60456360

CONTROL DATA CORPORATION

TERMINAL-INDEPENDENT GRAPHICS SYSTEM VERSION 1.1 INSTANT

CDC® COMPUTER SYSTEMS:
6000 SERIES
CYBER 70
MODELS 71, 72, 73, 74
CYBER 170 SERIES

60456360

CONTROL DATA CORPORATION

TERMINAL-INDEPENDENT GRAPHICS SYSTEM VERSION 1.1 INSTANT

CDC® COMPUTER SYSTEMS:
6000 SERIES
CYBER 70
MODELS 71, 72, 73, 74
CYBER 170 SERIES

REVISION RECORD	
REV	DESCRIPTION
A	Manual released.
(08-04-78)	
B	Manual revised to add version 1.1 features
(07-20-79)	and to make miscellaneous corrections.
	This edition obsoletes all previous
	editions.

Publication No.
60456360

Revision letters I, O, Q, and X are not used.

Address comments to:

Control Data Corporation
Publications and Graphics
4201 N. Lexington Avenue
St. Paul, Minnesota 55112

LIST OF EFFECTIVE PAGES

New features, as well as changes, deletions, and additions to information in this manual, are indicated by bars in the margins or by a dot near the page number if the entire page is affected. A bar by the page number indicates pagination rather than content has changed.

PAGE	REV
Front Cover	–
Title Page	–
ii	B
iii/iv	B
v/vi	B
vii	B
1	A
2	B
3	B
4	B
5	B
6	B
7	B
8	B
9	B
10	B
11	B
12	B
13	B
14	B
15	B
16	B
17	B
18	B
19	B
20	B
21	B
22	B
23	B
24	B
25	B
26	B
27	B
28	B
29	B
30	B
31	B

PAGE	REV
32	B
33	B
34	B
35	B
36	B
37	B
38	B
39	B
40	B
41	B
42	B
43	B
44	B
45	B
46	B
47	B
48	B
Back Cover	–

PREFACE

This instant manual provides calling formats, parameter descriptions, default values for all calling sequences, and error messages of the CDC® Terminal-Independent Graphics System (TIGS).

This manual is intended for use as a reference source for all users of TIGS. A familiarity with FORTRAN is assumed.

For further information on material in this manual, refer to the Control Data Terminal-Independent Graphics System (TIGS) Reference Manual, publication number 60455940.

CONTENTS

TIGS ROUTINES

The following are the user-level calling sequences for the Terminal-Independent Graphics System (TIGS). The routines are listed by type (primitives, interaction, and so on) with their parameters and a description of what function is performed.

TIGS CALLING SEQUENCES

The calling sequences are divided into the following categories.

- Primitives
- Terminal functions
- Segments and pictures
- Windows and viewports
- Transformation matrices
- Geometry utilities
- Interaction
- Error processing

PRIMITIVES

ARCxxx

ARCA(*cx,cy,x,y*)	Draw absolute 2-D arc to endpoint.
ARCA3(*cx,cy,cz,x,y,z, xdir,ydir,zdir*)	Draw absolute 3-D arc to endpoint.
ARCDA(*cx,cy,deg*)	Draw absolute 2-D arc through angle.
ARCDA3(*cx,cy,cz, ddeg,xdir,ydir,zdir*)	Draw absolute 3-D arc through angle.

ARCDR(*cdx,cdy,ddeg*)	Draw relative 2-D arc through angle.
ARCDR3(*cdx,cdy,cdz, ddeg,xdir,ydir,zdir*)	Draw relative 3-D arc through angle.
ARCR(*cdx,cdy,dx,dy*)	Draw relative 2-D arc to endpoint.
ARCR3(*cdx,cdy,cdz, dx,dy,dz,xdir,ydir, zdir*)	Draw relative 3-D arc to endpoint.

DOTxx

DOTA(*x,y*)	Draw dot at absolute 2-D position.
DOTA3(*x,y,z*)	Draw dot at absolute 3-D position.
DOTR(*dx,dy*)	Draw dot at relative 2-D position.
DOTR3(*dx,dy,dz*)	Draw dot at relative 3-D position.

DRAWxx

DRAWA(*x,y*)	Draw line to absolute 2-D endpoint.
DRAWA3(*x,y,z*)	Draw line to absolute 3-D endpoint.
DRAWR(*dx,dy*)	Draw line to relative 2-D endpoint.
DRAWR3(*dx,dy,dz*)	Draw line to relative 3-D endpoint.

MOVExx

MOVEA(*x,y*)	Move beam to absolute 2-D position.
MOVEA3(*x,y,z*)	Move beam to absolute 3-D position.
MOVER(*dx,dy*)	Move beam to relative 2-D position.
MOVER3(*dx,dy,dz*)	Move beam to relative 3-D position.

PLOTxx

PLOTA(*npoint,xary, yary,line*)	Plot symbols at absolute 2-D endpoints.
PLOTA3(*npoint,xary, yary,zary,line*)	Plot symbols at absolute 3-D endpoints.
PLOTR(*npoint,dxary, dyary,line*)	Plot symbols at relative 2-D endpoints.
PLOTR3(*npoint,dxary, dyary,dzary,line*)	Plot symbols at relative 3-D endpoints.

TEXTx

TEXT(*nchar,itext*)	Draw specified text string at current 2-D position.
TEXT3(*nchar,itext*)	Draw specified text string at current 3-D position.

TFNSIZ

TFNSIZ(*nsize*)	Test to determine number of discrete character sizes supported by terminal.

xxCSIZ

RACSIZ(*idseg,wide,high*)	Reset continuous character size attribute of a segment.
SMCSIZ(*wide,high*)	Modally set continuous character size.
TACSIZ(*idseg,widout,hiout*)	Test continuous character size attribute of a segment.
TFCSIZ(*lcchar*)	Test to see if continuous character sizes are supported.
TMCSIZ(*widout,hiout*)	Test current modal setting of continuous character size.

xxDSIZ

RADSIZ(*idseg,wide,high*)	Reset discrete character size attribute of a segment.
SMDSIZ(*wide,high*)	Modally set discrete character size.
TADSIZ(*idseg,widout,hiout*)	Test discrete character size attribute of a segment.
TFDSIZ(*wide,high, widout,hiout*)	Test to determine best character size to use.
TMDSIZ(*widout,hiout*)	Test current modal setting of discrete character size.

xxFONT

RAFONT(*idseg,ifont*)	Reset character font attribute of a segment.
SMFONT(*ifont*)	Modally set character font.

TAFONT(*idseg,ifont*)	Test character font attribute of a segment.
TFFONT(*nfont*)	Test to determine number of character fonts available on a terminal.
TMFONT(*ifont*)	Test current modal setting of character font.

xxHILT

RAHILT(*idseg,lhilit*)	Reset highlighting attribute of a segment.
SMHILT(*lhilit*)	Set highlighting mode on or off.
TAHILT(*idseg,lhilit*)	Test highlighting attribute of a segment.
TFHILT(*lhilt*)	Test to determine if highlighting of display is supported by postprocessor.
TMHILT(*lhilit*)	Test current highlighting mode.

xxINT

RAINT(*idseg,finten*)	Reset intensity attribute of a segment.
SMINT(*finten*)	Modally set intensity attribute of a segment.
TAINT(*idseg,finten*)	Test intensity attribute of a segment.
TFINT(*ninten*)	Test to determine number of intensity levels available on a terminal.
TMINT(*finten*)	Test current modal setting for intensity.

xxROT

RAROT(*idseg,deg*)	Reset character rotation attribute of a segment.
SMROT(*deg*)	Modally set angle of character rotation.
TAROT(*idseg,deg*)	Test character rotation attribute of a segment.
TFROT(*lninty,lcont*)	Test to determine if character rotation by 90-degree increments is supported, and if continuous character rotation is supported.
TMROT(*deg*)	Test current modal setting of angle of character rotation.

xxROT3

RAROT3(*idseg,xbase, ybase,zbase,xplane, yplane,zplane*)	Reset 3-D character rotation and plane attribute of a segment.
SMROT3(*xbase,ybase, zbase,xplane,yplane, zplane*)	Modally set rotation angle and plane for 3-D characters.
TAROT3(*idseg,xbase, ybase,zbase,xplane, yplane,zplane*)	Test 3-D character rotation and plane attribute of a segment.
TMROT3(*xbase,ybase, zbase,xplane,yplane, zplane*)	Test current modal setting of rotation angle and plane for 3-D characters.

xxSTYL

RASTYL(*idseg,istyle*)	Reset line style attribute of a segment.
SMSTYL(*istyle*)	Modally set line style.
TASTYL(*idseg,istyle*)	Test line style attribute of a segment.

TFSTYL(*lhard*)	Test for line styles supported by terminal hardware.
TMSTYL(*istyle*)	Test current modal setting of line style.

xxSYM

RASYM(*idseg,isym*)	Reset the plotting symbol attribute of a segment.
SMSYM(*isym*)	Modally set the selected plotting symbol for use with PLOTxx routines.
TASYM(*idseg,isym*)	Test the plotting symbol attribute of a segment.
TFSYM(*nsym*)	Test for maximum defined symbol number in a given postprocessor.
TMSYM(*isym*)	Test current modal setting for plot symbol.

xxVIS

RAVIS(*idseg,lvis*)	Reset visibility attribute of a segment.
SMVIS(*lvis*)	Set visibility mode for visible or invisible.
TAVIS(*idseg,lvis*)	Test visibility attribute of a segment.
TFVIS(*ltran*)	Test to determine if complete screen retransmission is required to make any one segment invisible.
TMVIS(*lvis*)	Test current setting of visibility mode.

TERMINAL FUNCTIONS

ALARM(*lon*)	Turn terminal alarm indicator on or off.
CLRSCR	Clear all displays from terminal screen.
DSPLAY	Display all pictures.
INITIG(*lsquar,lnwfil, filnam*)	Set all TIGS and terminal conditions to initial values.
QUITIG(*ldelet*)	Ensure orderly shutdown of terminal at end of program.
REMSCR	Copy contents of screen on remote hardcopier.
SCRNUR(*idwind,xscrn, yscrn,xuser,yuser*)	Convert screen coordinates to user coordinates; 2-D pictures only.
TFHARD(*lremot*)	Test for existence of remote hardcopiers.
TFSCRN(*lrtang,xll,yll, xur,yur,resltn*)	Test size, shape, and resolution of the terminal screen.
UDATA(*nwords,idat*)	Place user data in neutral display file.
UNISCR	Create picture on UNIPLOT NPFILE to reflect current state of neutral display file.
WHERE(*x,y*)	Obtain current beam position for 2-D picture.
WHERE3(*x,y,z*)	Obtain current beam position for 3-D picture.

SEGMENTS AND PICTURES

BLINDS(*idpict,ldown*)	Control display of all parts of picture through all windows with which the picture is associated.
CLSPIC	Close an open picture.
CLSSEG	Define end of the currently open segment.
COPY(*idseg,newseg*)	Generate a copy of the specified segment and assign it the segment identifier *newseg*.
DELPIC(*idpict*)	Delete specified picture.
DELSEG(*idseg*)	Delete specified segment.
EMPTY(*idseg*)	Delete contents of a segment.
EXTPIC(*idpict*)	Add segment to an existing picture.
EXTSEG(*idseg*)	Add primitive to an existing segment.
LCKSEG	Lock currently open segment.
OPNPIC(*idpict*)	Define beginning of a picture.
OPNSEG(*idseg*)	Define beginning of a segment.
RENAME(*idold,idnew*)	Replace old segment identifier *idold* with new identifier *idnew*.

xxPICT

RAPICT(*idseg,idpict*)	Reset picture attribute of a segment.
SMPICT(*idpict*)	Modally set the already-existing picture to which subsequent segments are added.
TAPICT(*idseg,idpict*)	Test picture attribute of a segment.
TFPICT(*npict*)	Test for maximum number of pictures postprocessor supports.
TMPICT(*idpict*)	Test for ID of current mode set picture.

xxPLIx

SMPLIM(*xll,yll,xur,yur*)	Modally set limits for 2-D picture coordinates.
SMPLI3(*xllh,yllh,zllh, xury,yury,zury*)	Modally set limits for 3-D picture coordinates.
TAPLIM(*idseg,xll,yll, xur,yur*)	Test limits of the 2-D picture which contains the specified segment.
TAPLI3(*idseg,xllh,yllh, zllh,xury,yury,zury*)	Test limits of the 3-D picture which contains the specified segment.
TMPLIM(*xll,yll,xur,yur*)	Test limits of the current modally-set 2-D picture.
TMPLI3(*xllh,yllh,zllh, xury,yury,zury*)	Test limits of the current modally-set 3-D picture.

xx3D

TA3D(*idseg,l3D*)	Test dimensionality attribute of given segment.
TM3D(*l3D*)	Test current mode for picture dimensionality.

WINDOWS AND VIEWPORTS

DELxxx

DELVUP(*idport*)	Delete specified viewport.
DELWIN(*idwind*)	Delete specified window.

VUPORx

VUPORT(*idport,xll,yll, xur,yur*)	Define 2-D viewport ID and limits.
VUPOR3(*idport,xllh, yllh,zllh,xury,yury, zury*)	Define 3-D viewport ID and limits.

WINxxx

WINCLP(*idwind,lclpnr, lclpfr,disner,disfar*)	Define 3-D window clipping.
WINDIR(*idwind,xeye, yeye,zeye,xat,yat,zat*)	Define 3-D line of vision.
WINDOW(*idwind,xll,yll, xur,yur*)	Define 2-D window ID and limits.
WINPER(*idwind,lpersp*)	Define 3-D window perspective.
WINPLN(*idwind,distat*)	Define 3-D projection plane.
WINSIZ(*idwind,width, height*)	Define 3-D window size.

WINUP(*idwind,dxup, dyup,dzup*)	Define 3-D window up direction.

xxPORT

TFPORT(*nport*)	Test for maximum number of viewports supported by a postprocessor.
TMPORT(*idport*)	Test for current modally set viewport.
SMPORT(*idport*)	Modally set viewport into which subsequent windows are mapped.

xxSVP

SMSVP(*lscren,xll,yll, xur,yur*)	Modally set viewport to be used for system messages and user entries.
TFSVP(*lsysvp*)	Test for default system viewport separate from graphics area.
TMSVP(*lscren,xll,yll, xur,yur*)	Test for viewport used for system messages and user entries.

TRANSFORMATION MATRICES

CLRSTx

CLRSTK	Clear 2-D transformation matrix storage stack.
CLRST3	Clear 3-D transformation matrix storage stack.

POPx

POP — Replace CTM with matrix on top of 2-D transformation matrix storage stack.

POP3 — Replace CTM3 with matrix on top of 3-D transformation matrix storage stack.

PUSHx

PUSH — Place copy of CTM on 2-D transformation matrix storage stack.

PUSH3 — Place copy of CTM3 on 3-D transformation matrix storage stack.

XIDNTx

XIDNT(*bmat23*) — Build 2-D identity matrix.

XIDNT3(*bmat34*) — Build 3-D identity matrix.

XINVRx

XINVR(*bmat23,binv23*) — Build 2-D inverse matrix.

XINVR3(*bmat34,binv34*) — Build 3-D inverse matrix.

XROTxx

XROTA(*deg,bmat23*) — 2-D absolute rotation.

XROTA3(*idaxis,deg, bmat34*) — 3-D absolute rotation.

XROTL(*ddeg,bmat23*) — 2-D relative rotation, left multiplication.

XROTL3(*idaxis,ddeg, bmat34*)	3-D relative rotation, left multiplication.
XROTR(*ddeg,bmat23*)	2-D relative rotation, right multiplication.
XROTR3(*idaxis,ddeg, bmat34*)	3-D relative rotation, right multiplication.

XSCLxx

XSCLA(*sx,sy,bmat23*)	2-D absolute scale.
XSCLA3(*sx,sy,sz,bmat34*)	3-D absolute scale.
XSCLL(*sdx,sdy,bmat23*)	2-D relative scale, left multiplication.
XSCLL3(*sdx,sdy,sdz, bmat34*)	3-D relative scale, left multiplication.
XSCLR(*sdx,sdy,bmat23*)	2-D relative scale, right multiplication.
XSCLR3(*sdx,sdy,sdz, bmat34*)	3-D relative scale, right multiplication.

XTRNxx

XTRNA(*x,y,bmat23*)	2-D absolute translation.
XTRNA3(*x,y,z,bmat34*)	3-D absolute translation.
XTRNL(*dx,dy,bmat23*)	2-D relative translation, left multiplication.
XTRNL3(*dx,dy,dz,bmat34*)	3-D relative translation, left multiplication.
XTRNR(*dx,dy,bmat23*)	2-D relative translation, right multiplication.
XTRNR3(*dx,dy,dz,bmat34*)	3-D relative translation, right multiplication.

xxXFxx

RAXFA(*idseg,bmat23*)	Absolutely reset transformation attribute of 2-D segment.
RAXFA3(*idseg,bmat34*)	Absolutely reset transformation attribute of 3-D segment.
RAXFL(*idseg,bmat23*)	Relatively reset transformation attribute of 2-D segment, left multiplication.
RAXFL3(*idseg,bmat34*)	Relatively reset transformation attribute of 3-D segment, left multiplication.
RAXFR(*idseg,bmat23*)	Relatively reset transformation attribute of 2-D segment, right multiplication.
RAXFR3(*idseg,bmat34*)	Relatively reset transformation attribute of 3-D segment, right multiplication.
SMXFA(*bmat23*)	Modally set absolute 2-D transformation.
SMXFA3(*bmat34*)	Modally set absolute 3-D transformation.
SMXFL(*bmat23*)	Modally set relative 2-D transformation, left multiplication.
SMXFL3(*bmat34*)	Modally set relative 3-D transformation, left multiplication.
SMXFR(*bmat23*)	Modally set relative 2-D transformation, right multiplication.
SMXFR3(*bmat34*)	Modally set relative 3-D transformation, right multiplication.

TAXFA(*idseg,bmat23*)
TAXFL(*idseg,bmat23*)
TAXFR(*idseg,bmat23*)

Test transformation attribute of 2-D segment; a copy of the matrix is placed in *bmat23*.

TAXFA3(*idseg,bmat34*)
TAXFL3(*idseg,bmat34*)
TAXFR3(*idseg,bmat34*)

Test transformation attribute of 3-D segment; a copy of the matrix is placed in *bmat34*.

TFXFA(*lxlat,lscal,lrot*)
TFXFL(*lxlat,lscal,lrot*)
TFXFR(*lxlat,lscal,lrot*)

Test for terminal capability to perform 2-D transformations.

TFXFA3(*lxfm3,lpersp, lpyram*)
TFXFL3(*lxfm3,lpersp, lpyram*)
TFXFR3(*lxfm3,lpersp, lpyram*)

Test for terminal capability to perform 3-D transformations.

TMXFA(*bmat23*)
TMXFL(*bmat23*)
TMXFR(*bmat23*)

Test for current modally set 2-D transformation matrix; a copy of CTM is placed in *bmat23*.

TMXFA3(*bmat34*)
TMXFL3(*bmat34*)
TMXFR3(*bmat34*)

Test for current modally set 3-D transformation matrix; a copy of CTM is placed in *bmat34*.

GEOMETRY UTILITIES

ENDPAR

ENDPAR(*cx,cy,x1,y1, x2,y2,narcs,xn1,yn1, xn2,yn2*)

Determine endpoints of specified 2-D arc.

ENDPLx

ENDPLN(*x1,y1,x2,y2, ishow,xn1,yn1,xn2, yn2*)

Determine endpoints of specified 2-D line.

ENDPL3(*x1,y1,z1,x2, y2,z2,ishow,xn1,yn1, zn1,xn2,yn2,zn2*)

Determine endpoints of specified 3-D line.

RTANGx

RTANGL(*xll,yll,xur,yur*) Define limits for 2-D endpoint calculations.

RTANG3(*xllh,yllh,zllh, xury,yury,zury*) Define limits for 3-D endpoint calculations.

INTERACTION

EVENT(*lky,ids,coords, iremng*) Report terminal input to application program.

KEYBRD(*maxchr,nchrs, itext*) Return to application program the text string entered from the terminal alphanumeric keyboard.

KYAC(*idky,iactn*) Assign individual function key to action type.

KYOFF Assign all function keys to ignore action type.

KYON Assign all function keys to terminate action type.

LOCATE(*x,y,iremng*) Report one or more sets of locator symbol coordinates.

PREEVN(*lucord,idvuwi*) Specify coordinate system and window/viewport ID for EVENT processing.

PRELOC(*lucord,idvuwi*) Specify coordinate system and window/viewport ID for LOCATE processing.

PROMPT(*nchar,itext*) Display message in system viewport area of screen.

xxAC

RAAC(*idseg,iactn*)	Reset action type attribute of segment.
SMAC(*iactn*)	Modally set action type.
TAAC(*idseg,iactn*)	Test action type attribute of segment.
TFAC(*lactn*)	Test for interaction support by postprocessor.
TMAC(*iactn*)	Test current modally set action type.

xxID

SMID(*idintr*)	Modally set intrasegment identifier.
TFID(*nid*)	Test for postprocessor support of return of intrasegment identifier.
TMID(*idintr*)	Test for current modally set intrasegment identifier.

xxINFO

RAINFO(*idseg,ninfo,info*)	Reset application information attribute of segment.
SMINFO(*ninfo,info*)	Modally set information stored with segments.
TAINFO(*idseg,ninfo, info*)	Test application attribute of segment.
TMINFO(*ninfo,info*)	Test current modal setting for information stored with segment.

xxLOCR

SMLOCR(*ilocr*)	Modally set locator device to be used.
TFLOCR(*maxloc,nlocrs, descrp,lone*)	Test for number of locators (if any) supported and their characteristics.
TMLOCR(*ilocr*)	Test for current modally set locator device.

ERROR PROCESSING

IERROR

IERROR(*ierr*)	Check current error status.

xxERR

SMERR(*routin*)	Modally set error processing routine to be used when error is detected.
TFERR(*lroutn*)	Test for postprocessor support of user-supplied error routine.
TMERR(*routin*)	Test current modally set error processing routine.

PARAMETERS

The following are the parameters used in TIGS calling sequences. After each parameter is a description. The parameters are listed in alphabetical order.

binv23	Output parameter; inverse matrix of *bmat23*; 2 x 3 array.
binv34	Output parameter; inverse matrix of *bmat34*; 3 x 4 array.

bmat23

Input parameter; 2 x 3 array used for 2-D building matrix.

bmat34

Input parameter; 3 x 4 array used for 3-D building matrix.

cdx,cdy,cdz

Input parameters; user coordinates of arc center relative to current beam position.

coords

Output parameter; three-word array containing best effort coordinate values of the location of the segment pick.

COORDS(1) = x coordinate
COORDS(2) = y coordinate
COORDS(3) = z coordinate (zero for 2-D picks)

cx,cy,cz

Input parameters; absolute user coordinates of arc center.

Input parameters (ENDPAR); coordinates of arc center to be checked against boundaries specified by RTANGx.

ddeg

Input parameter; relative angular position of arc endpoint, measured in degrees from the radius defined by arc center and current beam position. Positive values are counterclockwise (2-D) and in the direction of direction cosines (3-D). Negative values imply clockwise direction (2-D) and opposite to direction of the direction cosines (3-D).

Input parameter (XROTL, XROTR, XROTL3, and XROTR3); relative number of degrees to rotate segment; positive values indicate counterclockwise rotation.

deg	Input parameter; absolute angular position of arc endpoint, measured in degrees counterclockwise from X axis. If negative, angular measurement is clockwise.
	Input parameter (SMROT and RAROT) or output parameter (TMROT and TAROT); specifies magnitude of rotation in degrees measured counterclockwise from X axis.
	Input parameter (XROTA and XROTA3); absolute number of degrees to rotate segment; positive values indicate counterclockwise rotation.
descrp	Output array of size *nlocrs* describing the *nlocrs* locators for this postprocessor. Information for a given entry describes type of locator.
disner,disfar	Input parameters; specify clipping plane distances from center of attention.
	If LCLPNR = .FALSE., disner is meaningless.
	If LCLPFR = .FALSE., disfar is meaningless.
	Otherwise, these parameters specify directed distances along line of vision from center of attention to near and far clipping planes. Positive distances are away from the eye; negative distances are toward the eye. There are no defaults because overall defaults for WINCLP specify no clipping.
distat	Input parameter; specifies directed distance from center of attention to projection plane. Positive distances are away from the eye; negative distances are toward the eye.

dx,dy,dz

Input parameters; relative user coordinates of arc endpoint, dot, or line endpoint.

Input parameters (XTRNL, XTRNR, XTRNL3, and XTRNR3); relative displacement in x, y, and z direction.

dxary,dyary, dzary

Input array of length *npoint* containing relative user coordinates of each endpoint, in sequence. Coordinates of each point are given relative to last point in array, not relative to initial beam position.

dxup,dyup,dzup

Input parameters; specify coordinates of a point relative to center of attention. Directed line segment from center of attention to this point defines up direction for the window; this line segment becomes the vertical axis when window is mapped to viewport.

filnam

Input parameter; specifies name of neutral display file.

finten

Input parameter (SMINT and RAINT) or output parameter (TMINT and TAINT); indicates intensity level. Range is from 0. (dimmest) to 1. (brightest).

height

Input parameter; specifies height of viewing window on the projection plane, centered about the line of vision and parallel to the up direction.

iactn

Input or output parameter; specifies action to be performed when *idky* is pressed (KYAC) or action type (SMAC, RAAC, TMAC, TAAC) is selected.

IACTN = 1	Ignore.
IACTN = 2	Recognize.
IACTN = 3	Terminate.

idat	Input parameter; array containing user data.
idaxis	Input parameter; axis about which to perform rotation for 3-D; specify as 1HX, 1HY, or 1HZ.
idintr	Input parameter (SMID) or output parameter (TMID); specifies intrasegment identifier. If *idintr* > *nid,nid* is used.
idky	Input parameter; number of function key to be assigned. Number of function keys available is postprocessor dependent.
idnew	Input parameter; new ID of segment.
idold	Input parameter; ID of segment which is to be renamed.
idpict	Input parameter (SMPICT and RAPICT) or output parameter (TMPICT and TAPICT); specifies a picture identifier.
idport	Input parameter; specifies ID of viewport.
ids	Output parameter; five-word array containing information about event reported by current call to EVENT. If event was a function key press (LKY = .TRUE.), IDS(1) contains function key ID. Other elements of array ids have no meaning, and *coords* parameter has no meaning.

If event was a segment pick (LKY = .FALSE.), array *ids* has the following significance.

Array Element	Contents
1	ID of picked segment.
2	ID of window in which segment is displayed.
3	ID of picture containing segment.
4	ID of viewport in which segment is displayed.
5	Intrasegment identifier (zero if no intrasegment ID).

idseg — Input parameter; identifies segment whose attribute is to be tested, reset, deleted, or copied.

idvuwi — Input parameter; ID of window or viewport in which initially to display the locator symbol. ID pertains to a window or viewport, depending on the value given for *lucord*.

idwind — Input parameter; specifies ID of window.

ierr — Output parameter; TIGS error number.

If IERR = 0 [IERROR(IERR) = 0 if called as function], no error was detected.

If IERR ≥ 1 [IERROR(IERR) ≥ 1 if called as function], IERR contains error number of detected error.

ifont

Input parameter (SMFONT and RAFONT) or output parameter (TMFONT and TAFONT); specifies character font used.

Range is 0 to 63. If IFONT > NFONT, NFONT is used.

IFONT=1	Use normal font.
IFONT=2	Use italicized characters (if supported).
IFONT=3	Use third font (if supported).
.	
.	
.	
IFONT=n	Use nth font (if supported).
IFONT=0	Reserved for future expansion.

Number of supported fonts depends on postprocessor.

ilocr

Input parameter (SMLOCR) or output parameter (TMLOCR); specifies locator. Ordinals specify corresponding members of *descrp* array; that is, ILOCR=1 specifies locator described in DESCRP(1). If *ilocr* > *nlocrs*, the highest numbered locator is used.

info

Input parameter (SMINFO and RAINFO) or output parameter (TMINFO and TAINFO); name of array containing or to contain application information.

iremng

Output parameter; specifies number of events remaining on the event queue for EVENT calls or number of locations remaining on the locate queue for LOCATE calls.

ishow

Output parameter; specifies result of boundary check.

If ISHOW=0, line lies outside area or volume and endpoints are meaningless.

If ISHOW=1, line lies totally within area and new endpoints are same as original endpoints.

If ISHOW=2, first endpoint is clipped and at least one of the following relationships is true.

$x1 \neq xn1$
$y1 \neq yn1$
$z1 \neq zn1$

If ISHOW=3, second endpoint is clipped and at least one of the following relationships is true.

$x2 \neq xn2$
$y2 \neq yn2$
$z2 \neq zn2$

If ISHOW=4, both endpoints are clipped and at least one of the following relationships from each column is true.

$x1 \neq xn1$	$x2 \neq xn2$
$y1 \neq yn1$	$y2 \neq yn2$
$z1 \neq zn1$	$z2 \neq zn2$

istyle

An input parameter (SMSTYL and RASTYL) or output parameter (TMSTYL and TASTYL); specifies line style.

1	Solid lines.
2	Long dashed lines.
3	Short dashed lines.
4	Dash-dotted lines.
5	Dotted lines.
> 5	Bit pattern line style.

isym	Input or output parameter; specifies symbol used.
itext	Input array containing *nchar* characters to be drawn. Maximum characters per word is machine dependent. Output parameter (KEYBRD); first word of array containing text string.
lactn	Output parameter; if LACTN= .TRUE., EVENT subroutine is supported by postprocessor; otherwise, it is not.
lcchar	Output parameter; if LCCHAR= .TRUE., postprocessor supports continuous character sizes.
lclpnr,lclpfr	Input parameters; specify clipping planes. If LCLPNR=.TRUE., clipping is done to the near clipping plane; otherwise, it is not. If LCLPFR=.TRUE., clipping is done to the far clipping plane; otherwise, it is not.
lcont	Output parameter; if LCONT= .TRUE., continuous character rotation is supported by either terminal hardware or by TIGS software.
ldelet	Input parameter; specifies whether neutral display file used by program is to be saved or discarded when QUITIG is called. If LDELET=.TRUE., neutral display file is discarded. If LDELET=.FALSE., neutral display file is not discarded.

ldown

Input parameter; specifies whether picture will be seen in associated windows.

If LDOWN=.TRUE., blinds are down and no part of picture will be seen in any window.

If LDOWN=.FALSE., blinds are up and picture is visible in all associated windows.

lhard

Output array of six logical variables specifying what line styles are supported by terminal hardware.

LHARD(1)=.TRUE. Solid line style supported.

LHARD(2)=.TRUE. Long dashed line style is supported.

LHARD(3)=.TRUE. Short dashed line style is supported.

LHARD(4)=.TRUE. Dashed-dotted line style is supported.

LHARD(5)=.TRUE. Dotted line style is supported.

LHARD(6)=.TRUE. Bit pattern line style is supported.

lhilit

Input parameter (SMHILT and RAHILT) or output parameter (TMHILT and TAHILT); indicates highlighting mode or attribute. If LHILT=.TRUE., affected segments are highlighted; otherwise, they are not.

lhilt

Output parameter; if LHILT=.TRUE., highlighting is supported; otherwise, it is not.

line

Input logical variable specifying whether lines are to be drawn between plotted points. Applies to all points in plot.

If LINE=.TRUE., lines are drawn between endpoints, starting at current beam position.

If LINE=.FALSE., no lines are drawn.

Symbol is plotted regardless of value of *line*.

lky

Output parameter; indicates whether event was function key press or segment pick.

If LKY=.TRUE., event was a function key press.

If LKY=.FALSE., event was a segment pick.

lninty

Output parameter; if LNINTY= .TRUE., terminal hardware supports 90-degree character rotation.

lnwfil

Input parameter; specifies whether the neutral display file is new or old.

If LNWFIL=.TRUE., new neutral display file will be created by this graphics program.

If LNWFIL=.FALSE., this program will use old neutral display file.

lon

Input parameter; turn alarm on or off.

If LON=.TRUE., turn alarm on.

If LON=.FALSE., turn alarm off.

lone

Output logical array of size *nlocrs*; if LONE(1)=.TRUE., then DESCRP(1) is a one-shot locator.

lpersp

Input parameters; specify 3-D perspective type. If LPERSP=.TRUE., window has a perspective projection.

If LPERSP=.FALSE., window has a parallel (axonometric) projection.

Output parameter (TFXFA3, TFXFL3, and TFXFR3); if LPERSP=.TRUE., perspective preservation during 3-D transformation is hardware supported; otherwise, it is not.

lpyram

Output parameter; if LPYRAM=.TRUE., clipping to 3-D window pyramid during transformations is hardware supported; otherwise, it is not.

lremot

Output parameter; if LREMOT=.TRUE., remote hardcopier exists.

lrot

Output parameter; if LROT=.TRUE., 2-D rotation is hardware supported; otherwise, it is not.

lroutn

Output parameter; if LROUTN=.TRUE., user-supplied error routine is supported; otherwise, it is not.

lrtang

Output parameter; describes screen shape and dimensions.

If LRTANG=.TRUE., terminal screen is rectangular.

If LRTANG=.FALSE., terminal screen is circular.

lscal

Output parameter; if LSCAL=.TRUE., 2-D scaling is hardware supported; otherwise, it is not.

lscren	Input parameter (SMSVP) or output parameter (TMSVP). If LSCREN=.TRUE., system viewport coordinates are given in a terminal-independent coordinate system. If LSCREN=.FALSE., system viewport coordinates are given in terminal-dependent coordinate system.
lsquar	Input parameter; defines mapping of screen coordinates onto terminal display surface. If LSQUAR=.TRUE., map screen coordinates onto largest square possible on terminal display surface. If LSQUAR=.FALSE., use entire terminal display surface.
lsysvp	Output parameter. If LSYSVP=.TRUE., postprocessor supports a default system viewport that is guaranteed separate from the area of screen used for graphics displays. If LSYSVP=.FALSE., default viewport is not guaranteed separate. Dependent on *lsquar* value given in INITIG routine.
ltran	Output parameter; if LTRAN=.TRUE., retransmission of data to the screen is necessary to make a visible segment invisible.

lucord

Input parameter; specifies coordinate system for returned coordinates and type of ID specified by *idvuwi*.

If LUCORD=.TRUE., coordinates are returned as user coordinates; *idvuwi* specifies the ID of a window.

If LUCORD=.FALSE., coordinates are returned as screen coordinates; *idvuwi* specifies the ID of a viewport.

lvis

Input parameter (SMVIS and RAVIS) or output parameter (TMVIS and TAVIS); indicates visibility mode or attribute. If LVIS=.TRUE., affected segments are visible; otherwise, they are not. Visibility mode may not be changed during segment definition; entire segment is either visible or invisible.

lxfm3

Output parameter; if LXFM3=.TRUE., all 3-D transformation functions are hardware supported; otherwise, they are not.

lxlat

Output parameter; if LXLAT=.TRUE., 2-D translation is hardware supported; otherwise, it is not.

l3d

Output parameter; if L3D=.TRUE., the current modal setting or attribute is 3-D; otherwise, it is 2-D.

maxchr

Input parameter; maximum number of characters permitted in string.

maxloc

Input parameter; size of the arrays *descrp* and *lone*.

narcs	Output parameter; contains the number of arc sections that are inside area specified by RTANGL. If NARCS=0, *xn1,yn1,xn2,yn2* are meaningless.
nchar	Input parameter; number of characters in text string.
nchrs	Output parameter; number of characters contained in array *itext* up to and including last nonblank character.
newseg	Input parameter; identifier to be assigned to copy.
nfont	Output parameter; indicates number of character fonts supported.
nid	Output parameter; largest number supported by postprocessor for intrasegment identifiers; if NID=0, the feature is not supported.
ninfo	Input parameter; number of words in *info* array.
ninten	Output parameter; indicates number of discrete intensity levels supported.
nlocrs	Output parameter; total number of locator devices supported by postprocessor up to *maxloc*.
npict	Output parameter; indicates maximum number of pictures supported by postprocessor.
npoint	Input parameter; number of endpoints in each array (same for all arrays).
nport	Output parameter; specifies maximum number of viewports supported by postprocessor.
nsize	Output parameter; specifies number of discrete character sizes supported.

nsym	Output parameter; indicates maximum symbol number for which a symbol is defined.
nwords	Input parameter; number of words in *idat* array.
resltn	Output parameter; value expressing ability of the terminal screen to resolve images.
routin	Input parameter (SMERR) or output parameter (TMERR); specifies the address of the error routine.
sdx,sdy,sdz	Input parameters; relative scale factors in x, y, and z axes.
sx,sy,sz	Input parameters; absolute scale factors in x, y, and z axes.
wide,high	Input parameters; specifies width and height of desired character size as fractions of screen size for RADSIZ and SMDSIZ and in user coordinates for RACSIZ and SMCSIZ.
widout,hiout	Output parameters; specifies width and height of hardware discrete character size that best approximates desired size (TFDSIZ) or character size that is being used (TMDSIZ and TADSIZ) or specifies width and height of character box being used for continuous characters (TMCSIZ and TACSIZ).
width	Input parameter; specifies width of viewing window on projection plane, centered about the line of vision and perpendicular to the up direction.

x,y,z	Input parameters; absolute user coordinates of arc endpoint, dot, line endpoint, or locator symbol. Input parameters (XTRNA and XTRNA3); absolute displacement in x, y, and z direction.
xary,yary,zary	Input arrays of length *npoint* containing absolute user coordinates of each endpoint, in proper sequence.
xat,yat,zat	Input parameters; specify coordinates of center of attention. This point and the eye position point define a line of vision.
xbase,ybase, zbase,xplane, yplane,zplane	Input parameters (SMROT3 and RAROT3) or output parameters (TMROT3 and TAROT3); specify base and plane vectors in relative user coordinates to define 3-D character rotation and plane.
xdir,ydir,zdir	Input parameters; direction cosines defining direction of 3-D arcs. In special cases of semicircles and full circles, direction cosines also define the plane of the arc.
xeye,yeye,zeye	Input parameters; specify coordinates of a point to be used as the viewer's eye position.
xll,yll,xur,yur	Input parameters; specify user coordinates of lower left and upper right corners of a 2-D window. The conditions $xll < xur$ $yll < yur$ must be satisfied for WINDOW.

	Input parameters (SMPLIM) or output parameters (TMPLIM and TAPLIM); specify user coordinates of lower left and upper right corners of the picture, screen area (SMSVP and TMSVP), or rectangle used by ENDPLN and ENDPAR subroutines (RTANGL).
	Input parameters; specify user coordinates (WINDOW) or screen coordinates (VUPORT) of lower left and upper right corners of the 2-D window or viewport.
xllh,yllh,zllh, xury,yury,zury	Input parameters; specify screen coordinates of lower left hither and upper right yon corners of viewport space.
	Input parameters (SMPLI3) or output parameters (TMPLI3 and TAPLI3); specify user coordinates of lower left hither and upper right yon corners of picture or 3-D parallelepiped used by ENDPL3 subroutine (RTANG3).
xn1,yn1,zn1, xn2,yn2,zn2	Output parameters; coordinates of intersections of line with boundaries of area/volume (new endpoints). For ENDPAR, each is dimensioned as an array of five words.
xscrn,yscrn	Input parameters; screen coordinates to be converted.
xuser,yuser	Output parameters; user coordinates converted from screen coordinates.
x1,y1,z1,x2,y2, z2	Input parameters; endpoints of line to be checked against boundaries specified by RTANGx.
	Input parameters (ENDPAR); coordinates of the endpoints of the arc.

DEFAULT PARAMETER VALUES

The following are the default values assigned if the user does not assign a value.

Routine	Parameter(s)	Default Value	Description
BLINDS	*ldown*	FALSE	Blinds up - picture visible.
	idpict	0	Default picture.
INITIG	*lsquar*	TRUE	Map screen coordinates so that graphics area is square and there is room for a separate system viewport.
	lnwfil	TRUE	New neutral display file will be created.
	filnam		Postprocessor-dependent.
KYAC	*iactn*	3	Terminate.
PREEVN	*idvuwi*	0	Default viewport.
	lucord	FALSE	Coordinates return as screen coordinates.
QUITIG	*ldelet*	TRUE	Neutral display file is discarded.
WINxxx	*xll,yll*	0,0	x and y values of lower left corner (default picture only).
	xur,yur	1,1	x and y values of upper right corner (default picture only).
	lclpnr, lclpfr	FALSE	Clipping is not done.

Routine	Parameter(s)	Default Value	Description
	xeye	*xury+(xury-xllh)*	On positive x-axis at a distance from the picture limit box equal to the width of the box in the x- direction.
	yeye	$\frac{yury+yllh}{2}$	Center of y-axis picture limits.
	zeye	$\frac{zury+zllh}{2}$	Center of z-axis picture limits.
	xat,yat, zat		Center of picture limits.
	lpersp	TRUE	Window has perspective projection.
	distat	0	
	width	*yury-yllh*	Width of picture.
	height	*zury,zllh*	Height of picture.
	dxup,dyup, dzup	0,0,1	Parallel to positive z-axis.
xxAC	*iactn*	3	Terminate.
xxDSIZ	*wide,high*	.0125, .0167	Hardware character size most closely approximating these values.
xxERR	*routin*	NULL	Error routine named NULL.
xxFONT	*ifont*	1	Normal font.
xxHILT	*lhilit*	FALSE	Affected segments not highlighted.
xxINFO	*ninfo*	0	No information.

Routine	Parameter(s)	Default Value	Description
xxINT	*finten*	.5	Medium intensity.
xxLOCR	*ilocr*	1	Found in descrp(1).
xxPLIx	*xll,yll*	0,0	Lower left corner is (0,0).
	xur,yur	1,1	Upper right corner is (1,1).
xxROT	*deg*	0°	No rotation.
xxROT3	*xbase,* *ybase,* *zbase* *xplane,* *yplane,* *zplane*	1.,0.,0. 0.,1.,0.	X-axis for base and y-axis as up direction.
xxSTYL	*istyle*	1	Solid line.
xxSYM	*isym*	-1	No symbol assigned.
xxVIS	*lvis*	TRUE	Segments visible.

ERROR MESSAGES

WARNING MESSAGES

Illegal ID Errors (1-100)

Error Number	Cause
1	Attempt to use an *idseg* that is undefined or open when it should not be.
2	Attempt to change the picture attribute of a segment to a nonexistent *idpict*.
3	Attempt to delete or put blinds on a nonexistent picture.
4	ID of window is not defined.
5	ID of viewport is not defined.
6	Attempt to open or extend a segment when one is already open.
10	Attempt to extend a picture that does not exist.
11	Attempt to extend a segment that does not exist.

Character or Word Count Errors (101-200)

Error Number	Cause
102	*npoint* ≤ 0 on a plot call.
103	*maxchr* ≤ 0 on a request for keyboard input.
104	*nwords* < 0 or *nwords* > 26 on User Data specification.
105	Number of characters in prompting message of text string is out of range (*nchars* < 0 or *nchars* > 256).

Clipped and Unclipped Beam Position Errors (201-30

Error Number	Cause
201	Clipped (to picture limits) and unclipped beam positions unequal when picture was closed and now programmer wishes to extend picture.
202	Clipped and unclipped beam positions unequal when segment was closed and now programmer wishes to extend segment.

Unsupported Postprocessor Features (301-400)

Error Number	Cause
301	Request for unsupported font.
302	Request for unsupported intensity.
303	Request for unsupported plotting symbol.
304	Request for highlighting when highlighting is not supported.
305	Request for continuous characters when not supported.

Reserved Warning Numbers (401-900)

Reserved.

Miscellaneous Errors (901-1000)

Error Number	Cause
901	Primitive encountered and no open segment.
902	Mode set locator is 0 and user wishes to specify locations.

903	Attempt to pop an empty 2-D transformation stack.
904	Attempt to pop an empty 3-D transformation stack.

SERIOUS ERROR DIAGNOSTICS

Primitive Specification Errors (1001-1100)

Error Number	Cause
1001	Arc center outside picture limits.
1002	Radii not approximately equal.
1003	Radius ≤ 0.
1004	Relative degrees of arc approximately 0.
1005	Sum of squares of direction cosines $\neq 1$.
1006	Direction cosine vector is coincident with or parallel to the vector from the start point to the center point.

Segment, Picture, Window, or Viewport Errors (1101-12(

Error Number	Cause
1101	Attempt to extend a locked segment.
1102	*idwind* out of range (*idwind*<0 or *idwind*>32767).
1103	*idport* out of range (*idport*$\leq$0 or *idport*>32767).
1104	*idpict* out of range (*idpict*$\leq$0 or *idpict*>32767). For BLINDS legal range is 0$\leq$*idpict*$\leq$32767.
1105	*idseg* out of range (*idseg*$\leq$0 or *idseg*>32767).

1106	Viewport or system viewport limits are invalid.
1107	Window limits outside range of picture limits or lower left corner not less than upper right corner.
1108	Attempt to define a window on the default viewpoint.
1109	Attempt to delete the modally set picture.
1110	New segment ID is already defined.
1113	*idseg* already defined.
1114	*idpict* already defined.
1115	Eye position is the same as the center of attention.
1116	*width* or *height* of window<0.
1117	Zero-length up direction vector specified.
1118	Far clipping plane is closer to eye than near clipping plane or clipping planes are coincident.
1119	Up direction is parallel to the line of sight.
1120	Near or far clipping plane is not in front of the eye.
1121	Projection plane is not in front of the eye.

Two-/Three-Dimensional Intermixing Errors (1201-130

Error Number	Cause
1201	Attempt to put a 2-D window on a 3-D picture.
1202	Attempt to put a 2-D primitive or 2-D transformation in a 3-D segment.

1203	2-D locations requested when location queue is 3-D.
1205	Window or viewport is 3-D and both should be 2-D.
1206	Attempt to put a 3-D primitive or 3-D transformation in a 2-D segment.
1207	Attempt to put a 3-D segment in a 2-D picture or attempting to put a 2-D segment in a 3-D picture.
1208	Attempt to put a 3-D window on a 2-D picture.
1209	Attempt to put a 3-D window in a 2-D viewport.
1210	Attempt to modify a 2-D window with 3-D information.
1211	Attempt to "LOCATE" (2-D) in a 3-D window; conversion to user coordinates is not possible.
1212	Segment picked was in a 3-D window and EVENT (2-D) was called.
1213	Attempt to modify a 3-D window with 2-D information.
1214	Attempt to modify a 2-D viewport with 3-D information.
1215	Attempt to modify a 3-D viewport with 2-D information.
1216	3-D routine was called where a 2-D call should have been made.
1217	2-D routine called where a 3-D call should have been made.

Set Mode or Reset Attribute Errors (1301-1400)

Error Number	Cause
1301	Intensity out of range (*finten*<0. or *finten*>1.)
1302	Font value out of range (*ifont*<0 or *ifont*>63).
1303	Action type out of range (*iactn*≤0 or *iactn*>3).
1304	Function key number out of range (*idky*<0 or *idky*>255).
1305	Plot symbol number is out of range (*isym*<0 or *isym*>32767).
1306	Line thickness percentage is out of range (*percnt*<0. or *percnt*>100.)
1307	Intrasegment ID is out of range (*idintr*<0 or *idintr*>32767).
1308	Number of words of application-related information is out of range (*ninfo*<0 or *ninfo*>4).
1309	Locator code is out of range (*ilocr*<0 or *ilocr*>63).
1310	Line style values is out of range (*istyle*≤0 or *istyle*>4095).
1320	Attempt to change action attribute of a segment after a primitive has been defined.
1321	Attempt to set picture limits after a primitive or closed segment has been defined.
1322	Attempt to modify the visibility attribute of a segment after the first primitive has been defined.
1323	Attempt to change highlighting attribute of a segment after a primitive has been defined.

1324	Base and plane vectors are colinear.
1325	Character height or width≤0.

Symbol Table (IDLIST) Overflow (1401-1500)

Error Number	Cause
1401	Symbol table overflow on opening a picture.
1402	Symbol table overflow on opening a segment.
1403	Symbol table overflow on defining a viewport.
1404	Symbol table overflow on defining a window.

Reserved Serious Error Numbers (1501-3900)

Reserved.

Miscellaneous Errors (3901-4000)

Error Number	Cause
3901	*xllh*≥*xury* or *yllh*≥*yury* or *zllh*≥*zury*
3902	Requested locator does not exist.
3903	Attempt to hardcopy on a nonexistent hardcopier.
3904	Attempt to rotate about an undefined axis.
3905	Attempt to invert a 2-D singular matrix.
3906	Attempt to invert a 3-D singular matrix.

3907	Attempt to reset continuous character size attribute of a segment with a discrete character size attribute.
3908	Attempt to reset discrete character size attribute of a segment with a continuous character size attribute.
3909	Mode set or attribute character size is discrete and a continuous character size test routine was called.
3910	Mode set or attribute character size is continuous and a discrete character size test routine was called.

Data Manager Detected Errors (4001-5000)

Error Number	Cause
4001	No data file or in-core block space available.
4002	The version number of the old file does not match the current TIGS version number.

NOTE

The following error numbers, 4100-4109, occur if an application is run when a neutral display file from a previous run has not been returned. QUITIG called with LDELET=.TRUE. returns the NDF. Refer to Data Handler Reference Manual for details.

4100	Illegal bead address passed to internal routine DMDMP.
4101	The file to be dumped by internal routine DMDMP has not been initialized.

4102	File to be accessed has not been initialized.
4103	Block count limit exceeded.
4104	File is not in the correct format to be processed by the TIGS data manager.
4105	Internal data handler array, IBLK, is too short.
4106	Internal common block /ZLDMTB/ loaded after internal array IBLK.
4107	Illegal bead address passed to internal routine DMRLBD.
4108	Illegal bead address passed to internal routines DMSET or DMGET.
4109	Illegal component type code passed to internal routines DMSET or DMGET.

TIGS Reserved Error Numbers (5001-10000)

CORPORATE HEADQUARTERS
P.O. BOX 0
MINNEAPOLIS, MINNESOTA 55440

SALES OFFICES AND SERVICE CENTERS
IN MAJOR CITIES
THROUGHOUT THE WORLD

PRINTED IN U.S.A.

CONTROL DATA CORPORATION