

CHAPTER

32

DATA Step Graphics Interface Dictionary

<i>Overview</i>	1075
<i>Operating States</i>	1076
<i>Utility Functions</i>	1076
<i>GASK Routines</i>	1077
<i>GDRAW Functions</i>	1114
<i>GRAPH Functions</i>	1125
<i>GSET Functions</i>	1129
<i>Return Codes for DSGI Routines and Functions</i>	1165
<i>See Also</i>	1167
<i>References</i>	1167

Overview

This chapter contains detailed descriptions of each command used in the DATA Step Graphics Interface (DSGI).

The commands are discussed in the following order:

- 1 utility functions
 - GINIT
 - GPRINT
 - GTERM
- 2 GASK routines
- 3 GDRAW functions
- 4 GRAPH functions
- 5 GSET functions.

Each routine or function is followed by an alphabetical listing of the operators used with it. For each operator, this chapter provides the statement syntax, other argument definitions, and notes about using the functions and routines, operating states, and return codes. Operating states are summarized in “Operating States” on page 1033.

The syntax for all routines and functions contains the argument *return-code-variable*. This argument must be a numeric variable name and can be a different variable name for each routine.

The *return-code-variable* argument is used to debug DSGI programs. It contains the return code of the routine or function call. If the return code is any value other than 0, the routine or function did not execute properly.

Each routine and function has a different set of possible return codes. The return codes are listed in the heading for the routine or function. Refer to “Return Codes for DSGI Routines and Functions” on page 1165 for an explanation of the return codes.

Operating States

This list summarizes the operating states in DSGI. For a detailed discussion of operating states, see “Operating States” on page 1033.

GKCL	facility closed, initial state of DSGI.
GKOP	facility open. DSGI is open. You may check the settings of attributes.
SGOP	segment open. Graphics output can be generated.
WSAC	workstation active. You can issue DSGI statements.
WSOP	workstation open. The graphics catalog is opened or created.

Utility Functions

Utility functions enable you to initialize a session for DSGI, print error messages, and terminate the session.

GINIT

Initializes DSGI

Operating States: GKCL

Return Codes: 0, 1, 26, 301, 307

Resulting Operating State: WSAC

Syntax

return-code-variable=GINIT();

Description The GINIT function performs three functions: it readies the library that contains SAS/GRAPH graphics routines, it opens a workstation, and it activates it. A workstation is a Graphics Kernel Standard (GKS) concept. GKS allows for multiple workstations to be open at the same time; however, for DSGI applications, you always use exactly one workstation. This function moves the operating state from GKCL to WSAC.

See Also

GTERM utility function

GPRINT

Prints the specified interface error message

Operating States: All

Return Codes: 0

Syntax

return-code-variable=GPRINT(*code*);

Description The GPRINT function displays the message that corresponds to the error code entered. You can use this routine if you have disabled automatic error logging but still want to display the message associated with a return code you have received.

Argument Definitions

code numeric constant or numeric variable name; should be the value of a return code received from some previous function.

See Also

GSET('MESSAGE', . . .)function

GTERM

Terminates DSGI

Operating States: WSAC

Return Codes: 0, 3

Resulting Operating State: GKCL

Syntax

return-code-variable=GTERM();

Description The GTERM function performs three functions: it deactivates the workstation, closes the workstation, and closes the library that contains SAS/GRAPH routines. This function should be issued to free memory allocated by DSGI. This function moves the operating state from WSAC to GKCL.

See Also

GINIT utility function

GASK Routines

GASK routines enable you to check the current attribute settings. When you use GASK routines, remember the following:

- All arguments are required.
- Most arguments are expressed as variable names. You can use any valid SAS variable name.
- If character arguments are expressed as character strings, they must be enclosed in quotation marks.
- All character variable names used as arguments *must* be declared in a previous LENGTH statement.
- GASK routines do not change the operating state.
- PUT statements display a value returned by a routine in the SAS log.
- OUTPUT statements write a value that is returned by a routine to a data set.

ASF

Finds whether an aspect source flag is bundled or separate

Operating States: GKOP, SGOP, WSAC, WSOP

Return Codes: 0, 8

Syntax

CALL GASK('ASF', *attribute*, *status*, *return-code-variable*);

Description The GASK('ASF', . . .) routine returns the aspect source flag (ASF) of a particular attribute. Possible ASF values are BUNDLED (associated with a bundle index) and INDIVIDUAL (separate from a bundle index). GASK('ASF', . . .) returns the default value INDIVIDUAL if you have not set the ASF for an attribute.

Argument Definitions

<i>attribute</i>	character string enclosed in quotes or character variable name with one of the following values: <ul style="list-style-type: none"> <input type="checkbox"/> FILCOLOR <input type="checkbox"/> FILSTYLE <input type="checkbox"/> FILTYPE <input type="checkbox"/> LINCOLOR <input type="checkbox"/> LINTYPE <input type="checkbox"/> LINWIDTH <input type="checkbox"/> MARCOLOR <input type="checkbox"/> MARSIZE <input type="checkbox"/> MARTYPE <input type="checkbox"/> TEXCOLOR <input type="checkbox"/> TEXTFONT.
<i>status</i>	character variable name; returns either the value BUNDLED or INDIVIDUAL.
<i>return-code-variable</i>	numeric variable name; returns the return code of the routine call.

See Also

GSET('ASF', . . .)function
 GSET('FILCOLOR', . . .)function
 GSET('FILSTYLE', . . .)function
 GSET('FILTYPE', . . .)function
 GSET('LINCOLOR', . . .)function
 GSET('LINTYPE', . . .)function
 GSET('LINWIDTH', . . .)function
 GSET('MARCOLOR', . . .)function
 GSET('MARSIZE', . . .)function
 GSET('MARTYPE', . . .)function
 GSET('TEXCOLOR', . . .)function
 GSET('TEXFONT', . . .)function

ASPECT

Finds the aspect ratio

Operating States: All

Return Codes: 0

Syntax

CALL GASK('ASPECT', *aspect*, *return-code-variable*);

Description The GASK('ASPECT', . . .)routine returns the current aspect ratio used to draw graphics output. GASK('ASPECT', . . .)searches for the current aspect ratio in the following order:

- 1 the aspect ratio set with the GSET('ASPECT', . . .)function
- 2 the ASPECT= graphics option
- 3 the device's default aspect ratio found in the device entry. For more information about device entries, see Chapter 15, "The GDEVICE Procedure," on page 651.

Argument Definitions

aspect numeric variable name; returns the aspect ratio.

return-code-variable numeric variable name; returns the return code of the routine call.

See Also

ASPECT= graphics option (see “ASPECT” on page 305)
 GSET(‘ASPECT’, . . .)function

CATALOG

Finds the catalog for the graphs

Operating States: All

Return Codes: 0

Syntax

CALL GASK(‘CATALOG’, *libref*, *memname*, *return-code-variable*);

Description The GASK(‘CATALOG’, . . .)routine returns the libref and the name of the current output catalog. GASK(‘CATALOG’, . . .)returns the default catalog, WORK.GSEG, if no other catalog has been specified with the GSET(‘CATALOG’, . . .)function.

Argument Definitions

<i>libref</i>	character variable name; returns the libref of the library in which the current catalog is stored.
<i>memname</i>	character variable name; returns the name of the current output catalog.
<i>return-code-variable</i>	numeric variable name; returns the return code of the routine call.

See Also

GSET(‘CATALOG’, . . .)function
 GASK(‘NUMGRAPH’, . . .)routine
 GASK(‘OPENGRAPH’, . . .)routine

CBACK

Finds the background color

Operating States: All

Return Codes: 0

Syntax

CALL GASK('CBACK', *cback*, *return-code-variable*);

Description The GASK('CBACK', . . .)routine returns the current background color. GASK('CBACK', . . .)searches for the current background color in the following order:

- 1 the background color selected with the GSET('CBACK', . . .)function
- 2 the CBACK= graphics option
- 3 the default background color for the device found in the device entry. For more information about device entries, see Chapter 15, "The GDEVICE Procedure," on page 651.

Argument Definitions

cback character variable name; returns the background color name.

return-code-variable numeric variable name; returns the return code of the routine call.

See Also

CBACK= graphics option (see "CBACK" on page 309)

GSET('CBACK', . . .)function

CLIP

Finds whether clipping is on or off

Operating States: GKOP, SGOP, WSAC, WSOP

Return Codes: 55, 56

Syntax

CALL GASK('CLIP', *status*);

Description The GASK('CLIP', . . .)routine checks whether clipping outside of viewports is enabled or disabled. One of the two following messages is displayed when this routine is called:

NOTE: Clipping is ON.

or

NOTE: Clipping is OFF.

Clipping is OFF by default.

Argument Definitions

status numeric variable name; returns the current setting, 55 (ON) or 56 (OFF), for clipping.

See Also

GSET('CLIP', . . .)function

COLINDEX

Finds the color indexes that have colors associated with them

Operating States: SGOP

Return Codes: 0, 4, 86, 87

Syntax

CALL GASK('COLINDEX', *n*, *index-array*, *return-code-variable*);

Description The GASK('COLINDEX', . . .)routine returns the color indexes that currently have colors assigned to them.

Argument Definitions

<i>n</i>	numeric constant or numeric variable name; tells how many color indexes you want returned. If <i>n</i> is expressed as a variable, the variable must be initialized. The variable returns the number of colors currently assigned. If <i>n</i> is expressed as a constant, it will not return this value.
<i>index-array</i>	list of numeric variables into which the used color index numbers are returned. The list of variable names can be members of an array or OF argument lists (where the arguments are variables). If you are using an array, <i>index-array</i> must have been declared as an array. The dimension of the array is determined by the number of color indexes you want returned. Refer to the discussion of ARRAY in <i>SAS Language Reference: Dictionary</i> for more information about OF argument lists.
<i>return-code-variable</i>	numeric variable name; returns the return code of the routine call.

See Also

GASK('COLINDEX', . . .)routine

GASK('COLREP', . . .)routine

GSET('COLREP', . . .)function

COLREP

Finds the color name associated with a color index

Operating States: SGOP

Return Codes: 0, 4, 86, 87

Syntax

CALL GASK('COLREP', *color-index*, *color*, *return-code-variable*);

Description The GASK('COLREP', . . .)routine returns the predefined SAS color name associated with a color index. GASK('COLREP', . . .)searches for the current color assigned to a color index in the following order:

- 1 the color selected by the GSET('COLREP', . . .)function.
- 2 the COLORS= graphics option. If *color-index* is 2, the routine returns the second color from the colors list of the COLORS= graphics option.
- 3 the device's default colors list found in the device entry. If *color-index* is 2, the routine returns the second color from the default colors list.

See "Predefined SAS Colors" on page 145 for a list of SAS predefined color names.

Argument Definitions

<i>color-index</i>	numeric constant; indicates the color index for which you want to check the color. Valid values are 1 to 256, inclusive.
<i>color</i>	character variable name; returns the color name associated with <i>color-index</i> .
<i>return-code-variable</i>	numeric variable name; returns the return code of the routine call.

See Also

GASK('COLINDEX', . . .)routine
 GSET('COLREP', . . .)function

DEVICE

Finds the output graphics device

Operating States: All

Return Codes: 0

Syntax

CALL GASK('DEVICE', *device*, *return-code-variable*);

Description The GASK('DEVICE', . . .)routine returns the current device driver. This routine returns the device driver set by one of the following methods:

- the GSET('DEVICE', . . .)function
- the DEVICE= graphics option

- the device driver you entered in the DEVICE prompt window
- the device driver you entered in the OPTIONS window.

There is no default value for a device driver. To use DSGI, you must specify a device driver. For more information about setting device drivers, see “Selecting a Device Driver” on page 39.

Argument Definitions

<i>device</i>	character variable name; returns the name of the device driver.
<i>return-code-variable</i>	numeric variable name; returns the return code of the routine call.

See Also

DEVICE= graphics option (see “DEVICE” on page 321)
 GSET('DEVICE', . . .)function

FILCOLOR

Finds the color index of the color to be used to draw fill areas

Operating States: GKOP, SGOP, WSAC, WSOP

Return Codes: 0, 8

Syntax

CALL GASK('FILCOLOR', *color-index*, *return-code-variable*);

Description The GASK('FILCOLOR', . . .)routine returns the current fill color. If a GSET('FILCOLOR', . . .)function has not been previously submitted, GASK('FILCOLOR', . . .)returns the default value, 1. The color index returned corresponds to a color specification in the following order:

- 1 the color assigned to a color name with the GSET('COLREP', . . .)function
- 2 the *n*th color in the colors list of the COLORS= graphics option
- 3 the *n*th color in the device's default colors list found in the device entry.

Argument Definitions

<i>color-index</i>	numeric variable name; returns the color index of the fill color currently selected.
<i>return-code-variable</i>	numeric variable name; returns the return code of the routine call.

See Also

COLORS= graphics option (see “COLORS” on page 314)
 GASK('COLREP', . . .)
 GSET('COLREP', . . .)
 GSET('FILCOLOR', . . .)

FILINDEX

Finds the bundle of fill area attributes that is active

Operating States: GKOP, SGOP, WSAC, WSOP

Return Codes: 0, 8

Syntax

CALL GASK('FILINDEX', *index*, *return-code-variable*);

Description The GASK('FILINDEX', . . .)routine asks which fill bundle is active. If no fill bundles have been previously defined with GSET('FILREP', . . .)or activated with GSET('FILINDEX', . . .), GASK('FILINDEX', . . .)returns the default value, 1.

Argument Definitions

<i>index</i>	numeric variable name; returns the index of the fill bundle currently selected.
<i>return-code-variable</i>	numeric variable name; returns the return code of the routine call.

See Also

GASK('FILREP', . . .)
 GSET('FILREP', . . .)
 GSET('FILINDEX', . . .)

FILREP

Finds the fill area attributes associated with a bundle index

Operating States: GKOP, WSOP, WSAC, SGOP

Return Codes: 0, 8, 75, 76

Syntax

CALL GASK ('FILREP', *index*, *color-index*, *interior*, *style-index*, *return-code-variable*);

Description The GASK('FILREP', . . .)routine returns the color, type of interior, and fill pattern associated with a specific fill bundle. If the bundle indicated by *index* has not been previously defined with the GSET('FILREP', . . .)function, DSGI issues the following error message:

ERROR: A representation for the specified fill area index has not been defined on this workstation.

Argument Definitions

<i>index</i>	numeric constant or numeric variable name; indicates the fill bundle to check. Valid values are 1 to 20, inclusive. If <i>index</i> is expressed as a variable, the variable must be initialized to a value between 1 and 20.
<i>color-index</i>	numeric variable name; returns the color index of the fill color associated with the bundle. The color index that is returned corresponds to a color specification in the following order: <ol style="list-style-type: none"> 1 a color index assigned to a color name with the GSET('COLREP', . . .)function 2 the <i>n</i>th color in the colors list of the COLORS= graphics option 3 the <i>n</i>th color in the device's default colors list found in the device entry.
<i>interior</i>	character variable name; returns the style of the interior associated with the bundle index – that is, one of the following values: <ul style="list-style-type: none"> □ HATCH □ HOLLOW □ PATTERN □ SOLID.
<i>style-index</i>	numeric variable name; returns the index of the fill pattern associated with the bundle. See the GSET('FILSTYLE', . . .)function on page 1138 for the fill patterns represented by <i>style-index</i> .
<i>return-code-variable</i>	numeric variable name; returns the return code of the routine call.

See Also

COLORS= graphics option (see "COLORS" on page 314)
 GASK('FILINDEX', . . .)routine
 GSET('COLREP', . . .)function
 GSET('FILREP', . . .)function
 GSET('FILSTYLE', . . .)function

FILSTYLE

Finds the style of the fill area when FILTYPE is PATTERN or HATCH

Operating States: GKOP, SGOP, WSAC, WSOP

Return Codes: 0, 8

Syntax

CALL GASK('FILSTYLE', *style-index*, *return-code-variable*);

Description The GASK('FILSTYLE', . . .)routine returns the current fill style of the interior when FILTYPE is PATTERN or HATCH. If no fill style has been previously selected with the GSET('FILSTYLE', . . .)function, GASK('FILSTYLE', . . .)returns the default value, 1.

Argument Definitions

style-index numeric variable name; returns the index of the fill pattern associated with the bundle. See the GSET('FILSTYLE', . . .)function on page 1138 for the interior styles represented by *style-index*.

return-code-variable numeric variable name; returns the return code of the routine call.

See Also

GASK('FILSTYLE', . . .)routine
 GSET('FILSTYLE', . . .)function
 GSET('FILTYPE', . . .)function

FILTYPE

Finds the type of the interior of the fill area

Operating States: GKOP, SGOP, WSAC, WSOP

Return Codes: 0, 8

Syntax

CALL GASK('FILTYPE', *interior*, *return-code-variable*);

Description The GASK('FILTYPE', . . .)routine returns the current fill type. If no fill type has been previously selected with the GSET('FILTYPE', . . .)function, GASK('FILTYPE', . . .)returns the default value, HOLLOW.

Argument Definitions

interior character variable name; returns the fill type that is active, that is, one of the following values:

- HATCH
- HOLLOW
- PATTERN
- SOLID.

return-code-variable numeric variable name; returns the return code of the routine call.

See Also

GASK('FILSTYLE', . . .)routine

GSET('FILTYPE', . . .)function

GRAPHLIST

Finds the names of segments in the current catalog

Operating States: GKOP, SGOP, WSAC, WSOP

Return Codes: 0, 8

Syntax

CALL GASK('GRAPHLIST', *n*, *name-array*, *return-code-variable*);

Description The GASK('GRAPHLIST', . . .)routine lists the first *n* names of the graphs that are in the current catalog. If a catalog has not been previously specified with the GRAPH('CATALOG', . . .)function, the routine returns names from the default catalog, WORK.GSEG.

The names returned are any of the following:

- those specified in the GRAPH('CLEAR', . . .)function
- if the name is omitted from the GRAPH('CLEAR' . . .)function, some form of DSGI: for example, DSGI, DSGI1, or DSGI2.
- the name specified in the NAME= option of a graphics procedure
- graphs previously created by other graphics procedures and already in the catalog.

Argument Definitions

n numeric variable name; tells the maximum number of graph names you want returned. If you express *n* as a variable, the variable must be initialized to the maximum number of graph names you want returned.

name-array list of character variable names into which the graph names will be returned. The list of variable names can be members of an array or OF argument lists (where the arguments are variables). If you are using an array, *name-array* must be declared as an array. The dimension of the array is determined by the number of color indexes you want returned. See the discussion for ARRAY in *SAS Language Reference: Dictionary* for more information about OF argument lists.

return-code-variable numeric variable names; returns the return code of the routine call.

See Also

GRAPH ('CLEAR', . . .)function

HPOS

Finds the number of columns

Operating States: All

Return Codes: 0

Syntax

CALL GASK('HPOS',*hpos*, *return-code-variable*);

Description The GASK('HPOS', . . .)routine returns the number of columns currently in the graphics output area. GASK('HPOS', . . .)searches for the current number of columns in the following order:

- 1 the value selected in the GSET('HPOS', . . .)function
- 2 the value of the HPOS= graphics option
- 3 the device's default HPOS value found in the device entry.

Argument Definitions

hpos numeric variable name; returns the number of columns in the graphics output area.

return-code-variable numeric variable name; returns the return code of the routine call.

See Also

GASK('HSIZE', . . .)function

GSET('HPOS', . . .)routine

HPOS= graphics option (see "HPOS" on page 355)

HSIZE

Finds the horizontal dimension of the graphics output area

Operating States: All

Return Codes: 0

Syntax

CALL GASK('HSIZE', *hsize*, *return-code-variable*);

Description The GASK('HSIZE', . . .)routine returns the current horizontal dimension, in inches, of the graphics output area. GASK('HSIZE', . . .)searches for the current horizontal dimension in the following order:

- 1 the value selected in the GSET('HSIZE', . . .)function
- 2 the value of the HSIZE= graphics option
- 3 the device's default HSIZE found in the device entry.

Argument Definitions

hsize numeric variable name; the size of the graphics output area in the *x* dimension (in inches).

return-code-variable numeric variable name; returns the return code of the routine call.

See Also

GASK('HPOS', . . .)routine

GSET('HSIZE', . . .)function

HSIZE= graphics option (see "HSIZE" on page 356)

HTML

Finds the HTML string that is in effect when one of the following graphic elements is drawn: bar, ellipse, fill, mark, pie, and text.

Operating States: GKOP, SGOP, WSAC, WSOP

Return Codes: 0, 8

Syntax

CALL GASK('HTML', *string*, *return-code-variable*);

Description The GASK('HTML', . . .)routine returns the current HTML string. If a GSET('HTML', . . .)function has not been previously submitted, GASK('HTML', . . .)returns the default value, null.

Argument Definitions

string the HTML string invoked when an affected DSGI graphic element in a web page is clicked.

return-code-variable numeric variable name; returns the return code of the routine call.

See Also

GDRAW('BAR', . . .)function
 GDRAW('ELLIPSE', . . .)function
 GDRAW('FILL', . . .)function
 GDRAW('MARK', . . .)function
 GDRAW('PIE', . . .)function
 GDRAW('TEXT', . . .)function
 GSET('HTML', . . .)function

LINCOLOR

Finds the current setting of the color to be used to draw lines

Operating States: GKOP, SGOP, WSAC, WSOP

Return Codes: 0, 8

Syntax

CALL GASK('LINCOLOR', *color-index*, *return-code-variable*);

Description The GASK('LINCOLOR', . . .)routine returns the current line color. If a GSET('LINCOLOR', . . .)function has not been previously submitted, GASK('LINCOLOR', . . .)returns the default value, 1. The color index returned corresponds to a color specification in the following order:

- 1 the color specified in a GSET('COLREP', . . .)function
- 2 the *n*th color in the colors list of the COLORS= graphics option
- 3 the *n*th color in the device's default colors list.

Argument Definitions

color-index numeric variable name; returns the color index of the current line color.

return-code-variable numeric variable name; returns the return code of the routine call.

See Also

COLORS= graphics option (see “COLORS” on page 314)
 GASK('COLREP', . . .)routine
 GSET('COLREP', . . .)function
 GSET('LINCOLOR', . . .)function

LININDEX

Finds the index of the bundle of line attributes

Operating States: GKOP, SGOP, WSAC, WSOP

Return Codes: 0, 8

Syntax

CALL GASK('LININDEX', *index*, *return-code-variable*);

Description The GASK('LININDEX', . . .)routine returns the current line bundle. If no line bundles have been previously defined with GSET('LINREP', . . .)or activated with GSET('LININDEX', . . .), GASK('LININDEX', . . .)returns the default value, 1.

Argument Definitions

index numeric variable name; returns the index of the current line bundle.

return-code-variable numeric variable name; returns the return code of the routine call.

See Also

GASK('LINREP', . . .)routine
 GSET('LININDEX', . . .)function
 GSET('LINREP', . . .)function

LINREP

Finds the bundle of line attributes associated with an index

Operating States: GKOP, SGOP, WSAC, WSOP

Return Codes: 0, 8, 60, 61

Syntax

CALL GASK ('LINREP', *index*, *color-index*, *width*, *type*, *return-code-variable*);

Description The GASK('LINREP', . . .)routine returns the color, width, and line type associated with a specific line bundle. If the bundle indicated by *index* has not been previously defined with the GSET('LINREP', . . .)function, DSGI issues the following error message:

```
ERROR: A representation for the specified line type index has
       not been defined on this workstation.
```

Argument Definitions

<i>index</i>	numeric constant or numeric variable name; indicates the fill bundle to check. Valid values are 1 to 20, inclusive. If <i>index</i> is expressed as a variable, the variable must be initialized to a value between 1 and 20.
<i>color-index</i>	numeric variable name; returns the color index of the fill color associated with the bundle. The color index returned corresponds to a color specification in the following order: <ol style="list-style-type: none"> 1 a color index assigned with the GSET('COLREP', . . .)function 2 the <i>n</i>th color in the colors list of the COLORS= graphics option 3 the <i>n</i>th color in the device's default colors list.
<i>width</i>	numeric variable name; returns the line width (in pixels) associated with the bundle.
<i>type</i>	numeric variable name; returns the index of the line type associated with the bundle. Refer to Figure 8.22 on page 249 for representations of the line types.
<i>return-code-variable</i>	numeric variable name; returns the return code of the routine call.

See Also

COLORS= graphics option (see "COLORS" on page 314)
GASK('COLREP', . . .)routine
GASK('LININDEX', . . .)routine
GSET('COLREP', . . .)function
GSET('LINREP', . . .)function

LINTYPE

Finds the line type

Operating States: GKOP, SGOP, WSAC, WSOP

Return Codes: 0, 8

Syntax

CALL GASK('LINTYPE', *type*, *return-code-variable*);

Description The GASK('LINTYPE', . . .)routine returns the current line type. If no line type was previously selected with the GSET('LINTYPE', . . .)function, GASK('LINTYPE', . . .)returns the default value, 1.

Argument Definitions

type numeric variable name; returns the index of the line type currently selected. Refer to Figure 8.22 on page 249 for representations of the line types.

return-code-variable numeric variable name; returns the return code of the routine call.

See Also

GSET('LINTYPE', . . .)function

LINWIDTH

Finds the line thickness

Operating States: GKOP, SGOP, WSAC, WSOP

Return Codes: 0, 8

Syntax

CALL GASK('LINWIDTH', *width*, *return-code-variable*);

Description The GASK('LINWIDTH', . . .)routine returns the current line width. If a line width has not been previously selected with the GSET('LINWIDTH', . . .)function, GASK('LINWIDTH', . . .)returns the default value, 1.

Argument Definitions

width numeric variable name; returns the current line width (in units of pixels).

return-code-variable numeric variable name; returns the return code of the routine call.

See Also

GSET('LINWIDTH', . . .)function

MARCOLOR

Finds the color index of the color to be used to draw markers

Operating States: GKOP, SGOP, WSAC, WSOP

Return Codes: 0, 8

Syntax

CALL GASK('MARCOLOR', *color-index*, *return-code-variable*);

Description The GASK('MARCOLOR', . . .)routine returns the current marker color. If a GSET('MARCOLOR', . . .)function has not been previously submitted, GASK('MARCOLOR', . . .)returns the default value, 1. The color index returned corresponds to a color specification in the following order:

- 1 the color selected in a GSET('COLREP', . . .)function
- 2 the *n*th color in the colors list of the COLORS= graphics option
- 3 the *n*th color in the device's default colors list.

Argument Definitions

color-index numeric variable name; returns the color index of the current marker color.

return-code-variable numeric variable name; returns the return code of the routine call.

See Also

COLORS= graphics option (see "COLORS" on page 314)

GASK('COLREP', . . .)routine

GSET('COLREP', . . .)function

GSET('MARCOLOR', . . .)function

MARINDEX

Finds the index of the bundle of marker attributes currently selected

Operating States: GKOP, SGOP, WSAC, WSOP

Return Codes: 0, 8

Syntax

CALL GASK('MARINDEX', *index*, *return-code-variable*);

Description The GASK('MARINDEX', . . .)routine returns the current marker bundle. If no marker bundles have been previously defined with GSET('MARREP', . . .)or activated with GSET('MARINDEX', . . .), GASK('MARINDEX', . . .)returns the default value, 1.

Argument Definitions

index numeric variable name; returns the index of the marker bundle currently selected.

return-code-variable numeric variable name; returns the return code of the routine call.

See Also

GASK('MARREP', . . .)routine
 GSET('MARINDEX', . . .)function
 GSET('MARREP', . . .)function

MARREP

Finds the bundle of marker attributes associated with an index

Operating States: GKOP, SGOP, WSAC, WSOP

Return Codes: 0, 8, 64, 65

Syntax

CALL GASK('MARREP', *index*, *color-index*, *size*, *type*, *return-code-variable*);

Description The GASK('MARREP' . . .)routine returns the color, size, and type of marker associated with a specific marker bundle. If the bundle indicated by *index* has not been previously defined with the GSET('MARREP', . . .)function, DSGI issues the following error message:

ERROR: A representation for the specified marker index has
 not been defined on this workstation.

Argument Definitions

index numeric constant or numeric variable name; indicates the index of the fill bundle to check. Valid values are 1 to 20, inclusive. If *index* is expressed as a variable, the variable must be initialized to a value between 1 and 20.

<i>color-index</i>	numeric variable name; returns the color index of the fill color associated with the bundle. The color index returned corresponds to a color specification in the following order: <ol style="list-style-type: none"> 1 a color index assigned with the GSET('COLREP', . . .)function 2 the <i>n</i>th color in the colors list of the COLORS= graphics option 3 the <i>n</i>th color in the device's default colors list.
<i>size</i>	numeric variable name; returns the marker size in units of the current window system.
<i>type</i>	numeric variable name; the index of the marker type associated with the bundle. See the GSET('MARTYPE', . . .)function on page 1149 for an explanation of the marker indexes.
<i>return-code-variable</i>	numeric variable name; returns the return code of the routine call.

See Also

COLORS= graphics option (see "COLORS" on page 314)

GASK('COLREP', . . .)routine

GSET('COLREP', . . .)function

GSET('MARINDEX', . . .)function

GSET('MARREP', . . .)function

GSET('MARTYPE', . . .)function

MARSIZE

Finds the size of markers

Operating States: GKOP, SGOP, WSAC, WSOP

Return Codes: 0, 8

Syntax

CALL GASK('MARSIZE', *size*, *return-code-variable*);

Description The GASK('MARSIZE', . . .)routine returns the current marker size. If no marker size has been previously selected with the GSET('MARSIZE', . . .)function, GASK('MARSIZE', . . .)returns the default value, 1.

Argument Definitions

<i>size</i>	numeric variable name; returns the marker size in units of the current window system.
<i>return-code-variable</i>	numeric variable name; returns the return code of the routine call.

See Also

GSET('MARSIZE', . . .)function

MARTYPE

Finds the kind of markers

Operating States: GKOP, SGOP, WSAC, WSOP

Return Codes: 0, 8

Syntax

CALL GASK('MARTYPE', *type*, *return-code-variable*);

Description The GASK('MARTYPE', . . .)routine returns the current marker type. If no marker type has been previously selected with the GSET('MARTYPE', . . .)function, GASK('MARTYPE', . . .)returns the default value, 1.

Argument Definitions

<i>type</i>	numeric variable name; returns the index of the marker type currently selected. See the function GSET('MARTYPE', . . .) on page 1149 for an explanation of the indexes for markers.
<i>return-code-variable</i>	numeric variable name; returns the return code of the routine call.

See Also

GSET('MARTYPE', . . .)function

MAXDISP

Finds the maximum display area size

Operating States: GKOP, SGOP, WSAC, WSOP

Return Codes: 0, 8

Syntax

CALL GASK ('MAXDISP', *units*, *x-dim*, *y-dim*, *x-pixels*, *y-pixels*, *return-code-variable*);

Description The GASK('MAXDISP', . . .)routine returns the dimensions of the maximum display area for the device. This routine is useful when you need to know the maximum display area in order to determine the aspect ratio or to scale a graph.

There is a difference between the maximum display size returned when the operating state is not SGOP and when it is SGOP. The full addressable display area is returned when the operating state is not SGOP, and the display area minus room for titles and footnotes is returned when the operating state is SGOP.

Argument Definitions

<i>units</i>	numeric variable name; returns a 1 to show that <i>x-dim</i> and <i>y-dim</i> are in meters.
<i>x-dim</i>	numeric variable name; returns the dimension, in meters, in the <i>x</i> direction.
<i>y-dim</i>	numeric variable name; returns the dimension, in meters, in the <i>y</i> direction.
<i>x-pixels</i>	numeric variable name; returns the number of pixels in the <i>x</i> direction.
<i>y-pixels</i>	numeric variable name; returns the number of pixels in the <i>y</i> direction.
<i>return-code-variable</i>	numeric variable name; returns the return code of the routine call.

See Also

GASK('HSIZE', . . .)routine
 GASK('VSIZE', . . .)routine
 GSET('HSIZE', . . .)function
 GSET('VSIZE', . . .)function

NUMGRAPH

Finds the number of graphs in the current catalog

Operating States: GKOP, SGOP, WSAC, WSOP

Return Codes: 0, 8

Syntax

CALL GASK('NUMGRAPH', *n*, *return-code-variable*);

Description The GASK('NUMGRAPH', . . .)routine returns how many graphs are in the current catalog. The catalog checked is the catalog selected in the GSET('CATALOG', . . .)function, if specified; otherwise, it is the default catalog, WORK.GSEG.

Argument Definitions

<i>n</i>	numeric variable name; returns the number of graphs in the current catalog.
<i>return-code-variable</i>	numeric variable name; returns the return code of the routine call.

See Also

GASK('CATALOG', . . .)routine
 GSET('CATALOG', . . .)function

OPENGRAPH

Finds the name of the segment currently open

Operating States: SGOP

Return Codes: 0, 4

Syntax

CALL GASK('OPENGRAPH', *name*, *return-code-variable*);

Description The GASK('OPENGRAPH', . . .)routine returns the name of the graph that is currently open.

The name returned is one of the following:

- the name specified in the GRAPH('CLEAR', . . .)function
- if the name is omitted from the GRAPH('CLEAR', . . .)function, some form of DSGI: for example, DSGI, DSGI1, and DSGI2.

Argument Definitions

<i>name</i>	character variable name; returns the name of the graph that is currently open.
<i>return-code-variable</i>	numeric variable name; returns the return code of the routine call.

See Also

GRAPH('CLEAR', . . .)function

STATE

Finds the current operating state of DSGI

Operating States: All

Return Codes: 0

Syntax

CALL GASK('STATE', *status*);

Description The GASK('STATE', . . .)routine returns the current operating state of DSGI.

Argument Definitions

status character variable name; returns one of the following values:

- GKCL
- GKOP
- SGOP
- WSAC
- WSOP.

See Also

GASK('WSACTIVE', . . .)routine

GASK('WSOPEN', . . .)routine

TEXALIGN

Finds the horizontal and vertical alignment of the text string

Operating States: GKOP, SGOP, WSAC, WSOP

Return Codes: 0, 8

Syntax

CALL GASK('TEXALIGN', *halign*, *valign*, *return-code-variable*);

Description The GASK('TEXALIGN', . . .)routine returns the current horizontal and vertical text alignment. If no values have been previously selected with the GSET('TEXALIGN', . . .)function, GASK('TEXALIGN', . . .)returns the default value NORMAL for both *halign* and *valign*.

Argument Definitions

halign character variable name; indicates the horizontal alignment set by the GSET('TEXALIGN', . . .)function; returns one of the following values:

- CENTER

- LEFT
- NORMAL
- RIGHT.

valign character variable name; indicates the vertical alignment set by the GSET("TEXALIGN', . . .)function; returns one of the following values:

- BASE
- BOTTOM
- HALF
- NORMAL
- TOP.

return-code-variable numeric variable name; returns the return code of the routine call.

See Also

GASK("TEXPATH', . . .)routine
 GASK("TEXUP', . . .)routine
 GSET("TEXALIGN', . . .)function

TEXCOLOR

Finds the color index of the color currently selected to draw text strings

Operating States: GKOP, SGOP, WSAC, WSOP

Return Codes: 0, 8

Syntax

CALL GASK("TEXCOLOR', *color-index*, *return-code-variable*);

Description The GASK("TEXCOLOR', . . .)routine returns the current text color. If a GSET("TEXCOLOR', . . .)function has not been previously submitted, GASK("TEXCOLOR', . . .)returns the default value, 1. The color index returned corresponds to a color specification in the following order:

- 1 the color specified in a GSET('COLREP', . . .)function
- 2 the *n*th color in the colors list of the COLORS= graphics option
- 3 the *n*th color in the device's default colors list.

Argument Definitions

color-index numeric variable name; returns the color index of the color used to draw text.

return-code-variable numeric variable name; returns the return code of the routine call.

See Also

COLORS= graphics option (see "COLORS" on page 314)
 GASK('COLREP', . . .)routine
 GSET('COLREP', . . .)function
 GSET('TEXCOLOR', . . .)function

TEXEXTENT

Finds the text extent rectangle and concatenation point for a specified text string

Operating States: SGOP, WSAC, WSOP

Return Codes: 0, 8

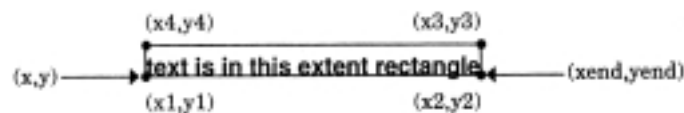
Syntax

CALL GASK ("TEXEXTENT", *x*, *y*, *string*, *x-end*, *y-end*, *x1*, *x2*, *x3*, *x4*, *y1*, *y2*, *y3*, *y4*, *return-code-variable*);

Description The GASK("TEXEXTENT", . . .)routine returns the text extent rectangle and text concatenation point for a specified text string. All text extent coordinates returned are in units of the current window system. If no text string is specified for *string*, GASK("TEXEXTENT", . . .)does not return values for the other arguments.

The text attributes and bundles affect the values returned by this query. See Figure 32.1 on page 1103 for a diagram of the text extent rectangle (in the figure, *x,y* is always the place where the text string starts).

Figure 32.1 Text Extent Diagram



Argument Definitions

- x* numeric variable name; *x* coordinates are in units based on the current window system; returns *x* coordinate after justification. The variable used to specify *x* must be initialized.
- y* numeric variable name; *y* coordinates are in units based on the current window system; returns *y* coordinate after justification. The variable used to specify *y* must be initialized.
- string* character string enclosed in single quotation marks or a character variable name; a set of characters for which the text extent rectangle and text concatenation point are calculated.

<i>x-end</i>	numeric variable name; returns the <i>x</i> coordinate of the point at which the next text string may be concatenated.
<i>y-end</i>	numeric variable name; returns the <i>y</i> coordinate of the point at which the next text string may be concatenated.
<i>x1, x2, x3, x4,</i> <i>y1, y2, y3, y4</i>	numeric variable names; return the text extent rectangles of the text strings as shown in Figure 32.1 on page 1103.
<i>return-code-</i> <i>variable</i>	numeric variable name; returns the return code of the routine call.

See Also

GASK('WINDOW', . . .)routine

GDRAW('TEXT', . . .)function

TEXFONT

Finds the font used to draw text strings

Operating States: GKOP, SGOP, WSAC, WSOP

Return Codes: 0, 8

Syntax

CALL GASK('TEXFONT', *font*, *return-code-variable*);

Description The GASK('TEXFONT' . . .)routine returns the current text font. GASK('TEXFONT', . . .)searches for the current font in the following order:

- 1 the value selected in the GSET('TEXFONT', . . .)function, if specified
- 2 the value of the FTEXT= graphics option, if specified
- 3 the device's default hardware font if the device supports a hardware font
- 4 the SIMULATE font.

Argument Definitions

font character variable name; returns the font name.

return-code-
variable numeric variable name; returns the return code of the routine call.

See Also

FTEXT= graphics options in (see “FTEXT” on page 335)
 GSET(“TEXTFONT”, . . .)function

TEXHEIGHT

Finds the character height of the text strings

Operating States: GKOP, SGOP, WSAC, WSOP

Return Codes: 0, 8

Syntax

CALL GASK(“TEXHEIGHT”, *height*, *return-code-variable*);

Description The GASK(“TEXHEIGHT”, . . .)routine returns the current text height. GASK(“TEXHEIGHT”, . . .)searches for the current text height in the following order:

- 1 the value selected in the GSET(“TEXHEIGHT”, . . .)function, if specified
- 2 the value of the HTEXT= graphics option, if specified
- 3 the default text height, 1.

Argument Definitions

height numeric variable name; returns the character height in units of the current window system.

return-code-variable numeric variable name; returns the return code of the routine call.

See Also

GSET(“TEXHEIGHT”, . . .)function
 HTEXT= graphics options (see “HTEXT” on page 357)

TEXINDEX

Finds the index of the bundle of text attributes currently selected

Operating States: GKOP, SGOP, WSAC, WSOP

Return Codes: 0, 8

Syntax

CALL GASK('TEXINDEX', *index*, *return-code-variable*);

Description The GASK('TEXINDEX', . . .)routine returns the current text bundle. If no text bundles have been previously defined with GSET('TEXREP', . . .)or activated with GSET('TEXINDEX', . . .), GASK('TEXINDEX', . . .)returns the default value, 1.

Argument Definitions

index numeric variable name; returns the text bundle index.

return-code-variable numeric variable name; returns the return code of the routine call.

See Also

GASK('TEXREP', . . .)routine
 GSET('TEXREP', . . .)function
 GSET('TEXINDEX', . . .)function

TEXPATH

Finds the direction of the text string

Operating States: GKOP, SGOP, WSAC, WSOP

Return Codes: 0, 8

Syntax

CALL GASK('TEXPATH', *path*, *return-code-variable*);

Description The GASK('TEXPATH', . . .)routine returns the current text path (reading direction). If TEXPATH has not been previously selected with the GSET('TEXPATH', . . .)function, GASK('TEXPATH', . . .)returns the default value, RIGHT. See the GSET('TEXPATH', . . .)function on page 1156 for an illustration of text paths.

Argument Definitions

path character variable name; returns one of the following values:

- DOWN
- LEFT
- RIGHT
- UP.

return-code-variable numeric variable name; returns the return code of the routine call.

See Also

GASK('TEXALIGN', . . .)routine

GASK('TEXUP', . . .)routine

GSET('TEXPATH', . . .)function

TEXREP

Finds the attribute settings associated with a text bundle

Operating States: GKOP, SGOP, WSAC, WSOP

Return Codes: 0, 8, 68, 69

Syntax

CALL GASK('TEXREP', *index*, *color-index*, *font*, *return-code-variable*);

Description The GASK('TEXREP', . . .)routine returns the color and font associated with a specific text bundle. If the bundle indicated by *index* has not been previously defined with the GSET('TEXREP', . . .)function, DSGI issues the following error message:

```
ERROR: A representation for the specified text index has
not been defined on this workstation.
```

Argument Definitions

<i>index</i>	numeric constant or numeric variable name; indicates the fill bundle to check. Valid values are 1 to 20, inclusive. If <i>index</i> is expressed as a variable, the variable must be initialized to a value between 1 and 20.
<i>color-index</i>	numeric variable name; returns the color index of the fill color associated with the bundle. The color index that is returned corresponds to a color specification in the following order: <ol style="list-style-type: none"> 1 a color index assigned with the GSET('COLREP', . . .)function 2 the <i>n</i>th color in the colors list of the COLORS= graphics option 3 the <i>n</i>th color in the device's default colors list.
<i>font</i>	character variable name; returns the text font associated with the bundle.
<i>return-code-variable</i>	numeric variable name; returns the return code of the routine call.

See Also

COLORS= graphics option (see "COLORS" on page 314)
 GASK('COLREP', . . .)routine
 GSET('COLREP', . . .)function
 GSET('TEXREP', . . .)function

TEXUP

Finds the orientation (angle) of the text string

Operating States: GKOP, SGOP, WSAC, WSOP

Return Codes: 0, 8

Syntax

CALL GASK('TEXUP', *up-x*, *up-y*, *return-code-variable*);

Description The GASK('TEXUP', . . .)routine returns the character up vector values. If TEXUP has not been previously selected with the GSET('TEXUP', . . .)function, GASK('TEXUP', . . .)returns the default values for *x* and *y*, 0 and 1. See the GSET('TEXUP', . . .)function on page 1158 for an explanation of the vector values.

Argument Definitions

<i>up-x</i>	numeric variable name; returns the <i>x</i> component of the vector.
<i>up-y</i>	numeric variable name; returns the <i>y</i> component of the vector.
<i>return-code-variable</i>	numeric variable name; returns the return code of the routine call.

See Also

GASK('TEXALIGN', . . .)routine
 GASK('TEXPATH', . . .)routine
 GSET('TEXUP', . . .)function

TRANS

Finds the viewport and window coordinates associated with a transformation number

Operating States: GKOP, SGOP, WSAC, WSOP

Return Codes: 0, 8, 50

Syntax

CALL GASK ('TRANS', *n*, *vllx*, *vly*, *vurx*, *vury*, *wllx*, *wly*, *wurx*, *wury*,
return-code-variable);

Description The GASK('TRANS', . . .)routine returns the viewport and window coordinates associated with a particular transformation number. GASK('TRANS', . . .)returns the default coordinates for viewports and windows if other coordinates have not been defined for the transformation specified.

Argument Definitions

<i>n</i>	numeric constant or numeric variable name; indicates the number of the transformation to check. Valid values are 0 to 20, inclusive. If <i>n</i> is expressed as a variable, the variable must be initialized to a value between 0 and 20.
<i>vllx</i>	numeric variable name; returns the <i>x</i> coordinate of the lower-left viewport corner.
<i>vly</i>	numeric variable name; returns the <i>y</i> coordinate of the lower-left viewport corner.
<i>vurx</i>	numeric variable name; returns the <i>x</i> coordinate of the upper-right viewport corner.
<i>vury</i>	numeric variable name; returns the <i>y</i> coordinate of the upper-right viewport corner.
<i>wllx</i>	numeric variable name; returns the <i>x</i> coordinate of the lower-left window corner.
<i>wly</i>	numeric variable name; returns the <i>y</i> coordinate of the lower-left window corner.
<i>wurx</i>	numeric variable name; returns the <i>x</i> coordinate of the upper-right window corner.
<i>wury</i>	numeric variable name; returns the <i>y</i> coordinate of the upper-right window corner.
<i>return-code-variable</i>	numeric variable name; returns the return code of the routine call.

See Also

GASK('TRANSNO', . . .)routine
 GASK('VIEWPORT', . . .)routine
 GASK('WINDOW', . . .)routine
 GSET('TRANSNO', . . .)function
 GSET('VIEWPORT', . . .)function
 GSET('WINDOW', . . .)function

TRANSNO

Finds the number of the transformation to be used

Operating States: GKOP, SGOP, WSAC, WSOP

Return Codes: 0, 8

Syntax

CALL GASK('TRANSNO', *n*, *return-code-variable*);

Description The GASK('TRANSNO', . . .)routine returns the current transformation. If a transformation has not been previously selected with the GSET('TRANSNO', . . .)function, GASK('TRANSNO', . . .)returns the number of the default transformation, 0.

Argument Definitions

n numeric variable name; returns the number of the current transformation.

return-code-variable numeric variable name; returns the return code of the routine call.

See Also

GASK('TRANS', . . .)
 GASK('VIEWPORT', . . .)
 GASK('WINDOW', . . .)
 GSET('VIEWPORT', . . .)
 GSET('WINDOW', . . .)
 GSET('TRANSNO', . . .)

VIEWPORT

Finds coordinates of the viewport associated with a transformation number

Operating States: GKOP, SGOP, WSAC, WSOP

Return Codes: 0, 8, 50

Syntax

CALL GASK('VIEWPORT', *n*, *llx*, *lly*, *urx*, *ury*, *return-code-variable*);

Description The GASK('VIEWPORT', . . .)routine returns the coordinates of the viewport associated with the specified transformation. If a viewport has not been defined with the GSET('VIEWPORT', . . .)function for the specified transformation, *n*, GASK('VIEWPORT', . . .)returns the default coordinates for the viewport, (0,0) and (1,1).

Argument Definitions

n numeric constant or numeric variable name; indicates the transformation number assigned to the viewport to check. Valid

	values are 0 to 20, inclusive. If <i>n</i> is expressed as a variable, the variable must be initialized to a value between 0 and 20.
<i>llx</i>	numeric variable name; returns the <i>x</i> coordinate of the lower-left corner.
<i>lly</i>	numeric variable name; returns the <i>y</i> coordinate of the lower-left corner.
<i>urx</i>	numeric variable name; returns the <i>x</i> coordinate of the upper-right corner.
<i>ury</i>	numeric variable name; returns the <i>y</i> coordinate of the upper-right corner.
<i>return-code-variable</i>	numeric variable name; returns the return code of the routine call.

See Also

GASK('TRANS', . . .)routine
 GASK('TRANSNO', . . .)routine
 GASK('WINDOW', . . .)routine
 GSET('TRANSNO', . . .)function
 GSET('VIEWPORT', . . .)function
 GSET('WINDOW', . . .)function

VPOS

Finds the number of rows

Operating States: All

Return Codes: 0

Syntax

CALL GASK('VPOS', *vpos*, *return-code-variable*);

Description The GASK('VPOS', . . .)routine returns the current number of rows in the graphics output area. GASK('VPOS', . . .)searches for the current number of rows in the following order:

- 1 the value selected in the GSET('VPOS', . . .)function
- 2 the value of the VPOS= graphics option
- 3 the device's default VPOS value found in the device entry.

Argument Definitions

vpos numeric variable name; returns the number of rows in the graphics output area.

return-code-variable numeric variable name; returns the return code of the routine call.

See Also

GASK('HPOS', . . .)routine
 GASK('VSIZE', . . .)routine
 GSET('VPOS', . . .)function
 VPOS= graphics option (see "VPOS" on page 397)

VSIZE

Finds the vertical dimension of the graphics output area

Operating States: All

Return Codes: 0

Syntax

CALL GASK('VSIZE', *vsize*, *return-code-variable*);

Description The GASK('VSIZE', . . .)routine returns the current vertical dimension, in inches, of the graphics output area. GASK('VSIZE', . . .)searches for the current vertical dimension in the following order:

- 1 the value selected in the GSET('VSIZE', . . .)function
- 2 the value of the VSIZE= graphics option
- 3 the device's default VSIZE found in the device entry.

Argument Definitions

vsize numeric variable name; returns the size of the graphics output area in the *y* dimension (in inches).

return-code-variable numeric variable name; returns the return code of the routine call.

See Also

GASK('HSIZE', . . .)routine
 GASK('VPOS', . . .)routine
 GSET('VSIZE', . . .)function
 VSIZE= graphics option (see "VSIZE" on page 398)

WINDOW

Finds the coordinates of the window associated with a transformation number

Operating States: GKOP, SGOP, WSAC, WSOP

Return Codes: 0, 8, 50

Syntax

CALL GASK('WINDOW', *n*, *llx*, *lly*, *urx*, *ury*, *return-code-variable*);

Description The GASK('WINDOW', . . .)routine returns the coordinates of the window associated with the specified transformation number. If no window has been defined with the GSET('WINDOW', . . .)function for transformation *n*, GASK('WINDOW', . . .)returns the default window coordinates, which are device dependent.

Argument Definitions

<i>n</i>	numeric constant or numeric variable name; indicates the transformation number of the window to check. Valid values are 0 to 20, inclusive. If <i>n</i> is expressed as a variable, the variable must be initialized to a value between 0 and 20.
<i>llx</i>	numeric variable name; returns the <i>x</i> coordinate of the lower-left corner.
<i>lly</i>	numeric variable name; returns the <i>y</i> coordinate of the lower-left corner.
<i>urx</i>	numeric variable name; returns the <i>x</i> coordinate of the upper-right corner.
<i>ury</i>	numeric variable name; returns the <i>y</i> coordinate of the upper-right corner.
<i>return-code-variable</i>	numeric variable name; returns the return code of the routine call.

See Also

GASK('TRANS', . . .)routine
 GASK('TRANSNO', . . .)routine
 GASK('VIEWPORT', . . .)routine
 GSET('TRANSNO', . . .)function
 GSET('VIEWPORT', . . .)function
 GSET('WINDOW', . . .)function

WACTIVE

Finds whether the interface is active

Operating States: All

Return Codes: 29, 30

Syntax

```
CALL GASK('WSACTIVE', status);
```

Description The GASK('WSACTIVE', . . .)routine asks if the workstation is active. When the workstation is active, you can execute certain DSGI routines and functions.

Argument Definitions

status numeric variable name; returns either 29 (active) or 30 (inactive).

See Also

GASK('STATE', . . .)routine

GASK('WSOPEN', . . .)routine

WSOPEN

Finds whether the interface is open

Operating States: All

Return Codes: 24, 25

Syntax

```
CALL GASK('WSOPEN', status);
```

Description The GASK('WSOPEN', . . .)routine asks if the workstation is open. If a workstation is open, the graphics catalog can be accessed.

Argument Definitions

status numeric variable name; returns either 24 (open) or 25 (closed).

See Also

GASK('WSACTIVE', . . .)routine

GASK('WSOPEN', . . .)routine

GDRAW Functions

GDRAW functions create graphics elements. Each GDRAW operator is associated with a set of GSET operators that control its attributes. For example, the color, height,

and font for the GDRAW('TEXT', . . .)function are controlled by GSET('TEXCOLOR', . . .), GSET('TEXHEIGHT', . . .), and GSET('TEXFONT', . . .), respectively. For a complete list of the attributes associated with each GDRAW function, see Table 31.2 on page 1038. The complete graph is displayed after the GRAPH('UPDATE', . . .)function is submitted.

When using GDRAW functions, remember the following:

- All arguments must be specified.
- All arguments are specified as variables or constants. If you express an argument as a variable, the variable must be initialized.
- All character arguments that are expressed as character strings must be enclosed in quotes.
- All character variable names used as arguments *must* be declared in a LENGTH statement.
- All character constants must be enclosed in single or double quotes.

ARC

Draws a circular arc

Operating States: SGOP

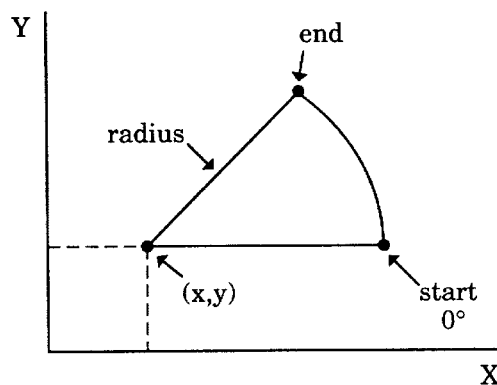
Return Codes: 0, 4, 61, 86

Syntax

return-code-variable=GDRAW('ARC', *x*, *y*, *radius*, *start*, *end*);

Description The GDRAW('ARC', . . .)function draws a circular arc. The line attributes and bundles affect the appearance of this primitive. See Table 31.2 on page 1038 for a list of these attributes. Figure 32.2 on page 1115 illustrates the arguments used with GDRAW('ARC', . . .).

Figure 32.2 Arguments Used with the GDRAW('ARC', ...) Function



Argument Definitions

<i>x</i>	numeric constant or numeric variable name; specifies the <i>x</i> coordinate of the position of the arc on the display; <i>x</i> coordinates are in units based on the current window system.
<i>y</i>	numeric constant or numeric variable name; specifies the <i>y</i> coordinate of the position of the arc on the display; <i>y</i> coordinates are in units based on the current window system;
<i>radius</i>	numeric constant or numeric variable name; the arc radius size is in units based on the current window system.
<i>start</i>	numeric constant or numeric variable name; the starting angle of the arc is in degrees, with 0 degrees at 3 o'clock.
<i>end</i>	numeric constant or numeric variable name; the ending angle of the arc is in degrees, with 0 degrees at 3 o'clock.

See Also

GDRAW('ELLARC', . . .)function
 GDRAW('PIE', . . .)function
 GSET('LINCOLOR', . . .)function
 GSET('LININDEX', . . .)function
 GSET('LINREP', . . .)function
 GSET('LINTYPE', . . .)function
 GSET('LINWIDTH', . . .)function

BAR

Draws a rectangle

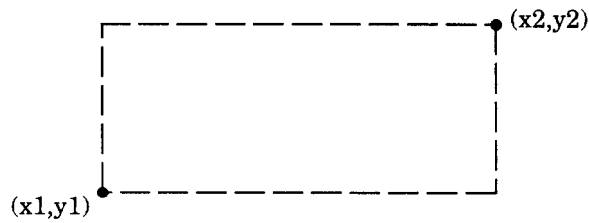
Operating States: SGOP

Return Codes: 0, 4, 76, 79, 80, 86

Syntax

return-code-variable=GDRAW('BAR', *x1*, *y1*, *x2*, *y2*);

Description The GDRAW('BAR', . . .)function draws a rectangular bar whose sides are parallel to the sides of the display area. The fill attributes and bundles affect the appearance of this graphics element. See Table 31.2 on page 1038 for a list of these attributes. Figure 32.3 on page 1117 illustrates the arguments used with GDRAW('BAR', . . .).

Figure 32.3 Points that Draw a Bar

Argument Definitions

$x1$	numeric constant or numeric variable name; refers to the x coordinate of one corner of the bar.
$y1$	numeric constant or numeric variable name; refers to the y coordinate of one corner of the bar.
$x2$	numeric constant or numeric variable name; refers to the x coordinate of the corner of the bar that is diagonally opposite to the corner of $(x1, y1)$.
$y2$	numeric constant or numeric variable name; refers to the y coordinate of the corner of the bar that is diagonally opposite to the corner of $(x1, y1)$.

See Also

GDRAW('FILL', . . .)function
 GSET('FILCOLOR', . . .)function
 GSET('FILINDEX', . . .)function
 GSET('FILREP', . . .)function
 GSET('FILTYPE', . . .)function
 GSET('FILSTYLE', . . .)function
 GSET('HTML', . . .)function

ELLARC

Draws an elliptical arc

Operating States: SGOP

Return Codes: 0, 4, 61, 86

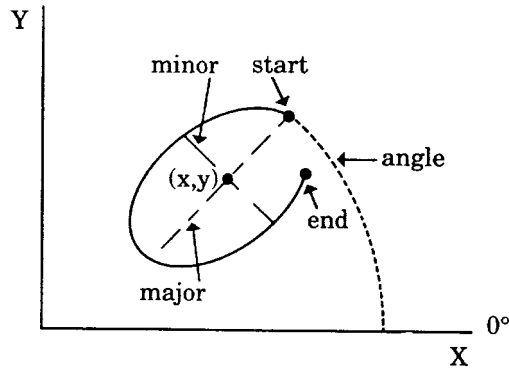
Syntax

return-code-variable =GDRAW('ELLARC', x , y , *major*, *minor*, *start*, *end*, *angle*);

Description The GDRAW('ELLARC', . . .)function draws a hollow section of an ellipse. The line attributes and bundles affect the appearance of this primitive. See

Table 31.2 on page 1038 for a list of these attributes. Figure 32.4 on page 1118 illustrates the arguments used with GDRAW('ELLARC', . . .) and GDRAW('ELLIPSE', . . .).

Figure 32.4 Arguments Used with GDRAW('ELLARC',...) function and GDRAW('ELLIPSE',...) function



Argument Definitions

<i>x</i>	numeric constant or numeric variable name; <i>x</i> coordinates are in units based on the current window system.
<i>y</i>	numeric constant or numeric variable name; <i>y</i> coordinates are in units based on the current window system.
<i>major</i>	numeric constant or numeric variable name; the major axis lengths for the elliptical arc.
<i>minor</i>	numeric constant or numeric variable name; the minor axis lengths for the elliptical arc.
<i>start</i>	numeric constant or numeric variable name; the starting angle from the major axis, in degrees, for the elliptical arc with 0 degrees beginning at the major axis.
<i>end</i>	numeric constant or numeric variable name; the ending angle from the major axis, in degrees, for the elliptical arc with 0 degrees at 3 o'clock.
<i>angle</i>	numeric constant or numeric variable name; the angle that the major axis of the elliptical arc has to 0 degrees (with 0 degrees at 3 o'clock).

See Also

GDRAW('ELLIPSE', . . .)function

GSET('LINCOLOR', . . .)function

GSET('LINTYPE', . . .)function

GSET('LINWIDTH', . . .)function

GSET('LINREP', . . .)function

GSET('LININDEX', . . .)function

ELLIPSE

Draws an ellipse

Operating States: SGOP

Return Codes: 0, 4, 76, 79, 80, 86

Syntax

return-code-variable =GDRAW('ELLIPSE', *x*, *y*, *major*, *minor*, *start*, *end*, *angle*);

Description The GDRAW('ELLIPSE', . . .)function draws a filled section of an ellipse. The fill attributes and bundles affect the appearance of this primitive. See Table 31.2 on page 1038 for a list of these attributes. Figure 32.4 on page 1118 illustrates the arguments used with GDRAW('ELLARC', . . .)and GDRAW('ELLIPSE', . . .).

Argument Definitions

<i>x</i>	numeric constant or numeric variable name; the <i>x</i> coordinate of the position of the ellipse on the display.
<i>y</i>	numeric constant or numeric variable name; the <i>y</i> coordinate of the position of the ellipse on the display.
<i>major</i>	numeric constant or numeric variable name; the major axis length for the ellipse.
<i>minor</i>	numeric constant or numeric variable name; the minor axis length for the ellipse.
<i>start</i>	numeric constant or numeric variable name; the starting angle for the ellipse from the major axis, with 0 degrees beginning at the major axis.
<i>end</i>	numeric constant or numeric variable name; the ending angle for the ellipse from the major axis, with 0 degrees at 3 o'clock.
<i>angle</i>	numeric constant or numeric variable name; the angle that the major axis of the ellipse has to 0 degrees, with 0 degrees at 3 o'clock.

See Also

GDRAW('ELLARC', . . .)function
 GSET('FILCOLOR', . . .)function
 GSET('FILINDEX', . . .)function
 GSET('FILREP', . . .)function
 GSET('FILTYPE', . . .)function
 GSET('FILWIDTH', . . .)function
 GSET('HTML', . . .)function

FILL

Draws a filled area

Operating States: SGOP

Return Codes: 0, 4, 76, 79, 80, 86, 100, 301

Syntax

return-code-variable=GDRAW('FILL', *n*, *x-values*, *y-values*);

Description The GDRAW('ILL' . . .)function draws a filled polygon. The fill attributes and bundles affect the appearance of this primitive. See Table 31.2 on page 1038 for a list of these attributes.

Note: All of the *x* coordinates are listed in the function first, followed by the *y* coordinates. This primitive takes the first *n* values and stores them as *x* coordinates. The next *n* values are stored as *y* coordinates. Δ

Argument Definitions

<i>n</i>	numeric constant or numeric variable name; the number of vertices (<i>x</i> and <i>y</i> pairs) in the polygon. You can specify a missing value (.) for <i>n</i> . If <i>n</i> is missing, the number of vertices is computed from the number of <i>x</i> and <i>y</i> arguments.
<i>x-values</i>	list of numeric constants, variables, or OF arguments that describe the <i>x</i> coordinates for the vertices in units based on the current window system.
<i>y-values</i>	list of numeric constants, variables, or OF arguments that describe the <i>y</i> coordinates for the vertices in units based on the current window system.

See Also

GDRAW('BAR', . . .)function
 GSET('FILCOLOR', . . .)function
 GSET('FILINDEX', . . .)function
 GSET('FILREP', . . .)function
 GSET('FILTYPE', . . .)function
 GSET('FILSTYLE', . . .)function
 GSET('HTML', . . .)function

LINE

Draws a polyline

Operating States: SGOP

Return Codes: 0, 4, 61, 86, 100, 301

Syntax

return-code-variable=GDRAW('LINE', *n*, *x-values*, *y-values*);

Description The GDRAW('LINE' . . .)function draws one line, a series of connected lines, or a dot. The line attributes and bundles affect the appearance of this primitive. See Table 31.2 on page 1038 for a list of these attributes.

Note: All of the *x* coordinates are listed in the function first, followed by the *y* coordinates. This primitive takes the first *n* values and stores them as *x* coordinates and the next *n* values and stores them as *y* coordinates. Δ

Argument Definitions

<i>n</i>	numeric constant or numeric variable name; the number of vertices (<i>x</i> and <i>y</i> pairs) in the polygon. You can specify a missing value (.) for <i>n</i> . If <i>n</i> is missing, the number of vertices is computed from the number of <i>x</i> and <i>y</i> pairs.
<i>x-values</i>	list of numeric constants, variables, or OF arguments that describe the <i>x</i> coordinates for the vertices in units based on the current window system.
<i>y-values</i>	list of numeric constants, variables, or OF argument lists that describe the <i>y</i> coordinates for the vertices in units based on the current window system.

See Also

GSET('LINCOLOR', . . .)function
 GSET('LININDEX', . . .)function
 GSET('LINREP', . . .)function
 GSET('LINTYPE', . . .)function
 GSET('LINWIDTH', . . .)function

MARK

Draws a polymarker

Operating States: SGOP

Return Codes: 0, 4, 65, 86, 100, 301

Syntax

return-code-variable=GDRAW ('MARK', *n*, *x-values*, *y-values*);

Description The GDRAW('MARK', . . .)function draws a series of symbols. The marker attributes and bundles affect the appearance of this primitive. See Table 31.2 on page 1038 for a list of these attributes. Refer to the GSET('MARTYPE', . . .)function on page 1149 for a list of symbols that you can draw with GDRAW('MARK', . . .).

Note: All of the *x* coordinates are listed in the function first, followed by the *y* coordinates. This primitive takes the first *n* values and stores them as *x* coordinates and the next *n* values and stores them as *y* coordinates. Δ

Argument Definitions

<i>n</i>	numeric constant or numeric variable name; the number of times the symbol is drawn. You can specify a missing value (.) for <i>n</i> . If <i>n</i> is missing, the number of vertices is calculated from the number of <i>x</i> and <i>y</i> pairs.
<i>x-values</i>	list of numeric constants, variables, or OF arguments that describe the <i>x</i> coordinates of the symbols in units based on the current window system.
<i>y-values</i>	list of numeric constants, variables, or OF arguments that describe the <i>y</i> coordinates of the symbols in units based on the current window system.

See Also

GDRAW('TEXT', . . .)function
 GSET('HTML', . . .)function
 GSET('MARCOLOR', . . .)function
 GSET('MARINDEX', . . .)function
 GSET('MARREP', . . .)function
 GSET('MARTYPE', . . .)function
 GSET('MARWIDTH', . . .)function

MESSAGE

Prints a message in the SAS log

Operating States: All

Return Codes: 0

Syntax

return-code-variable=GDRAW('MESSAGE', *message*);

Description The GDRAW('MESSAGE', . . .)function prints a message in the SAS log. This function may be used for debugging applications or for printing custom messages for your application.

Argument Definitions

message character string enclosed in quotes or character variable name; the text to be printed in the log.

See Also

GSET('MESSAGE', . . .)function
 GPRINT('code') utility function

PIE

Draws a filled circle or section of a filled circle

Operating States: SGOP

Return Codes: 0, 4, 76, 79, 80, 86

Syntax

```
return-code-variable=GDRAW('PIE', x, y, radius, start, end);
```

Description The GDRAW('PIE', . . .)function draws a filled section of a circular arc. The fill attributes and bundles affect the appearance of this primitive. See Table 31.2 on page 1038 for a list of these attributes.

Argument Definitions

<i>x</i>	numeric constant or numeric variable name; <i>x</i> coordinates are in units based on the current window system.
<i>y</i>	numeric constant or numeric variable name; <i>y</i> coordinates are in units based on the current window system.
<i>radius</i>	numeric constant or numeric variable name; the pie radius size in units based on the current window system.
<i>start</i>	numeric constant or numeric variable name; the starting angle of the pie, with 0 degrees at 3 o'clock on the unit circle.
<i>end</i>	numeric constant or numeric variable name; the ending angle of the pie, with 0 degrees at 3 o'clock on the unit circle.

See Also

GDRAW('ARC', . . .)function
 GSET('FILCOLOR', . . .)function
 GSET('FILINDEX', . . .)function
 GSET('FILREP', . . .)function
 GSET('FILTYPE', . . .)function
 GSET('FILSTYLE', . . .)function
 GSET('HTML', . . .)function

TEXT

Draws a text string

Operating States: SGOP

Return Codes: 0, 4, 69, 86

Syntax

```
return-code-variable=GDRAW('TEXT', x, y, string);
```

Description The GDRAW('TEXT', . . .)function draws a text string. The text attributes and bundles affect the appearance of this primitive. See Table 31.2 on page 1038 for a list of these attributes.

Argument Definitions

<i>x</i>	numeric constant or numeric variable name; <i>x</i> coordinates are in units based on the current window system.
<i>y</i>	numeric constant or numeric variable name; <i>y</i> coordinates are in units based on the current window system.
<i>string</i>	character string enclosed in quotes or character variable name; a set of characters to be drawn on the output beginning at position (<i>x,y</i>).

See Also

GDRAW('MARK', . . .)function
 GSET('HTML', . . .)function
 GSET('TEXCOLOR', . . .)function
 GSET('TEXINDEX', . . .)function
 GSET('TEXREP', . . .)function
 GSET('TEXTYPE', . . .)function
 GSET('TEXHEIGHT', . . .)function

GRAPH Functions

GRAPH functions perform library management tasks from within the DATA Step Graphics Interface. These functions can only be performed on one catalog at a time. They cannot be performed across catalogs. For example, you cannot copy a graph from one catalog to another.

When using GRAPH functions, remember the following:

- All arguments are specified as variables or constants. If you express an argument as a variable, the variable must be initialized.
- All character arguments expressed as character strings must be enclosed in quotes.
- All character variable names used as arguments *must* be declared in a LENGTH statement.
- All character constants must be enclosed in single or double quotes.

CLEAR

Opens a graphics segment for output

Operating States: WSAC

Return Codes: 0, 3, 301, 302

Resulting Operating State: SGOP

Syntax

return-code-variable=GRAPH ('CLEAR'<, *name*> <, *des*><, *byline*>);

Description The GRAPH('CLEAR', . . .)function opens a graphics segment for output in the current catalog. The first parameter, 'CLEAR', is the only required one. The values of *name*, *des*, and *byline* are displayed in catalog listings and in catalog information in the GREPLAY procedure.

If the name specified is an existing graph, DSGI will suffix the name with a number. For example, if PIE is chosen for the name and it already exists, DSGI will name the output PIE1; the next time the code is submitted, DSGI names the output PIE2, and so forth.

This function moves the operating state from WSAC to SGOP.

Argument Definitions

<i>name</i>	character string enclosed in quotes or character variable name; gives a name to the graph to be opened. If <i>name</i> is not specified, DSGI assigns the graph a name that is some form of DSGI: for example, DSGI, DSGI1, and DSGI2.
<i>des</i>	character string enclosed in quotes or character variable name; gives a description to the graph to be opened. If <i>des</i> is not specified, DSGI assigns the following description to the catalog entry: Graph from DATA Step Graphics Interface.
<i>by-line</i>	character string enclosed in quotes or character variable name; gives another line of description for the graph. The byline appears under the titles on the graph. DSGI does not provide a default byline.

See Also

GASK('OPENGRAPH', . . .)routine
 GRAPH('UPDATE', . . .)function

COPY

Copies a graph

Operating States: GKOP, WSOP, WSAC, SGOP

Return Codes: 0, 8, 307

Syntax

return-code-variable=GRAPH('COPY', *name*, *new-name*);

Description The GRAPH('COPY', . . .)function copies a graph to another catalog entry. The graph to be copied must be closed and be in the current catalog. You cannot copy from one catalog to another. The new graph will also be in the current catalog.

Argument Definitions

<i>name</i>	character string enclosed in quotes or character variable name; name of the graph to be copied.
-------------	---

new-name character string enclosed in quotes or character variable name;
name of the graph to be created.

See Also

GASK('CATALOG', . . .)routine
 GRAPH('DELETE', . . .)function
 GRAPH('INSERT', . . .)function
 GSET('CATALOG', . . .)function

DELETE

Deletes a graph

Operating States: GKOP, SGOP, WSAC, WSOP

Return Codes: 0, 4, 8, 307

Syntax

return-code-variable=GRAPH('DELETE', *name*);

Description The GRAPH('DELETE', . . .)function deletes a graph in the current catalog. The graph does not have to be closed to be deleted.

Argument Definitions

name character string enclosed in quotes or character variable name; the
name of the graph to delete.

See Also

GASK('CATALOG', . . .)routine
 GRAPH('COPY', . . .)function
 GSET('CATALOG', . . .)function

INSERT

Inserts a previously created segment into the currently open graph

Operating States: SGOP

Return Codes: 0, 4, 302, 307

Syntax

return-code-variable=GRAPH('INSERT', *name*);

Description The GRAPH('INSERT', . . .)function inserts a graph into the currently open graph. The graph to be inserted must be closed and be in the current catalog.

Argument Definitions

name character string enclosed in quotes or character variable name; the name of a graph to be inserted.

See Also

GASK('CATALOG', . . .)routine

GRAPH('COPY', . . .)function

GSET('CATALOG', . . .)function

RENAME

Renames a graph

Operating States: GKOP, SGOP, WSAC, WSOP

Return Codes: 0, 8, 307

Syntax

return-code-variable=GRAPH('RENAME', *name*, *new-name*);

Description The GRAPH('RENAME', . . .)function renames a graph. The graph to be renamed must be in the current catalog and be closed.

Argument Definitions

name character string enclosed in quotes or character variable name; the name of the closed graph that is to be changed.

new-name character string enclosed in quotes or character variable name; the new name for the graph.

See Also

GASK('CATALOG', . . .)routine
 GRAPH('INSERT', . . .)function
 GSET('CATALOG', . . .)function

UPDATE

Completes the currently open graph and (optionally) displays it

Operating States: SGOP

Return Codes: 0, 4

Resulting Operating State: WSAC

Syntax

return-code-variable=GRAPH('UPDATE' <, 'show'>);

Description The GRAPH('UPDATE', . . .)function closes the graph currently open and displays it. DSGI operates in buffered mode, so the picture is never displayed until this function is called.

This function can be called only once for the currently open graph. Therefore, you cannot incrementally build a graph; however, you can close the currently open graph and later insert it into another graph within the same DATA step.

This function moves the operating state from SGOP to WSAC.

Argument Definitions

show character string, optional; valid values are SHOW and NOSHOW. If SHOW is specified, the graph is displayed. If NOSHOW is specified, the graph is closed and not displayed.

See Also

GRAPH('CLEAR', . . .)function

GSET Functions

GSET functions allow you to set attributes for the graphics elements. Some GSET functions set the attributes for a subset of graphics primitives. For example, attributes prefixed by FIL control the appearance of the graphics primitives GDRAW('BAR', . . .), GDRAW('ELLIPSE', . . .), GDRAW('FILL', . . .), and GDRAW('PIE', . . .). See Table 31.2 on page 1038 for a complete list of the attributes that control the appearance of the graphics primitives.

Some GSET functions affect the appearance of the entire graphics output. For example, GSET('HPOS', . . .)and GSET('VPOS', . . .)set the number of columns and

rows for the output. See each GSET function for the aspect of the graphics output that it controls.

When using GSET functions, remember the following:

- All arguments must be specified.
- All arguments are specified as variables or constants. If you express an argument as a variable, the variable must be initialized.
- All character arguments that are expressed as character strings must be enclosed in quotation marks.
- All character variable names used as arguments *must* be declared in a LENGTH statement.
- All character constants must be enclosed in single or double quotation marks.

ASF

Specifies an aspect source flag to bundle or separate attributes

Operating States: GKOP, SGOP, WSAC, WSOP

Return Codes: 0, 8

Default Value: INDIVIDUAL

Syntax

return-code-variable=GSET('ASF', *attribute*, *status*);

Description The GSET('ASF', . . .)function sets an attribute's aspect source flag (ASF) so that it can be used in a bundle (BUNDLED) or individually (INDIVIDUAL).

If an attribute's ASF is set to 'BUNDLED', it cannot be used outside of a bundle. It must be defined in a GSET('xxxREP', . . .)function and activated with a GSET('xxxINDEX', . . .)function, where *xxx* can have one of the following values: FIL, LIN, MAR, TEX.

If an attribute's ASF is set to 'INDIVIDUAL', it cannot be used with a bundle. In this case, the attribute is set with a GSET('attribute', . . .). The values of *attribute* are listed in "Argument Definitions."

Argument Definitions

attribute character string enclosed in quotes or character variable name with one of the following values:

- FILCOLOR
- FILSTYLE
- FILTYPE
- LINCOLOR
- LINTYPE
- LINWIDTH
- MARCOLOR
- MARSIZE
- MARTYPE

- `TEXCOLOR`
- `TEXTFONT`.

status character string enclosed in quotation marks or character variable name; accepts either the value `BUNDLED` or `INDIVIDUAL`.

See Also

`GASK('ASF', . . .)` routine
`GSET('FILCOLOR', . . .)` function
`GSET('FILSTYLE', . . .)` function
`GSET('FILTYPE', . . .)` function
`GSET('LINCOLOR', . . .)` function
`GSET('LINTYPE', . . .)` function
`GSET('LINWIDTH', . . .)` function
`GSET('MARCOLOR', . . .)` function
`GSET('MARSIZE', . . .)` function
`GSET('MARTYPE', . . .)` function
`GSET('TEXCOLOR', . . .)` function
`GSET('TEXTFONT', . . .)` function

ASPECT

Specifies the aspect ratio

Operating States: GKCL

Return Codes: 0, 1, 90, 307

Default Value: 0.0

Syntax

return-code-variable=`GSET('ASPECT', aspect)`;

Description The `GSET('ASPECT', . . .)` function sets the aspect ratio used to draw graphics output. `GSET('ASPECT', . . .)` affects only pies, arcs, and software text.

Argument Definitions

aspect numeric constant or numeric variable name; specifies the aspect ratio and cannot be less than 0.

See Also

ASPECT= graphics option (see “ASPECT” on page 305)
 GASK('ASPECT', . . .)routine

CATALOG

Specifies the catalog for the graphs

Operating States: GKCL

Return Codes: 0, 1

Default Values: *libref* = WORK, *catalog-name*=GSEG

Syntax

return-code-variable=GSET('CATALOG', *libref*, *catalog-name*);

Description The GSET('CATALOG', . . .)function makes the specified catalog the current catalog in which to store graphs generated with DSGI. GSET('CATALOG', . . .)creates the catalog if it does not exist.

The values of *libref* and *catalog-name* cannot exceed eight characters. The number of characters allowed for a catalog name varies across operating environments; see the SAS companion for your operating system. *Libref* should have been defined through the LIBNAME statement.

Argument Definitions

<i>libref</i>	character string enclosed in quotation marks or character variable name; points to the library that contains the catalog.
<i>catalog-name</i>	character string enclosed in quotation marks or character variable name; specifies the catalog name to be used.

See Also

GASK('CATALOG', . . .)routine
 GASK('GRAPHLIST', . . .)routine
 GASK('NUMGRAPH', . . .)routine

CBACK

Specifies the background color

Operating States: GKCL

Return Codes: 0, 1

Default Value: 1. CBACK= graphics option, if specified; 2. device's default background color.

Syntax

return-code-variable=GSET('CBACK', *cback*);

Description The GSET('CBACK', . . .)function sets the background color. GSET('CBACK', . . .)has the same effect as the CBACK= graphics option.

Argument Definitions

cback character string enclosed in quotation marks or character variable name; can contain any predefined SAS color name. See "Predefined SAS Colors" on page 145 for a list of predefined SAS color names.

See Also

CBACK= graphics option (see "CBACK" on page 309)

GASK('CBACK', . . .)routine

CLIP

Specifies whether clipping is on or off

Operating States: GKOP, SGOP, WSAC, WSOP

Return Codes: 0

Default Value: OFF

Syntax

return-code-variable=GSET('CLIP', *status*);

Description The GSET('CLIP', . . .)function activates or suppresses clipping around the current viewport.

Argument Definitions

status character string enclosed in quotation marks or character variable name; valid values are ON and OFF. When ON is used, the graphics elements outside of the specified viewport are not displayed. If you turn clipping OFF, the graphics elements outside of the defined viewport are displayed.

See Also

GASK('CLIP', . . .)routine
 GASK('VIEWPORT', . . .)routine
 GSET('VIEWPORT', . . .)function

COLREP

Associates a color name with a certain color index

Operating States: SGOP

Return Codes: 0, 4, 86

Default Values: 1. colors list of COLORS= graphics option; 2. device's default colors list

Syntax

return-code-variable=GSET('COLREP', *color-index*, *color*);

Description The GSET('COLREP', . . .)function associates a predefined SAS color name with a color index. Many of the GASK routines and GSET functions use *color-index* as an argument.

If this function is not used, DSGI searches for a color specification in the following order:

- 1 the *n*th color in the colors list of the COLORS= graphics option
- 2 the *n*th color in the device's default colors list.

Argument Definitions

color-index numeric constant or numeric variable name; a number from 1 to 256 that identifies a color.

color character string enclosed in quotation marks or character variable name; a predefined SAS color name. See "Predefined SAS Colors" on page 145 for a list of predefined SAS color names.

See Also

COLORS= graphics option (see "COLORS" on page 314)
 GASK('COLINDEX', . . .)routine
 GASK('COLREP', . . .)routine

DEVICE

Specifies the output graphics device

Operating States: GKCL

Return Codes: 0, 1

Default Value: 1. DEVICE= graphics option, if specified; 2. value entered in DEVICE prompt window; 3. value entered in OPTIONS window

Syntax

return-code-variable=GSET('DEVICE', *device*);

Description The GSET('DEVICE', . . .)function selects the device driver.

Argument Definitions

device character string enclosed in quotation marks or character variable name; the name of the driver you will be using. *Device* must match one of the device entries in the SASHELP.DEVICES catalog or one of your personal device catalogs, GDEVICE0.DEVICES through GDEVICE9.DEVICES. Refer to "About Device Catalogs" on page 652 for more information about catalogs that store device entries.

See Also

DEVICE= graphics option (see "DEVICE" on page 321)

GASK('DEVICE', . . .)routine

FILCOLOR

Specifies the color index of the color used to draw fill areas

Operating States: GKOP, SGOP, WSAC, WSOP

Return Codes: 0, 8, 85

Default Value: 1

Syntax

return-code-variable=GSET('FILCOLOR', *color-index*);

Description The GSET('FILCOLOR', . . .)function selects the color index of the color used to draw fill areas. The aspect source flag (ASF) of FILCOLOR must be set to 'INDIVIDUAL' for this attribute to be used outside of a fill bundle.

DSGI searches for a color to assign to the index in the following order:

- 1 the color specified for the index in a GSET('COLREP', . . .)function
- 2 the *n*th color in the colors list of the COLORS= graphics option
- 3 the *n*th color in the device's default colors list found in the device entry.

Argument Definitions

color-index numeric constant or numeric variable name; indicates the index of the color to be used. Valid values are 1 to 256, inclusive.

See Also

COLORS= graphics option (see “COLORS” on page 314)
 GSET('ASF', . . .)routine
 GSET('COLREP', . . .)function
 GASK('FILCOLOR', . . .)function
 GSET('FILREP', . . .)function

FILINDEX

Specifies the index of the bundle of fill area attributes

Operating States: GKOP, SGOP, WSAC, WSOP

Return Codes: 0, 8, 75

Default Value: 1

Syntax

return-code-variable=GSET('FILINDEX', *index*);

Description The GSET('FILINDEX', . . .)function activates a particular fill bundle. The aspect source flag (ASF) for FILCOLOR, FILSTYLE, and FILTYPE must be set to 'BUNDLED' before the associated GDRAW function is executed if you want the bundled values to be used when the affected graphics element is drawn.

Argument Definitions

index numeric constant or numeric variable name; specifies the index number of the fill bundle. Valid values are 1 to 20, inclusive.

See Also

GASK('FILINDEX', . . .)routine
 GSET('ASF', . . .)function
 GSET('FILREP', . . .)function

FILREP

Associates a bundle of fill attributes with an index

Operating States: GKOP, SGOP, WSAC, WSOP

Return Codes: 0, 8, 75, 78, 85

Default Value: none

Syntax

return-code-variable =GSET('FILREP', *index*, *color-index*, *interior*, *style-index*);

Description The GSET('FILREP', . . .)function assigns a color, type of interior, and style of the interior to a specific fill bundle. The aspect source flags for FILCOLOR, FILTYPE, and FILSTYLE must be set to 'BUNDLED' before the associated GDRAW function is executed if you want the bundled values to be used when the affected graphics element is drawn.

Argument Definitions

- | | |
|--------------------|--|
| <i>index</i> | numeric constant or numeric variable name; indicates the index to be used with the bundle. Valid values are 1 to 20, inclusive. If <i>index</i> is expressed as a variable, the variable name must be initialized to a value between 1 and 20. |
| <i>color-index</i> | numeric constant or numeric variable name; indicates the index of the color to be used. Valid values are 1 to 256, inclusive. The color index should represent one of the following: <ul style="list-style-type: none"> □ a color index assigned with the GSET('COLREP', . . .)function □ the <i>n</i>th color in the colors list of the COLORS= graphics option □ the <i>n</i>th color in the device's default colors list. |
| <i>interior</i> | character string enclosed in quotation marks or character variable name; indicates the type of interior. Valid values are <ul style="list-style-type: none"> □ HATCH □ HOLLOW □ PATTERN □ SOLID. |
| <i>style-index</i> | numeric constant or numeric variable name; indicates the index of the style to be used. Valid values are 1 to 15, inclusive, when FILTYPE is PATTERN, or 1 to 60, inclusive, when FILTYPE is HATCH. See the GSET('FILSTYLE', . . .)function on page 1138 for a table of the patterns used for each style index. If <i>interior</i> is HOLLOW or SOLID, <i>style-index</i> is ignored. |

See Also

GASK('FILREP', . . .)routine
 GSET('ASF', . . .)function
 GSET('COLREP', . . .)function
 GSET('FILCOLOR', . . .)function
 GSET('FILINDEX', . . .)function
 GSET('FILSTYLE', . . .)function
 GSET('FILTYPE', . . .)function

FILSTYLE

Specifies the style of the interior of the fill area when the FILTYPE is PATTERN or HATCH

Operating State: GKOP, SGOP, WSAC, WSOP

Return Codes: 0, 8, 78

Default Value: 1

Syntax

return-code-variable=GSET('FILSTYLE', *style-index*);

Description The GSET('FILSTYLE', . . .)function activates a particular fill pattern when FILTYPE is specified as either PATTERN or HATCH. The aspect source flag (ASF) must be set to 'INDIVIDUAL' for this attribute to be used outside of a fill bundle.

Table 32.1 Style Index Table

Value	PATTERN	HATCH	Value	PATTERN	HATCH
1	X1	M1X	31		M3N045
2	X2	M1X030	32		M3N060
3	X3	M1X045	33		M3N090
4	X4	M1X060	34		M3N120
5	X5	M1N	35		M3N135
6	L1	M1N030	36		M3N150
7	L2	M1N045	37		M4X
8	L3	M1N060	38		M4X030
9	L4	M1N090	39		M4X045
10	L5	M1N120	40		M4X060
11	R1	M1N135	41		M4N
12	R2	M1N150	42		M4N030

Value	PATTERN	HATCH	Value	PATTERN	HATCH
13	R3	M2X	43		M4N045
14	R4	M2X030	44		M4N060
15	R5	M2X045	45		M4N090
16		M2X060	46		M4N120
17		M2N	47		M4N135
18		M2N030	48		M4N150
19		M2N045	49		M5X
20		M2N060	50		M5X030
21		M2N090	51		M5X045
22		M2N120	52		M5X060
23		M2N135	53		M5N
24		M2N150	54		M5N030
25		M3X	55		M5N045
26		M3X030	56		M5N060
27		M3X045	57		M5N090
28		M3X060	58		M5N120
29		M3N	59		M5N135
30		M3N030	60		M5N150

Argument Definitions

style-index numeric constant or numeric variable name. Valid values are 1 to 15, inclusive, when FILTYPE is PATTERN, or 1 to 60, inclusive, when FILTYPE is HATCH. See Table 31.1 on page 1034 for value specifications.

See Also

GASK('FILSTYLE', . . .)routine
 GSET('ASF', . . .)function
 GSET('FILREP', . . .)function
 GSET('FILTYPE', . . .)function

FILTYPE

Specifies the type of the interior of the fill area

Operating States: GKOP, SGOP, WSAC, WSOP

Return Codes: 0, 8, 78

Default Value: HOLLOW

Syntax

return-code-variable=GSET('FILTYPE', *interior*);

Description The GSET('FILTYPE', . . .)function selects a particular type of interior fill. If FILTYPE is set to HATCH or PATTERN, the GSET('FILSTYLE', . . .)function determines the type of hatch or pattern fill used. The aspect source flag (ASF) for FILTYPE must be set to 'INDIVIDUAL' for this attribute to be used outside of a fill bundle.

Argument Definitions

interior character string or character variable name; indicates the type of interior fill. Valid values are

- HATCH
- HOLLOW
- PATTERN
- SOLID.

See Also

GSET('ASF', . . .)function
 GSET('FILREP', . . .)function
 GSET('FILSTYLE', . . .)function

HPOS

Specifies the number of columns

Operating States: GKCL

Return Codes: 0, 1, 90, 307

Default Value: 1. HPOS= graphics option, if specified; 2. device's default HPOS setting

Syntax

return-code-variable=GSET('HPOS', *hpos*);

Description The GSET('HPOS', . . .)function sets the number of columns in the graphics output area. GSET('HPOS', . . .)has the same effect as the HPOS= graphics option. See "HPOS" on page 355 for more information. You can reset the HPOS value by submitting one of the following statements:

```
goptions reset=goptions;
goptions reset=all;

goptions hpos=0;
```

Argument Definitions

hpos numeric constant or numeric variable name; specifies the number of horizontal columns; must be greater than 0.

See Also

GASK('HPOS', . . .)routine

GSET('HSIZE', . . .)function

GSET('VPOS', . . .)function

HPOS= graphics option (see "HPOS" on page 355)

HSIZE

Specifies the horizontal dimension of the graphics output area

Operating States: GKCL

Return Codes: 0, 1, 90, 307

Default Value: 1. HSIZE= graphics option, if specified; 2. HSIZE device parameter

Syntax

return-code-variable=GSET('HSIZE', *hsize*);

Description The GSET('HSIZE', . . .)function sets the horizontal dimension, in inches, of the graphics output area. GSET('HSIZE', . . .)affects the dimensions of the default window. You can reset the HSIZE value by submitting one of the following statements:

```
goptions reset=goptions;
goptions reset=all;

goptions hsize=0;
```

Argument Definitions

hsize numeric constant or numeric variable name; specifies the horizontal dimension, in inches, of the graphics output area; must be greater than 0.

See Also

GASK('HSIZE', . . .)routine
 GSET('HPOS', . . .)function
 GSET('VSIZE', . . .)function
 HSIZE= graphics option (see "HSIZE" on page 356)

HTML

Specifies the HTML string to invoke when an affected DSGI graphic element in a web page is clicked

Operating States: GKOP, SGOP, WSAC, WSOP

Return Codes: 0, 8

Default Value: null

Syntax

return-code-variable=GSET('HTML', 'string');

Description The GSET('HTML', . . .)function sets the HTML string to invoke when an affected DSGI graphic element in a web page is clicked. The HTML string is used with ODS processing to create a drill-down graph. The string value is used as the value for the HREF= attribute in the image map that implements the drill-down capability.

The value for *string* must be *href=* followed by a valid URL that is specified in double quotation marks, as in

```
rc = GSET('HTML', 'HREF="http://www.sas.com/"');
```

The HTML string can be used by any of the following graphic element types drawn in the code: BAR, ELLIPSE, FILL, MARK, PIE, and TEXT. The string applies to all of these element types that are drawn *after* the string is set. To change the HTML string, set a new value. To turn off the HTML string, specify a null string:

```
rc = GSET('HTML', '');
```

For more information on drill-down graphs, see "About Drill-down Graphs" on page 90. For an example of how to use DSGI to generate a drill-down graph, see "Generating a Drill-down Graph Using DSGI" on page 1069.

Argument Definitions

string the HTML string. The string must be enclosed in single quotation marks and must begin with *href=* followed by a URL that is enclosed in double quotation marks.

See Also

GASK('HTML', . . .)function
 GDRAW('BAR', . . .)function
 GDRAW('ELLIPSE', . . .)function
 GDRAW('FILL', . . .)function
 GDRAW('MARK', . . .)function
 GDRAW('PIE', . . .)function
 GDRAW('TEXT', . . .)function

LINCOLOR

Specifies the color index of the color used to draw lines

Operating States: GKOP, SGOP, WSAC, WSOP

Return Codes: 0, 8, 85

Default Value: 1

Syntax

return-code-variable=GSET('LINCOLOR', *color-index*);

Description The GSET('LINCOLOR', . . .)function selects the index of the color used to draw lines. The aspect source flag (ASF) for LINCOLOR must be set to 'INDIVIDUAL' for this attribute to be used outside of a line bundle.

DSGI searches for a color specification in the following order:

- 1 the color specified for the index in a GSET('COLREP', . . .)function
- 2 the *n*th color in the colors list of the COLORS= graphics option
- 3 the *n*th color in the device's default colors list found in the device entry.

Argument Definitions

color-index numeric constant or numeric variable name; indicates the index of the color to use. Valid values are 1 to 256, inclusive.

See Also

COLORS= graphics option (see "COLORS" on page 314)
 GASK('LINCOLOR', . . .)routine
 GSET('ASF', . . .)function
 GSET('COLREP', . . .)function
 GSET('LINREP', . . .)function

LININDEX

Specifies the index of the bundle of line attributes

Operating States: GKOP, SGOP, WSAC, WSOP

Return Codes: 0, 8, 60

Default Value: 1

Syntax

return-code-variable=GSET('LININDEX', *index*);

Description The GSET('LININDEX', . . .)function activates a particular line bundle. The aspect source flags (ASF) of LINCOLOR, LINTYPE, and LINWIDTH must be set to 'BUNDLED' before the associated GDRAW function is executed if you want the bundled values to be used when the affected graphics element is drawn.

Argument Definitions

index numeric constant or numeric variable name; indicates the index of the bundle to activate. Valid values are 1 to 20, inclusive.

See Also

GASK('LININDEX', . . .)routine

GSET('ASF', . . .)function

GSET('LINREP', . . .)function

LINREP

Associates a bundle of line attributes with an index

Operating States: GKOP, SGOP, WSAC, WSOP

Return Codes: 0, 8, 60, 62, 85, 90

Default Value: none

Syntax

return-code-variable=GSET ('LINREP',*index*, *color-index*, *width*, *type*);

Description The GSET('LINREP', . . .)function assigns a color, width, and line type to a specific line bundle. The aspect source flags (ASF) for LINCOLOR, LINWIDTH, and LINTYPE must be set to 'BUNDLED' before the associated GDRAW function is executed if you want the bundled values to be used when the affected graphics element is drawn.

Argument Definitions

index numeric constant or numeric variable name; indicates the number for the bundle to use as an index. Valid values are 1 and 20, inclusive. If *index* is expressed as a variable, the variable must be initialized to a value between 1 and 20.

<i>color-index</i>	numeric constant or numeric variable name; specifies the index of the color to use. Valid values are 1 to 256, inclusive. The color index should represent one of the following: <ul style="list-style-type: none"> <input type="checkbox"/> a color index assigned with the GSET('COLREP', . . .)function <input type="checkbox"/> the <i>n</i>th color in the colors list of the COLORS= graphics option <input type="checkbox"/> the <i>n</i>th color in the device's default colors list.
<i>width</i>	numeric constant or numeric variable name; indicates the width of the line; must be greater than 0.
<i>type</i>	numeric constant or numeric variable name; indicates the type of line. Valid values are 1 to 46, inclusive. See Figure 8.22 on page 249 for representations of the different line types.

See Also

GSET('ASF', . . .)function
GSET('COLREP', . . .)function
GSET('LINCOLOR', . . .)function
GSET('LININDEX', . . .)function
GSET('LINREP', . . .)function
GSET('LINTYPE', . . .)function
GSET('LINWIDTH', . . .)function

LINTYPE

Specifies the line type

Operating States: GKOP, SGOP, WSAC, WSOP

Return Codes: 0, 8, 62

Default Value: 1

Syntax

return-code-variable=GSET('LINTYPE', *type*);

Description The GSET('LINTYPE', . . .)function selects a line type. See Figure 8.22 on page 249 for representations of the different line types. The aspect source flag (ASF) for LINTYPE must be set to 'INDIVIDUAL' for this attribute to be used outside of a line bundle.

Argument Definitions

type numeric constant or numeric variable name; indicates the type of line to use. Valid values are 1 to 46, inclusive.

See Also

GASK('LINTYPE', . . .)routine
 GSET('ASF', . . .)function
 GSET('LINREP', . . .)function

LINWIDTH

Specifies the thickness of the line

Operating States: GKOP, SGOP, WSAC, WSOP

Return Codes: 0, 8, 90

Default Value: 1

Syntax

return-code-variable=GSET('LINWIDTH', *width*);

Description The GSET('LINWIDTH', . . .)function selects a line width in units of pixels. The aspect source flag (ASF) for LINWIDTH must be set to 'INDIVIDUAL' for this attribute to be used outside of a line bundle.

Argument Definitions

width numeric constant or numeric variable name; specifies the width of the line in pixels; must be greater than 0.

See Also

GASK('LINWIDTH', . . .)routine
 GSET('ASF', . . .)function
 GSET('LINREP', . . .)function

MARCOLOR

Specifies the color index of the color used to draw markers

Operating States: GKOP, SGOP, WSAC, WSOP

Return Codes: 0, 8, 85

Default Value: 1

Syntax

return-code-variable=GSET('MARCOLOR', *color-index*);

Description The GSET('MARCOLOR', . . .)function selects the color index of the color used to draw markers. The aspect source flag (ASF) of MARCOLOR must be set to 'INDIVIDUAL' for this attribute to be used outside of a marker bundle.

DSGI searches for a color specification in the following order:

- 1 the color specified for the index in a GSET('COLREP', . . .)function
- 2 the *n*th color in the colors list of the COLORS= graphics option
- 3 the *n*th color in the device's default colors list found in the device entry.

Argument Definitions

color-index numeric constant or numeric variable name; indicates the index of the color to use. Valid values are 1 to 256, inclusive.

See Also

COLORS= graphics option (see "COLORS" on page 314)

GASK('MARCOLOR', . . .)routine

GSET('ASF', . . .)function

GSET('COLREP', . . .)function

GSET('MARREP', . . .)function

MARINDEX

Specifies the index of the bundle of marker attributes

Operating States: GKOP, SGOP, WSAC, WSOP

Return Codes: 0, 8, 64

Default Value: 1

Syntax

return-code-variable=GSET('MARINDEX', *index*);

Description The GSET('MARINDEX', . . .)function activates the marker bundle indicated by *index*. The aspect source flag (ASF) for MARCOLOR, MARTYPE, and MARSIZE must be set to 'BUNDLED' before the GDRAW('MARK', . . .)function is executed if you want the bundled values to be used when the marker is drawn.

Argument Definitions

index numeric constant or numeric variable name; the number of the bundle to activate. Valid values are 1 to 20, inclusive.

See Also

GASK('MARINDEX', . . .)routine

GSET('ASF', . . .)function

GSET('MARREP', . . .)function

MARREP

Associates a bundle of marker attributes with an index

Operating States: GKOP, SGOP, WSAC, WSOP

Return Codes: 0, 8, 64, 66, 85, 90

Default Value: none

Syntax

return-code-variable=GSET ('MARREP',*index*, *color-index*, *size*, *type*);

Description The GSET('MARREP', . . .)function assigns a color, size, and type of marker to a specific marker bundle. The aspect source flag (ASF) of MARCOLOR, MARSIZE, and MARTYPE must be set to 'BUNDLED' before the GDRAW('MARK', . . .)function is executed if you want the bundled values to be used when the marker is drawn.

Argument Definitions

<i>index</i>	numeric constant or numeric variable name; defines the bundle index number. Valid values are 1 to 20, inclusive.
<i>color-index</i>	numeric constant or numeric variable name; indicates the color index of the color to use. Valid values are 1 to 256, inclusive. The color index should represent one of the following: <ul style="list-style-type: none"> <input type="checkbox"/> a color index assigned to a color name with the GSET('COLREP', . . .)function <input type="checkbox"/> the <i>n</i>th color in the colors list of the COLORS= graphics option <input type="checkbox"/> the <i>n</i>th color in the device's default colors list.
<i>size</i>	numeric constant or numeric variable name; indicates the size of the marker in units of the current window system; must be greater than 0.
<i>type</i>	numeric constant or numeric variable name; specifies the type of marker to use; valid values are 1 to 67, inclusive. See Table 32.2 on page 1150 for a table of the symbols used for each marker type.

See Also

GSET('ASF', . . .)function
 GSET('COLREP', . . .)function
 GSET('MARCOLOR', . . .)function
 GSET('MARINDEX', . . .)function
 GSET('MARREP', . . .)function
 GSET('MARSIZE', . . .)function
 GSET('MARTYPE', . . .)function

MARSIZE

Selects the size of markers

Operating States: GKOP, SGOP, WSAC, WSOP

Return Codes: 0, 8, 90

Default Value: 1

Syntax

return-code-variable=GSET('MARSIZE', *size*);

Description The GSET('MARSIZE', . . .)function sets the marker size in units of the current window system. The aspect source flag (ASF) of MARSIZE must be set to 'INDIVIDUAL' for this attribute to be used outside of a marker bundle.

Argument Definitions

size numeric constant or numeric variable name; indicates the size of the marker in units of the current window system; must be greater than 0.

See Also

GASK('MARSIZE', . . .)routine
 GSET('ASF', . . .)function
 GSET('MARREP', . . .)function

MARTYPE

Selects the kind of markers

Operating States: GKOP, SGOP, WSAC, WSOP

Return Codes: 0, 8, 66

Default Value: 1

Syntax

return-code-variable=GSET('MARTYPE', *type*);

Description The GSET('MARTYPE', . . .)function determines the type of marker drawn. See Figure 8.21 on page 243 for representations of the symbols described in Table 32.2 on page 1150. The aspect source flag (ASF) of MARTYPE must be set to 'INDIVIDUAL' for this attribute to be used outside of a marker bundle.

Table 32.2 Symbol Indexes Used with DSGI

<i>Values and Markers</i>					
1	plus	24	K	46	9
2	x	25	L	47	lozenge
3	star	26	M	48	spade
4	square	27	N	49	heart
5	diamond	28	O	50	diamond
6	triangle	29	P	51	club
7	hash	30	Q	52	shamrock
8	Y	31	R	53	fleur-de-lis
9	Z	32	S	54	star
10	paw	33	T	55	sun
11	point	34	U	56	Mercury
12	dot	35	V	57	Venus
13	circle	36	W	58	Earth
14	A	37	0	59	Mars
15	B	38	1	60	Jupiter
16	C	39	2	61	Saturn
17	D	40	3	62	Uranus
18	E	41	4	63	Neptune
19	F	42	5	64	Pluto
20	G	43	6	65	moon
21	H	44	7	66	comet

Values and Markers

22	I	45	8	67	asterisk
23	J				

Argument Definitions

type numeric constant or numeric variable name; indicates the index of the marker to draw. Valid values are 1 to 67, inclusive. See Table 32.2 on page 1150 for value specifications.

See Also

GASK('MARTYPE', . . .)routine
 GSET('ASF', . . .)function
 GSET('MARREP', . . .)function

MESSAGE

Specifies whether the interface error message system is enabled or disabled

Operating States: All
 Return Codes: 0
 Default Value: ON

Syntax

return-code-variable=GSET('MESSAGE', *status*);

Description The GSET('MESSAGE', . . .)function activates or suppresses automatic error logging.

Argument Definitions

status character string enclosed in quotation marks or character variable name; indicates whether messages should be displayed. Valid values are ON and OFF. When ON is used, messages are automatically generated by the DSGI based on the return code from the function. If you set MESSAGE to OFF, no messages are automatically printed. You may choose to do this if you want to print custom messages for your application or decide which error message you want printed.

See Also

GDRAW('MESSAGE', . . .)function

GPRINT(*code*) utility function

TEXALIGN

Specifies the horizontal and vertical alignment of the text string

Operating States: GKOP, SGOP, WSAC, WSOP

Return Codes: 0, 8

Default values: *halign*=NORMAL, *valign*=NORMAL

Syntax

return-code-variable=GSET('TEXALIGN', *halign*, *valign*);

Description The GSET('TEXALIGN', . . .)function sets a particular type of horizontal and vertical alignment for text strings. Figure 32.5 on page 1152 illustrates *halign*.

Figure 32.5 Halign Values

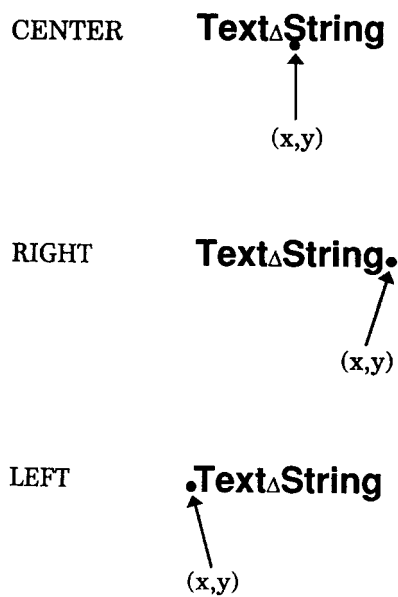
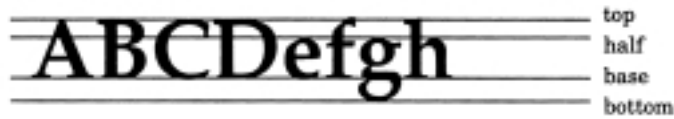


Figure 32.6 on page 1153 illustrates *valign*.

Figure 32.6 Valign Values

Argument Definitions

halign character string enclosed in quotation marks or character variable name. Valid values are

CENTER

LEFT

NORMAL (the natural alignment based on the text path); alignment is chosen according to the following logic:

- 1 If `TEXPATH` is 'RIGHT', then NORMAL is 'LEFT'.
- 2 Otherwise, if `TEXPATH` is 'LEFT', then NORMAL is 'RIGHT'.
- 3 Otherwise, the text string is centered.

RIGHT.

valign character string enclosed in quotation marks or character variable name. Valid values are

BASE (alignment based on the baseline of the text string)

BOTTOM (alignment based on the bottom of the text string)

HALF (alignment based on the vertical midpoint of the string)

NORMAL (natural alignment based on the text path); alignment is chosen according to the following logic:

- 1 If `TEXPATH` is 'RIGHT' or `TEXPATH` is 'LEFT', then NORMAL is 'BASE'.
- 2 Otherwise, if `TEXPATH` is 'UP', then NORMAL is 'BOTTOM'.
- 3 Otherwise, if `TEXPATH` is 'DOWN', then NORMAL is 'TOP'.

TOP (alignment based on the top of the string).

See Also

GASK("TEXALIGN", . . .)routine

GDRAW("TEXT", . . .)function

GSET("TEXPATH", . . .)function

GSET("TEXUP", . . .)function

TEXCOLOR

Specifies the color index of the color used to draw text strings

Operating States: GKOP, SGOP, WSAC, WSOP

Return Codes: 0, 8, 85

Default Value: 1

Syntax

return-code-variable=GSET("TEXCOLOR', *color-index*);

Description The GSET("TEXCOLOR', . . .)function selects the color for text. The aspect source flag (ASF) of TEXCOLOR must be set to 'INDIVIDUAL' for this attribute to be used outside of a text bundle.

The value of GSET("TEXCOLOR', . . .)can be used in a text bundle. See the GSET("TEXREP', . . .)function on page 1157 for information on how to define a text bundle.

DSGI searches for a color specification in the following order:

- 1 the color specified for the index in a GSET('COLREP', . . .)function
- 2 the *n*th color from the colors list of the COLORS= graphics option
- 3 the *n*th color in the device's default colors list found in the device entry.

Argument Definitions

color-index numeric constant or numeric variable name; indicates the color index of the color to be used. Valid values are 1 to 256, inclusive.

See Also

COLORS= graphics option (see "COLORS" on page 314)

GASK("TEXCOLOR', . . .)routine

GSET('ASF', . . .)function

GSET('COLREP', . . .)function

GSET("TEXREP', . . .)function

TEXFONT

Specifies the font used to draw text strings

Operating States: GKOP, SGOP, WSAC, WSOP

Return Codes: 0, 8

Default values: 1. FTEXT= graphics option, if specified; 2. hardware font, if possible; 3. SIMULATE font

Syntax

return-code-variable=GSET("TEXFONT", *font*);

Description The GSET("TEXTFONT", . . .)function selects a SAS/GRAPH font for the text. The aspect source flag (ASF) of TEXTFONT must be set to 'INDIVIDUAL' for this attribute to be used outside of a text bundle. See "Font Lists" on page 131 for a list of valid SAS/GRAPH fonts. You may also use fonts you have created using the GFONT procedure.

Argument Definitions

font character string enclosed in quotation marks or character variable name; the name of a font that can be accessed by SAS/GRAPH software. If you want to use the hardware font, submit

```
rc=gset('textfont', ' ');
```

When DSGI is used with long font names, the font name must be in double quotation marks that are embedded in single quotation marks, as in "HW font name".

See Also

FTEXT= graphics options (see "FTEXT" on page 335)
 GASK("TEXTFONT", . . .)routine
 GSET('ASF', . . .)function
 GSET("TEXTREP", . . .)function

TEXHEIGHT

Specifies the character height of the text string

Operating States: GKOP, SGOP, WSAC, WSOP

Return Codes: 0, 8, 73

Default Value: 1. HTEXT= graphics option, if specified; 2. 1 unit

Syntax

return-code-variable=GSET("TEXHEIGHT", *height*);

Description The GSET("TEXHEIGHT", . . .)function sets the height for text. GSET("TEXHEIGHT", . . .)affects text the same way as the HTEXT= graphics option.

Argument Definitions

height numeric constant or numeric variable name; indicates height in units based on the current window system; must be greater than 0.

See Also

GASK("TEXHEIGHT", . . .)routine
 HTEXT= graphics options (see "HTEXT" on page 357)

TEXINDEX

Specifies the index of the bundle of text attributes

Operating States: GKOP, SGOP, WSAC, WSOP

Return Codes: 0, 8, 68

Default Value: 1

Syntax

return-code-variable=GSET("TEXINDEX', *index*);

Description The GSET("TEXINDEX', . . .)function activates the text bundle indicated by *index*. The aspect source flag (ASF) for TEXCOLOR and TEXFONT must be set to 'BUNDLED' before the GDRAW("TEXT', . . .)function is executed if you want the bundled values to be used when the text is drawn.

Argument Definitions

index numeric constant or numeric variable name; indicates the number of the bundle to activate. Valid values are 1 to 20, inclusive.

See Also

GASK("TEXINDEX', . . .)routine

GSET('ASF', . . .)function

GSET("TEXREP', . . .)function

TEXPATH

Specifies the direction of the text string

Operating States: GKOP, SGOP, WSAC, WSOP

Return Codes: 0, 8

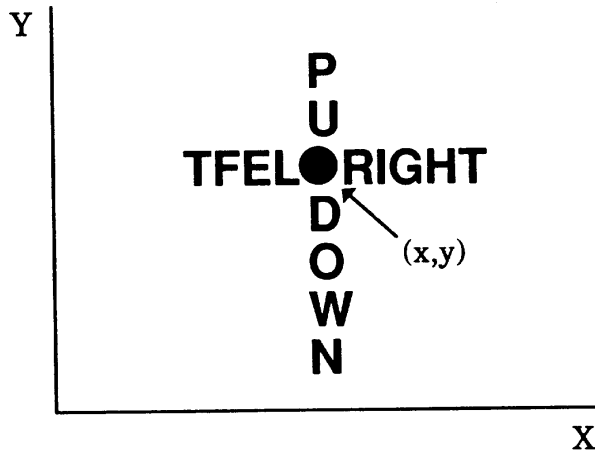
Default Value: RIGHT

Syntax

return-code-variable=GSET("TEXPATH', *path*);

Description The GSET("TEXPATH', . . .)function selects a particular type of text path. Text path determines the direction in which the text string reads. Figure 32.7 on page 1157 illustrates the text paths that can be used with DSGI.

Figure 32.7 TEXPATH Values



Argument Definitions

path character string enclosed in quotation marks or character variable name; specifies the direction in which the text will read. Valid values are

- DOWN
- LEFT
- RIGHT
- UP.

See Also

GASK("TEXPATH', . . .)routine
 GDRAW("TEXT", . . .)function
 GSET("TEXALIGN', . . .)function
 GSET("TEXUP', . . .)function

TEXREP

Associates a bundle of text attributes with an index

Operating States: GKOP, SGOP, WSAC, WSOP

Return Codes: 0, 8, 68, 85

Default Value: none

Syntax

return-code-variable=GSET ('TEXREP',*index*, *color-index*, *font*);

Description The GSET('TEXREP', . . .)function assigns a color and font to a particular text bundle. The aspect source flags (ASF) of TEXCOLOR and TEXFONT must be set to 'BUNDLED' before the GDRAW('TEXT', . . .)function is executed if you want the bundled values to be used when the text is drawn.

Argument Definitions

<i>index</i>	numeric constant or numeric variable name; specifies the number to use as an index for the bundle; valid values are 1 to 20, inclusive. If <i>index</i> is expressed as a variable, the variable must be initialized to a value between 1 and 20.
<i>color-index</i>	numeric constant or numeric variable name; indicates the color to use; valid values are 1 to 256, inclusive. The color index should represent one of the following: <ul style="list-style-type: none"> □ a color index assigned with the GSET('COLREP', . . .)function □ the <i>n</i>th color in the colors list of the COLORS= graphics option □ the <i>n</i>th color in the device's default colors list.
<i>font</i>	character string enclosed in quotation marks or character variable name; names the font to use with the bundle. See "Font Lists" on page 131 for a list of valid SAS/GRAPH fonts. You may also use fonts you have created using the GFONT procedure.

See Also

COLORS= graphics option (see "COLORS" on page 314)
 GASK('TEXREP', . . .)routine
 GSET('ASF', . . .)function
 GSET('COLREP', . . .)function
 GSET('TEXINDEX', . . .)function

TEXUP

Specifies the orientation (angle) of the text string

Operating States: GKOP, SGOP, WSAC, WSOP

Return Codes: 0, 8, 74

Default Values: *upx*=0, *upy*=1

Syntax

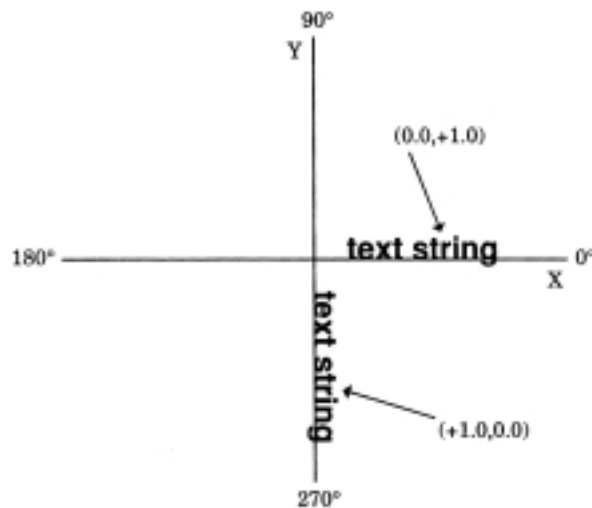
return-code-variable=GSET('TEXUP',*upx*, *upy*);

Description The GSET('TEXUP', . . .)function sets the angle of the text string. DSGI uses the values of character up vectors to determine the angle of a text string. The character up vector has two components, upx and upy , that describe the angle at which the text string is placed. The angle is calculated with the following formula:

$$\text{angle} = \text{atan}(upx/upy)$$

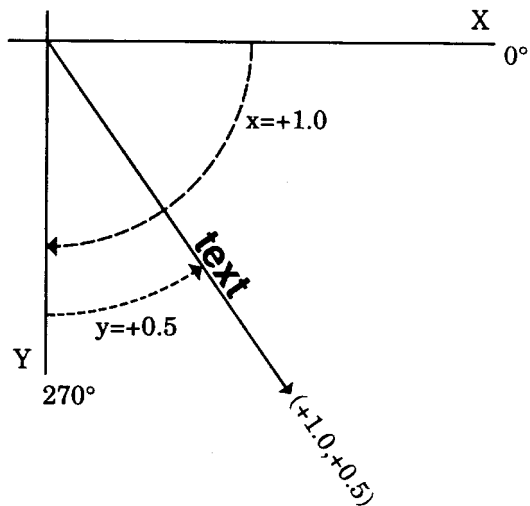
Effectively, when DSGI is calculating the angle for the text, it uses upx and upy as forces that are pushing the string toward an angle. The natural angle of text in the upx direction is toward the 6 o'clock position. In the upy direction, text naturally angles at the 3 o'clock position. If upx is greater than upy , the text is angled toward 6 o'clock. If upy is greater than upx , the text is angled toward 3 o'clock. Figure 32.8 on page 1159 shows the angle of text when the values for upx and upy are (0.0, 1.0) and (1.0, 0.0).

Figure 32.8 Natural Angle of Text



As you change the values of upx and upy , the coordinate that has the highest value is taken as the angle, and the lowest value as the offset. Figure 32.9 on page 1160 shows the angle of text when the character vector values (+1.0, +0.5) are used.

Figure 32.9 Varying the Angle of Text



You can use the following macro to convert angles measured in degrees to character up vectors:

```
%macro angle(x);
  if mod(&x, 180)=90 then do;
    if mod(&x,270) = 0 then
      xup = 1.0;
    else
      xup = -1.0;
    rc = gset('texup', xup, 0.0);
    end;
  else do;
    b = mod(&x, 360);
    /* adjust y vector for 2nd and 3rd quadrants */
    if b > 90 and b lt 270 then
      yup = -1.0;
    else
      yup = 1.0;
    a=&x*1.7453292519943300e-002;
    xup = tan(-a);
    /* adjust x vector for 3rd quadrant */
    if b > 180 and b le 270 then
      xup = -xup;
    rc = gset('texup', xup, yup);
    end;
%mend angle;

data _null_;
  rc = ginit();
  rc = graph('clear', 'angle');
  rc = gset('texalign', 'left', 'base');
  rc = gset('texheight', 5);
  rc = gset('texfont', 'swissl');
  %angle(180);
  rc = gdraw('text', 50, 50, '180');
  %angle(80);
```

```
rc = gdraw('text', 50, 50, '80');
%angle(600);
rc = gdraw('text', 50, 50, '600');
rc = graph('update');
rc = gterm();
run;
```

Argument Definitions

upx numeric constant or numeric variable name; if *upy* is 0, *upx* cannot be 0.

upy numeric constant or numeric variable name; if *upx* is 0, *upy* cannot be 0.

See Also

GASK("TEXUP", . . .)routine
 GDRAW("TEXT", . . .)function
 GSET("TEXALIGN", . . .)function
 GSET("TEXPATH", . . .)function

TRANSNO

Specifies the number of the transformation to be used

Operating States: GKOP, SGOP, WSAC, WSOP

Return Codes: 0, 8, 50

Default Value: 0

Syntax

return-code-variable=GSET("TRANSNO", *n*);

Description The GSET("TRANSNO", . . .)function activates the viewport and/or window you have defined for the specified transformation number. If you have not defined both a viewport and window for a transformation, the default is used for the one missing.

You can select 0 as the active transformation, but you cannot define a viewport or window for that transformation number. A transformation of 0 activates the default viewport, (0,0) to (1,1), and window, which is device dependent.

Argument Definitions

n numeric constant or numeric variable name; indicates the viewport and/or window to activate; should correspond to the *n* used in the GSET("VIEWPORT", . . .)and/or GSET("WINDOW", . . .)functions. Valid values are 0 to 20, inclusive.

See Also

GASK('TRANS', . . .)routine
 GASK('TRANSNO', . . .)routine
 GASK('VIEWPORT', . . .)routine
 GASK('WINDOW', . . .)routine
 GSET('VIEWPORT', . . .)function
 GSET('WINDOW', . . .)function

VIEWPORT

Associates a viewport with a transformation number

Operating States: GKOP, SGOP, WSAC, WSOP

Return Codes: 0, 8, 50, 51, 52

Default Values: *llx=0, lly=0, urx=1, ury=1*

Syntax

return-code-variable=GSET('VIEWPORT', *n*, *llx*, *lly*, *urx*, *ury*);

Description The GSET('VIEWPORT', . . .)function defines a viewport and associates it with the transformation number, *n*. See the GSET('TRANSNO', . . .)function on page 1161 for information on how to activate the viewport. See the GSET('WINDOW', . . .)function on page 1164 for information on how to define a window to be used within the viewport.

Argument Definitions

<i>n</i>	numeric constant or numeric variable name; specifies the transformation number of the viewport. Valid values are 1 to 20, inclusive.
<i>llx</i>	numeric constant or numeric variable name; defines the <i>x</i> component of the lower-left corner of the viewport; must not exceed the value of <i>urx</i> ; cannot be less than 0. Units are based on percent of the graphics output area.
<i>lly</i>	numeric constant or numeric variable name; defines the <i>y</i> component of the lower-left corner of the viewport; must not exceed the value of <i>ury</i> ; cannot be less than 0. Units are based on percent of the graphics output area.
<i>urx</i>	numeric constant or numeric variable name; defines the <i>x</i> component of the upper-right corner of the viewport; cannot be greater than 1. Units are based on percent of the graphics output area.
<i>ury</i>	numeric constant or numeric variable name; defines the <i>y</i> component of the upper-right corner of the viewport; cannot be greater than 1. Units are based on percent of the graphics output area.

See Also

GASK('VIEWPORT', . . .)routine
 GSET('WINDOW', . . .)function
 GSET('TRANSNO', . . .)function
 GASK('TRANSNO', . . .)routine
 GASK('TRANS', . . .)routine
 GASK('WINDOW', . . .)routine

VPOS

Specifies the number of rows

Operating States: GKCL

Return Codes: 0, 1, 90, 307

Default Values: 1. VPOS=graphics option, if specified; 2. device's default VPOS value

Syntax

return-code-variable=GSET('VPOS', *vpos*);

Description The GSET('VPOS', . . .)function sets the number of rows in the graphics output area. GSET('VPOS', . . .)has the same effect on graphics output as the VPOS=graphics option.

You can reset the VPOS value by submitting one of the following statements:

```
goptions reset=goptions;
goptions reset=all;
```

```
goptions vpos=0;
```

Argument Definitions

vpos numeric constant or numeric variable name; specifies the number of rows in the graphics output area; must be greater than 0.

See Also

GASK('VPOS', . . .)routine
 GSET('HPOS', . . .)function
 GSET('VSIZE', . . .)function
 VPOS= graphics option (see "VPOS" on page 397)

VSIZE

Specifies the vertical dimension of the graphics output area

Operating States: GKCL

Return Codes: 0, 1, 90, 307

Default Values: 1. VSIZE= graphics option, if specified; 2. device's default VSIZE value

Syntax

return-code-variable=GSET('VSIZE', *vsize*);

Description The GSET('VSIZE', . . .)function sets the vertical dimension, in inches, of the graphics output area. GSET('VSIZE', . . .)affects the dimensions of the default window.

You can reset the VSIZE value by submitting one of the following statements:

```
goptions reset=goptions;
goptions reset=all;
goptions vsize=0;
```

Argument Definitions

vsize numeric constant or numeric variable name; indicates the vertical dimension for the graph in inches; must be greater than 0.

See Also

GASK('VSIZE', . . .)routine

GSET('HSIZE', . . .)function

GSET('VPOS', . . .)function

VSIZE= graphics option (see "VSIZE" on page 398)

WINDOW

Associates a window with a transformation number

Operating States: GKOP, SGOP, WSAC, WSOP

Return Codes: 0, 8, 50, 51

Default Values: *llx*=0, *lly*=0; *urx* and *ury* are device dependent

Syntax

return-code-variable=GSET ('WINDOW', *n*, *llx*, *lly*, *urx*, *ury*);

Description The GSET('WINDOW', . . .)function defines a window and associates it with a transformation number. See the GSET('TRANSNO', . . .)function on page 1161 for information on how to activate a window. See the GSET('VIEWPORT', . . .)function on page 1162 for information on how to define a viewport for a window.

Argument Definitions

<i>n</i>	numeric constant or numeric variable name; specifies the transformation number of the window. Valid values are 1 to 20, inclusive.
<i>llx</i>	numeric constant or numeric variable name; defines the <i>x</i> component of the lower-left corner of the window; must not exceed the value of <i>urx</i> . Units are based on percent of the active viewport.
<i>lly</i>	numeric constant or numeric variable name; defines the <i>y</i> component of the lower-left corner of the window; must not exceed the value of <i>ury</i> . Units are based on percent of the active viewport.
<i>urx</i>	numeric constant or numeric variable name; defines the <i>x</i> component of the upper-right corner of the window. Units are based on percent of the active viewport.
<i>ury</i>	numeric constant or numeric variable name; defines the <i>y</i> component of the upper-right corner of the window. Units are based on percent of the active viewport.

See Also

GASK('TRANS', . . .)routine
 GASK('TRANSNO', . . .)routine
 GASK('VIEWPORT', . . .)routine
 GASK('WINDOW', . . .)routine
 GSET('TRANSNO', . . .)function
 GSET('VIEWPORT', . . .)function

Return Codes for DSGI Routines and Functions

0	Function completed successfully.
1	DATA Step Graphics Interface should be in GKCL state; the statement is out of place within the DATA step.
3	DATA Step Graphics Interface should be in WSAC state; the statement is out of place within the DATA step.
4	DATA Step Graphics Interface should be in SGOP state; the statement is out of place within the DATA step.
7	DATA Step Graphics Interface should be in WSOP, WSAC, or SGOP state; the statement is out of place within the DATA step.
8	DATA Step Graphics Interface should be in GKOP, WSOP, WSAC, or SGOP state; the statement is out of place within the DATA step.
24	Workstation is open.
25	Workstation is not open.

26	Workstation cannot be opened.
29	Workstation is active.
30	Workstation is not active.
50	Invalid transformation number; transformation numbers must be in the range 0 to 20; viewports and windows cannot be defined for transformation 0.
51	Transformation is not a well-defined rectangle; transformations must have coordinates for four vertices.
52	Viewport coordinates are out of range; coordinates must be within dimensions of graphics output area for the device.
55	Clipping is on.
56	Clipping is off.
60	Bad line index; index numbers must be in the range 1 to 20.
61	No bundle defined for the line index; a GSET('LINREP', . . .)function has not been submitted for the referenced line index.
62	Line type is less than or equal to 0 or greater than 46; type must be in the range 1 to 46.
64	Invalid marker index; index numbers must be in the range 1 to 20.
65	No bundle defined for the polymarker index; a GSET('MARREP', . . .)function has not been submitted for the referenced marker index.
66	Marker type is less than or equal to 0 or greater than 67; type must be in the range 1 to 67.
68	Invalid text index; index numbers must be in the range 1 to 20.
69	No bundle defined for the text index; a GSET('TEXREP', . . .)function has not been submitted for the referenced text index.
73	Character height is less than or equal to 0; height must be greater than 0.
74	Both components of the character up vector are 0; both X and Y of a character up vector cannot be 0.
75	Invalid fill index; index numbers must be in the range 1 to 20.
76	No bundle defined for the fill index; a GSET('FILREP', . . .)function has not been submitted for the referenced fill index.
78	Style index is less than or equal to 0 or greater than 60; style indexes must be in the range of 1 to 60.
86	Invalid color index; color index is out of the range 1 to 256 or is not numeric.
87	No color name defined for the color index
90	Value is less than 0; value must be greater than or equal to 0.
301	Out of memory; your workstation does not have enough memory to generate the graph.
302	Out of room for graph; your device cannot display the size of the graph.

307 Error occurred in program library management; a GRAPH function did not execute properly.

See Also

- Chapter 3, “Device Drivers,” on page 37
for information about specifying device drivers.
- Chapter 9, “Graphics Options and Device Parameters Dictionary,” on page 301
for descriptions of graphics options and device parameters
- Chapter 6, “SAS/GRAPH Fonts,” on page 125
for information about the fonts available in SAS/GRAPH software
- Chapter 7, “SAS/GRAPH Colors,” on page 139
for information about specifying colors in SAS/GRAPH programs
- “GOPTIONS Statement” on page 182
for an explanation of setting graphics options with the GOPTIONS statement
- “PATTERN Statement” on page 211
for information about specifying patterns with DSGI
- “SYMBOL Statement” on page 226
for representations of the markers that can be used with DSGI
- Chapter 31, “The DATA Step Graphics Interface,” on page 1027
for a complete explanation of using DSGI statements to produce graphs
- Chapter 15, “The GDEVICE Procedure,” on page 651
for information about device entries
- The discussion for ARRAY in *SAS Language Reference: Dictionary*
for an explanation of OF argument lists

References

- Enderle, G.; Kansy, K.; and Pfaff, G. (1985), *Computer Graphics Programming: GKS—The Graphics Standard* Springer-Verlag New York, Inc.

Glossary

absolute coordinates

coordinates measured from the origin of the coordinate system. In two-dimensional graphs, the origin is (0,0). In three-dimensional graphs, the origin is (0,0,0). See also relative coordinates.

aspect ratio

the ratio of width to height (that is, width divided by height) in an output area such as a display, plotter, or film recorder. In SAS/GRAPH software, the ASPECT= graphics option simulates a change in the aspect ratio of the display, causing fonts and circles to be compressed horizontally or vertically or both.

axis

a one-dimensional line representing the zero point on a scale used to plot values of x , y , or z coordinates. In SAS/GRAPH software, in two dimensions, the X axis represents the horizontal plane, and the Y axis represents the vertical plane. In three dimensions, the X axis represents width, the Y axis represents depth, and the Z axis represents height. See also Cartesian coordinate system. The term *axis* may also refer collectively to the axis line, the major and minor tick marks, the major tick mark values, and the axis label.

axis area

an area bounded by axes. In SAS/GRAPH software, this area may be enclosed by an axis frame. See also frame.

baseline

in a font, the imaginary line upon which the characters rest.

block map

a three-dimensional map that uses blocks of varying heights to represent the value of a variable for each map area.

border

in SAS/GRAPH software, the line drawn around the entire graphics output area. This area includes the title and footnote areas as well as the procedure output area. See also frame.

boundary

in the GMAP procedure, a separating line or point that distinguishes between two or more unit areas or segments.

BY group

all observations with the same values for all BY variables.

BY-group processing

the process of using the BY statement to process observations that are ordered, grouped, or indexed according to the values of one or more variables. Many SAS procedures and the DATA step support BY-group processing.

BY variable

a variable named in a BY statement whose values define groups of observations to process.

Cartesian coordinate system

the two- or three-dimensional coordinate system in which perpendicular axes meet at the origin (0,0) or (0,0,0). Typically, Cartesian coordinate axes are called X, Y, and Z. See also axis.

Cartesian coordinates

values that locate a point in two- or three-dimensional space. Each value represents units measured along an X, Y, or Z axis. See also Cartesian coordinate system.

capline

the highest point of a normal uppercase letter. In some fonts, the capline may be above the top of the letter to allow room for an accent.

catalog

See SAS catalog.

catalog entry

See entry type and SAS catalog entry.

cell

a unit of measure defined by the number of rows and the number of columns in the graphics output area. See also aspect ratio.

CGM

an abbreviation for computer graphics metafile. A CGM is a graphics output file written in the internationally recognized format for describing computer graphics images. This standardization allows any image in a CGM to be imported and exported among different systems without error or distortion.

character string

one or more alphanumeric or other keyboard characters or both.

character value

a value that can contain alphabetic characters, numeric characters 0 through 9, and other special characters. See also character variable.

character variable

a variable whose values can consist of alphabetic and special characters as well as numeric characters.

chart

a graph in which graphics elements (bars, pie slices, and so on) show the magnitude of a statistic. The graphics elements can represent one data value or a range of data values.

chart statistic

the statistical value calculated for the chart variable: frequency, cumulative frequency, percentage, cumulative percentage, sum, or mean.

chart variable

a variable in the input data set whose values are categories of data represented by bars, blocks, slices, or spines.

choropleth map

a two-dimensional map that uses color and fill pattern combinations to represent different categories or levels of magnitude.

class variable

in some SAS procedures, a variable used to group, or classify, data. Class variables can be character or numeric. Class variables can have continuous values, but they typically have a few discrete values that define the classifications of the variable.

classification variable

See class variable.

CMYK

a color-coding scheme that specifies a color in terms of levels of cyan, magenta, yellow, and black components. The levels of each component range from 0 to 255. See also HLS, HSV, and RGB.

color map

a table that is used to translate the original colors in graphics output to different colors when replaying graphics output using the GREPLAY procedure. The table is contained in a catalog entry.

color, predefined

one of the set of colors for which SAS/GRAPH software defines and recognizes names, for example, BLACK, BLUE, and CYAN.

color, user-defined

a color expressed in CMYK, HLS, HSV, RGB, or gray-scale format. See also CMYK, HLS, HSV, RGB, and gray scale.

colors list

the list of foreground colors available for the graphics output. The colors list is either the default list established from the device entry or the list established from the colors specified with the COLORS= graphics option.

computer graphics metafile

See CGM.

confidence limits

the upper and lower values of a confidence interval. There is a percentage of confidence (typically 95%) that the true value of the parameter being estimated lies within the interval.

contour plot

a three-variable plot that uses line styles or patterns to represent levels of magnitude of z corresponding to x and y coordinates.

coordinate system

the context in which to interpret coordinates. Coordinate systems vary according to their origin, limits, and units. See also Cartesian coordinate system.

coordinates

the values representing the location of a data point or a graphics element along the X, Y, and Z axes. Coordinate values are measured from the origin of the coordinate system.

data area

the portion of the graphics output area in which data values are displayed. In the Annotate facility, the data area defines a coordinate system. In plots and bar charts, the data area is bounded by axes; in choropleth maps, the data area is bounded by the edge of the unit areas. See also graphics output area, procedure output area, and coordinate system.

data value

a unit of character or numeric information in a SAS data set. A data value represents one variable in an observation. In the rectangular structure of a SAS data set, intersection of a row and a column.

date value

See SAS date value.

default

(1) The setting of a value, parameter, or argument used by the SAS System if the user does not specify a setting.

(2) the value, parameter, or option setting used by the SAS System if the user specifies no particular setting.

density value

a value assigned to each observation in a map data set reflecting the amount of detail (resolution) contributed by the observation.

dependent variable

a variable whose value is determined by the value of another variable or set of variables.

device driver

a routine that generates the specific machine-language commands needed to display graphics output on a particular device. SAS/GRAPH device drivers take device-independent graphics information produced by SAS/GRAPH procedures and create the commands required to produce the graph on the particular device.

device entry

a SAS catalog entry that stores the values of device parameters (or the characteristics) that are used with a particular output device.

device map

a catalog entry used to convert the SAS/GRAPH internal encoding for one or more characters to the device-specific encoding needed to display the character(s) in hardware text on a particular graphics output device. See also hardware character set.

device parameter

a value in a device entry that defines a default behavior or characteristic of a device driver. Some device parameters can be overridden by graphics options. See also graphics option.

display

the area of the monitor that displays what the software presents to you.

display manager

See SAS Display Manager System.

entry type

a characteristic of a SAS catalog entry that identifies its structure and attributes to the SAS System. When you create an entry, the SAS System automatically assigns the entry type as part of the name.

export

to put a SAS catalog entry containing graphics output into a format that can be moved to another software product.

fileref

a name temporarily assigned to an external file or to an aggregate storage location that identifies it to the SAS System. You assign a fileref with a FILENAME statement or with an operating system command.

fill pattern

a design of parallel or crosshatched lines, solid colors, or empty space used to fill an area in a graph.

font

a complete set of all the characters of the same design and style. The characters in a font can be figures or symbols as well as alphanumeric characters. See also type style.

font maximum

in the GFONT procedure, the highest vertical coordinate in a font.

font minimum

in the GFONT procedure, the lowest vertical coordinate in a font.

font units

in the GFONT procedure, units defined by the range of coordinates specified in the font data set. For example, a font in which the vertical coordinates range from 10 to 100 has 90 font units.

font, hardware

a font stored in an output device. See also font, software.

font, software

a font in which the characters are drawn by graphics software. See also font, hardware.

format

an instruction the SAS System uses to display or write each value of a variable. Some formats are supplied by SAS software. Other formats can be written by the user with the FORMAT procedure in base SAS software or with SAS/TOOLKIT software.

frame

a box enclosing a group of graphics elements. In GSLIDE procedure output, the frame encloses the procedure output area. In GPLOT, GCHART, and GCONTOUR procedure output, the frame encloses the axis area. In a legend, the frame encloses the legend label and entries. See also border.

global statement

a SAS statement that you can specify anywhere in a SAS program.

graph

a visual representation of data showing the variation of a variable in comparison to one or more other variables.

graphics element

a discrete visual part of a picture. For example, a bar in a chart and a plot's axis label are both graphics elements.

graphics device

See graphics output device.

graphics option

a value specified in a GOPTIONS statement that controls some attribute of the graphics output. The values specified remain in effect only for the duration of the SAS session. Some graphics options override device parameters.

graphics output

output from a graphics program that can be stored as a catalog entry of type GRSEG, or as a graphics stream file. Graphics output can be displayed or printed on a graphics output device. See also graphics output device and graphics stream file (GSF).

graphics output area

the area of a graphics output device where the graphics output is displayed or drawn. Typically, the graphics output area occupies the full drawing area of the device, but the dimensions of the graphics output area can be changed with graphics options or device parameters. See also procedure output area and graphics output device.

graphics output device

any terminal, printer, or other output device capable of displaying or producing graphics output. See also graphics output.

graphics stream file (GSF)

a file containing device-dependent graphics commands from a SAS/GRAPH device driver. This file can be sent to a graphics device or to other software packages.

gray scale

a color-coding scheme that specifies a color in terms of gray components. Gray-scale color codes are commonly used with some laser printers and PostScript devices.

grid request

in the G3GRID procedure, the request specified in a GRID statement that identifies the horizontal variables that identify the x - y plane and one or more z variables for the interpolation.

group variable

a variable in the input data set used to categorize chart variable values into groups.

GSF

See graphics stream file (GSF).

HLS

a color-coding scheme that specifies a color in terms of its hue, lightness, and saturation components. Hue is the color, lightness is the percentage of white, and saturation is the attribute of a color that determines its relative strength and its departure from gray. Lightness and saturation added to the hue produce a specific shade. See also CMYK, HSV, and RGB.

HSV (or HSB)

a color-coding scheme that specifies a color in terms of its hue, saturation, and value (or brightness) components. Hue is the color, saturation is the attribute of a color that determines its relative strength and its departure from gray, and value or brightness is its departure from black. See also CMYK, HLS, and RGB.

identification variable

a variable common to both the map data set and the response data set that the GMAP procedure uses to associate each pair of map coordinates and each response value with a unique map area.

import

(1) to read a computer graphics metafile (CGM) and store the graphics output in a SAS catalog. Use the GIMPORT procedure to import the CGM. (2) to restore a SAS transport file to its original form (a SAS data library, a SAS catalog, or a SAS data set) in the format appropriate to the host operating system. Use the CIMPORT procedure to import a SAS transport file created by the CPORT procedure.

independent variable

a variable that does not depend on the value of another variable; in a two-dimensional plot, the independent variable is usually plotted on the x (horizontal) axis.

interpolate

to estimate values between two or more known values.

justify

to position text in relation to the left or right margin or the center of the line.

key map

a SAS catalog entry used to translate the codes generated by the keys on a keyboard into their corresponding SAS/GRAPH internal character encoding. See also device map.

label

(1) in the AXIS and LEGEND statements and GPLOT and GCHART procedures, the text that names the variable associated with an axis, a legend, or a bubble in a bubble plot. By default, this text is the name of a variable or of a label previously assigned with a LABEL statement. The text of a label also can be specified with the LABEL= option. (2) in special cases of pie charts and star charts in the GCHART procedure, the midpoint value and the value of the chart statistic for a slice or spine. (3) in the Annotate facility, the text displayed by the LABEL function or macro.

latitude

the angular measure between the equator and the circle of parallel on which a point lies.

legend

refers collectively to the legend value, the legend value description, the legend label, and the legend frame.

libref

the name temporarily associated with a SAS data library. For example, in the name SASUSERS.ACCOUNTS, the name SASUSER is the libref. You assign a libref with a LIBNAME statement or with operating system control language. See also first-level name.

longitude

the angular measure between the reference meridian and the plane intersecting both poles and a point. The reference meridian, called the prime meridian, is assigned a longitude of 0, and other longitude values are measured from there in appropriate angular units (degrees or radians, for example).

major tick marks

the points on an axis that mark the major divisions of the axis scale. See also minor tick marks.

map

a graphic representation of an area, often a geographic area, but also any other area of any size. See also device map and key map.

map area

a polygon or group of polygons on a map, for example, a state, province, or country. In a map data set, a map area consists of all the observations with the same values for the identification variable or variables. A map area and a unit area are the same things. See also identification variable.

map data set

a SAS data set that contains information the GMAP procedure uses to draw a map. Each observation contains variables whose values are the x,y coordinates of a point on the boundary of a map area, and an identification variable whose value identifies the map area to which the point belongs.

meridian

an imaginary circle of constant longitude around the surface of the earth perpendicular to the equator. See also parallel.

midpoint

a value that represents one data value or the middle of a range of data values. When a midpoint represents a range of values, the algorithm used to calculate it depends on the procedure.

minor tick marks

the divisions of the axis scale that fall between major tick marks. See also major tick marks.

needle plot

a plot in which a vertical line connects each data point to the horizontal axis (two dimensions) or the horizontal plane (three dimensions).

numeric variable

a variable that can contain only numeric values. By default, the SAS System stores all numeric variables in floating-point representation.

observation

a row in a SAS data set. An observation is a collection of data values associated with a single entity, such as a customer or state. Each observation contains one data value for each variable. See also variable.

offset

(1) in a legend, the distance between the edge of the legend or the edge of the legend frame and the axis frame or the border surrounding the graphics output area. (2) on an axis, the distance from the origin to either the first major tick mark or the midpoint of the first bar, or the distance from the last major tickmark or the midpoint of the last bar to the end of the axis.

origin

(1) in a three-dimensional coordinate system, the point at which the X, Y, and Z axes intersect, defined by the coordinates (0,0,0). In a two-dimensional coordinate system, the point at which the X and Y axes intersect, defined by the coordinates (0,0). (2) in the AXIS statement, the origin is the point at which the axis line begins (the left end of the horizontal axis or the bottom of the vertical axis). In the LEGEND statement, the origin is the location of the lower-left corner of the legend. (3) in the graphics output area, the lower-left corner.

palette

the range of colors that can be generated on a graphics device. See also colors list.

panel

in the GREPLAY procedure, a part of the template in which one or more pictures can be displayed. A template can contain one or more panels.

parallel

an imaginary circle of constant latitude around the surface of the earth parallel to the equator. See also meridian.

pattern type

the set of fill patterns that are valid for a particular type of graph. The PATTERN statement supports three pattern types: bar and block patterns, map and plot patterns, and pie and star patterns. See also fill pattern.

pen mounts

on a pen plotter, the holders for the drawing pens.

pie chart

a chart made up of a circle divided by radial lines used to display the relative contribution of each part to the whole.

plot

a graph showing the relationship between variables. The coordinates of each point on the graph represent the values you plot. See also coordinates.

plot line

the line joining the data points in a plot.

plotter

a class of graphics devices that typically use pens to draw hardcopy output.

polygon

a closed, geometric figure bounded by lines or arcs. Polygons can be filled in to represent a surface.

polygon font

a font in which the characters are drawn with enclosed areas that can be filled or empty. See also stroked font.

prism map

a three-dimensional map that uses prisms (polyhedrons with two parallel surfaces) of varying height to indicate the ordinal magnitude of a response variable.

procedure output area

the portion of the graphics output area where the output from a graphics procedure is displayed. See also graphics output area and data area.

projection

a two-dimensional map representation of unit areas on the surface of a sphere, for example, geographic regions on the surface of the Earth.

regression analysis

an analysis of the nature of the relationship between two or more variables, expressed as a mathematical function. On a scatter plot, this relationship is diagrammed as a line drawn through data points. A straight line indicates simple regression; a curve indicates a higher-order regression.

relative coordinates

the coordinates measured from a point other than the origin, usually the endpoint of the last graphics element drawn. See also absolute coordinates.

relative move

a move that repositions the graphics element by a specified distance from its current location. See also absolute move.

replay

to display graphics output that is stored in a catalog entry.

response data set

a SAS data set the GMAP procedure uses that contains data values associated with map areas and one or more identification variables. See also identification variable, response values, and response variable.

response levels

the individual values or ranges of values into which the GMAP or GCHART procedure divides the response variable. See also midpoint.

response values

values of a response variable that the GMAP procedure represents on a map as different pattern/color combinations, or as raised map areas (prisms), spikes, or blocks of different heights. The GCHART procedure represents response values as bars, slices, spines, or blocks. See also midpoint.

response variable

the SAS data set variable in a response data set the GMAP procedure uses that contains data values associated with a map area. Response variables used by the GCHART procedure contain data values associated with bars, slices, spines, or blocks. See also chart variable, response data set, response levels, and response values.

RGB

a color-coding scheme that specifies a color in terms of levels of red, green, and blue components. The levels of each component range from 0 to 255. See also CMYK, HLS, and HSV.

rotate

in the graphics editor, to turn a graphics object about its axis.

RUN group

in SAS procedures, a set of statements ending with a RUN statement.

SAS catalog

a SAS file that stores many different kinds of information in smaller units called catalog entries. A single SAS catalog can contain several different types of catalog entries.

SAS catalog entry

a separate storage unit within a SAS catalog. Each entry has an entry type that identifies its purpose to the SAS System. See also entry type.

SAS data library

a collection of one or more SAS files that are recognized by the SAS System and that are referenced and stored as a unit. Each file is a member of the library.

SAS data set

descriptor information and its related data values organized as a table of observations and variables that can be processed by the SAS System. A SAS data set can be either a SAS data file or a SAS data view.

SAS date value

an integer representing a date in the SAS System. The integer represents the number of days between January 1, 1960, and another specified date. (For example, the SAS date value 366 represents the calendar date January 1, 1961.)

SAS Display Manager System

an interactive, windowing interface to SAS System software. Display manager commands can be issued by typing them on the command line, pressing function keys, or selecting items from the PMENU facility. Within one session, many different tasks can be accomplished, including preparing and submitting programs, viewing and printing results, and debugging and resubmitting programs.

scatter plot

a two- or three-dimensional plot showing the joint variation of two (or three) variables from a group of observations. The coordinates of each point in the plot correspond to the data values for a single observation.

segment

in the GMAP procedure, a polygon that is a part of a unit area consisting of more than one polygon. For example, consider a map of Hawaii. The representation of the single unit area (the state) consists of a group of individual segments (the islands), each of which is a separate polygon. In the GFONT procedure, a segment is a single continuous line that forms part of all of a character or symbol.

software font

See font, software.

spine

a line on a star chart used to represent the relative value of the chart statistic for a midpoint. Spines are drawn outward from the center of the chart.

spline

a method of interpolation in which a smooth line or surface connects data points.

standard deviation

a statistical measure of the variability of a group of data values. This measure, which is the most widely used measure of the dispersion of a frequency distribution, is equal to the positive square root of the variance.

string

See character string.

stroked font

a font in which the characters are drawn with discrete line segments or circular arcs. See also polygon font.

subgroup variable

the variable in the input data set for a chart that is used to proportionally fill areas of the bars or blocks on the chart.

summary variable

a variable in an input data set whose values the GCHART procedure totals or averages to produce the sum or mean statistics, respectively.

surface map

a three-dimensional map that uses spikes of varying heights to indicate levels of relative magnitude.

surface plot

a three-dimensional graph that displays a grid-like surface formed by the values of the vertical (Z) variable plotted on a plane specified by the X and Y variables.

template

in the GREPLAY procedure, a framework that enables you to display one or more pictures on a page.

text string

See character string.

tilt angle

the measure in degrees from the horizontal axis to the major axis of an object.

type style

a typeface design and its variations, for example, Swiss, Swiss Bold, and Swiss Italic. See also font.

unit area

See map area.

user-definable colors

the colors that can be defined using SAS color names, or CMYK (cyan, magenta, yellow, black), RGB (red, green, blue), HLS (hue, lightness, saturation), HSV (hue, saturation, value), or gray-scale color equivalents.

value

the text that labels a major tick mark on an axis. Also, in a legend, a value is a line, bar, or shape that the legend explains.

variable

a column in a SAS data set. A variable is a set of data values that describe a given characteristic across all observations. See also macro variable.

variable type

the classification of a variable as either numeric or character. Type is an attribute of SAS variables.

WORK data library

the SAS data library automatically defined by the SAS System at the beginning of each SAS session or SAS job. It contains SAS files that are temporary by default. When the libref USER is not defined, the SAS System uses WORK as the default library for SAS files created with one-level names.

X axis

in a two-dimensional plot, the horizontal axis. In a three-dimensional plot, the X axis is the axis perpendicular to the Y-Z plane.

Y axis

in a two-dimensional plot, the vertical axis. In a three-dimensional plot, the Y axis is the axis perpendicular to the X-Z plane.

Z axis

in a three-dimensional plot, the axis perpendicular to the X-Y plane.

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.