating an ADMPRINT file for output on 3287-type printers.

GDDMNICKNAME

Selects a GDDM nickname for the device to which output is sent.

Alias: GDDMN

Used in: GOPTIONS statement

Restriction: GDDM device drivers on IBM mainframe systems only

Syntax

GDDMNICKNAME=*nickname*

Details

Refer to the SAS Help facility for details on using GDDM drivers and options.

GDDMTOKEN

Selects a GDDM token for the device to which output is sent.

Alias: GDDMT

Used in: GOPTIONS statement

Restriction: GDDM device drivers on IBM mainframe systems only

Syntax

GDDMTOKEN=*token*

Details

Refer to the SAS Help facility for details on using GDDM drivers and options.

GDEST

Specifies the JES SYSOUT destination for IBM printers.

Used in: GOPTIONS statement

Default: LOCAL

Restriction: used only with IBM3287 and IBM3268 device drivers on OS/390 systems

Syntax

GDEST=*destination*

GEND

Appends an ASCII string to every graphics data record that is sent to a device or file.

Used in: GOPTIONS statement; GDEVICE procedure; GDEVICE Gend window

See also: GSTART

Syntax

GEND=*'string'* <...*'string-n'*>

'string'

can be either of the following:

*'hex-string'*X

'character-string'

In a GOPTIONS statement or in the GDEVICE procedure ADD or MODIFY statement, you can specify multiple strings with the GEND= option. In this case, you can mix the formats, specifying some as ASCII hexadecimal strings and some as character strings. Multiple strings are concatenated automatically.

In the GEND window, enter the hexadecimal string without either quotation marks or a trailing x. Note, however, that the string must be entered as a hexadecimal string.

PROC GOPTIONS always reports the value as a hexadecimal string.

Details

GEND is useful if you are creating a file and want to insert a carriage return at the end of every record. You can also use GEND in conjunction with the GSTART= graphics option or device parameter.

If you must specify the long and complicated initialization strings required by some devices (for example, PostScript printers), it is easier to use the GOPTIONS GEND= option rather than the GDEVICE Gend window because it is easier to code the string as text with GEND= than it is to convert the string to its ASCII representation, which is required to enter the string in the GDEVICE Gend window.

Note: On non-ASCII hosts, only ASCII hexadecimal strings produce consistent results in all instances because of the way the character strings are translated. In addition, the only way to specify a value for GEND that can be used by all hosts is to use an ASCII hexadecimal string; therefore, using an ASCII hexadecimal string to specify a value for GEND is the recommended method. △

GEPILOG

Sends a string to a device or file after all graphics commands are sent.

Used in: GOPTIONS statement; GDEVICE procedure; GDEVICE Gepilog window

See also: PREGEPILOG, POSTGEPILOG

Syntax

GEPILOG=*'string'* <...*'string-n'*>

'string'

can be either of the following:

'hex-string'X

'character-string'

In a GOPTIONS statement or in the GDEVICE procedure ADD or MODIFY statement, you can specify multiple strings with the GEPILOG= option. In this case, you can mix the formats, specifying some as ASCII hexadecimal strings and some as character strings. Multiple strings are concatenated automatically.

In the Gepilog window, enter the hexadecimal string without either quotation marks or a trailing x. Note, however, that the string must be entered as a hexadecimal string.

PROC GOPTIONS always reports the value as a hexadecimal string.

Details

GEPILOG can be used in conjunction with the GPROLOG= graphics option or device parameter.

If you must specify the long and complicated initialization strings required by some devices (for example, PostScript printers), it is easier to use the GOPTIONS GEPILOG= option rather than the Gepilog window because it is easier to code the string as text with GEPILOG= than it is to convert the string to its ASCII representation, which is required to enter the string in the Gepilog window.

Note: On non-ASCII hosts, only ASCII hexadecimal strings produce consistent results in all instances because of the way the character strings are translated. In addition, the only way to specify a value for GEPILOG that can be used by all hosts is

to use an ASCII hexadecimal string; therefore, using an ASCII hexadecimal string to specify a value for GEPILOG is the recommended method. △

GFORMS

Specifies the JES form name for IBM printers.

Used in: GOPTIONS statement

Default: STD

Restriction: used only with IBM3287 and IBM3268 device drivers on OS/390 systems only

Syntax

GFORMS='forms-code'

GOUTMODE

Appends to or replaces the graphics output catalog.

Used in: GOPTIONS statement

Default: APPEND

Syntax

GOUTMODE=APPEND | REPLACE

APPEND

adds each new graph to the end of the current catalog.

REPLACE

replaces the contents of the catalog with the graph or graphs produced by a single procedure.

CAUTION:

If you specify REPLACE, the *entire contents* of the catalog are replaced, not just graphs of the same name. Graphs are added to the catalog for the duration of the procedure, but when the procedure ends and a new procedure begins, the contents of the catalog are deleted and the new graph or graphs are added. △

GPROLOG

Sends a string to device or file before graphics commands are sent.

Used in: GOPTIONS statement; GDEVICE procedure; GDEVICE Gprolog window

See also: PREGPROLOG, POSTGPROLOG

Syntax

GPROLOG=*'string'* <...*'string-n'*>

'string'

can be either of the following:

*'hex-string'*X

'character-string'

In a GOPTIONS statement or in the GDEVICE procedure ADD or MODIFY statement, you can specify multiple strings with the GPROLOG= option. In this case, you can mix the formats, specifying some as ASCII hexadecimal strings and some as character strings. Multiple strings are concatenated automatically.

In the GPROLOG window, enter the hexadecimal string without either quotation marks or a trailing x. Note, however, that the string must be entered as a hexadecimal string.

PROC GOPTIONS always reports the value as a hexadecimal string.

Details

GPROLOG can be used in conjunction with the GEPiLOG= graphics option or device parameter.

If you must specify the long and complicated initialization strings required by some devices (for example, PostScript printers), it is easier to use the GOPTIONS GPROLOG= option rather than the GDEVICE Gprolog window because it is easier to code the string as text with GPROLOG= than it is to convert the string to its ASCII representation, which is required to enter the string in the GDEVICE Gprolog window.

Note: On non-ASCII hosts, only ASCII hexadecimal strings produce consistent results in all instances because of the way the character strings are translated. In addition, the only way to specify a value for GEND that can be used by all hosts is to use an ASCII hexadecimal string; therefore, using an ASCII hexadecimal string to specify a value for GEND is the recommended method. Δ

GPROTOCOL

Specifies the protocol module to use when routing output directly to a printer or creating a graphics stream file (GSF) to send to a device attached to your host by a protocol converter.

Used in: GOPTIONS statement; GDEVICE procedure; GDEVICE Host File Options window

Default: host dependent

Syntax

GPROTOCOL=*module-name*

module-name can be one of these

SASGPADE*

SASGPAGL*

SASGPASC

SASGPAXI*

SASGPCAB*

SASGPCHK*

SASGPDAT*

SASGPDCA*

SASGPHEX

SASGPHYD*

SASGPIDA*

SASGPIDX*

SASGPIMP*

SASGPIOC*

SASGPISI*

SASGPI24*

SASGPLCL*

SASGPNET*

SASGPMIC*

SASGPRTM*

SASGPSCS*

SASGPSTD

SASGPSTE*

SASGPTCX*

SASGPVAT*

SASGP497*

SASGP71

*Valid only for IBM mainframe systems.

Details

GPROTOCOL= specifies whether the graphics data generated by the SAS/GRAPH device driver should be altered and how the data should be altered. Unless you are using a protocol converter on an IBM mainframe, most devices do not require that the data be altered, and ordinarily, you do not have to change the default of GPROTOCOL.

On IBM hosts, the protocol module converts the graphics output to a format that can be processed by protocol converters. On other hosts, it can be used to produce a file in ASCII hexadecimal format.

Refer to the SAS Help facility for descriptions of these protocol modules.

Operating Environment Information: GPROTOCOL is valid only in certain operating environments. △

GRAPHRC

Specifies whether to return a step code at graphics procedure termination.

Used in: GOPTIONS statement

Default: GRAPHRC

Syntax

GRAPHRC | NOGRAPHRC

GRAPHRC

allows a return code at procedure termination. If the return code is not 0, the entire job may terminate.

NOGRAPHRC

always returns a step code of 0, even if the SAS/GRAPH program produced errors. As a result, the entire job's return code is unaffected by errors in any graphics procedure. NOGRAPHRC also overrides the ERRABEND system option.

Details

You typically use this option when you are running multiple jobs in a batch environment. It is useful primarily in an OS/390 batch environment.

GSFLEN

Controls the length of records written to the graphics stream file (GSF).

Used in: GOPTIONS statement; GDEVICE procedure; GDEVICE Host File Options window

Default: device dependent

See also: PROMPTCHARS

Syntax

GSFLEN=*record-length*

record-length

must be a nonnegative integer up to five digits long (0...99999). GSFLEN= specifies the length of the records written by the driver to a GSF or to the device.

If GSFLEN is 0, SAS/GRAPH uses the sixth byte of the PROMPTCHARS string to determine the length of the records. If the sixth byte of the PROMPTCHARS string is 00, the device driver sets the record length.

If you specify GACCESS=SASGAFIX and omit GSFLEN=, SAS/GRAPH uses the default length for the device.

Some values of the GPROTOCOL device parameter cause each byte in the data stream to be expanded to two bytes. This expansion is done after the length of the record is set by GSFLEN. If you are specifying a value for GPROTOCOL that does this (for example, SASGPHEX, SASGPLCL, or SASGPAGL), specify a value for GSFLEN that is half of the actual record length desired. For example, a value of 64 produces a 128-byte record after expansion by the GPROTOCOL module.

GSFMODE

Specifies the disposition of records written to a graphics stream file (GSF) or to a device or communications port by the device driver.

Used in: GOPTIONS statement; GDEVICE procedure; GDEVICE Host File Options window

Default: REPLACE

See also: GACCESS, GSFNAME

Syntax

GSFMODE=APPEND | PORT | REPLACE

APPEND

adds the records to the end of a GSF designated by the GACCESS= or GSFNAME= graphics option or device parameter. If the file does not already exist, it is created.

The destination can be either a specific file or an aggregate file storage location.

If the destination of the GSF is a specific file and you specify APPEND, SAS/GRAPH will add the new records to an existing GSF of the same name.

If the destination of the GSF is a file location and not a specific file, SAS/GRAPH will add the records to an external file whose name matches the name of the newly created catalog entry. For more information on how SAS/GRAPH names catalog entries, see “Exporting SAS/GRAPH Output with Program Statements” on page 58.

Note: Some viewers of bitmapped output can view only one graph, even though multiple graphs are stored in the file. Therefore it may appear that a file contains only one graph when in fact it contains multiple graphs. △

PORT

sends the records to a device or communications port. The GACCESS= graphics option or device parameter should point to the desired port or device.

REPLACE

replaces the existing contents of a GSF designated by the GACCESS= or GSFNAME= graphics option or device parameter. If the file does not exist, it is created. REPLACE is always the default, regardless of the destination of the GSF.

If the destination of the GSF is a specific file and you specify REPLACE, SAS/GRAPH will replace an existing GSF with the contents of a newly created GSF of the same name.

If the destination of the GSF is a file location and not a specific file, SAS/GRAPH will replace an external file whose name matches the name of the newly created catalog entry. For more information on how SAS/GRAPH names catalog entries, see “Exporting SAS/GRAPH Output with Program Statements” on page 58.

Details

When you create a GSF, the GSFNAME= or GACCESS= graphics option or device parameter controls where the output goes, and GSFMODE= controls how the driver writes graphics output records. If the output is to go to a file, specify APPEND or REPLACE. If the output is to go directly to a device or to a communications port, specify PORT. See “About Graphics Stream Files” on page 56 for more information on creating a graphics stream file.

GSFNAME

Specifies the fileref of the file or aggregate file location to which graphics stream file records are written.

Used in: GOPTIONS statement; GDEVICE procedure; GDEVICE Host File Options window

Restriction: Not valid for IBM32xx, linkable, or Metagraphics drivers.

See also: GACCESS, GSFMODE

Syntax

GSFNAME=*fileref*

fileref

specifies a fileref that points to the destination for the graphics stream file (GSF) output. *Fileref* must be a valid SAS fileref up to eight characters long and must be assigned with a FILENAME statement prior to running a SAS/GRAPH procedure that uses that fileref. The destination specified by the FILENAME statement can be either a specific file or an aggregate file location. See “FILENAME Statement” on page 24 for additional information on the FILENAME statement.

Details

Whether the resulting graphs are stored as one file or many files depends on both the type of destination and the setting of the GSFMODE= option.

If you specify a fileref with GSFNAME= and forget the FILENAME statement that defines the fileref, and if a destination is specified by the GACCESS= graphics option or

device parameter, SAS/GRAPH assigns that destination to the fileref and sends the graphics output there. See also “GACCESS” on page 337.

See “About Graphics Stream Files” on page 56 for more information on creating graphics stream files.

GSFPROMPT

Specifies whether to write prompt messages to the graphics stream file (GSF).

Used in: GOPTIONS statement

Default: NOGSFPROMPT

Syntax

GSFPROMPT | NOGSFPROMPT

Details

When the GSF is processed by another program, that program can display the prompt messages. The default, NOGSFPROMPT, is compatible with Release 6.06.

Although the prompt messages appear if the graphics device is in eavesdrop mode, they do not wait for user response. If GSFPROMPT is on, the prompt messages are sent with the GSF to the device, regardless of the status of the graphics options PROMPT, GACCESS=, GSFMODE=, or GSFNAME=.

GSIZE

Sets the number of lines of display used for graphics for devices whose displays can be divided into graphics and text areas.

Used in: GOPTIONS statement; GDEVICE procedure; GDEVICE Parameters window

Default: device dependent

Syntax

GSIZE=*lines*

lines

specifies the number of lines to be used for graphics. *Lines* is a nonnegative integer up to three digits long (0...999), and can be larger or smaller than the total number of lines that can be displayed at one time. If the number is larger, scroll the graph to see it all. If GSIZE is 0, all lines are used for text.

GSTART

Prefixes every record of graphics data sent to a device or file with a string of characters.

Used in: GOPTIONS statement; GDEVICE procedure; GDEVICE Gstart window

Default: none

See also: GEND

Syntax

GSTART=*'string <... 'string-n'>*

'string'

can be either of the following:

'hex-string'X

'character-string'

In a GOPTIONS statement or in the GDEVICE procedure ADD or MODIFY statement, you can specify multiple strings with the GSTART= option. In this case, you can mix the formats, specifying some as ASCII hexadecimal strings and some as character strings. Multiple strings are concatenated automatically.

In the GSTART window, enter the hexadecimal string without either quotation marks or a trailing x. Note, however, that the string must be entered as a hexadecimal string.

PROC GOPTIONS always reports the value as a hexadecimal string.

Details

GSTART is useful when sending a file to a device that requires each record be prefixed with some character. You can use GSTART= in conjunction with the GEND= graphics option or device parameter.

If you must specify the long and complicated initialization strings required by some devices (for example, PostScript printers), it is easier to use the GOPTIONS GSTART= option rather than the GDEVICE Gstart window because it is easier to code the string as text with GSTART= than it is to convert the string to its ASCII representation, which is required to enter the string in the GDEVICE Gstart window.

Note: On non-ASCII hosts, only ASCII hexadecimal strings produce consistent results in all instances because of the way the character strings are translated. In addition, the only way to specify a value for GEND that can be used by all hosts is to use an ASCII hexadecimal string; therefore, using an ASCII hexadecimal string to specify a value for GEND is the recommended method. △

GUNIT

Specifies the default unit of measure to use with height specifications.

Used in: GOPTIONS statement

Default: CELLS

Syntax

GUNIT=*units*

units must be one of

CELLS	character cells
CM	centimeters
IN	inches
PCT	percentage of the graphics output area.
PT	points (there are approximately 72 points in an inch)

Featured in: “Example 1. Ordering Axis Tick Marks with SAS Datetime Values” on page 266 and “Example 3. Rotating Plot Symbols through the Colors List” on page 271

Details

Used with options in the AXIS, FOOTNOTE, LEGEND, NOTE, SYMBOL, and TITLE statements and in some graphics options. If you specify a value but do not specify an explicit unit, the value of the GUNIT= graphics option is used.

GWAIT

Specifies the time between each graph displayed in a series.

Used in: GOPTIONS statement

Default: GWAIT=0

Syntax

GWAIT=*seconds*

seconds

specifies the number of seconds between graphs. *Seconds* can be any reasonable positive integer. By default, GWAIT=0, which means that you must press the RETURN key between each display in a series of graphs.

Details

GWAIT= enables you to view a series of graphs without having to press the ENTER key (or the RETURN or END key, depending on your device) between each display. For example, if you specify GWAIT=5, five seconds elapse between the display of each graph in a series. If you use the NOPROMPT graphics option, the GWAIT= graphics option is disabled.

GWRITER

Specifies the name of the external writer used with IBM printers.

Used in: GOPTIONS statement

Default: SASWTR

Restriction: Used only with IBM3287 and IBM3268 device drivers on OS/390 systems

Syntax

GWRITER=*'writer-name'*

HANDSHAKE

Specifies the type of flow control used to regulate the flow of data to a hardcopy device.

Used in: GOPTIONS statement; GDEVICE procedure; GDEVICE Parameters window

Default: host dependent

Syntax

HANDSHAKE=HARDWARE | NONE | SOFTWARE | XONXOFF

HARDWARE

HARD

specifies that SAS/GRAPH instruct the device to use the hardware CTS and RTS signals. (This is not appropriate for some devices.)

NONE

specifies that SAS/GRAPH send data without providing flow control. Specify NONE only if the hardware or interface program you are using provides its own flow control.

SOFTWARE

SOFT

specifies that SAS/GRAPH use programmed flow control with plotters in eavesdrop mode.

XONXOFF

X

specifies that SAS/GRAPH instruct the device to use ASCII characters DC1 and DC3. (This is not appropriate for some devices.)

Details

HANDSHAKE regulates flow of control by specifying how and if a device can signal to the host to temporarily halt transmission and then resume it. Flow control is important

because it is possible to send commands to a hardcopy device faster than they can be executed.

HANDSHAKE can be used when you are using a protocol converter, interface program, or host computer that can perform XONXOFF or hardware handshaking. You also can use this option if you are routing output through flow-control programs of your own, as in a multiple-machine personal computer environment where the graphics plotter is a shared resource. SAS/GRAPH software sends output to a server (the file transfer does not require flow control). The server queues incoming graphs and sends them to the plotter. The server, rather than SAS/GRAPH software, is responsible for handling flow control.

If you do not use HANDSHAKE, the value in the driver entry is used.

If you use HANDSHAKE=XONXOFF or HANDSHAKE=HARDWARE, SAS/GRAPH does not actually do the handshaking. It tells the device which type of handshake is being used. The protocol converter, interface program, or host computer actually does the handshake.

Note: If you are creating a graphics stream file using a driver for a plotter and you specify HANDSHAKE=SOFTWARE, the software that you use to send the file to the plotter must be able to perform a software handshake. You will probably want to specify one of the alternative values if you route output to a file. △

HBY

Specifies the height of BY lines generated when you use BY-group processing.

Used in: GOPTIONS statement

Default: 1 cell unless HTEXT= is used

See also: "BY Statement" on page 177

Syntax

HBY=BY-line-height <units>

BY-line-height <units>

specifies the height of BY-line text; by default *BY-line-height* is 1. If you specify *HBY=0*, the BY headings are suppressed. For a description of *units*, see "Dictionary of Graphics Options and Device Parameters" on page 304.

Note: If a value for *units* is not specified, the current units associated with the GUNIT graphics option are used. △

Details

When you use a BY statement with a SAS/GRAPH procedure to process a data set in subgroups, each graph produced by that procedure is headed by a BY line that displays the BY variables and their values that define the current subgroup.

HEADER

Specifies the command that executes a user-supplied program to create HEADER records for the driver.

Used in: GDEVICE procedure; GDEVICE Metagraphics window

Restriction: Used only with user-supplied Metagraphics drivers.

See also: HEADERFILE

Syntax

HEADER=*command*

command

specifies a command that runs a user-written program that creates the file of HEADER records. *Command* is a string up to 40 characters long.

Details

For information about Metagraphics drivers, contact Technical Support.

HEADERFILE

Specifies the fileref for the file from which the Metagraphics driver reads HEADER records.

Used in: GDEVICE procedure; GDEVICE Metagraphics window

Restriction: Used only with user-supplied Metagraphics drivers.

See also: HEADER

Syntax

HEADERFILE=*fileref*

fileref

specifies a valid SAS fileref up to eight characters long. *Fileref* must have been previously assigned with a FILENAME statement or a host command before running the Metagraphics driver. See “FILENAME Statement” on page 24 for details.

Details

For information about Metagraphics drivers, contact Technical Support.

HORIGIN

Sets the horizontal offset from the lower-left corner of the display area to the lower-left corner of the graph.

Used in: GOPTIONS statement; GDEVICE procedure; GDEVICE Detail window

See also: VORIGIN

Syntax

HORIGIN=*horizontal-offset* <IN | CM | PT>

horizontal-offset <IN | CM | PT>

must be a nonnegative number and may be followed by a unit specification, either IN for inches (default), or CM for centimeters, or PT for points. If you do not specify HORIGIN, a default offset is searched for in this order:

- 1 the left margin specification on an OPTIONS LEFTMARGIN setting
- 2 HORIGIN setting in the device catalog.

Details

The display area is defined by the XMAX and YMAX device parameters. By default, the origin of the graphics output area is the lower-left corner of the display area; the graphics output is offset from the lower-left corner of the display area by the values of HORIGIN and VORIGIN. HORIGIN + HSIZE cannot exceed XMAX. See “About the Graphics Output Area” on page 29 for details.

HOSTSPEC

Stores FILENAME statement options in the device entry.

Used in: GDEVICE procedure; GDEVICE Host File Options window

Syntax

HOSTSPEC=*'text-string'*

text-string

specifies FILENAME statement options that are valid for the operating environment. *Text-string* accepts characters in upper or lower case. See the SAS documentation for your operating environment for details.

Details

HOSTSPEC may be used when the driver dynamically allocates a graphics stream file or spool file. It can specify the attributes of the file, such as record format or record length. It cannot be used with Metagraphics drivers.

HPOS

Specifies the number of columns in the graphics output area.

Used in: GOPTIONS statement

Default: device dependent: the value of the LCOLS or PCOLS device parameter

See also: PCOLS, LCOLS, VPOS

Syntax

HPOS=*columns*

columns

specifies the number of columns in the graphics output area, which is equivalent to the number of hardware characters that can be displayed horizontally. Specifying HPOS=0 causes the device driver to use the default hardware character cell width for the device.

Details

The HPOS= graphics option overrides the values of the LCOLS or PCOLS device parameters and temporarily sets the number of columns in the graphics output area. HPOS= does not affect the width of the graphics output area but merely divides it into columns. Therefore, you can use HPOS= to control cell width.

The values specified in the HPOS= and VPOS= graphics options determine the size of a character cell for the graphics output area and consequently the size of many graphics elements, such as hardware text. The larger the size of the HPOS= and VPOS= values, the smaller the size of each character cell.

See “Procedure Output and the Graphics Output Area” on page 29 for more information.

HSIZE

Sets the horizontal size of the graphics output area.

Used in: GOPTIONS statement; GDEVICE procedure; GDEVICE Detail window

See also: VSIZE, XMAX

Syntax

HSIZE=*horizontal-size* <IN | CM | PT>

horizontal-size <IN | CM | PT>

specifies the width of the graphics output area; *horizontal-size* must be a positive number and may be followed by a unit specification, either IN for inches (default), or CM for centimeters, or PT for points.

If you do not specify HSIZE=, a default size is searched for in this order:

- 1 the horizontal size is calculated as

$$\text{XMAX} - \text{LEFTMARGIN} - \text{RIGHTMARGIN}$$

Note that LEFTMARGIN and RIGHTMARGIN are used in the OPTIONS statement.

- 2 HSIZE setting in the device catalog.

Featured in: “Example 3. Rotating Plot Symbols through the Colors List” on page 271

HTEXT

Specifies the default height of the text in the graphics output.

Used in: GOPTIONS statement

Default: 1 cell

Syntax

HTEXT=*text-height* <units>

text-height <units>

specifies the height of the text; by default *text-height* is 1. For a description of *units*, see “Dictionary of Graphics Options and Device Parameters” on page 304.

Note: If a value for *units* is not specified, the current units associated with the GUNIT graphics option are used. △

Featured in: “Example 1. Ordering Axis Tick Marks with SAS Datetime Values” on page 266

Details

HTEXT= is overridden by the HTITLE= graphics option for the *first* TITLE line.

HTITLE

Selects the default height used for the first TITLE line.

Used in: GOPTIONS statement

Default: 2 cells unless HTEXT= is used

Syntax

HTITLE=*title-height* <units>

title-height <units>

specifies the height of the text in the TITLE1 statement. By default, *title-height* is 2. For a description of *units*, see “Dictionary of Graphics Options and Device Parameters” on page 304.

Note: If a value for *units* is not specified, the current units associated with the GUNIT graphics option are used. Δ

Featured in: “Example 1. Ordering Axis Tick Marks with SAS Datetime Values” on page 266

Details

If you omit HTITLE=, TITLE1 uses the height specified by the HTEXT= graphics option, if used.

ID

Specifies the description string used by the Metagraphics driver.

Used in: GDEVICE procedure; GDEVICE Metagraphics window

Restriction: Used only with user-supplied Metagraphics drivers.

Syntax

ID=*description*

description

is a character string up to 70 characters long. If this field is blank, the name and description of the graph as specified in the PROC GREPLAY window of the GREPLAY procedure are used.

Details

For information about Metagraphics drivers, contact Technical Support.

INTERACTIVE

Sets level of interactivity for Metagraphics driver.

Used in: GDEVICE procedure; GDEVICE Metagraphics window

Default: USER

Restriction: Used only with user-supplied Metagraphics drivers.

Syntax

INTERACTIVE=USER | GRAPH | PROC

USER

specifies that the user-written part of the driver be executed outside of SAS/GRAPH.

PROC

specifies that the user-written part of the Metagraphics driver be invoked after the procedure is complete.

GRAPH

specifies that the user-written part be invoked for each graph.

Details

For information about Metagraphics drivers, contact Technical Support.

INTERLACED

Specifies whether images are to be displayed as they are received in the browser.

Used in: GOPTIONS statement

Default: NONINTERLACED

Restriction: driver dependent, GIF series of drivers only

Syntax

INTERLACED | NONINTERLACED

Details

With interlacing it is possible to get a rough picture of what a large image will look like before it is completely drawn in your browser. Your browser may allow you to set an option that will determine how images are displayed.

INTERPOL

Sets the default interpolation value for the SYMBOL statement.

Used in: GOPTIONS statement

Syntax

INTERPOL=*interpolation-method*

interpolation-method

specifies the default interpolation to be used when the INTERPOL= option is not specified in the SYMBOL statement. See “SYMBOL Statement” on page 226 for the complete syntax of all interpolation methods.

ITERATION

Specifies the number of times to repeat the animation loop.

Used in: GOPTIONS statement

Default: 0

Restriction: GIFANIM driver only

Syntax

ITERATION=*iteration-count*

iteration-count

specifies the number of times that your complete GIF animation loop is repeated.

Iteration-count can be a number from 0...65535. A value of 0 causes the animation to loop continuously.

Details

In Version 6, the GCOPIES graphics option controlled iteration for the GIFANIM driver.

KEYMAP

Selects the keymap to use.

Used in: GOPTIONS statement

Default: installation dependent

Syntax

KEYMAP=*key-map-name* | NONE

key-map-name

specifies the name of a keymap. See Chapter 18, “The GKEYMAP Procedure,” on page 719 for details.

NONE

suppresses the keymap assigned by default to a non-U.S. keyboard. If you specify KEYMAP=NONE, text may display incorrectly or not at all.

Details

Non-default key maps usually are used only with non-U.S. keyboards.

LCOLS

Sets the number of columns in the graphics output area for landscape orientation.

Used in: GDEVICE procedure; GDEVICE Detail window

Default: device dependent

See also: HPOS, LROWS, PCOLS

Syntax

LCOLS=*landscape-columns*

landscape-columns

must be a nonnegative integer up to three digits long (0...999).

Details

Either the LROWS and LCOLS pair of device parameters or the PROWS and PCOLS pair of device parameters are required and must be nonzero.

The HPOS= graphics option overrides the value of LCOLS.

See “Procedure Output and the Graphics Output Area” on page 29 for more information.

LFACTOR

Selects the default hardware line thickness.

Used in: GOPTIONS statement; GDEVICE procedure; GDEVICE Parameters window

Default: device dependent

Restriction: Used only with devices that can draw hardware lines of varying thicknesses.

Syntax

LFACTOR=*line-thickness-factor*

line-thickness-factor

can range from 0 through 9999. A value of 0 for LFACTOR is the same as a factor of 1. Lines are drawn *line-thickness-factor* times as thick as normal.

Details

LFACTOR is useful when you are printing graphics output on a plotter. Depending on the orientation and type of device, some plotters may require LFACTOR=10 to get the same thickness of lines as on the display of some devices.

LROWS

Sets the number of rows in the graphics output area for landscape orientation.

Used in: GDEVICE procedure; GDEVICE Detail window

Default: device dependent

See also: LCOLS, PROWS, VPOS

Syntax

LROWS=*landscape-rows*

landscape-rows

is a nonnegative integer up to three digits long (0...999).

Details

Either the LROWS and LCOLS pair of device parameters or the PROWS and PCOLS pair of device parameters are required and must be nonzero.

The VPOS= graphics option overrides the value of LROWS.

See “Procedure Output and the Graphics Output Area” on page 29 for more information.

MAXCOLORS

Sets the total number of colors that can be displayed at once.

Used in: GDEVICE procedure; GDEVICE Parameters window

Default: device dependent

See also: PENMOUNTS

Syntax

MAXCOLORS=*number-of-colors*

number-of-colors

must be an integer in the range 2 through 256. The total number of colors includes the foreground colors plus the background color.

Details

The PENMOUNTS= graphics option overrides the value of MAXCOLORS.

MAXPOLY

Sets the maximum number of vertices for hardware-drawn polygons.

Used in: GDEVICE procedure; GDEVICE Parameters window

Default: device dependent

Syntax

MAXPOLY=*number-of-vertices*

number-of-vertices

is a nonnegative integer up to four digits long. A value of 0 means that there is no limit to the number of vertices that can be specified in the hardware's polygon-drawing command. The maximum value of MAXPOLY depends on the number of vertices your device can process.

MODEL

Specifies the model number of the output device.

Used in: GDEVICE procedure; GDEVICE Detail window

Default: device dependent

Syntax

MODEL=*model-number*

model-number

is a nonnegative integer up to five digits long that is the Institute-designated model number for the corresponding device. It is not the same as a manufacturer's model number.

Details

Do not change this field in Institute-supplied drivers or in drivers that you copy from Institute-supplied drivers.

MODULE

Specifies the name of the corresponding executable driver module for the device.

Used in: GDEVICE procedure; GDEVICE Detail window

Default: device dependent

Syntax

MODULE=*driver-module*

driver-module

is a literal string up to eight characters long. All standard driver modules begin with the characters SASGD.

Details

Do not change this field in Institute-supplied drivers or in drivers that you copy from Institute-supplied drivers.

NAK

Specifies the negative response for software handshaking for Metagraphics drivers.

Used in: GDEVICE procedure; GDEVICE Metagraphics window

Restriction: Used only with user-supplied Metagraphics drivers.

Syntax

NAK='negative-handshake-response'X

negative-handshake-response

is a hexadecimal string up to 16 characters long.

Details

For information about Metagraphics drivers, contact Technical Support.

OFFSHADOW

Controls the width and depth of the drop shadow in legend frames.

Used in: GOPTIONS statement

Default: (0.0625, - 0.0625) IN

Syntax

OFFSHADOW=(*x* <units>, *y* <units>) | (*x,y*) <units>

x,y

specify the width (*x*) and depth (*y*) of the drop shadow generated by the LEGEND statement.

If a value for *units* is not specified, the current units associated with the GUNIT graphics option are used. For a description of *units*, see “Dictionary of Graphics Options and Device Parameters” on page 304.

Details

The values specified by OFFSHADOW= are used with the CSHADOW= and CBLOCK= options in a LEGEND statement. For details, see “LEGEND Statement” on page 187.

PAPERDEST

Specifies which output bin the printer should use if multiple bins are available on the device.

Used in: GOPTIONS statement; OPTIONS statement

Default: 1 (the upper output bin)

Restriction: hardware dependent, PostScript printers require a PPD file

See also: PAPERSOURCE, PPDFILE

Syntax

PAPERDEST=*bin*

bin

specifies the name or number of the output bin. Values for *bin* depend on the type of printer and can be one of the following:

bin the name or number of the output bin – for example, PAPERDEST=4, PAPERDEST=BIN2, PAPERDEST=SIDE

'*long bin name*' a character string that is the name of the output bin – for example, PAPERDEST='Top Output Bin'. Names with blanks or special characters must be quoted.

For PostScript printers, the value for *bin* must correspond to an OutputBin value in the PPD file.

For PCL printers, consult the printer's documentation for valid bin values. If a numeric value exceeds the maximum bin value allowed for the printer, a warning message is issued. For string values, the string is checked against a list of strings that are valid for the driver (for example, 'UPPER', 'LOWER', or 'OPTIONALOUTBIN*n*', where *n* is the bin number). If the string is not valid for the driver, a warning message is issued.

PAPERFEED

Specifies the increment of paper that is ejected when a graph is completed.

Used in: GOPTIONS statement; GDEVICE procedure; GDEVICE Detail window

Default: PAPERFEED=0.0 IN

Restriction: device dependent

Syntax

PAPERFEED=*feed-increment* <IN | CM>

***feed-increment* <IN | CM>**

must be a nonnegative number and may be followed by a unit specification, either IN for inches (default) or CM for centimeters.

Details

PAPERFEED does not control the total length of the ejection. If you specify PAPERFEED=1, the driver ejects paper in 1-inch increments until the total amount of paper ejected is at least half an inch greater than the size of the graph last printed. If you specify PAPERFEED=8.5 IN, the paper is ejected in increments of 8.5 inches, measuring from the origin of the first graph.

PAPERFEED is provided mainly for plotters that use fanfold or roll paper. If you are using fanfold paper, specify a value for PAPERFEED that is equal to the distance between the perforations.

PAPERLIMIT

Sets the width of the paper used with plotters.

Used in: GOPTIONS statement

Default: maximum dimensions specified in the device driver

Restriction: ZETA plotters and KMW rasterizers

Syntax

PAPERLIMIT=*width* <IN | CM>

***width* <IN | CM>**

specifies the paper width in IN for inches (default) or CM for centimeters. If PAPERLIMIT= is not specified, the maximum dimensions of the graph are restricted by the hardware limits of the graphics device.

Details

If you want to use a driver with a device that has a larger plotting area than the device for which the driver is intended (for example, using the ZETA887 driver with a ZETA 836 plotter), the PAPERLIMIT= graphics option can be used to override the size limit of the driver.

PAPERSIZE

Specifies the name of a paper size.

Used in: GOPTIONS statement; OPTIONS statement

Default: device dependent

Restriction: hardware dependent, PostScript printers require a PPD file

See also: PAPERSOURCE, PPDFILE

Syntax

PAPERSIZE=*size-name*

size-name

specifies the name of a paper size, such as LETTER, LEGAL, or A4.

If you do not specify PAPERSIZE=, the PAPERSIZE= setting on an OPTIONS statement is used. If no OPTIONS statement sets a paper size, the value for paper size is device dependent:

- For the Windows and OS/2 operating environments, the host printer devices (WINPRTx and OS2PRTx) use the setting in the SAS Print Setup window
- All other printer devices in all operating environments use the LETTER paper size.

Details

Typically, you might use PAPERSIZE= with the Output Delivery System (ODS). For some printers, PAPERSIZE= overrides the PAPERSOURCE= selection.

For PostScript devices, the name must match the name of a paper size in the PPD file. Refer to the PPD file for a list of valid names. *Size-name* is case-insensitive and can contain a subset of the full name. For example, if the name in the PPD file is *PageSize A4/A4, you can specify PAPERSIZE='A4'. If a PPD file is not specified, PAPERSIZE= is ignored.

For PCL devices, the device driver searches the SAS Registry for supported paper size values. To see the supported list of sizes, submit the following statements:

```
proc registry listhelp
    startat='options\papersize';
run;
```

For more information about the SAS Registry, refer to the SAS Help facility.

PAPERSOURCE

Specifies which paper tray the printer should use if multiple trays are available on the device.

Used in: GOPTIONS statement; OPTIONS statement

Default: device dependent

Restriction: hardware dependent, PostScript printers require a PPD file

See also: PAPERDEST, PAPERSIZE, PPDFILE

Syntax

PAPERSOURCE=*tray*

tray

specifies the name or number of the paper tray. Values for *tray* depend on the type of printer and can be one of the following:

- | | |
|---------------------------|---|
| <i>tray</i> | the name or number of the paper tray, for example, PAPERSOURCE=3, PAPERSOURCE=TRAY3, PAPERSOURCE=Upper |
| ' <i>long tray name</i> ' | a character string that is the name of the paper tray, for example, PAPERSOURCE='Optional Output Tray'. Names with blanks or special characters must be quoted. |

Details

On some printers, if PAPERSIZE= is also specified, it overrides the setting on PAPERSOURCE=.

For PostScript printers, a tray number, such as PAPERSOURCE='tray3', must correspond to an InputSlot value in the PPD file.

For PCL printers, consult the printer's documentation for valid tray values. If a numeric value exceeds the maximum tray value allowed for the printer, a warning message is issued. For string values, the string is checked against a list of strings that are valid for the driver:

- 'AUTO'
- 'HCI' or 'HCI*n*', where *n* is a number from 2 to 21
- 'MANUAL'
- 'MANUAL_ENVELOPE'
- 'TRAY*n*', where *n* is 1, 2, or 3.

If the string is not valid for the driver, a warning message is issued.

PAPERTYPE

Specifies the name of a paper type.

Used in: GOPTIONS statement; OPTIONS statement

Default: PLAIN

Restriction: hardware dependent, PostScript printers require a PPD file

See also: PPDFILE

Syntax

PAPERTYPE='*type-name*'

type-name

specifies the name of a paper type. Valid values depend on the type of printer.

For PostScript devices, *type-name* must match the name of a paper type in the PPD file, such as TRANSPARENCY or PLAIN. Refer to the PPD file for a list of valid names. *Type-name* is case-insensitive and can contain a subset of the full name. For

example, if the name in the PPD file is *MediaType Plain/Paper you can specify PAPERTYPE='PLAIN/PAPER'.

For PCL devices, *type-name* specifies the name of a paper type that is available on the current printer, such as GLOSSY, PLAIN, SPECIAL, or TRANSPARENCY. Consult your printer's user manual for the complete list of available paper types on your printer.

Details

For PostScript devices, if a PPD file is not specified, PAPERTYPE= is ignored.

PATH

Sets the increment of the angle for hardware text rotation.

Used in: GDEVICE procedure; GDEVICE Metagraphics window

Default: PATH=0

Restriction: Used only with user-supplied Metagraphics drivers.

Syntax

PATH=*angle-increment*

angle-increment

is an integer in the range 0 to 360 that specifies the angle at which to rotate the text baseline. A value of 0 means that the device uses its default orientation. Specify 0 if your device does not perform string angling in hardware.

Details

For information about Metagraphics drivers, contact Technical Support.

PCLIP

Specifies whether a clipped polygon is stored in its clipped or unclipped form.

Used in: GOPTIONS statement

Default: NOPCLIP

See also: POLYGONCLIP

Syntax

PCLIP | NOPCLIP

PCLIP

stores clipped polygons with the graph in the default catalog WORK.GSEG, or in the catalog you specify.

NOPClip

stores the unclipped form of the polygon and causes the polygon to be clipped when replayed.

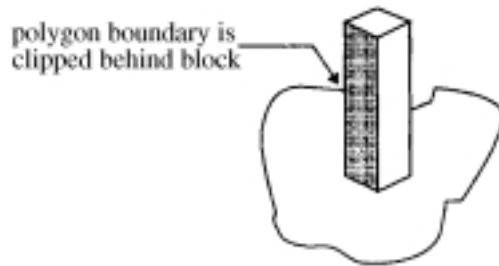
Details

The effects of this option are only seen when you use the graphics editor to edit a graph.

When a procedure produces a graph with intersecting polygons or blanking areas, it clips portions of the polygons to prevent the ones behind from showing through. When the graph is created and stored in a catalog, if PCLIP is in effect, the clipped form of the polygon is stored with it. If NOPCLIP is specified, the complete polygon is stored in the catalog and the graph is clipped each time it is replayed.

For example, suppose you create a block map like the one in Figure 9.1 on page 370.

Figure 9.1 Intersecting Polygons



The block clips the boundary of the map area polygon. If you specify PCLIP, the map area polygon is stored in its clipped form, as shown in Figure 9.2 on page 370.

Figure 9.2 Clipped Polygon with PCLIP Option



NOPClip stores the map area in its unclipped form, as shown in Figure 9.3 on page 370.

Figure 9.3 Polygon with NOPCLIP Option



In this case, when the graph is recalled from the catalog, the map area polygon must be clipped before it is displayed with the block. If you plan to edit the graph with the graphics editor, specify NOPCLIP so polygons retain their original form.

PCOLS

Sets the number of columns in the graphics output area for portrait orientation.

Used in: GDEVICE procedure; GDEVICE Detail window

Default: device dependent

See also: HPOS, LCOLS, PROWS

Syntax

PCOLS=*portrait-columns*

portrait-columns

must be a nonnegative integer up to three digits long (0...999).

Details

Either the LROWS and LCOLS pair of device parameters or the PROWS and PCOLS pair of device parameters are required and must be nonzero.

The HPOS= graphics option overrides the value of PCOLS.

See "Procedure Output and the Graphics Output Area" on page 29 for more information.

PENMOUNTS

Specifies the number of active pens or colors.

Used in: GOPTIONS statement

Default: device dependent

See also: MAXCOLORS

Syntax

PENMOUNTS=*active-pen-mounts*

active-pen-mounts

specifies the number of pens for a plotter with multiple pens. After the specified number of pens have been used, you are prompted to change the pens.

Details

For devices that are not pen plotters, PENMOUNTS= can be used to indicate the number of colors that can be displayed at one time. In this case, PENMOUNTS= performs the same function as the MAXCOLORS device parameter except that the value specified for MAXCOLORS includes the background color and PENMOUNTS only refers to foreground colors. Thus, PENMOUNTS=4 implies MAXCOLORS=5.

PENMOUNTS= overrides the value of the MAXCOLORS device parameter. You can specify MAXCOLORS= in a GOPTIONS statement as a synonym for PENMOUNTS=.

PENSORT

Specifies whether plotters draw graphics elements in order of color.

Used in: GOPTIONS statement; GDEVICE procedure; GDEVICE Parameters window

Default: device dependent

Syntax

GOPTIONS: PENSORT | NOPENSORT

GDEVICE: PENSORT=Y | N

PENSORT

PENSORT=Y

causes the plotter to draw all graphics elements of one color at one time. For example, it draws all the red elements in the output, then all the blue elements, and so forth. This specification is compatible with previous releases. Use it for plotters with real pens.

NOPENSORT

PENSORT=N

causes the plotter to draw each element as it is encountered, regardless of its color. For example, the plotter might draw a red circle, then a blue line, and then a red line, and so forth. This method is best for electrostatic printers implemented with Metagraphics drivers of TYPE=PLOTTER. In addition, NOPENSORT enables you to specify non-standard color names.

PIEFILL

Specifies whether to use the device's hardware pie-fill capability.

Used in: GOPTIONS statement; GDEVICE procedure; GDEVICE Parameters window

Default: device dependent

Syntax

GOPTIONS: PIEFILL | NOPIEFILL

GDEVICE: PIEFILL=Y | N

PIEFILL

PIEFILL=Y

causes SAS/GRAPH to use the built-in hardware capability of the device, if available, to fill pies and pie sections. A blank `Piefill` field in the Parameters window is the same as `PIEFILL=Y`.

Hardware drawing is faster, but not all devices have the capability. SAS/GRAPH does not try to use the capability if your device does not support it.

NOPIEFILL

PIEFILL=N

causes SAS/GRAPH to fill pies and pie sections using software pie fills.

POLYGONCLIP

Specifies the type of clipping used when two polygons overlap.

Used in: GOPTIONS statement

Default: device dependent

See also: PCLIP

Syntax

POLYGONCLIP | NOPOLYGONCLIP

POLYGONCLIP

specifies polygon clipping, which enables a clipped polygon to be filled with a hardware pattern. `POLYGONCLIP` affects only graphs that have blanking areas or intersecting polygons.

NOPOLYGONCLIP

specifies line clipping; a polygon that has been line-clipped cannot use a hardware pattern.

Details

Clipping is the process of removing part of one polygon when two polygons intersect. For example, in a block map, a block may overlap the boundary of its map area. In this case, the polygon that makes up the map area is clipped so that you do not see the boundary line behind the block. (See Figure 9.1 on page 370 for an illustration of a clipped polygon.) The type of clipping used by a graph affects whether a clipped area can use hardware patterns.

`POLYGONCLIP` is affected by the `PCLIP` graphics option:

`POLYGONCLIP` with `PCLIP` or `NOPClip`

all areas can use hardware patterns

`NOPOLYGONCLIP` with `NOPClip`

all areas use only software patterns

NOPOLYGONCLIP with PCLIP

areas may use either hardware or software patterns depending on the nature of the clipped polygons.

Under some conditions the polygons may not be clipped correctly. Specifying both POLYGONCLIP and NOPCLIP will produce the correct graph.

POLYGONFILL

Specifies whether to use the device's hardware polygon-fill capability.

Used in: GOPTIONS statement; GDEVICE procedure; GDEVICE Parameters window

Default: device dependent

Syntax

GOPTIONS: POLYGONFILL | NOPOLYGONFILL

GDEVICE: POLYFILL=Y | N

POLYGONFILL**POLYFILL=Y**

causes SAS/GRAPH to use the built-in hardware capability of the device to fill polygons. A blank `polyfill` field in the Parameters window is the same as POLYGONFILL.

Hardware drawing is faster, but not all devices have the capability. SAS/GRAPH does not try to use the capability if your device does not support it.

NOPOLYGONFILL**POLYFILL=N**

causes SAS/GRAPH to use software fills to fill polygons.

POSTGEPiLOG

Specifies data to send immediately after the data that is stored in the Gepilog field of the device entry is sent.

Used in: GOPTIONS statement

Default: Null string

See also: GEPILOG, PREGEPILOG

Syntax

POSTGEPiLOG=*'string'*

'string'

can be either of the following:

*'hex-string'*X

'character-string'

PROC GOPTIONS always reports the value as a hexadecimal string.

POSTGPROLOG

Specifies the data to send immediately after the data that is stored in the Gprolog field of the device entry is sent.

Used in: GOPTIONS statement

Default: Null string

See also: GPROLOG, PREGPROLOG

Syntax

POSTGPROLOG=*'string'*

'string'

can be either of the following:

*'hex-string'*X

'character-string'

PROC GOPTIONS always reports the value as a hexadecimal string.

POSTGRAPH

Specifies host commands to be executed after the graph is produced.

Used in: GOPTIONS statement; GDEVICE procedure; GDEVICE Host Commands window

See also: FILECLOSE

Syntax

POSTGRAPH1=*'system-command(s)'*

POSTGRAPH2=*'system-command(s)'*

system-command(s)

specifies one or more valid system commands. The string can contain upper- or lowercase characters. Separate multiple commands with a command delimiter, which is host-specific; for example, some operating environments use a semicolon. The total

length of the string cannot exceed 72 characters. The commands are executed right after the graph is produced.

Details

If you want to use a host command to send output to the device after each graph executes, use the POSTGRAPH parameter with FILECLOSE=GRAPHEND.

PPDFILE

Specifies the location of an external file containing PostScript Printer Description (PPD) information.

Used in: GOPTIONS statement

Restriction: PostScript printers only

See also: BINDING, COLLATE, DUPLEX, PAPERDEST, PAPERSIZE, PAPERSOURCE, PAPERTYPE, REVERSE

Syntax

PPDFILE=*fileref* | '*external-file*'

fileref

specifies a fileref that points to the PPD file you want to use. *Fileref* must be a valid SAS fileref up to eight characters long and must have been previously assigned with a FILENAME statement. See “FILENAME Statement” on page 24 for additional information.

external-file

specifies the complete filename of the PPD file you want to use. The format of *external-file* varies across operating environments. For details, see the SAS documentation for your operating environment.

Details

A PostScript Printer Description (PPD) file is a text file that contains commands required to access features of the device. These files are available from Adobe. Also, many printer manufacturers provide the appropriate PPD file for their PostScript printers.

PREGEPILOG

Specifies data to send immediately before the data that is stored in the Gepilog field of the device entry is sent.

Used in: GOPTIONS statement

Default: Null string

See also: GEPILOG, POSTGEPILOG

Syntax

PREGEPILOG=*'string'*

'string'

can be either of the following:

*'hex-string'*X

'character-string'

PROC GOPTIONS always reports the value as a hexadecimal string.

PREGPROLOG

Specifies the data to send immediately before the data that is stored in the Gprolog field of the device entry is sent.

Used in: GOPTIONS statement

Default: Null string

See also: GPROLOG, POSTGPROLOG

Syntax

PREGPROLOG=*'string'*

'string'

can be either of the following:

*'hex-string'*X

'character-string'

PROC GOPTIONS always reports the value as a hexadecimal string.

PREGRAPH

Specifies host commands to be executed before the graph is produced.

Used in: GOPTIONS statement; GDEVICE procedure; GDEVICE Host Commands window

See also: FILECLOSE

Syntax

PREGRAPH1=*'system-command(s)'*

PREGRAPH2=*system-command(s)*

system-command(s)

specifies one or more valid system commands. The string can contain upper- or lowercase characters. Separate multiple commands with a command delimiter, which is host-specific; for example, some operating environments use a semicolon. The total length of the string cannot exceed 72 characters. The commands are executed immediately before the graph is produced.

Details

The PREGRAPH parameter should be used with FILECLOSE=GRAPHEND.

PROCESS

Specifies the command that translates the metafile into commands for the device.

Used in: GDEVICE procedure; GDEVICE Metagraphics window

Restriction: Used only with user-supplied Metagraphics drivers.

See also: INTERACTIVE

Syntax

PROCESS=*command*

command

specifies the command that translates the metafile produced by the Metagraphics driver into commands for the device. The command runs your program to produce the output. *Command* is a string up to 40 characters long.

Details

PROCESS is required if the value of the INTERACTIVE device parameter is PROC or GRAPH.

For information about Metagraphics drivers, contact Technical Support.

PROCESSINPUT

Specifies the fileref for the file that contains input for the user-written part of the Metagraphics driver.

Used in: GDEVICE procedure; GDEVICE Metagraphics window

Restriction: Used only with user-supplied Metagraphics drivers.

Syntax

PROCESSINPUT=*fileref*

fileref

specifies a valid SAS fileref up to eight characters long. *Fileref* must be assigned with a FILENAME statement or a host command before running the Metagraphics driver. See “FILENAME Statement” on page 24 for additional information.

Details

For information about Metagraphics drivers, contact Technical Support.

PROCESSOUTPUT

Specifies the fileref for the file that receives output from the user-written part of the Metagraphics driver.

Used in: GDEVICE procedure; GDEVICE Metagraphics window

Restriction: Used only with user-supplied Metagraphics drivers.

Syntax

PROCESSOUTPUT=*fileref*

fileref

specifies a valid SAS fileref up to eight characters long. *Fileref* must be assigned with a FILENAME statement or a host command before running the Metagraphics driver. See “FILENAME Statement” on page 24 for additional information.

Details

For information about Metagraphics drivers, contact Technical Support.

PROMPT

Specifies whether prompts are issued.

Used in: GOPTIONS statement; GDEVICE procedure; GDEVICE Parameters window

Default: device dependent

Syntax

GOPTIONS: PROMPT | NOPROMPT

GDEVICE: PROMPT=0...7

PROMPT

causes all prompts to be displayed.

NOPROMPT

suppresses all prompts. NOPROMPT overrides the GWAIT= graphics option.

PROMPT=0...7

in the GDEVICE procedure, specifies the level of prompting:

- | | |
|---|--|
| 0 | provides no prompting |
| 1 | issues startup messages only. Startup messages are messages such as PLEASE PRESS RETURN TO CONTINUE. |
| 2 | signals end of graph if device is a video display or sends message to change paper if device is a plotter. |
| 3 | combines the effects of 1 and 2. |
| 4 | sends a message to mount pens if the device is a plotter. |
| 5 | combines the effects of 4 and 1. |
| 6 | combines the effects of 4 and 2. |
| 7 | sends all prom |

Note: If you specify either 0 for the PROMPT device parameter or NOPROMPT in a GOPTIONS statement for a display device, the display clears immediately after the graph is drawn. Δ

In the GDEVICE Parameters window, the PROMPT parameter consists of four fields that describe the type of prompt:

start up

issues a message to turn the device on (if the device is a hardcopy device) or the message PLEASE PRESS RETURN AFTER EACH BELL TO CONTINUE.

end of graph

signals, usually by a bell, when the graph is complete (valid for video displays only).

mount pens

issues a message to mount pens in a certain order and (for certain devices only) to ask for pen priming strokes for plotters.

change paper

prompts the user to change the paper (valid for plotters only).

Enter an X for each prompt that you want to be given. If no Xs appear in these fields, no prompt messages are issued, and the device does not wait for you to respond between graphs.

PROMPTCHARS

Selects the prompt characters to be used by SAS/GRAPH device drivers.

Used in: GOPTIONS statement; GDEVICE procedure; GDEVICE Parameters window

Default: host dependent

See also: GSFLEN, HANDSHAKE

Syntax

PROMPTCHARS='prompt-chars-hex-string'X

prompt-chars-hex-string

is an 8-byte hexadecimal string that is specified as 16 hexadecimal characters. In GDEVICE procedure statements, enclose the string in single quotation marks, followed by an X. In the Parameters window, enter the hexadecimal string without either quotation marks or a trailing X.

Note: Bytes 1, 4, and 5 are the safest for you to change because you are most likely to know the correct value for them. Check with Technical Support before changing any of the other bytes. △

The following list describes each byte in the string:

byte 1

is the ASCII code of the system prompt character (for software handshaking). The system prompt character is the last character that the host sends before waiting for a response from the plotter. For example, 11 means the host sends an XON or DC1 character as a prompt. If the host does not send a special character for a prompt, set this byte to 00.

byte 2

is the ASCII code of the echo-terminator character (for software handshaking). This character is sent at the beginning of each record.

byte 3

prevents splitting commands across records if the value is 01. If you are creating a graphics stream file to send to a device at a later time, and there is the possibility that extra characters will be added between records during transmission, setting the third byte to 01 reduces the likelihood that the extra characters will be interpreted as graphics commands and cause stray lines or other device characters. If the third byte is set to 00, the driver makes the records as long as possible and splits device commands across records if necessary. Setting the third byte to 00 is more efficient but is more likely to result in device errors if output is written to a file and later transmitted to the device.

byte 4

is the line-end character (for software handshaking). It indicates that more data can be sent. This character is almost always a carriage-return character, 0D.

byte 5

specifies turnaround delay in tenths of a second (for software handshaking). The turnaround delay is the amount of time the device waits after receiving the prompt character before sending the line-end character. For example, a value of 05 represents a half-second delay.

byte 6

sets default record length using a hexadecimal value 00–FF. This byte sets the length of the records sent to the device or to a file. If this byte is set to 00 (the default), SAS/GRAPH uses the longest record length possible for the device. To specify an alternate length, set the sixth byte to the hexadecimal value for the

desired length. For example, to generate records of length 80, specify 50 for the sixth byte. If the GSFLLEN device parameter or graphics option is specified, its value overrides the value of the sixth prompt character.

Some values of the GPROTOCOL device parameter cause each byte in the data stream to be expanded to two bytes. This expansion is done after the length of the record is set by PROMPTCHARS. If you are specifying a value for GPROTOCOL that does this (for example, SASGPHX, SASGPLCL, or SASGPAGL), specify a value for the sixth byte of PROMPTCHARS that is half of the actual record length desired. For example, a hexadecimal value of 40 (64 decimal) produces a 128-byte record after expansion by the GPROTOCOL module.

bytes 7 and 8
are unused and should be set to 0000.

Details

PROMPTCHARS is most commonly used to specify parameters used in software handshaking (see “HANDSHAKE” on page 352), but it also can be used to control the length of records written by most drivers. You also can use the GSFLLEN= graphics option for this purpose.

PROWS

Sets the number of rows in the graphics output area for portrait orientation.

Used in: GDEVICE procedure; GDEVICE Detail window

Default: device dependent

See also: LROWS, PCOLS, VPOS

Syntax

PROWS=*portrait-rows*

portrait-rows

is a nonnegative integer up to three digits long (0...999).

Details

Either the LROWS and LCOLS pair of device parameters or the PROWS and PCOLS pair of device parameters are required and must be nonzero.

The VPOS= graphics option overrides the value of PROWS.

See “Procedure Output and the Graphics Output Area” on page 29 for more information.

QMSG

Specifies whether log messages are held until after the graphics output is displayed.

Used in: GDEVICE procedure; GDEVICE Detail window

Default: device dependent

Syntax

GOPTIONS: QMSG | NOQMSG

GDEVICE: QMSG=Y | N

QMSG QMSG=Y

queues driver messages while the device is in graphics mode (default for video devices).

NOQMSG QMSG=N

prevents the queuing of messages (default for plotters, cameras, and printers).

Details

Message queuing is desirable on display devices that do not have a separate dialog and graphics area. If messages are not queued, they are written to the log as the graphics output is being generated. This behavior may cause problems on some devices.

A blank **Queued messages** field in the Parameters window can mean either Y or N, depending on the device.

RECTFILL

Specifies which rectangle fills should be performed by hardware.

Used in: GDEVICE procedure; GDEVICE Parameters window

Default: device dependent

See also: FILL

Syntax

RECTFILL=*'rectangle-fill-hex-string'*X

rectangle-fill-hex-string

is a hexadecimal string that is 16 characters long. In GDEVICE procedure statements, enclose the string in single quotation marks, followed by an X. In the Parameters window, enter the hexadecimal string without either quotation marks or a trailing X.

The following table shows which bit position (left-to-right) within the hexadecimal string controls each fill pattern.

Bit	Fill pattern	Bit	Fill pattern
1	R1	9	L4
2	R2	10	L5
3	R3	11	X1

Bit	Fill pattern	Bit	Fill pattern
4	R4	12	X2
5	R5	13	X3
6	L1	14	X4
7	L2	15	X5
8	L3	16	S

For example, if you want the driver to use only the L1 and R1 fills in hardware, the first and sixth bits of the first byte of the hexadecimal string should be turned on, which corresponds to a value of '8400000000000000'X ('84'X is equivalent to '1 0 0 0 0 1 0 0' in binary). If a particular hardware rectangle fill is not available or not to be used (as indicated by the value of RECTFILL), the fill is generated by the software.

See "PATTERN Statement" on page 211 for an illustration of the fill patterns.

Details

Note: Not all devices support this capability. If FILL=N is specified or the NOFILL option is used in a GOPTIONS statement, RECTFILL is ignored. Δ

RENDER

Controls the creation and disposition of rendered Bitstream fonts.

Used in: GOPTIONS statement

Default: MEMORY

See also: RENDERLIB

Syntax

RENDER=APPEND | DISK | MEMORY | NONE | READ

APPEND

creates files to store rendered versions of Bitstream fonts if the files do not already exist, reads previously rendered characters from the font files, and appends rendered versions of new characters to the font files when the SAS/GRAPH procedure terminates.

DISK

creates files to store rendered versions of Bitstream fonts if the files do not already exist, reads previously rendered characters from the font files, and appends rendered versions of new characters to the font files as they are encountered. This method is slower on some hosts, but it may work in memory-constrained conditions where the other rendering methods fail.

MEMORY

renders all fonts in memory without creating any font files on disk. Font files are not used even if they already exist. New characters are not written to existing font files when SAS/GRAPH procedures terminate.

This is the default and should be the fastest method on hosts that support virtual memory.

NONE

disables the font rendering features.

READ

reads existing rendered font files but does not create new font files or write new characters to existing font files. This is useful only when font files already exist in the rendered font library.

Details

The memory capacity and input/output characteristics of your host system determine which value for the RENDER= option provides the best performance.

RENDERLIB

Specifies the SAS data library in which rendered font files are stored.

Used in: GOPTIONS statement

Default: WORK

See also: RENDER

Syntax

RENDERLIB=*libref*

libref

specifies a previously defined libref that identifies the SAS library. The default library is WORK. See “LIBNAME Statement” on page 25 for more information on assigning a libref.

REPAINT

Specifies how many times to redraw the graph.

Used in: GOPTIONS statement; GDEVICE procedure; GDEVICE Parameters window

Default: device dependent

Syntax

REPAINT=*redraw-factor*

redraw-factor

is a nonnegative integer up to three digits long (0...999).

Details

Use this option with printers that produce light images after only one pass. This option also is useful for producing transparencies; multiple passes make the colors more solid or more intense.

Not all devices have this capability.

RESET

Resets graphics options to their defaults and/or cancels global statements.

Used in: GOPTIONS statement

Syntax

RESET=ALL | GLOBAL | *statement-name* | (*statement-name(s)*)

ALL

sets all graphics options to defaults and cancels all global statements.

GLOBAL

cancels all global statements (AXIS, FOOTNOTE, LEGEND, PATTERN, SYMBOL, and TITLE). Options in the GOPTIONS statement are unaffected.

statement-name

resets or cancels only the specified global statements. For example, RESET=PATTERN cancels all PATTERN statements only. To cancel several statements at one time, enclose the statement names in parentheses. For example, RESET=(TITLE FOOTNOTE AXIS).

Note: RESET=GOPTIONS sets all graphics options to defaults but does not cancel any global statements. Δ

Featured in: “Example 10. Creating a Bar Chart with Drill-down for the Web” on page 294

Details

RESET=ALL or RESET=GOPTIONS must be the first option specified in the GOPTIONS statement; otherwise, the graphics options that precede RESET= in the GOPTIONS statement are reset. Other options can follow the RESET= graphics option in the statement.

REVERSE

Specifies whether to print the output in reverse order, if reverse printing is supported by the device.

Used in: GOPTIONS statement

Default: NOREVERSE

Restriction: hardware dependent, PostScript printers require a PPD file

See also: PPDFILE

Syntax

REVERSE | NOREVERSE

Details

The purpose of REVERSE is to control the stacking order of printer output, depending on how the printer stacks paper. On some printers, reverse implies using the alternate output bin (back of the printer).

For PCL devices, REVERSE sends output to the LOWER out bin, which is the face-up output bin.

For PostScript devices, if the PPD file has an “OutputOrder” entry and one of its entries is “Reverse,” the device supports reverse order printing and the appropriate PostScript code to activate reverse will be used. If the PPD file does not have an “OutputOrder” entry but does have a “PageStackOrder” entry and corresponding OutputBin value, then reverse order printing is supported indirectly, using the PPD file’s PageStackOrder/OutputBin entries.

Note: Some PostScript devices implement Reverse as the default output mode for one of the output bins. In this case, selecting either the “reverse” output bin or specifying REVERSE mode produces identical results. △

ROTATE

Specifies whether and how to rotate the graph.

Used in: GOPTIONS statement; GDEVICE procedure; GDEVICE Detail window

Syntax

GOPTIONS: ROTATE=LANDSCAPE | PORTRAIT ROTATE | NOROTATE

GDEVICE: ROTATE=LANDSCAPE | PORTRAIT

ROTATE | NOROTATE

specifies whether to rotate the graph 90 degrees from its default orientation.

ROTATE=LANDSCAPE

specifies landscape orientation (the graph is wider than it is high).

ROTATE=PORTRAIT

specifies portrait orientation (the graph is higher than it is wide).

If you do not specify a rotation, a default is searched for in this order:

- 1 the ORIENTATION setting on an OPTIONS statement

2 device-dependent default.

ROTATION

Sets the increment of the angle by which the device can rotate any given letter in a string of text in a Metagraphics driver.

Used in: GDEVICE procedure; GDEVICE Metagraphics window

Default: ROTATION=0

Restriction: Used only with user-supplied Metagraphics drivers.

Syntax

ROTATION=*angle-increment*

angle-increment

specifies the increment of the angle at which to rotate individual characters, for example, every 5 degrees, every 45 degrees, and so on. *Angle-increment* is an integer in the range 0 to 360. A value of 0 means that the device uses its default character rotation. Specify 0 if your device does not perform hardware character rotation.

Details

For information about Metagraphics drivers, contact Technical Support.

ROWS

Specifies the number of rows the hardware font uses in graphics output.

Used in: GDEVICE Chartype window; GDEVICE procedure; CHARREC= option

Default: 0

See also: CHARREC

Syntax

See "CHARREC" on page 311 for syntax.

Details

If you are using a device driver from SASHELP.DEVICES, this parameter already is set for hardware fonts that have been defined for your installation. For scalable fonts, you can specify 1 for ROWS, and the actual number of rows will be computed based on the current text width. If you are adding to or modifying hardware fonts available for a particular device driver, specify a positive value for the ROWS device parameter. If ROWS is greater than 0, it overrides the values of the LROWS and PROWS device parameters.

SCALABLE

Specifies whether a font is scalable.

Used in: GDEVICE Chartype window; GDEVICE procedure; CHARREC= option

Default: device dependent

See also: CHARTYPE

Syntax

See “CHARREC” on page 311 for syntax.

Details

A hardware font is scalable if it can be used with any combination of rows and columns. Use the SCALABLE device parameter if you are adding to or modifying the fonts available for a particular device driver. If you are using a device driver from SASHELP.DEVICES, this parameter already is set for hardware fonts that have been defined for your installation.

SIMFONT

Specifies a software font to use if the default hardware font cannot be used.

Used in: GOPTIONS statement

Default: SIMULATE

Syntax

SIMFONT=*software-font*

software-font

specifies a software font to use instead of the default hardware font. By default, this is the SIMULATE font, which is stored in the SASHELP.FONTS catalog.

Details

SAS/GRAPH substitutes the software font specified by the SIMFONT= option for the default hardware font in these cases:

- when you use the NOCHARACTERS option in a GOPTIONS statement
- when you specify a non-default value for the HPOS= or VPOS= graphics option and your device does not have scalable hardware characters
- when you replay a graph using a device driver other than the one used to create the graph
- when you specify an angle or rotation for your hardware text that the device is not capable of producing

- when you specify a hardware font that is not supported by your device.
See Chapter 6, “SAS/GRAPH Fonts,” on page 125 for details.

SPEED

Selects pen speed for plotters with variable speed selection.

Used in: GOPTIONS statement; GDEVICE procedure; GDEVICE Parameters window

Default: device dependent

Syntax

SPEED=*pen-speed*

pen-speed

specifies a percentage (1 through 100) of the maximum pen speed for the device. For example, SPEED=50 slows the drawing speed by half. In general, slowing the drawing speed produces better results.

By default, the value of SPEED is the normal speed for the device.

SWAP

Specifies whether to reverse BLACK and WHITE in the graphics output.

Used in: GOPTIONS statement; GDEVICE procedure; GDEVICE Parameters window

Defaults: GOPTIONS: NOSWAP; GDEVICE: SWAP=N

Syntax

GOPTIONS: SWAP | NOSWAP

GDEVICE: SWAP=Y | N

SWAP

SWAP=Y

swaps BLACK for WHITE and vice versa.

NOSWAP

SWAP=N

does not swap the colors. A blank **swap** field in the Parameters window is the same as SWAP=N.

Details

SWAP does not affect the background color and only affects BLACK and WHITE foreground colors specified as predefined SAS color names. SWAP ignores BLACK and

WHITE specified in HLS, RGB, or gray-scale format. This option is useful when you want to preview a graph on a video device and send the final copy to a printer that uses a white background.

```
goptions reset=all cback=blue ctitle=black swap;
title1 h=8 'swap test';
title2 h=8 'another title';
proc gslide border;
run;
```

SYMBOL

Specifies whether to use the device's hardware symbol-drawing capability.

Used in: GOPTIONS statement; GDEVICE procedure; GDEVICE Parameters window

Default: device dependent

See also: SYMBOLS

Syntax

GOPTIONS: SYMBOL | NOSYMBOL

GDEVICE: SYMBOL=Y | N

SYMBOL

SYMBOL=Y

causes SAS/GRAPH to use the built-in symbol-drawing capability of the device, if available. A blank **Symbol** field in the Parameters window is the same as SYMBOL=Y.

Hardware drawing is faster, but not all devices have the capability. SAS/GRAPH does not try to use the capability if your device does not support it.

NOSYMBOL

SYMBOL=N

causes SAS/GRAPH to draw the symbols using software fonts.

SYMBOLS

Specifies which symbols can be generated by hardware.

Used in: GDEVICE procedure; GDEVICE Parameters window

Default: device dependent

See also: SYMBOL "SYMBOL Statement" on page 226

Syntax

SYMBOLS='hardware-symbols-hex-string'X

hardware-symbols-hex-string

is a hexadecimal string that is 16 characters long and must be completely filled. This table shows which bit position (left-to-right) within the hexadecimal string controls each hardware symbol.

Bit to turn on	Symbol Description	Symbol
1	PLUS	+
2	X	×
3	STAR	*
4	SQUARE	□
5	DIAMOND	◇
6	TRIANGLE	△
7	HASH	#
8	Y	Y
9	Z	Z
10	PAW	⋮
11	POINT	.
12	DOT	●
13	CIRCLE	○

For example, if you want the driver to do only the PLUS and X symbols in hardware, the first and second bits of the first byte of the hexadecimal string should be turned on, which would correspond to a value of 'C000000000000000'X ('C0'X is equivalent to '1 1 0 0 0 0 0 0' in binary).

Details

These are not the only symbols that can be generated for graphics output but are the symbols that can be drawn by the hardware. SAS/GRAPH can draw other symbols.

Note: Not all devices are capable of drawing every symbol. If a particular hardware symbol is not available or not to be used (as indicated by the value of SYMBOLS), the symbol is generated by the software. If the value of the SYMBOL device parameter in the device entry is N or the NOSYMBOL graphics option is used, the value of SYMBOLS is ignored. Δ

TARGETDEVICE

Displays the output as it would appear on a different device. Also, specifies the device driver for the PRINT command and for ODS HTML output.

Alias: TARGET

Used in: GOPTIONS statement

Syntax

TARGETDEVICE=*target-device-entry*

target-device-entry

specifies the name of a device entry in a catalog.

Featured in: “Example 8. Creating a Simple Web Page with the ODS HTML Statement” on page 284 and “Example 9. Combining Graphs and Reports in a Web Page” on page 287

Details

Use TARGETDEVICE= to specify a device driver when you want to:

- preview graphics output on your monitor as it would appear on a different output device. For details, see “Previewing Output” on page 48.
- print output from the Graph window or the Graphics Editor window with the PRINT command. For details, see “Printing Graphics Output” on page 47.
- specify a device driver for graphics output created by the ODS HTML statement.

TRAILER

Specifies the command that creates TRAILER records for the Metagraphics driver.

Used in: GDEVICE procedure; GDEVICE Metagraphics window

Restriction: Used only with user-supplied Metagraphics drivers

See also: TRAILERFILE

Syntax

TRAILER=*command*

command

specifies a command that runs a user-written program that creates the TRAILER file. *Command* is a string up to 40 characters long.

Details

For information about Metagraphics drivers, contact Technical Support.

TRAILERFILE

Specifies the fileref of the file from which the Metagraphics driver reads TRAILER records.

Used in: GDEVICE procedure GDEVICE Metagraphics window

Restriction: Used only with user-supplied Metagraphics drivers

See also: TRAILER

Syntax

TRAILERFILE=*fileref*

fileref

specifies a valid SAS fileref up to eight characters long. *Fileref* must have been previously assigned with a FILENAME statement or a host command before running the Metagraphics driver. See “FILENAME Statement” on page 24 for additional information on the FILENAME statement.

Details

For information about Metagraphics drivers, contact Technical Support.

TRANSPARENCY

Specifies whether the background of the image should appear to be transparent when the image is displayed in the browser.

Used in: GOPTIONS statement

Default: NOTTRANSPARENCY

Restriction: GIF series of drivers only

Syntax

TRANSPARENCY | NOTTRANSPARENCY

Featured in: “Example 8. Creating a Simple Web Page with the ODS HTML Statement” on page 284

Details

When the image is displayed and TRANSPARENCY is in effect, the browser’s background color replaces the driver’s background color, causing the image to appear transparent.

TRANTAB

Selects a translate table for your system that performs ASCII-to-EBCDIC translation.

Used in: GOPTIONS statement; GDEVICE procedure; GDEVICE Host File Options window

Default: host dependent

Syntax

TRANTAB=*table* | *user-defined-table*

table

specifies a translate table stored as a SAS/GRAPH catalog entry. *Table* can be one of the following:

SASGTAB0 (default translate table for your operating environment)

GTABVTAM

GTABTCAM

GTABCMS

user-defined-table

specifies the name of a user-created translate table.

Details

TRANTAB is set by the SAS Installation Representative and is needed when an EBCDIC host sends data to an ASCII graphics device. See the SAS/GRAPH installation instructions for details. You can also create your own translate table using the TRANTAB procedure. For a description of the TRANTAB Procedure, see *SAS Procedures Guide*.

TYPE

Specifies the type of output device to which graphics commands are sent.

Used in: GDEVICE procedure; GDEVICE Detail window

Default: device dependent

Syntax

TYPE=CAMERA | CRT | EXPORT | PLOTTER | PRINTER

CAMERA

specifies a film-recording device.

CRT

specifies a monitor or terminal.

EXPORT

identifies the list in which the device appears under SAS/ASSIST software. This is used for drivers that produce output to be exported to other software applications, such as CGM or HPGL.

PLOTTER

specifies a pen plotter.

PRINTER

specifies a printer

Details

You should not modify this value for Institute-supplied device drivers.

UCC

Sets the user-defined control characters for the device.

Used in: GOPTIONS statement; GDEVICE procedure; GDEVICE Parameters window

Restriction: device dependent

Syntax

`UCC='control-characters-hex-string'X`

control-characters-hex-string

is a hexadecimal string that can be up 32 bytes (64 characters) long. You only need to specify up to the last non-zero byte; the remaining bytes will be set to zero.

Details

Not all devices support this feature, and the meaning of each byte of the string varies from device to device.

UCC values for specific devices are listed in the SAS Help facility for SAS/GRAPH. Typically the UCC byte position is indicated by a bracketed value. For example, UCC[2] refers to the second byte of the string. Refer to the Help for the UCC device parameter for details.

VORIGIN

Sets the vertical offset from the lower-left corner of the display area to the lower-left corner of the graph.

Used in: GOPTIONS statement; GDEVICE procedure; GDEVICE Detail window

See also: HORIGIN

Syntax

VORIGIN=*vertical-offset* <IN | CM | PT>

vertical-offset <IN | CM | PT>

must be a nonnegative number and may be followed by a unit specification, either IN for inches (default), or CM for centimeters, or PT for points. If you do not specify VORIGIN, a default offset is searched for in this order:

- 1 the bottom margin specification on an OPTIONS BOTTOMMARGIN setting
- 2 VORIGIN setting in the device catalog.

Details

The display area is defined by the XMAX and YMAX device parameters. By default, the origin of the graphics output area is the lower-left corner of the display area; the graphics output is offset from the lower-left corner of the display area by the values of HORIGIN and VORIGIN. VORIGIN + VSIZE cannot exceed YMAX. See “About the Graphics Output Area” on page 29 for details.

VPOS

Sets the number of rows in the graphics output area.

Used in: GOPTIONS statement

Default: device dependent: the value of the LROWS or PROWS device parameter

See also: HPOS, LROWS, PROWS

Syntax

VPOS=*rows*

rows

specifies the number of rows in the graphics output area, which is equivalent to the number of hardware characters that can be displayed vertically. Specifying VPOS=0 causes the device driver to use the default hardware character cell height for the device.

Details

The VPOS= graphics option overrides the values of the LROWS or PROWS device parameters and temporarily sets the number of columns in the graphics output area. VPOS= does not affect the height of the graphics output area but merely divides it into rows. Therefore, you can use VPOS= to control cell height.

The values specified in the HPOS= and VPOS= graphics options determine the size of a character cell for the graphics output area and consequently the size of many graphics elements, such as hardware text. The larger the size of the HPOS= and VPOS= values, the smaller the size of each character cell.

See “Procedure Output and the Graphics Output Area” on page 29 for more information.

VSIZE

Sets the vertical size of the graphics output area.

Used in: GOPTIONS statement; GDEVICE procedure; GDEVICE Detail window

see also: HSIZE, YMAX

Syntax

VSIZE=*vertical-size* <IN | CM | PT>

vertical-size <IN | CM | PT>

specifies the height of the graphics output area; *vertical-size* must be a positive number and may be followed by a unit specification, either IN for inches (default), or CM for centimeters, or PT for points. If you do not specify VSIZE=, a default size is searched for in this order:

- 1 the vertical size is calculated as

$$\text{YMAX} - \text{BOTTOMMARGIN} - \text{TOPMARGIN}$$

Note that BOTTOMMARGIN and TOPMARGIN are used in the OPTIONS statement.

- 2 VSIZE setting in the device catalog.

V6COMP

Allows programs developed in Version 6 to run with selected defaults for the current SAS version.

Used in: GOPTIONS statement

Default: NOV6COMP

Syntax

V6COMP | NOV6COMP

V6COMP

causes SAS/GRAPH programs to use these Version 6 behaviors:

- By default, patterns are hatched patterns, not solid, and the default outline color matches the pattern color.
- By default, the GCHART and GPLOT procedures do not draw a frame around the axis area.

NOV6COMP

causes SAS/GRAPH programs to use all the features of the current SAS version.

Details

V6COMP performs the necessary conversions so that you get the same results in the current SAS version that you did in Version 6.

Note: V6COMP does not convert Version 6 catalogs to catalogs with the current SAS catalog format. Δ

XMAX

Specifies the width of the addressable graphics display area; affects the horizontal resolution of the device and the horizontal dimension of the graphics output area.

Used in: GOPTIONS statement; GDEVICE procedure; GDEVICE Detail window

Restriction: Ignored by default display drivers and host printing drivers

See also: HSIZE, PAPERSIZE, XPIXELS

Syntax

XMAX=*width* <IN | CM | PT>

width

is a positive number that may be followed by a unit specification, either IN for inches (default), or CM for centimeters, or PT for points. If you do not specify XMAX, a default width is searched for in this order:

- 1 the *width* specification on an OPTIONS PAPERSIZE setting
- 2 XMAX in the device entry catalog.

If XMAX=0, default behavior is used. If both XMAX and PAPERSIZE have been specified on GOPTIONS, the last request is used.

Details

Like the XPIXELS device parameter, XMAX controls the width of the display area, but the width is in inches, centimeters, or points rather than pixels. Typically, you might use XMAX to change the width of the display area for a hardcopy device.

SAS/GRAPH uses the value of XMAX in calculating the horizontal resolution of the device:

$$x\text{-resolution} = \text{XPIXELS} / \text{XMAX}$$

However, changing XMAX does not necessarily change the resolution:

- If you use the GOPTIONS statement to change only the value of XMAX= and do not change XPIXELS=, SAS/GRAPH retains the default resolution of the device and recalculates XPIXELS, temporarily changing the width.
- If you specify values for both XMAX= and XPIXELS=, SAS/GRAPH recalculates the resolution of the device using both of the specified values. The new resolution may or may not be different. For example, both of these pairs of values produce the same resolution, 300dpi:

XPIXELS=1500 and XMAX=5

XPIXELS=1800 and XMAX=6

XMAX also affects the value of HSIZE, which controls the horizontal dimension of the graphics output area.

- If you change the value of XMAX and do not change HSIZE=, SAS/GRAPH calculates a new value for HSIZE=, using this formula:

$$\text{HSIZE} = \text{XMAX} - \text{margins}$$

Note: The *margins* quantity, here, is not a device parameter. It represents the value of the left margin plus the right margin. The left margin is the value of HORIGIN. The right margin is whatever is left over when you subtract HSIZE and HORIGIN from XMAX. The value of *margins* is always based on the original XMAX and HSIZE values that are stored in the device entry. Δ

- If you specify values for both XMAX= and HSIZE=, SAS/GRAPH uses the specified values plus the value of device parameter HORIGIN. Anything left over is added to the right margin. For example, if XMAX=6IN and HSIZE=4IN and HORIGIN=.5IN, the right margin will be 1.5in. If HSIZE= is larger than XMAX=, HSIZE= is ignored.

To permanently change the value of the XMAX device parameter in the device entry, use the GDEVICE procedure. This can change the resolution.

To temporarily change the size of the display and the resolution of the device for the current graph or for the duration of your SAS session, use XMAX= and XPIXELS= in the GOPTIONS statement.

To reset the value of XMAX to the default, specify XMAX=0. To return to the default resolution for the device, specify both XMAX=0 and XPIXELS=0.

See “Procedure Output and the Graphics Output Area” on page 29 for more information.

XPIXELS

Specifies the width of the addressable display area in pixels and in conjunction with XMAX determines the horizontal resolution for the device.

Used in: GOPTIONS statement; GDEVICE procedure; GDEVICE Detail window

Default: device dependent

See also: XMAX

Restriction: Ignored by default display drivers and host printing drivers.

Syntax

XPIXELS=*width-in-pixels*

width-in-pixels

is a positive integer up to eight digits long (0...99999999).

Details

Like the XMAX device parameter, XPIXELS controls the width of the display area, but the width is in pixels rather than inches, centimeters, or points. Typically, you might use XPIXELS to change the width of the display area for an image format device.

The value of XPIXELS is used in calculating the resolution of the device:

$$x\text{-resolution} = \text{XPIXELS} / \text{XMAX}$$

However, changing XPIXELS does not necessarily change the device resolution:

- If you use the GOPTIONS statement to change only the value of XPIXELS= and do not change XMAX=, SAS/GRAPH retains the default resolution of the device and recalculates XMAX, temporarily changing the width of the display. If HSIZE= is also not specified, SAS/GRAPH uses the new XMAX value to calculate a new HSIZE value, using this formula:

$$\text{HSIZE} = \text{XMAX} - \text{margins}$$

Note: Margins are not device parameters, but represent the value of HORIGIN (the left margin) plus the right margin. The right margin is whatever is left over when you subtract HSIZE and HORIGIN from XMAX. The values of margins is always based on the original XMAX and HSIZE values that are stored in the device entry. Δ

If HSIZE= is specified and its value is larger than XMAX, HSIZE= is ignored.

- If you use the GDEVICE procedure to permanently change the value of the XPIXELS device parameter in the device entry, SAS/GRAPH automatically recalculates the resolution of the device is using the value of XMAX device parameter.
- If you change the values of both XMAX= and XPIXELS=, SAS/GRAPH recalculates the resolution of the device using both of the specified values.

Note: When SAS/GRAPH recalculates the resolution, the resolution does not necessarily change. For example, both of these pairs of values produce the same resolution, 300dpi: Δ

XPIXELS=1500 and XMAX=5

XPIXELS=1800 and XMAX=6

To reset the value of XPIXELS to the default, specify XPIXELS=0. To return to the default resolution for the device, specify both XPIXELS=0 and XMAX=0.

YMAX

Specifies the height of the addressable graphics display area; affects the vertical resolution of the device and the vertical dimension of the graphics output area.

Used in: GOPTIONS statement; GDEVICE procedure; GDEVICE Detail window

Restriction: Ignored by default display drivers and host printing drivers.

See also: PAPERSIZE, VSIZE, YPIXELS

Syntax

YMAX=*height* <IN | CM | PT>

height

is a positive number that may be followed by a unit specification, either IN for inches (default), or CM for centimeters, or PT for points. If you do not specify YMAX, a default height is searched for in this order:

- 1 the *height* specification on an OPTIONS PAPERSIZE setting
- 2 YMAX in the device entry catalog.

If YMAX=0, default behavior is used. If both YMAX and PAPERSIZE have been specified on GOPTIONS, the last request is used.

Details

See “XMAX” on page 399.

YPIXELS

Specifies the height of the addressable display area in pixels and in conjunction with YMAX determines the horizontal resolution for the device.

Used in: GOPTIONS statement; GDEVICE procedure; GDEVICE Detail window

Default: device dependent

See also: YMAX

Restriction: Ignored by default display drivers and host printing drivers

Syntax

YPIXELS=*height-in-pixels*

height-in-pixels

is a positive integer up to eight digits long (0...99999999).

Details

See “XPIXELS” on page 400.

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.