

DEC GKS

FORTRAN Binding Reference Manual, Part 2

Order Number: AA-PQP6A-TE

June 1992

This manual describes the FORTRAN binding functions provided for DEC GKS™.

Revision/Update Information: This is a new manual.

**Digital Equipment Corporation
Maynard, Massachusetts**

First Printing, June 1992

The information in this document is subject to change without notice and should not be construed as a commitment by Digital Equipment Corporation. Digital Equipment Corporation assumes no responsibility for any errors that may appear in this document.

The software described in this document is furnished under a license and may be used or copied only in accordance with the terms of such license.

No responsibility is assumed for the use or reliability of software on equipment that is not supplied by Digital Equipment Corporation or its affiliated companies.

Restricted Rights: Use, duplication, or disclosure by the U.S. Government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013.

© Digital Equipment Corporation 1992.

All Rights Reserved.

The postpaid Reader's Comments forms at the end of this document request your critical evaluation to assist in preparing future documentation.

The following are trademarks of Digital Equipment Corporation: DDIF, DEC, DEC GKS, DEC GKS-3D, DEC FORTRAN, DECnet, DECstation, DECwindows, LA75, LVP16, MicroVAX, ReGIS, VAX, VAX Ada, VAX BASIC, VAX C, VAX COBOL, VAX FORTRAN, VAX Pascal, VAXstation, VAXstation II, VAXstationII/GPX, VMS, VT125, VT240, VT241, VT330, VT340, ULTRIX, ULTRIX Worksystem Software, and the DIGITAL logo.

BASIC is a registered trademark of Dartmouth College. HP-GL, HP7475, HP7550, HP7580, HP7585, and Hewlett-Packard are trademarks of Hewlett-Packard Company. Motif and OSF/Motif are registered trademarks of Open Software Foundation, Inc. MPS-2000 is a trademark of Laser Graphics, Inc. PostScript is a registered trademark of Adobe Systems, Incorporated. Tektronix is a registered trademark of Tektronix, Inc.

ZK5683

This manual is available on CDROM.

This document was prepared using VAX DOCUMENT, Version 2.1.

Contents

Preface	ix
11 Inquiry Functions	
11.1 Using the Inquiry Functions	11-2
11.1.1 The Error Status Argument	11-3
11.1.2 The Value Type Argument	11-4
11.2 Function Descriptions	11-4
INQUIRE ASPECT SOURCE FLAGS	11-5
INQUIRE ASPECT SOURCE FLAGS 3	11-7
INQUIRE CHARACTER BASE VECTOR	11-9
INQUIRE CHARACTER EXPANSION FACTOR	11-10
INQUIRE CHARACTER HEIGHT	11-11
INQUIRE CHARACTER SPACING	11-12
INQUIRE CHARACTER UP VECTOR	11-13
INQUIRE CHARACTER WIDTH	11-14
INQUIRE CHOICE DEVICE STATE	11-15
INQUIRE CHOICE DEVICE STATE 3	11-17
INQUIRE CLIPPING	11-19
INQUIRE CLIPPING 3	11-20
INQUIRE COLOUR FACILITIES	11-21
INQUIRE COLOUR MODEL	11-22
INQUIRE COLOUR MODEL FACILITIES	11-23
INQUIRE COLOUR REPRESENTATION	11-24
INQUIRE CURRENT EDGE COLOUR INDEX	11-25
INQUIRE CURRENT EDGE FLAG	11-26
INQUIRE CURRENT EDGE INDEX	11-27
INQUIRE CURRENT EDGETYPE	11-28
INQUIRE CURRENT EDGEWIDTH SCALE FACTOR	11-29
INQUIRE CURRENT HLHSR IDENTIFIER VALUE	11-30
INQUIRE CURRENT NORMALIZATION TRANSFORMATION NUMBER	11-31
INQUIRE CURRENT PATTERN REFERENCE POINT AND VECTORS	11-32
INQUIRE CURRENT PICK IDENTIFIER VALUE	11-33
INQUIRE CURRENT VIEW INDEX	11-34
INQUIRE DEFAULT CHOICE DEVICE DATA	11-35
INQUIRE DEFAULT CHOICE DEVICE DATA 3	11-36
INQUIRE DEFAULT DEFERRAL STATE VALUES	11-37

INQUIRE DEFAULT LOCATOR DEVICE DATA	11-38
INQUIRE DEFAULT LOCATOR DEVICE DATA 3	11-39
INQUIRE DEFAULT PICK DEVICE DATA	11-40
INQUIRE DEFAULT PICK DEVICE DATA 3	11-41
INQUIRE DEFAULT STRING DEVICE DATA	11-42
INQUIRE DEFAULT STRING DEVICE DATA 3	11-43
INQUIRE DEFAULT STROKE DEVICE DATA	11-44
INQUIRE DEFAULT STROKE DEVICE DATA 3	11-45
INQUIRE DEFAULT VALUATOR DEVICE DATA	11-46
INQUIRE DEFAULT VALUATOR DEVICE DATA 3	11-47
INQUIRE DISPLAY SPACE SIZE	11-48
INQUIRE DISPLAY SPACE SIZE 3	11-49
INQUIRE DYNAMIC MODIFICATION OF SEGMENT ATTRIBUTES	11-50
INQUIRE DYNAMIC MODIFICATION OF WORKSTATION ATTRIBUTES	11-52
INQUIRE DYNAMIC MODIFICATION OF WORKSTATION ATTRIBUTES 3	11-54
INQUIRE EDGE FACILITIES	11-57
INQUIRE EDGE REPRESENTATION	11-58
INQUIRE FILL AREA COLOUR INDEX	11-59
INQUIRE FILL AREA FACILITIES	11-60
INQUIRE FILL AREA INDEX	11-62
INQUIRE FILL AREA INTERIOR STYLE	11-63
INQUIRE FILL AREA REPRESENTATION	11-64
INQUIRE FILL AREA STYLE INDEX	11-65
INQUIRE GENERALIZED DRAWING PRIMITIVE	11-66
INQUIRE GENERALIZED DRAWING PRIMITIVE 3	11-67
INQUIRE HLHSR FACILITIES	11-68
INQUIRE HLHSR MODE	11-69
INQUIRE INPUT QUEUE OVERFLOW	11-70
INQUIRE LEVEL OF GKS	11-71
INQUIRE LINETYPE	11-72
INQUIRE LINEWIDTH SCALE FACTOR	11-73
INQUIRE LIST OF AVAILABLE GENERALIZED DRAWING PRIMITIVES	11-74
INQUIRE LIST OF AVAILABLE GENERALIZED DRAWING PRIMITIVES 3	11-75
INQUIRE LIST OF AVAILABLE WORKSTATION TYPES	11-76
INQUIRE LIST OF COLOUR INDICES	11-77
INQUIRE LIST OF EDGE INDICES	11-78
INQUIRE LIST OF FILL AREA INDICES	11-79
INQUIRE LIST OF NORMALIZATION TRANSFORMATION NUMBERS	11-80
INQUIRE LIST OF PATTERN INDICES	11-81
INQUIRE LIST OF POLYLINE INDICES	11-82
INQUIRE LIST OF POLYMARKER INDICES	11-83
INQUIRE LIST OF TEXT INDICES	11-84

INQUIRE LIST OF VIEW INDICES	11-85
INQUIRE LOCATOR DEVICE STATE	11-86
INQUIRE LOCATOR DEVICE STATE 3	11-88
INQUIRE MARKER SIZE SCALE FACTOR	11-90
INQUIRE MARKER TYPE	11-91
INQUIRE MAXIMUM LENGTH OF WORKSTATION STATE TABLES	11-92
INQUIRE MAXIMUM LENGTH OF WORKSTATION STATE TABLES 3	11-93
INQUIRE MAXIMUM NORMALIZATION TRANSFORMATION NUMBER	11-94
INQUIRE MORE SIMULTANEOUS EVENTS	11-95
INQUIRE NAME OF OPEN SEGMENT	11-96
INQUIRE NORMALIZATION TRANSFORMATION	11-97
INQUIRE NORMALIZATION TRANSFORMATION 3	11-98
INQUIRE NUMBER OF AVAILABLE LOGICAL INPUT DEVICES	11-99
INQUIRE NUMBER OF SEGMENT PRIORITIES SUPPORTED	11-100
INQUIRE OPERATING STATE VALUE	11-101
INQUIRE PATTERN FACILITIES	11-102
INQUIRE PATTERN REFERENCE POINT	11-103
INQUIRE PATTERN REPRESENTATION	11-104
INQUIRE PATTERN SIZE	11-105
INQUIRE PICK DEVICE STATE	11-106
INQUIRE PICK DEVICE STATE 3	11-108
INQUIRE PIXEL	11-110
INQUIRE PIXEL ARRAY	11-111
INQUIRE PIXEL ARRAY DIMENSIONS	11-113
INQUIRE POLYLINE COLOUR INDEX	11-114
INQUIRE POLYLINE FACILITIES	11-115
INQUIRE POLYLINE INDEX	11-116
INQUIRE POLYLINE REPRESENTATION	11-117
INQUIRE POLYMARKER COLOUR INDEX	11-118
INQUIRE POLYMARKER FACILITIES	11-119
INQUIRE POLYMARKER INDEX	11-120
INQUIRE POLYMARKER REPRESENTATION	11-121
INQUIRE PREDEFINED COLOUR REPRESENTATION	11-123
INQUIRE PREDEFINED EDGE REPRESENTATION	11-124
INQUIRE PREDEFINED FILL AREA REPRESENTATION	11-125
INQUIRE PREDEFINED PATTERN REPRESENTATION	11-126
INQUIRE PREDEFINED POLYLINE REPRESENTATION	11-127
INQUIRE PREDEFINED POLYMARKER REPRESENTATION	11-128
INQUIRE PREDEFINED TEXT REPRESENTATION	11-129
INQUIRE PREDEFINED VIEW REPRESENTATION	11-130
INQUIRE SEGMENT ATTRIBUTES	11-131
INQUIRE SEGMENT ATTRIBUTES 3	11-133
INQUIRE SET OF ACTIVE WORKSTATIONS	11-135

INQUIRE SET OF ASSOCIATED WORKSTATIONS	11-136
INQUIRE SET OF OPEN WORKSTATIONS	11-137
INQUIRE SET OF SEGMENT NAMES IN USE	11-138
INQUIRE SET OF SEGMENT NAMES ON WORKSTATION	11-139
INQUIRE STRING DEVICE STATE	11-140
INQUIRE STRING DEVICE STATE (FORTRAN-77 Subset)	11-142
INQUIRE STRING DEVICE STATE 3	11-144
INQUIRE STRING DEVICE STATE 3 (FORTRAN-77 Subset)	11-146
INQUIRE STROKE DEVICE STATE	11-148
INQUIRE STROKE DEVICE STATE 3	11-150
INQUIRE TEXT ALIGNMENT	11-152
INQUIRE TEXT COLOUR INDEX	11-153
INQUIRE TEXT EXTENT	11-154
INQUIRE TEXT EXTENT (FORTRAN-77 Subset)	11-155
INQUIRE TEXT EXTENT 3	11-156
INQUIRE TEXT EXTENT 3 (FORTRAN-77 Subset)	11-157
INQUIRE TEXT FACILITIES	11-158
INQUIRE TEXT FONT AND PRECISION	11-160
INQUIRE TEXT INDEX	11-162
INQUIRE TEXT PATH	11-163
INQUIRE TEXT REPRESENTATION	11-164
INQUIRE VALUATOR DEVICE STATE	11-166
INQUIRE VALUATOR DEVICE STATE 3	11-168
INQUIRE VIEW FACILITIES	11-170
INQUIRE VIEW REPRESENTATION 3	11-171
INQUIRE WORKSTATION CATEGORY	11-173
INQUIRE WORKSTATION CLASSIFICATION	11-174
INQUIRE WORKSTATION CONNECTION AND TYPE	11-175
INQUIRE WORKSTATION DEFERRAL AND UPDATE STATES	11-176
INQUIRE WORKSTATION MAXIMUM NUMBERS	11-178
INQUIRE WORKSTATION STATE	11-179
INQUIRE WORKSTATION TRANSFORMATION	11-180
INQUIRE WORKSTATION TRANSFORMATION 3	11-182

12 Error-Handling Functions

12.1	Function Descriptions	12-2
	EMERGENCY CLOSE GKS	12-3
	ERROR HANDLING	12-4
	ERROR LOGGING	12-5
12.2	Program Example	12-6

A DEC GKS Error Messages

A.1	Operating State Errors	A-1
A.2	Workstation Errors	A-2
A.3	Transformation Function Errors	A-5
A.4	Attribute Function Errors	A-6
A.5	Output Function Errors	A-10
A.6	Segment Function Errors	A-11
A.7	Input Function Errors	A-12
A.8	Metafile Function Errors	A-14
A.9	Escape Function Errors	A-15
A.10	Miscellaneous Errors	A-15
A.11	System Errors	A-16
A.12	FORTRAN Binding Errors	A-19
A.13	Implementation-Specific Errors	A-20
A.14	Fatal Errors	A-36

B Constants

C Program Example

D Function Identifiers

E Initial Attribute Values

E.1	Initial Polyline Attributes	E-1
E.2	Initial Polymarker Attributes	E-2
E.3	Initial Text Attributes	E-2
E.4	Initial Fill Area Attributes	E-3
E.5	Initial Fill Area Set Attributes	E-3
E.6	Initial Segment Attributes	E-4
E.7	Initial Normalization Transformation Settings	E-4

F Differences in GKS Implementations

F.1	Global Differences	F-1
-----	--------------------------	-----

Index

Examples

12-1	Creating an Error Handler	12-6
------	---------------------------------	------

Tables

A-1	Operating State Errors	A-1
A-2	Workstation Errors	A-2
A-3	Transformation Function Errors	A-5
A-4	Attribute Function Errors	A-6
A-5	Output Function Errors	A-10
A-6	Segment Function Errors	A-11

A-7	Input Function Errors	A-12
A-8	Metafile Function Errors	A-14
A-9	Escape Function Errors	A-15
A-10	Miscellaneous Errors	A-16
A-11	System Errors	A-16
A-12	FORTRAN Binding Errors	A-19
A-13	Implementation-Specific Errors	A-20
A-14	Fatal Errors	A-36
B-1	Constants	B-1
C-1	Binding-Specific Code Examples	C-1
D-1	FORTRAN Binding Function Identifiers	D-1
F-1	Global Differences	F-1

Preface

This manual contains complete descriptions for the FORTRAN binding functions provided for DEC GKS. Use this reference material to program DEC GKS on any supported operating system, using any of the languages supported by DEC GKS.

Intended Audience

This manual is for programmers who have experience developing graphics applications in one of the languages supported by DEC GKS. They also should be familiar with the principles of programming DEC GKS, as described in the *DEC GKS User's Guide*.

Structure of This Document

This manual is divided into two parts. Each chapter deals with a specific subject or group of functions, describing the syntax and arguments for each function. The appendixes provide additional information you may find useful. Part 1 includes the following chapters:

- Chapter 1 provides an introduction to DEC GKS.
- Chapter 2 provides information about DEC GKS and the VMS™ operating system.
- Chapter 3 provides information about DEC GKS and the ULTRIX™ operating system.
- Chapter 4 describes the functions you use to control DEC GKS and workstation environments.
- Chapter 5 describes the functions you use to generate output primitives.
- Chapter 6 describes the functions you use to generate attributes.
- Chapter 7 describes the functions you use to set up and perform normalization and workstation transformations.
- Chapter 8 describes the functions you use to store output primitives in segments.
- Chapter 9 describes the functions you use to accept input from workstations.
- Chapter 10 describes the functions you use to store graphic images as metafiles.

Part 2 includes the following chapters and appendixes:

- Chapter 11 describes the functions you use to inquire for information about the capabilities and state of the DEC GKS system.
- Chapter 12 describes the functions you use to handle errors.

- Appendix A lists DEC GKS error codes, along with the corresponding severity code and message for each one.
- Appendix B lists constants defined for the FORTRAN binding interface.
- Appendix C provides a list of code examples available throughout this manual, listed alphabetically, by function.
- Appendix D lists the DEC GKS functions and the corresponding FORTRAN binding names.
- Appendix E lists specific input values that apply to all DEC GKS workstations that perform both input and output.
- Appendix F provides implementation-specific information about DEC GKS.

Associated Documents

You may find the following documents useful when using DEC GKS:

- *DEC GKS User's Guide*—for programmers who need information that supplements the DEC GKS binding manuals
- *DEC GKS GKS\$ Binding Reference Manual*—for programmers who need specific syntax and argument descriptions for the GKS\$ binding
- *DEC GKS GKS3D\$ Binding Reference Manual*—for programmers who need specific syntax and argument descriptions for the GKS3D\$ binding
- *DEC GKS C Binding Reference Manual*—for programmers who need specific syntax and argument descriptions for the C binding
- *Device Specifics Reference Manual for DEC GKS and DEC PHIGS*—for programmers who need information about specific devices
- *Building a Device Handler System for DEC GKS and DEC PHIGS*—for programmers who need to build workstation graphics handlers

Conventions

The following conventions are used in this manual:

Convention	Meaning
<code>Return</code>	The symbol <code>RETURN</code> represents a single stroke of the Return key on a terminal.
Boldface text	Boldface text represents the introduction of a new term. In interactive examples, user input appears in boldface type.
<i>Italic text</i>	Italic text indicates a parameter name or a book name. DEC GKS description table and state list entry names, and workstation description table and state list entry names are also italicized.
UPPERCASE TEXT	Uppercase text indicates a DEC GKS function or symbol name.
.	A vertical ellipsis indicates that not all of the text of a program or program output is illustrated. Only relevant material is shown in the example.
...	A horizontal ellipsis indicates that additional arguments, options, or values can be entered. A comma preceding the ellipsis indicates that successive items must be separated by commas. Horizontal ellipses in illustrations indicate that there is information not illustrated that either precedes or follows the information included in the illustration itself.
[]	Square brackets, in function synopses and a few other contexts, indicate that a syntactic element is optional.

Inquiry Functions

Insert tabbed divider here. Then discard this sheet.



Inquiry Functions

The DEC GKS inquiry functions allow you to obtain current and default values for the operating state, output function attributes, deferral and regeneration modes, transformations, segments, and device capabilities. DEC GKS writes the values from the state lists and description tables to the inquiry function arguments.

The following list describes the tables and lists that are sources of information for many of the inquiry functions:

Table/List	Description
GKS description table	<p>This table contains information about the DEC GKS implementation you are using, such as the level of GKS (with DEC GKS, level 2c), the number of available workstation types, the list of workstation types, and the maximum allowable open workstations.</p> <p>If you are transporting your programs from one implementation of GKS to another, you may need to inquire about the implementation level of GKS on a given system, so your program does not call unsupported functions.</p>
Workstation description table	<p>This type of table contains information about one particular workstation, such as the workstation type, the workstation category, the device-specific maximum coordinate values, and the different bundled output attribute values. Each graphics handler contains a workstation description table describing that particular device.</p> <p>If your DEC GKS application uses more than one workstation at a time, or if you are unsure of the capabilities of your workstation, you may need to inquire about the values contained in the workstation description table.</p>
GKS state list	<p>This list contains entries that specify the current DEC GKS values such as the set of open workstations (if any), the current normalization transformation number, and the current character height.</p> <p>If you need to check the alterable DEC GKS values, you may need to inquire about the values contained in the GKS state list.</p>

Inquiry Functions

Table/List	Description
Workstation state list	<p>For each workstation you open, DEC GKS allocates space for a workstation state list. This list contains entries that specify whether output is “on hold” (deferred), whether or not the surface has to be redrawn to fulfill an output request, whether the workstation surface is “empty” as defined by GKS, and whether the picture on the surface represents all the requests for output by the application program.</p> <p>If you need information concerning the current state of a particular workstation, you may need to inquire about the values contained in the workstation state list.</p>
Segment state list	<p>When you create a segment, DEC GKS creates a segment state list. The segment state list contains entries that specify the segment name, the set of associated workstations, and the detectability of the segment.</p> <p>If you need information concerning a particular segment, you may need to inquire about the values contained in the segment state list.</p>

Note

You cannot inquire from a VWS (VAXstation Workstation Software) workstation description table unless you are logged onto a system running DEC GKS and VWS.

The only other type of information obtained by the inquiry functions is information concerning the color and dimensions of one or more pixels on the workstation surface. To obtain this information, you can use the pixel inquiry functions.

Inquiry function calls are similar; consequently, only the INQUIRE . . . DEVICE STATE functions are illustrated in program examples. For complete examples that use calls to these input inquiry functions, see Chapter 9.

To gain an understanding of when to call certain DEC GKS inquiry functions, see the *DEC GKS User's Guide*. For more information concerning state lists and description tables, see Chapter 4.

11.1 Using the Inquiry Functions

The DEC GKS inquiry functions return information about the DEC GKS tables, lists, and state of the pixels on a given device, by writing values to arguments passed to the function. For example, consider the call:

```
GQLVKS ( ERRIND, LEVEL )
```

The two arguments to the function INQUIRE LEVEL OF GKS are passed as write-only parameters. If this function completes its task successfully, DEC GKS returns the value 0 in the write-only argument *ERRIND*. If this function encounters an error condition (see Section 11.1.1 for detailed information), DEC GKS returns an error status code in the *ERRIND* argument. This function returns the level of the DEC GKS implementation with which you are working in the write-only argument *LEVEL*.

Inquiry Functions

11.1 Using the Inquiry Functions

Some of the inquiry functions have read-only arguments as well. For example, review the following syntax:

`GQLCS (WKID, DEVNUM, TYPE, MLDR, ERRIND, MODE, ESW,
TNR, IPX, IPY, PET, EAREA, LDR, DATREC)`

The first three arguments (*WKID*, *DEVNUM*, *TYPE*) are all read only; DEC GKS needs to know the workstation identifier, the device type, and the type of values to be returned to this function to return the proper values to the other arguments (see Section 11.1.2 for detailed information concerning the argument value type).

The function `INQUIRE LOCATOR DEVICE STATE` illustrates the usefulness of the inquiry functions when requesting input. If you wish to change one of the default input values, you have to assign values to all the input variables, one by one. This can be tedious if you only want to change one or two of the default variable values.

A practical way to initialize all the necessary variables with default input values is to pass the variables to the function `INQUIRE LOCATOR DEVICE STATE`. To initialize the values, do the following:

1. Call the function `INQUIRE LOCATOR DEVICE STATE` to initialize all the input variables.
2. Change the values of the variables you wish to change.
3. Pass all the variables to `INITIALIZE LOCATOR`.

For more information concerning the workstation identifier, see Chapter 4. For more information concerning the input device type or general input concepts, see Chapter 9.

11.1.1 The Error Status Argument

DEC GKS inquiry functions never generate an error, but they can encounter error conditions. The value passed to the *ERRIND* argument determines whether the values passed to the remaining write-only arguments are valid.

Because the inquiry functions obtain values from the description tables and state lists, and because the description tables and state lists are not accessible unless you have called the proper DEC GKS control functions, the inquiry functions may or may not be able to access the values you need. There are other device-dependent situations that would cause a DEC GKS inquiry function to encounter an error condition.

If all values are available, the inquiry function returns the value 0 in the *ERRIND* argument.

If a value is not presently available, the inquiry function returns a number, corresponding to an appropriate DEC GKS error message, in the *ERRIND* argument. If the value passed to the *ERRIND* argument is anything other than the value 0, the values that the inquiry function passed to the remaining arguments are invalid.

For more information concerning the DEC GKS error messages and their numbers, see Appendix A. For more information concerning DEC GKS error handling, see Chapter 12.

Inquiry Functions

11.1 Using the Inquiry Functions

11.1.2 The Value Type Argument

Several of the inquiry functions that take their values from the workstation state list have a return type argument. This argument determines whether DEC GKS returns the exact values that you previously set in the application program, or returns the values the DEC GKS device handlers determine closely approximate the values you requested.

The possible values for this argument are:

Value	Description
GSET	If you specify this constant (or the value 0), the inquiry function returns the requested values exactly as specified in the application program. If you did not assign any values in the application program, the inquiry function returns the default values.
GREALI	If you specify this constant (or the value 1), and if you specified values in your application program that a particular workstation cannot fully support, the inquiry function returns the realized values that closely approximate the values you specified in the application program. If you did not assign any values in the application program, the inquiry function returns the default values.

For example, some devices support a limited number of pick aperture sizes (the size of the tracking prompt used for picking segments). A set aperture size is one set by the application program, and a realized size is used by the graphics handler. Using the function `INQUIRE PICK DEVICE STATE`, you can inquire about both types of values.

For more information concerning pick input, see Chapter 9.

11.2 Function Descriptions

This section describes the DEC GKS inquiry functions in detail.

INQUIRE ASPECT SOURCE FLAGS

Operating States

GKOP, WSOP, WSAC, SGOP

Syntax

GQASF (ERRIND, LASF)

Argument	Data Type	Access	Description
ERRIND	Integer	Write	Error indicator
LASF(13)	Array of integers (constant)	Write	Aspect source flags array

Constants

Defined Argument	Constant	Description
LASF	GBUNDL	Bundled attributes
	GINDIV	Individual attributes

Description

The INQUIRE ASPECT SOURCE FLAGS function returns the value for the *current list of aspect source flags* entry in the GKS state list.

The ASF indicates whether a particular primitive attribute is selected from an attribute bundle, or as an individual attribute selection.

There are 13 nongeometric ASFs. These flags are as follows:

1. Polyline type
2. Polyline width
3. Polyline color
4. Polymarker type
5. Polymarker size
6. Polymarker color
7. Text font and precision
8. Character expansion
9. Character spacing
10. Text color
11. Fill area interior style
12. Fill area style index
13. Fill area color

INQUIRE ASPECT SOURCE FLAGS

See Also

SET ASPECT SOURCE FLAGS

INQUIRE ASPECT SOURCE FLAGS 3
Operating States

GKOP, WSOP, WSAC, SGOP

Syntax

GQASF3 (ERRIND, ASF3)

Argument	Data Type	Access	Description
ERRIND	Integer	Write	Error indicator
ASF3(17)	Array of integers (constant)	Write	Aspect source flags array

Constants

Defined Argument	Constant	Description
ASF3	GBUNDL	Bundled attributes
	GINDIV	Individual attributes

Description

The INQUIRE ASPECT SOURCE FLAGS 3 function returns the value for the *current list of aspect source flags 3* entry in the GKS state list.

The ASF indicates whether a particular primitive attribute is selected from an attribute bundle, or as an individual attribute selection.

There are 17 nongeometric ASFs. These flags are as follows:

1. Polyline type
2. Polyline width
3. Polyline color
4. Polymarker type
5. Polymarker size
6. Polymarker color
7. Text font and precision
8. Character expansion
9. Character spacing
10. Text color
11. Fill area interior style
12. Fill area style index
13. Fill area color

INQUIRE ASPECT SOURCE FLAGS 3

- 14. Edge flag
- 15. Edge type
- 16. Edge width
- 17. Edge color

See Also

SET ASPECT SOURCE FLAGS 3

INQUIRE CHARACTER BASE VECTOR

Operating States

GKOP, WSOP, WSAC, SGOP

Syntax

GQCHB (ERRIND, CHBX, CHBY)

Argument	Data Type	Access	Description
ERRIND	Integer	Write	Error indicator
CHBX, CHBY	Real	Write	Character base vector

Description

The INQUIRE CHARACTER BASE VECTOR function returns the principle base direction value for the text string.

The direction is a two-dimensional vector in the text plane specified in the text output.

See Also

SET CHARACTER UP VECTOR

INQUIRE CHARACTER EXPANSION FACTOR

INQUIRE CHARACTER EXPANSION FACTOR

Operating States

GKOP, WSOP, WSAC, SGOP

Syntax

GQCHXP (ERRIND, CHXP)

Argument	Data Type	Access	Description
ERRIND	Integer	Write	Error indicator
CHXP	Real	Write	Character expansion factor

Description

The INQUIRE CHARACTER EXPANSION FACTOR function returns the value for the *current character expansion factor* entry in the GKS state list.

The character expansion factor specifies the deviation of character width from the defined nominal value.

See Also

SET CHARACTER EXPANSION FACTOR

INQUIRE CHARACTER HEIGHT**Operating States**

GKOP, WSOP, WSAC, SGOP

Syntax

GQCHH (ERRIND, CHH)

Argument	Data Type	Access	Description
ERRIND	Integer	Write	Error indicator
CHH	Real	Write	Character height

Description

The INQUIRE CHARACTER HEIGHT function returns the value for the text attribute *current character height* entry in the GKS state list.

The character height value is workstation independent, expressed in WC values, and used when creating subsequent TEXT output primitives.

See Also

SET CHARACTER HEIGHT

INQUIRE CHARACTER SPACING

INQUIRE CHARACTER SPACING

Operating States

GKOP, WSOP, WSAC, SGOP

Syntax

GQCHSP (ERRIND, CHSP)

Argument	Data Type	Access	Description
ERRIND	Integer	Write	Error indicator
CHSP	Real	Write	Character spacing

Description

The INQUIRE CHARACTER SPACING function returns the value for the *current text spacing* entry in the GKS state list.

DEC GKS measures the spacing between characters as a fraction of the character height; adjusting character height proportionately adjusts spacing. The character spacing value 0.0 places the character bodies next to each other without any separating space contained in the font specification for the letter bodies. Whether the characters actually touch depends on the type of font you use. Positive spacing values increase the space between characters; negative values decrease the space. Using negative spacing values, you can overlap characters or reverse the text so that characters are written in the opposite direction.

See Also

SET CHARACTER SPACING

INQUIRE CHARACTER UP VECTOR
Operating States

GKOP, WSOP, WSAC, SGOP

Syntax

GQCHUP (ERRIND, CHUX, CHUY)

Argument	Data Type	Access	Description
ERRIND	Integer	Write	Error indicator
CHUX, CHUY	Real	Write	Character-up vector

Description

The INQUIRE CHARACTER UP VECTOR function returns the value for the geometric attribute *current character up vector* entry in the GKS state list. The character up vector gives the "up" direction of a character.

DEC GKS uses the value specified in the call to SET CHARACTER UP VECTOR for all subsequent calls to TEXT until another value is specified.

See Also

SET CHARACTER UP VECTOR
TEXT

INQUIRE CHARACTER WIDTH

INQUIRE CHARACTER WIDTH

Operating States

GKOP, WSOP, WSAC, SGOP

Syntax

GQCHW (ERRIND, CHW)

Argument	Data Type	Access	Description
ERRIND	Integer	Write	Error indicator
CHW	Real	Write	Character width

Description

The INQUIRE CHARACTER WIDTH function returns the value for the geometric attribute *current character width* entry in the GKS state list. The character width is calculated using the value set by the SET CHARACTER EXPANSION FACTOR or SET CHARACTER HEIGHT function.

DEC GKS uses the value specified in the calls to SET CHARACTER HEIGHT or SET CHARACTER EXPANSION FACTOR for all subsequent calls to TEXT until another value is specified.

See Also

SET CHARACTER EXPANSION FACTOR
SET CHARACTER HEIGHT
SET CHARACTER SPACING
TEXT

INQUIRE CHOICE DEVICE STATE

Operating States

WSOP, WSAC, SGOP

Syntax

GQCHS (WKID, DEVNUM, MLDR, ERRIND, MODE, ESW, ISTAT, ICHNR, PET, EAREA, LDR, DATREC)

Argument	Data Type	Access	Description
WKID	Integer	Read	Workstation identifier
DEVNUM	Integer	Read	Choice device number
MLDR	Integer	Read	Number of elements in output data record
ERRIND	Integer	Write	Error indicator
MODE	Integer (constant)	Write	Operating mode
ESW	Integer (constant)	Write	Echo flag
ISTAT	Integer (constant)	Write	Initial choice status
ICHNR	Integer	Write	Initial choice number
PET	Integer	Write	Prompt and echo type
EAREA(4)	Array of reals	Write	Device coordinate echo area in the order XMIN, XMAX, YMIN, YMAX
LDR	Integer	Write	Number of array elements used in the output data record
DATREC(MLDR)	Character*80	Write	Output data record

Constants

Defined Argument	Constant	Description
MODE	GREQU	Request mode
	GSAMPL	Sample mode
	GEVENT	Event mode
ESW	GNECHO	Echo disabled
	GECHO	Echo enabled

INQUIRE CHOICE DEVICE STATE

Defined Argument	Constant	Description
ISTAT	GOK	Valid choice number
	GNCHOI	No choice

Description

The INQUIRE CHOICE DEVICE STATE function returns the current state of the given choice-class logical input device.

The data record returned by this function is the input data record associated with the PET returned in the argument *PET*. Use the UNPACK DATA RECORD function to access this information.

The data record components for each of the possible PETs are listed in the introduction to Chapter 9. Note that this information refers to arguments in the PACK DATA RECORD function. Because the listed arguments have the same names and meanings in both the PACK DATA RECORD and UNPACK DATA RECORD functions, you can associate the data record components easily with either function.

See Also

INITIALIZE CHOICE

SET CHOICE MODE

Example 9–3 for a program example using an INQUIRE . . . DEVICE STATE function

INQUIRE CHOICE DEVICE STATE 3

Operating States

WSOP, WSAC, SGOP

Syntax

GQCHS3 (WKID, DEVNUM, MLDR, ERRIND, MODE, ESW, ISTAT, ICHNR, PET, EVOL, LDR, DATREC)

Argument	Data Type	Access	Description
WKID	Integer	Read	Workstation identifier
DEVNUM	Integer	Read	Choice device number
MLDR	Integer	Read	Number of elements in output data record
ERRIND	Integer	Write	Error indicator
MODE	Integer (constant)	Write	Operating mode
ESW	Integer (constant)	Write	Echo flag
ISTAT	Integer (constant)	Write	Initial choice status
ICHNR	Integer	Write	Default choice state
PET	Integer	Write	Prompt and echo type
EVOL(6)	Array of reals	Write	Device coordinate echo volume in the order XMIN, XMAX, YMIN, YMAX, ZMIN, ZMAX
LDR	Integer	Write	Number of array elements used in the output data record
DATREC(MLDR)	Character*80	Write	Output data record

Constants

Defined Argument	Constant	Description
MODE	GREQU	Request mode
	GSAMPL	Sample mode
	GEVENT	Event mode
ESW	GNECHO	Echo disabled
	GECHO	Echo enabled

INQUIRE CHOICE DEVICE STATE 3

Defined Argument	Constant	Description
ISTAT	GOK	Valid choice number
	GNCHOI	No choice

Description

The INQUIRE CHOICE DEVICE STATE 3 function returns the current state of the given choice-class logical input device.

The data record returned by this function is the input data record associated with the PET returned in the argument *PET*. Use the UNPACK DATA RECORD function to access this information.

The data record components for each of the possible PETs are listed in the introduction to Chapter 9. Note that this information refers to arguments in the PACK DATA RECORD function. Because the listed arguments have the same names and meanings in both the PACK DATA RECORD and UNPACK DATA RECORD functions, you can associate the data record components easily with either function.

See Also

INITIALIZE CHOICE 3
SET CHOICE MODE

Example 9–3 for a program example using an INQUIRE . . . DEVICE STATE function

INQUIRE CLIPPING

Operating States

GKOP, WSOP, WSAC, SGOP

Syntax

GQCLIP (ERRIND, CLSW, CLRECT)

Argument	Data Type	Access	Description
ERRIND	Integer	Write	Error indicator
CLSW	Integer (constant)	Write	Clipping flag
CLRECT(4)	Array of reals	Write	Clipping rectangle in NDC points in the order XMIN, XMAX, YMIN, YMAX

Constants

Defined Argument	Constant	Description
CLSW	GNCLIP	Clipping disabled
	GCLIP	Clipping enabled

Description

The INQUIRE CLIPPING function returns the value in NDC points of the current clipping rectangle.

See Also

SET CLIPPING INDICATOR

INQUIRE CLIPPING 3

INQUIRE CLIPPING 3

Operating States

GKOP, WSOP, WSAC, SGOP

Syntax

GQCLP3 (ERRIND, CLSW, CLVOL)

Argument	Data Type	Access	Description
ERRIND	Integer	Write	Error indicator
CLSW	Integer (constant)	Write	Clipping flag
CLVOL(6)	Array of reals	Write	Clipping volume in NDC points in the order XMIN, XMAX, YMIN, YMAX, ZMIN, ZMAX

Constants

Defined Argument	Constant	Description
CLSW	GNCLIP	Clipping disabled
	GCLIP	Clipping enabled

Description

The INQUIRE CLIPPING 3 function returns the value in NDC points of the current clipping volume.

See Also

SET CLIPPING INDICATOR

INQUIRE COLOUR FACILITIES

Operating States

GKOP, WSOP, WSAC, SGOP

Syntax

GQCF (WTYPE, ERRIND, NCOLI, COLA, NPCI)

Argument	Data Type	Access	Description
WTYPE	Integer	Read	Workstation type
ERRIND	Integer	Write	Error indicator
NCOLI	Integer	Write	Number of colors
COLA	Integer (constant)	Write	Color or monochrome flag
NPCI	Integer	Write	Number of predefined color indexes

Constants

Defined Argument	Constant	Description
COLA	GMONOC	Monochrome device
	GCOLOR	Color device

Description

The INQUIRE COLOUR FACILITIES function returns the number of available colors, the color availability, and the number of predefined color bundles of a specified workstation.

INQUIRE COLOUR MODEL

INQUIRE COLOUR MODEL

Operating States

WSOP, WSAC, SGOP

Syntax

GQCMD (WKID, ERRIND, CMODEL)

Argument	Data Type	Access	Description
WKID	Integer	Read	Workstation identifier
ERRIND	Integer	Write	Error indicator
CMODEL	Integer	Write	Color model used

Description

The INQUIRE COLOUR MODEL function returns the number of the current color model.

INQUIRE COLOUR MODEL FACILITIES

Operating States

GKOP, WSOP, WSAC, SGOP

Syntax

GQC MDF (WTYPE, N, ERRIND, OL, CMOD, DFCMOD)

Argument	Data Type	Access	Description
WTYPE	Integer	Read	Workstation type
N	Integer	Read	List element requested
ERRIND	Integer	Write	Error indicator
OL	Integer	Write	Number of available color models
CMOD	Integer	Write	Nth element of list of available color models
DFCMOD	Integer	Write	Default color model

Description

The INQUIRE COLOUR MODEL FACILITIES function returns the number of available color models, the requested element in the list of color models, and the default color model for the specified workstation type.

INQUIRE COLOUR REPRESENTATION

INQUIRE COLOUR REPRESENTATION

Operating States

WSOP, WSAC, SGOP

Syntax

GQCR (WKID, COLI, TYPE, ERRIND, COMP1, COMP2, COMP3)

Argument	Data Type	Access	Description
WKID	Integer	Read	Workstation identifier
COLI	Integer	Read	Color index
TYPE	Integer (constant)	Read	Type of values returned
ERRIND	Integer	Write	Error indicator
COMP1, COMP2, COMP3	Real	Write	Color components

Constants

Defined Argument	Constant	Description
TYPE	GSET	Use exact state list values.
	GREALI	Use values approximated by the graphics handler.

Description

The INQUIRE COLOUR REPRESENTATION function returns the color triplet values, which are the color coordinates of the current color model on the workstation. (See the SET COLOUR REPRESENTATION function in Chapter 6.)

If the specified color index is not in the color table on the specified workstation, and the specified type of returned values is REALIZED, the representation for color 1 is returned.

See Also

SET COLOUR REPRESENTATION

INQUIRE CURRENT EDGE COLOUR INDEX

Operating States

GKOP, WSOP, WSAC, SGOP

Syntax

GQEDCI (ERRIND, COLI)

Argument	Data Type	Access	Description
ERRIND	Integer	Write	Error indicator
COLI	Integer	Write	Edge color index

Description

The INQUIRE CURRENT EDGE COLOUR INDEX function queries the GKS state list and returns the current edge color index.

INQUIRE CURRENT EDGE FLAG

INQUIRE CURRENT EDGE FLAG

Operating States

GKOP, WSOP, WSAC, SGOP

Syntax

GQEDFG (ERRIND, EDFLAG)

Argument	Data Type	Access	Description
ERRIND	Integer	Write	Error indicator
EDFLAG	Integer (constant)	Write	Edge flag

Constants

Defined Argument	Constant	Description
EDFLAG	GOFF	Edge off
	GON	Edge on

Description

The INQUIRE CURRENT EDGE FLAG function returns the value for the *current edge flag* entry in the GKS state list.

This flag enables the display of subsequent fill area set output primitives when the current edge ASF has been set to INDIVIDUAL by the function SET ASPECT SOURCE FLAGS 3. If this ASF is set to BUNDLED, the current edge flag has no effect.

INQUIRE CURRENT EDGE INDEX

Operating States

GKOP, WSOP, WSAC, SGOP

Syntax

GQEDI (ERRIND, EDI)

Argument	Data Type	Access	Description
ERRIND	Integer	Write	Error indicator
EDI	Integer	Write	Edge index

Description

The INQUIRE LIST OF EDGE INDICES function returns the number of defined edge index values and the index value of the requested element in the list of defined edge indexes.

INQUIRE CURRENT EDGETYPE

INQUIRE CURRENT EDGETYPE

Operating States

GKOP, WSOP, WSAC, SGOP

Syntax

GQEDT (ERRIND, EDTYPE)

Argument	Data Type	Access	Description
ERRIND	Integer	Write	Error indicator
EDTYPE	Integer	Write	Edge type

Description

The INQUIRE CURRENT EDGETYPE function queries the GKS state list and returns the current edge type.

INQUIRE CURRENT EDGEWIDTH SCALE FACTOR

INQUIRE CURRENT EDGEWIDTH SCALE FACTOR

Operating States

GKOP, WSOP, WSAC, SGOP

Syntax

GQEWSC (ERRIND, EDWSF)

Argument	Data Type	Access	Description
ERRIND	Integer	Write	Error indicator
EDWSF	Real	Write	Edge width scale factor

Description

The INQUIRE CURRENT EDGEWIDTH SCALE FACTOR function returns the current edgewidth scale factor value from the GKS state list.

INQUIRE CURRENT HLHSR IDENTIFIER VALUE

INQUIRE CURRENT HLHSR IDENTIFIER VALUE

Operating States

GKOP, WSOP, WSAC, SGOP

Syntax

GQHRIV (ERRIND, HRID)

Argument	Data Type	Access	Description
ERRIND	Integer	Write	Error indicator
HRID	Integer	Write	HLHSR identifier

Description

The INQUIRE CURRENT HLHSR IDENTIFIER VALUE function returns the current hidden line and hidden surface removal (HLHSR) identifier. Implementation of HLHSR is described in the *DEC GKS User's Guide*.

INQUIRE CURRENT NORMALIZATION TRANSFORMATION NUMBER

INQUIRE CURRENT NORMALIZATION TRANSFORMATION NUMBER

Operating States

GKOP, WSOP, WSAC, SGOP

Syntax

GQCNTN (ERRIND, CTNR)

Argument	Data Type	Access	Description
ERRIND	Integer	Write	Error indicator
CTNR	Integer	Write	Current normalization transformation number

Description

The INQUIRE CURRENT NORMALIZATION TRANSFORMATION NUMBER function returns the normalization transformation number currently in effect.

See Also

SELECT NORMALIZATION TRANSFORMATION

INQUIRE CURRENT PATTERN REFERENCE POINT AND VECTORS

INQUIRE CURRENT PATTERN REFERENCE POINT AND VECTORS

Operating States

GKOP, WSOP, WSAC, SGOP

Syntax

GQPRPV (ERRIND, RFX, RFY, RFZ, RFVX, RFVY, RFVZ)

Argument	Data Type	Access	Description
ERRIND	Integer	Write	Error indicator
RFX, RFY, RFZ	Real	Write	Current pattern reference point coordinates
RFVX(2), RFVY(2), RFVZ(2)	Real	Write	Current pattern reference vector coordinates

Description

The INQUIRE CURRENT PATTERN REFERENCE POINT AND VECTORS function returns the value for the geometric attributes *current pattern reference point* and *current pattern reference vectors* entries in the GKS state list.

The point attribute represents the starting point for a pattern used to fill a designated area. When the currently selected fill area interior style is PATTERN, the vectors attribute is used in conjunction with the current pattern width and height vectors to display the FILL AREA and FILL AREA SET output primitives.

INQUIRE CURRENT PICK IDENTIFIER VALUE

Operating States

GKOP, WSOP, WSAC, SGOP

Syntax

GQPKID (ERRIND, PKID)

Argument	Data Type	Access	Description
ERRIND	Integer	Write	Error indicator
PKID	Integer	Write	Pick identifier

Description

The INQUIRE CURRENT PICK IDENTIFIER VALUE function queries the GKS state list and returns the current GKS pick identifier.

See Also

SET PICK IDENTIFIER

INQUIRE CURRENT VIEW INDEX

INQUIRE CURRENT VIEW INDEX

Operating States

GKOP, WSOP, WSAC, SGOP

Syntax

GQVWI (ERRIND, VIEWI)

Argument	Data Type	Access	Description
ERRIND	Integer	Write	Error indicator
VIEWI	Integer	Write	View index

Description

The INQUIRE CURRENT VIEW INDEX function returns the current view index value from the GKS state list.

INQUIRE DEFAULT CHOICE DEVICE DATA

Operating States

GKOP, WSOP, WSAC, SGOP

Syntax

GQDCH (WTYPE, DEVNUM, N, MLDR, ERRIND, MALT, OL, PET, EAREA, LDR, DATREC)

Argument	Data Type	Access	Description
WTYPE	Integer	Read	Workstation type
DEVNUM	Integer	Read	Choice device number
N	Integer	Read	Requested list element from the list of available prompt and echo types
MLDR	Integer	Read	Number of elements in output data record
ERRIND	Integer	Write	Error indicator
MALT	Integer	Write	Maximum number of alternatives
OL	Integer	Write	Number of available prompt and echo types
PET	Integer	Write	<i>N</i> th element of the list of available prompt and echo types
EAREA(4)	Array of reals	Write	Device coordinate echo area in the order XMIN, XMAX, YMIN, YMAX
LDR	Integer	Write	Number of array elements used in the output data record
DATREC(MLDR)	Character*80	Write	Output data record

Description

The INQUIRE DEFAULT CHOICE DEVICE DATA function returns the default values for the specified choice-class logical input device on the specified workstation.

The default data record returned by this function is the input data record associated with PET 1. Use the UNPACK DATA RECORD function to access the information.

The data record components for each of the possible PETs are listed in the introduction to Chapter 9. Note that this information refers to arguments in the PACK DATA RECORD function. Because the listed arguments have the same names and meanings in both the PACK DATA RECORD and UNPACK DATA RECORD functions, you can associate the data record components easily with either function.

INQUIRE DEFAULT CHOICE DEVICE DATA 3

INQUIRE DEFAULT CHOICE DEVICE DATA 3

Operating States

GKOP, WSOP, WSAC, SGOP

Syntax

GQDCH3 (WTYPE, DEVNUM, N, MLDR, ERRIND, MALT, NPE, PE, EVOL, LDR, DATREC)

Argument	Data Type	Access	Description
WTYPE	Integer	Read	Workstation type
DEVNUM	Integer	Read	Choice device number
N	Integer	Read	Requested list element from the list of available prompt and echo types
MLDR	Integer	Read	Number of elements in output data record
ERRIND	Integer	Write	Error indicator
MALT	Integer	Read	Maximum number of choice alternatives
NPE	Integer	Write	Number of available prompt and echo types
PE	Integer	Write	<i>N</i> th element of the list of available prompt and echo types
EVOL(6)	Array of reals	Write	Device coordinate echo volume in the order XMIN, XMAX, YMIN, YMAX, ZMIN, ZMAX
LDR	Integer	Write	Number of array elements used in the output data record
DATREC(MLDR)	Character*80	Write	Output data record

Description

The INQUIRE DEFAULT CHOICE DEVICE DATA 3 function returns the default values for the specified choice-class logical input device on the specified workstation.

The default data record returned by this function is the input data record associated with PET 1. Use the UNPACK DATA RECORD function to access the information.

The data record components for each of the possible PETs are listed in the introduction to Chapter 9. Note that this information refers to arguments in the PACK DATA RECORD function. Because the listed arguments have the same names and meanings in both the PACK DATA RECORD and UNPACK DATA RECORD functions, you can associate the data record components easily with either function.

INQUIRE DEFAULT DEFERRAL STATE VALUES

Operating States

GKOP, WSOP, WSAC, SGOP

Syntax

GQDDS (WTYPE, ERRIND, DEFMOD, REGMOD)

Argument	Data Type	Access	Description
WTYPE	Integer	Read	Workstation type
ERRIND	Integer	Write	Error indicator
DEFMOD	Integer (constant)	Write	Default value for deferral mode
REGMOD	Integer (constant)	Write	Default value for implicit regeneration mode

Constants

Defined Argument	Constant	Description
DEFMOD	GASAP	Generate images as soon as possible.
	GBNIG	Generate images before input is requested globally.
	GBNIL	Generate images before input is requested locally.
	GASTI	Generate images sometime. Exact time is not guaranteed.
REGMOD	GSUPPD	Image regeneration is suppressed.
	GALLOW	Image regeneration is allowed.

Description

The INQUIRE DEFAULT DEFERRAL STATE VALUES function returns the default deferral and implicit regeneration modes.

See Also

SET DEFERRAL STATE

INQUIRE DEFAULT LOCATOR DEVICE DATA

INQUIRE DEFAULT LOCATOR DEVICE DATA

Operating States

GKOP, WSOP, WSAC, SGOP

Syntax

GQDLC (WTYPE, DEVNUM, N, MLDR, ERRIND, DPX, DPY, OL, PET, EAREA, LDR, DATREC)

Argument	Data Type	Access	Description
WTYPE	Integer	Read	Workstation type
DEVNUM	Integer	Read	Locator device number
N	Integer	Read	Requested list element from the list of available prompt and echo types
MLDR	Integer	Read	Number of elements in output data record
ERRIND	Integer	Write	Error indicator
DPX, DPY	Integer	Write	Default initial locator position
OL	Integer	Write	Number of available prompt and echo types
PET	Integer	Write	<i>N</i> th element of the list of available prompt and echo types
EAREA(4)	Array of reals	Write	Device coordinate echo area in the order XMIN, XMAX, YMIN, YMAX
LDR	Integer	Write	Number of array elements used in the output data record
DATREC(MLDR)	Character*80	Write	Output data record

Description

The INQUIRE DEFAULT LOCATOR DEVICE DATA function returns the default values for the specified locator-class logical input device on the specified workstation.

The default data record returned by this function is the input data record associated with PET 1. There are no data record components associated with locator-class PET 1. All the default information is returned in the other function arguments. The number of array elements used in the data record is 0.

INQUIRE DEFAULT LOCATOR DEVICE DATA 3

Operating States

GKOP, WSOP, WSAC, SGOP

Syntax

GQDLC3 (WTYPE, DEVNUM, N, MLDR, ERRIND, DPX, DPY, DPZ, NPE, PE, EVOL, LDR, DATREC)

Argument	Data Type	Access	Description
WTYPE	Integer	Read	Workstation type
DEVNUM	Integer	Read	Locator device number
N	Integer	Read	Requested list element from the list of available prompt and echo types
MLDR	Integer	Read	Number of elements in output data record
ERRIND	Integer	Write	Error indicator
DPX, DPY, DPZ	Real	Write	Default initial locator position
NPE	Integer	Write	Number of available prompt and echo types in array
PE	Integer	Write	<i>N</i> th element of the list of available prompt and echo types
EVOL(6)	Array of reals	Write	Device coordinate echo volume in the order XMIN, XMAX, YMIN, YMAX, ZMIN, ZMAX
LDR	Integer	Write	Number of array elements used in the output data record
DATREC(MLDR)	Character*80	Write	Output data record

Description

The INQUIRE DEFAULT LOCATOR DEVICE DATA 3 function returns the default values for the specified locator-class logical input device on the specified workstation.

The default data record returned by this function is the input data record associated with PET 1. There are no data record components associated with locator-class PET 1. All the default information is returned in the other function arguments. The number of array elements used in the data record is 0.

INQUIRE DEFAULT PICK DEVICE DATA

INQUIRE DEFAULT PICK DEVICE DATA

Operating States

GKOP, WSOP, WSAC, SGOP

Syntax

GQDPK (WTYPE, DEVNUM, N, MLDR, ERRIND, OL, PET, EAREA, LDR, DATREC)

Argument	Data Type	Access	Description
WTYPE	Integer	Read	Workstation type
DEVNUM	Integer	Read	Pick device number
N	Integer	Read	Requested list element from the list of available prompt and echo types
MLDR	Integer	Read	Number of elements in output data record
ERRIND	Integer	Write	Error indicator
OL	Integer	Write	Number of available prompt and echo types
PET	Integer	Write	Nth element of the list of available prompt and echo types
EAREA(4)	Array of reals	Write	Device coordinate echo area in the order XMIN, XMAX, YMIN, YMAX
LDR	Integer	Write	Number of array elements used in the output data record
DATREC(MLDR)	Character*80	Write	Output data record

Description

The INQUIRE DEFAULT PICK DEVICE DATA function returns the default values for the specified pick-class logical input device on the specified workstation.

The default data record returned by this function is the input data record associated with PET 1. Use the UNPACK DATA RECORD function to access the information.

The data record components for each of the possible PETs are listed in the introduction to Chapter 9. Note that this information refers to arguments in the PACK DATA RECORD function. Because the listed arguments have the same names and meanings in both the PACK DATA RECORD and UNPACK DATA RECORD functions, you can associate the data record components easily with either function.

INQUIRE DEFAULT PICK DEVICE DATA 3

Operating States

GKOP, WSOP, WSAC, SGOP

Syntax

GQDPK3 (WTYPE, DEVNUM, N, MLDR, ERRIND, NPE, PE, EVOL, LDR, DATREC)

Argument	Data Type	Access	Description
WTYPE	Integer	Read	Workstation type
DEVNUM	Integer	Read	Pick device number
N	Integer	Read	Requested list element from the list of available prompt and echo types
MLDR	Integer	Read	Number of elements in output data record
ERRIND	Integer	Write	Error indicator
NPE	Integer	Write	Number of available prompt and echo types
PE	Integer	Write	Nth element of the list of available prompt and echo types
EVOL(6)	Array of reals	Write	Device coordinate echo volume in the order XMIN, XMAX, YMIN, YMAX, ZMIN, ZMAX
LDR	Integer	Write	Number of array elements used in the output data record
DATREC(MLDR)	Character*80	Write	Output data record

Description

The INQUIRE DEFAULT PICK DEVICE DATA 3 function returns the default values for the specified pick-class logical input device on the specified workstation.

The default data record returned by this function is the input data record associated with PET 1. Use the UNPACK DATA RECORD function to access the information.

The data record components for each of the possible PETs are listed in the introduction to Chapter 9. Note that this information refers to arguments in the PACK DATA RECORD function. Because the listed arguments have the same names and meanings in both the PACK DATA RECORD and UNPACK DATA RECORD functions, you can associate the data record components easily with either function.

INQUIRE DEFAULT STRING DEVICE DATA

INQUIRE DEFAULT STRING DEVICE DATA

Operating States

GKOP, WSOP, WSAC, SGOP

Syntax

GQDST (WTYPE, DEVNUM, N, MLDR, ERRIND, MBUFF, OL, PET, EAREA, BUFLLEN, LDR, DATREC)

Argument	Data Type	Access	Description
WTYPE	Integer	Read	Workstation type
DEVNUM	Integer	Read	String device number
N	Integer	Read	Requested list element from the list of available prompt and echo types
MLDR	Integer	Read	Number of elements in output data record
ERRIND	Integer	Write	Error indicator
MBUFF	Integer	Write	Maximum string buffer size
OL	Integer	Write	Number of available prompt and echo types
PET	Integer	Write	<i>N</i> th element of the list of available prompt and echo types
EAREA(4)	Array of reals	Write	Device coordinate echo area in the order XMIN, XMAX, YMIN, YMAX
BUFLLEN	Integer	Write	Input buffer size
LDR	Integer	Write	Number of array elements used in the output data record
DATREC(MLDR)	Character*80	Write	Output data record

Description

The INQUIRE DEFAULT STRING DEVICE DATA function returns the default values for the specified string-class logical input device on the specified workstation.

The default data record returned by this function is the input data record associated with PET 1. There are no data record components associated with string-class PET 1. All the default information is returned in the other function arguments. The number of array elements used in the data record is 0.

INQUIRE DEFAULT STRING DEVICE DATA 3

Operating States

GKOP, WSOP, WSAC, SGOP

Syntax

GQDST3 (WTYPE, DEVNUM, N, MLDR, ERRIND, MBUFF, NPE, PE, EVOL, BUFLLEN, LDR, DATREC)

Argument	Data Type	Access	Description
WTYPE	Integer	Read	Workstation type
DEVNUM	Integer	Read	String device number
N	Integer	Read	Requested list element from the list of available prompt and echo types
MLDR	Integer	Read	Number of elements in output data record
ERRIND	Integer	Write	Error indicator
MBUFF	Integer	Write	Maximum input buffer size
NPE	Integer	Write	Number of available prompt and echo types
PE	Integer	Write	Nth element of the list of available prompt and echo types
EVOL(6)	Array of reals	Write	Device coordinate echo volume in the order XMIN, XMAX, YMIN, YMAX, ZMIN, ZMAX
BUFLLEN	Integer	Write	Input buffer size
LDR	Integer	Write	Number of array elements used in the output data record
DATREC(MLDR)	Character*80	Write	Output data record

Description

The INQUIRE DEFAULT STRING DEVICE DATA 3 function returns the default values for the specified string-class logical input device on the specified workstation.

The default data record returned by this function is the input data record associated with PET 1. There are no data record components associated with string-class PET 1. All the default information is returned in the other function arguments. The number of array elements used in the data record is 0.

INQUIRE DEFAULT STROKE DEVICE DATA

INQUIRE DEFAULT STROKE DEVICE DATA

Operating States

GKOP, WSOP, WSAC, SGOP

Syntax

GQDSK (WTYPE, DEVNUM, N, MLDR, ERRIND, DBUFSK, OL, PET, EAREA, BUFLLEN, LDR, DATREC)

Argument	Data Type	Access	Description
WTYPE	Integer	Read	Workstation type
DEVNUM	Integer	Read	Stroke device number
N	Integer	Read	Requested list element from the list of available prompt and echo types
MLDR	Integer	Read	Number of elements in output data record
ERRIND	Integer	Write	Error indicator
DBUFSK	Integer	Write	Maximum input buffer size
OL	Integer	Write	Number of available prompt and echo types
PET	Integer	Write	<i>N</i> th element of the list of available prompt and echo types
EAREA(4)	Array of reals	Write	Device coordinate echo area in the order XMIN, XMAX, YMIN, YMAX
BUFLLEN	Integer	Write	Input buffer length
LDR	Integer	Write	Number of array elements used in the output data record
DATREC(MLDR)	Character*80	Write	Output data record

Description

The INQUIRE DEFAULT STROKE DEVICE DATA function returns the default values for the specified stroke-class logical input device on the specified workstation.

The default data record returned by this function is the input data record associated with PET 1. Use the UNPACK DATA RECORD function to access the information. The data record components for each of the possible PETs are listed in the introduction to Chapter 9. Note that this information refers to arguments in the PACK DATA RECORD function. Because the listed arguments have the same names and meanings in both the PACK DATA RECORD and UNPACK DATA RECORD functions, you can associate the data record components easily with either function.

INQUIRE DEFAULT STROKE DEVICE DATA 3

Operating States

GKOP, WSOP, WSAC, SGOP

Syntax

GQDSK3 (WTYPE, DEVNUM, N, MLDR, ERRIND, MXBFSZ, NPE, PE, EVOL, BUFLLEN, LDR, DATREC)

Argument	Data Type	Access	Description
WTYPE	Integer	Read	Workstation type
DEVNUM	Integer	Read	Stroke device number
N	Integer	Read	Requested list element from the list of available prompt and echo types
MLDR	Integer	Read	Number of elements in output data record
ERRIND	Integer	Write	Error indicator
MXBFSZ	Integer	Read	Maximum input buffer size
NPE	Integer	Write	Number of available prompt and echo types
PE	Integer	Write	<i>N</i> th element of the list of available prompt and echo types
EVOL(6)	Array of reals	Write	Device coordinate echo volume in the order XMIN, XMAX, YMIN, YMAX, ZMIN, ZMAX
BUFLLEN	Integer	Write	Input buffer length
LDR	Integer	Write	Number of array elements used in the output data record
DATREC(MLDR)	Character*80	Write	Output data record

Description

The INQUIRE DEFAULT STROKE DEVICE DATA 3 function returns the default values for the specified stroke-class logical input device on the specified workstation.

The default data record returned by this function is the input data record associated with PET 1. Use the UNPACK DATA RECORD function to access the information. The data record components for each of the possible PETs are listed in the introduction to Chapter 9. Note that this information refers to arguments in the PACK DATA RECORD function. Because the listed arguments have the same names and meanings in both the PACK DATA RECORD and UNPACK DATA RECORD functions, you can associate the data record components easily with either function.

INQUIRE DEFAULT VALUATOR DEVICE DATA

INQUIRE DEFAULT VALUATOR DEVICE DATA

Operating States

GKOP, WSOP, WSAC, SGOP

Syntax

GQDVL (WTYPE, DEVNUM, N, MLDR, ERRIND, DVAL, OL, PET, EAREA, LOVAL, HIVAL, LDR, DATREC)

Argument	Data Type	Access	Description
WTYPE	Integer	Read	Workstation type
DEVNUM	Integer	Read	Valuator device number
N	Integer	Read	Requested list element from the list of available prompt and echo types
MLDR	Integer	Read	Number of elements in output data record
ERRIND	Integer	Write	Error indicator
DVAL	Real	Write	Default initial value
OL	Integer	Write	Number of available prompt and echo types
PET	Integer	Write	<i>N</i> th element of the list of available prompt and echo types
EAREA(4)	Array of reals	Write	Device coordinate echo area in the order XMIN, XMAX, YMIN, YMAX
LOVAL, HIVAL	Real	Write	Minimum and maximum values
LDR	Integer	Write	Number of array elements used in the output data record
DATREC(MLDR)	Character*80	Write	Output data record

Description

The INQUIRE DEFAULT VALUATOR DEVICE DATA function returns the default values for the specified valuator-class logical input device on the specified workstation.

The default data record returned by this function is the input data record associated with PET 1. There are no data record components associated with valuator-class PET 1. All the default information is returned in the other function arguments. The number of array elements used in the data record is 0.

INQUIRE DEFAULT VALUATOR DEVICE DATA 3

Operating States

GKOP, WSOP, WSAC, SGOP

Syntax

GQDVL3 (WTYPE, DEVNUM, N, MLDR, ERRIND, DVAL, NPE, PE, EVOL, LOVAL, HIVAL, LDR, DATREC)

Argument	Data Type	Access	Description
WTYPE	Integer	Read	Workstation type
DEVNUM	Integer	Read	Valuator device number
N	Integer	Read	Requested list element from the list of available prompt and echo types
MLDR	Integer	Read	Number of elements in output data record
ERRIND	Integer	Write	Error indicator
DVAL	Real	Write	Default initial value
NPE	Integer	Write	Number of available prompt and echo types
PE	Integer	Write	Nth element of the list of available prompt and echo types
EVOL(6)	Array of reals	Write	Device coordinate echo volume in the order XMIN, XMAX, YMIN, YMAX, ZMIN, ZMAX
LOVAL, HIVAL	Real	Write	Minimum and maximum values
LDR	Integer	Write	Number of array elements used in the output data record
DATREC(MLDR)	Character*80	Write	Output data record

Description

The INQUIRE DEFAULT VALUATOR DEVICE DATA 3 function returns the default values for the specified valuator-class logical input device on the specified workstation.

The default data record returned by this function is the input data record associated with PET 1. There are no data record components associated with valuator-class PET 1. All the default information is returned in the other function arguments. The number of array elements used in the data record is 0.

INQUIRE DISPLAY SPACE SIZE

INQUIRE DISPLAY SPACE SIZE

Operating States

GKOP, WSOP, WSAC, SGOP

Syntax

GQDSP (WTYPE, ERRIND, DCUNIT, RX, RY, DVRUX, DVRUY)

Argument	Data Type	Access	Description
WTYPE	Integer	Read	Workstation type
ERRIND	Integer	Write	Error indicator
DCUNIT	Integer (constant)	Write	DC units
RX, RY	Real	Write	Display space size in DC units
DVRUX, DVRUY	Integer	Write	Display space size in raster units

Constants

Defined Argument	Constant	Description
DCUNIT	GMETRE	Meters
	GOTHU	Some unit other than meters

Description

The INQUIRE DISPLAY SPACE SIZE function returns, for the specified workstation type, the display size in device coordinates, a flag specifying whether the device coordinate units are returned in meters or in other units, and the display size in raster units.

By comparing a workstation's raster units with its maximum display coordinates, you can determine the resolution of the workstation surface, and how the device coordinates are mapped onto the pixels of the device.

See Also

Example 7-4 for a program example using the INQUIRE DISPLAY SPACE SIZE function

INQUIRE DISPLAY SPACE SIZE 3

Operating States

GKOP, WSOP, WSAC, SGOP

Syntax

GQDSP3 (WTYPE, ERRIND, DCUNIT, DVDCX, DVDCY, DVDCZ, DVRUX, DVRUY, DVRUZ)

Argument	Data Type	Access	Description
WTYPE	Integer	Read	Workstation type
ERRIND	Integer	Write	Error indicator
DCUNIT	Integer (constant)	Write	DC units
DVDCX, DVDCY, DVDCZ	Real	Write	Display space size in DC units
DVRUX, DVRUY, DVRUZ	Integer	Write	Display space size in raster units

Constants

Defined Argument	Constant	Description
DCUNIT	GMETRE	Meters
	GOTHU	Some unit other than meters

Description

The INQUIRE DISPLAY SPACE SIZE 3 function returns, for the specified workstation type, the display size in device coordinates, a flag specifying whether the device coordinate units are returned in meters or in other units, and the display size in raster units.

By comparing a workstation's raster units with its maximum display coordinates, you can determine the resolution of the workstation surface, and how the device coordinates are mapped onto the pixels of the device.

See Also

Example 7–4 for a program example using the INQUIRE DISPLAY SPACE SIZE function

INQUIRE DYNAMIC MODIFICATION OF SEGMENT ATTRIBUTES

INQUIRE DYNAMIC MODIFICATION OF SEGMENT ATTRIBUTES

Operating States

GKOP, WSOP, WSAC, SGOP

Syntax

GQDSGA (WTYPE, ERRIND, SGTR, VONOFF, VOFFON, HIGH, SGPR, ADD, SGDEL)

Argument	Data Type	Access	Description
WTYPE	Integer	Read	Workstation type
ERRIND	Integer	Write	Error indicator
SGTR	Integer (constant)	Write	Workstation capability for dynamic modification with a segment transformation
VONOFF, VOFFON	Integers (constant)	Write	Workstation capability for dynamic modification when visibility changes from on to off
HIGH	Integer (constant)	Write	Workstation capability for dynamic modification when highlighting changes from on to off
SGPR	Integer (constant)	Write	Workstation capability for dynamic modification when segment priority changes
ADD	Integer (constant)	Write	Workstation capability for dynamic modification when primitives are added to an open segment
SGDEL	Integer (constant)	Write	Workstation capability for dynamic modification when a segment is deleted

Constants

Defined Argument	Constant	Description
SGTR, VONOFF, VOFFON, HIGH, SGPR, ADD, SGDEL	GIRG GIMM	Implicit regeneration Immediate regeneration

INQUIRE DYNAMIC MODIFICATION OF SEGMENT ATTRIBUTES

Description

The INQUIRE DYNAMIC MODIFICATION OF SEGMENT ATTRIBUTES function returns information concerning the ability of the workstation to dynamically generate segment transformations, visibility changes, highlighting changes, priority changes, primitive additions, and segment deletions.

If the workstation can dynamically change the display surface, DEC GKS displays the results of the segment attribute changes immediately. If the workstation cannot dynamically change the display surface, and the implicit regeneration mode is set to SUPPRESSED, DEC GKS waits until the next update of the display surface to regenerate the primitives contained in segments. Implicit regeneration is described in the *DEC GKS User's Guide*.

If an implicit regeneration is required, all output primitives not contained in a segment are lost.

INQUIRE DYNAMIC MODIFICATION OF WORKSTATION ATTRIBUTES

INQUIRE DYNAMIC MODIFICATION OF WORKSTATION ATTRIBUTES

Operating States

GKOP, WSOP, WSAC, SGOP

Syntax

GQDWKA (WTYPE, ERRIND, PLBUN, PMBUN, TXBUN, FABUN, PAREP, COLREP, WKTR)

Argument	Data Type	Access	Description
WTYPE	Integer	Read	Workstation type
ERRIND	Integer	Write	Error indicator
PLBUN	Integer (constant)	Write	Workstation capability for dynamic modification when polyline representation changes
PMBUN	Integer (constant)	Write	Workstation capability for dynamic modification when the polymarker representation changes
TXBUN	Integer (constant)	Write	Workstation capability for dynamic modification when the text representation changes
FABUN	Integer (constant)	Write	Workstation capability for dynamic modification when the fill area representation changes
PAREP	Integer (constant)	Write	Workstation capability for dynamic modification when the pattern representation changes
COLREP	Integer (constant)	Write	Workstation capability for dynamic modification when the color representation changes
WKTR	Integer (constant)	Write	Workstation capability for dynamic modification when the workstation transformation changes

INQUIRE DYNAMIC MODIFICATION OF WORKSTATION ATTRIBUTES

Constants

Defined Argument	Constant	Description
PLBUN, PMBUN, TXBUN, FABUN, PAREP, COLREP, WKTR	GIRG GIMM	Implicit regeneration Immediate regeneration

Description

The INQUIRE DYNAMIC MODIFICATION OF WORKSTATION ATTRIBUTES function returns information describing the ability of the workstation to dynamically alter the display surface following a modification of a workstation attribute.

If the workstation can dynamically change the display surface, DEC GKS displays the results of attribute changes immediately. If the workstation cannot dynamically change the display surface and the implicit regeneration mode is set to SUPPRESSED, DEC GKS waits until the next update of the display surface to regenerate the primitives contained in segments. Implicit regeneration is described in the *DEC GKS User's Guide*.

If an implicit regeneration is required, all output primitives not contained in a segment are lost.

See Also

SET COLOUR REPRESENTATION
SET FILL AREA REPRESENTATION
SET PATTERN REPRESENTATION
SET POLYLINE REPRESENTATION
SET POLYMARKER REPRESENTATION
SET TEXT REPRESENTATION

Example 7-4 for a program example using the INQUIRE DYNAMIC MODIFICATION OF WORKSTATION ATTRIBUTES function

INQUIRE DYNAMIC MODIFICATION OF WORKSTATION ATTRIBUTES 3

INQUIRE DYNAMIC MODIFICATION OF WORKSTATION ATTRIBUTES 3

Operating States

GKOP, WSOP, WSAC, SGOP

Syntax

GQDWA3 (WTYPE, ERRIND, PLBUN, PMBUN, TBBUN, FABUN, PABUN, EDBUN, COLREP, WKTR, VWREP, HMODE)

Argument	Data Type	Access	Description
WTYPE	Integer	Read	Workstation type
ERRIND	Integer	Write	Error status
PLBUN	Integer (constant)	Write	Workstation capability for dynamic modification when the polyline representation changes
PMBUN	Integer (constant)	Write	Workstation capability for dynamic modification when the polymarker representation changes
TBBUN	Integer (constant)	Write	Workstation capability for dynamic modification when the text representation changes
FABUN	Integer (constant)	Write	Workstation capability for dynamic modification when the fill area representation changes
PABUN	Integer (constant)	Write	Workstation capability for dynamic modification when the pattern representation changes
EDBUN	Integer (constant)	Write	Workstation capability for dynamic modification when the edge representation changes
COLREP	Integer (constant)	Write	Workstation capability for dynamic modification when the color representation changes
WKTR	Integer (constant)	Write	Workstation capability for dynamic modification when the workstation transformation changes

INQUIRE DYNAMIC MODIFICATION OF WORKSTATION ATTRIBUTES 3

Argument	Data Type	Access	Description
VWREP	Integer (constant)	Write	Workstation capability for dynamic modification when the view transformation changes
HMODE	Integer (constant)	Write	Workstation capability for dynamic modification when the hidden line and hidden surface removal representation changes

Constants

Defined Argument	Constant	Description
PLBUN, PMBUN, TBBUN, FABUN, PABUN, EDBUN, COLREP, WKTR, VWREP, HMODE	GIRG GIMM	Implicit regeneration Immediate regeneration

Description

The INQUIRE DYNAMIC MODIFICATION OF WORKSTATION ATTRIBUTES 3 function returns information describing the ability of the workstation to dynamically alter the display surface following a modification of a workstation attribute.

If the workstation can dynamically change the display surface, DEC GKS displays the results of attribute changes immediately. If the workstation cannot dynamically change the display surface and the implicit regeneration mode is set to SUPPRESSED, DEC GKS waits until the next update of the surface to regenerate the primitives contained in segments. Implicit regeneration is described in the *DEC GKS User's Guide*.

If an implicit regeneration is required, all output primitives not contained in a segment are lost.

INQUIRE DYNAMIC MODIFICATION OF WORKSTATION ATTRIBUTES 3

See Also

SET COLOUR REPRESENTATION
SET FILL AREA REPRESENTATION
SET PATTERN REPRESENTATION
SET POLYLINE REPRESENTATION
SET POLYMARKER REPRESENTATION
SET TEXT REPRESENTATION

Example 7-4 for a program example using the INQUIRE DYNAMIC MODIFICATION OF WORKSTATION ATTRIBUTES function

INQUIRE EDGE FACILITIES

Operating States

GKOP, WSOP, WSAC, SGOP

Syntax

GQEDF (WTYPE, N, ERRIND, NEDT, EDT, NEDW, NOMEDW, MINEDW, MAXEDW, NPEDI)

Argument	Data Type	Access	Description
WTYPE	Integer	Read	Workstation type
N	Integer	Read	Requested list element of the list of available edge types
ERRIND	Integer	Write	Error indicator
NEDT	Integer	Write	Number of available edge types
EDT	Integer	Write	<i>Nth</i> element of list of available edge types
NEDW	Integer	Write	Number of available edge widths
NOMEDW	Integer	Write	Nominal edge width
MINEDW, MAXEDW	Real	Write	Range of edge widths from minimum to maximum
NPEDI	Integer	Write	Number of predefined edge indexes

Description

The INQUIRE EDGE FACILITIES function returns the edge flag; the number of edge types; the requested element in the list of edge types; the number of edge widths; the nominal, minimum, and maximum edge widths; and the number of predefined edge index values.

If the number of edge widths returned is 0, the workstation supports a continuous range of edge widths.

INQUIRE EDGE REPRESENTATION

INQUIRE EDGE REPRESENTATION

Operating States

WSOP, WSAC, SGOP

Syntax

GQEDR (WKID, EDI, TYPE, ERRIND, EDFLAG, EDTYPE, EDWSF, COLI)

Argument	Data Type	Access	Description
WKID	Integer	Read	Workstation identifier
EDI	Integer	Read	Edge index
TYPE	Integer (constant)	Read	Type of value to return
ERRIND	Integer	Write	Error indicator
EDFLAG	Integer (constant)	Write	Edge flag
EDTYPE	Integer (constant)	Write	Edge type
EDWSF	Real	Write	Edge width scale factor
COLI	Integer	Write	Edge color index

Constants

Defined Argument	Constant	Description
TYPE	GSET	Use exact state list values.
	GREALI	Use values approximated by the graphics handler.
EDFLAG	GOFF	Edge off.
	GON	Edge on.
EDTYPE	GESOLI	Solid edge.
	GEDASH	Dashed edge.
	GEDOT	Dotted edge.
	GEDASD	Dashed-dotted edge.

Description

The INQUIRE EDGE REPRESENTATION function returns the values associated with the given edge index value.

If the specified edge index is not in the edge bundle table on the specified workstation, and the type of returned values is REALIZED, this function returns the edge representation associated with edge index 1.

INQUIRE FILL AREA COLOUR INDEX

Operating States

GKOP, WSOP, WSAC, SGOP

Syntax

GQFACI (ERRIND, COLI)

Argument	Data Type	Access	Description
ERRIND	Integer	Write	Error indicator
COLI	Integer	Write	Fill area color index

Description

The INQUIRE FILL AREA COLOUR INDEX function returns the value for the *current fill area color index* entry in the GKS state list.

The value of the *current fill area color index* entry controls the display of subsequent fill area set output primitives when the current fill color index ASF has been set to INDIVIDUAL by the function SET ASPECT SOURCE FLAGS 3. If this ASF is set to BUNDLED, the current fill area colour index has no effect.

See Also

SET ASPECT SOURCE FLAGS 3
 SET FILL AREA COLOUR INDEX

INQUIRE FILL AREA FACILITIES

INQUIRE FILL AREA FACILITIES

Operating States

GKOP, WSOP, WSAC, SGOP

Syntax

GQFAF (WTYPE, NI, NH, ERRIND, NIS, IS, NHS, HS, NPFAI)

Argument	Data Type	Access	Description
WTYPE	Integer	Read	Workstation type
NI	Integer	Read	Requested list element from the list of available interior styles
NH	Integer	Read	Requested list element from the list of available hatch styles
ERRIND	Integer	Write	Error indicator
NIS	Integer	Write	Number of available fill area interior styles
IS	Integer (constant)	Write	<i>N</i> th element of list of available fill area interior styles
NHS	Integer	Write	Number of available fill area hatch styles
HS	Array of integers	Write	<i>N</i> th element of list of available fill area hatch styles
NPFAI	Integer	Write	Number of predefined fill area indexes

Constants

Defined Argument	Constant	Description
IS	GHOLLO	Hollow interior
	GSOLID	Solid interior
	GPATTR	Patterned interior
	GHATCH	Hatched interior

Description

The INQUIRE FILL AREA FACILITIES function returns the number of available interior styles, the requested list element from the list of available interior styles, the number of hatching styles, the requested list element from the list of available hatching styles, and the number of predefined fill area indexes available for a given workstation type.

INQUIRE FILL AREA FACILITIES

If the graphics handler supports the fill area interior style, it writes the style's constant value to the *IS* argument. If the style is not supported, DEC GKS writes a -1 to the *IS* argument.

INQUIRE FILL AREA INDEX

INQUIRE FILL AREA INDEX

Operating States

GKOP, WSOP, WSAC, SGOP

Syntax

GQFAI (ERRIND, FAI)

Argument	Data Type	Access	Description
ERRIND	Integer	Write	Error indicator
FAI	Integer	Write	Fill area index

Description

The INQUIRE FILL AREA INDEX function returns the value for the *current fill area index* entry in the GKS state list.

The fill area bundle table contains entries for the fill area interior style, fill area style index, and fill area color index attribute values. When calling FILL AREA, DEC GKS uses the bundle table only if the corresponding ASF has been set to BUNDLED.

See Also

FILL AREA
SET FILL AREA INDEX

INQUIRE FILL AREA INTERIOR STYLE

Operating States

GKOP, WSOP, WSAC, SGOP

Syntax

GQFAIS (ERRIND, INTS)

Argument	Data Type	Access	Description
ERRIND	Integer	Write	Error indicator
INTS	Integer (constant)	Write	Fill area interior style

Constants

Defined Argument	Constant	Description
INTS	GHOLLO	Hollow interior
	GSOLID	Solid interior
	GPATTR	Patterned interior
	GHATCH	Hatched interior

Description

The INQUIRE FILL AREA INTERIOR STYLE function queries the GKS state list and returns the current value for the fill area interior style index. The fill area interior style index determines whether a fill area will be drawn hollow, filled with a single color, or filled with a pattern or hatch design.

See Also

SET FILL AREA INTERIOR STYLE

INQUIRE FILL AREA REPRESENTATION

INQUIRE FILL AREA REPRESENTATION

Operating States

WSOP, WSAC, SGOP

Syntax

GQFAR (WKID, FAI, TYPE, ERRIND, INTS, STYLI, COLI)

Argument	Data Type	Access	Description
WKID	Integer	Read	Workstation identifier
FAI	Integer	Read	Fill area index
TYPE	Integer (constant)	Read	Type of values to return
ERRIND	Integer	Write	Error indicator
INTS	Integer (constant)	Write	Fill area interior style
STYLI	Integer	Write	Fill area style index
COLI	Integer	Write	Fill area color index

Constants

Defined Argument	Constant	Description
TYPE	GSET	Use exact state list values.
	GREALI	Use values approximated by the graphics handler.
INTS	GHOLLO	Hollow interior.
	GSOLID	Solid interior.
	GPATTR	Patterned interior.
	GHATCH	Hatched interior.

Description

The INQUIRE FILL AREA REPRESENTATION function returns the values associated with the given fill area index value.

See Also

SET FILL AREA REPRESENTATION

INQUIRE FILL AREA STYLE INDEX
Operating States

GKOP, WSOP

Syntax

GQFASI (ERRIND, STYLI)

Argument	Data Type	Access	Description
ERRIND	Integer	Write	Error indicator
STYLI	Integer	Write	Style index

Description

The INQUIRE FILL AREA STYLE INDEX function returns the *current fill area style index* entry in the GKS state list.

This function returns the value of the specific pattern or hatch style used to fill the interior of a polygonal fill area. If the interior style is hollow or solid, the current style index is ignored in a call to FILL AREA. If the interior style is pattern, a pattern index value is returned by this function. If the interior style is hatch, a hatch style value is returned by this function. For device-dependent hatch styles, the hatch style index is always a negative number.

See Also

FILL AREA
SET FILL AREA STYLE INDEX

INQUIRE GENERALIZED DRAWING PRIMITIVE

INQUIRE GENERALIZED DRAWING PRIMITIVE

Operating States

GKOP, WSOP, WSAC, SGOP

Syntax

GQGDP (WTYPE, N, ERRIND, NBND, BNDL)

Argument	Data Type	Access	Description
WTYPE	Integer	Read	Workstation type
N	Integer	Read	Generalized drawing primitive identifier
ERRIND	Integer	Write	Error indicator
NBND	Integer	Write	Number of sets of attributes used
BNDL(4)	Array of integers (constant)	Write	List of sets of attributes used

Constants

Defined Argument	Constant	Description
BNDL	GPLATT	GDP polyline attributes
	GPMATT	GDP polymarker attributes
	GTXATT	GDP text attributes
	GFAATT	GDP fill area attributes

Description

The INQUIRE GENERALIZED DRAWING PRIMITIVE function returns the number of attribute sets, and the list of those attribute sets that are associated with the specified two-dimensional GDP identifier for a given workstation type.

INQUIRE GENERALIZED DRAWING PRIMITIVE 3

Operating States

GKOP, WSOP, WSAC, SGOP

Syntax

GQGDP3 (WTYPE, N, ERRIND, NBND, BNDL)

Argument	Data Type	Access	Description
WTYPE	Integer	Read	Workstation type
N	Integer	Read	Generalized drawing primitive 3 identifier
ERRIND	Integer	Write	Error indicator
NBND	Integer	Write	Number of sets of attributes used
BNDL(5)	Array of integers (constant)	Write	List of sets of attributes used

Constants

Defined Argument	Constant	Description
BNDL	GPLATT	GDP polyline attributes
	GPMATT	GDP polymarker attributes
	GTXATT	GDP text attributes
	GFAATT	GDP fill area attributes
	GEDATT	GDP edge attributes

Description

The INQUIRE GENERALIZED DRAWING PRIMITIVE 3 function returns the number of attribute sets, and the list of those attribute sets that are associated with the specified three-dimensional GDP identifier for a given workstation type.

INQUIRE HLHSR FACILITIES

INQUIRE HLHSR FACILITIES

Operating States

GKOP, WSOP, WSAC, SGOP

Syntax

GQHRF (WTYPE, NI, NM, ERRIND, NHRID, HRID, NHRM, HRM)

Argument	Data Type	Access	Description
WTYPE	Integer	Read	Workstation type
NI	Integer	Read	Requested list element from the list of available HLHSR identifiers
NM	Integer	Read	Requested list element from the list of available HLHSR modes
ERRIND	Integer	Write	Error indicator
NHRID	Integer	Write	Number of available HLHSR identifiers
HRID	Integer	Write	<i>N</i> th element of list of available HLHSR identifiers
NHRM	Integer	Write	Number of available HLHSR modes
HRM	Integer	Write	<i>NM</i> th element of list of available HLHSR modes

Description

The INQUIRE HLHSR FACILITIES function returns the number of available hidden line and hidden surface removal (HLHSR) identifiers and HLHSR modes, the requested element in the list of available HLHSR identifiers, and the requested element in the list of available HLHSR modes for the specified workstation type.

INQUIRE HLHSR MODE

Operating States

WSOP, WSAC, WGOP

Syntax

GQHRM (WKID, ERRIND, HUPD, RHRM, CHRM)

Argument	Data Type	Access	Description
WKID	Integer	Read	Workstation identifier
ERRIND	Integer	Write	Error indicator
HUPD	Integer (constant)	Write	HLHSR mode update state
RHRM	Integer	Write	Requested HLHSR mode
CHRM	Integer	Write	Current HLHSR mode

Constants

Defined Argument	Constant	Description
HUPD	GNPEND	Action not pending
	GPEND	Action pending

Description

The INQUIRE HLHSR MODE function returns the hidden line and hidden surface removal (HLHSR) update state, and the requested and current HLHSR modes.

If an HLHSR mode change was requested but not yet provided at the time of the call to this function, the HLHSR update state is PENDING.

INQUIRE INPUT QUEUE OVERFLOW

INQUIRE INPUT QUEUE OVERFLOW

Operating States

WSOP, WSAC, SGOP

Syntax

GQIQOV (ERRIND, WKID, ICL, IDN)

Argument	Data Type	Access	Description
ERRIND	Integer	Write	Error indicator
WKID	Integer	Write	Workstation identifier
ICL	Integer (constant)	Write	Input class
IDN	Integer	Write	Input device number

Constants

Defined Argument	Constant	Description
ICL	GLOCAT	Locator input
	GSTROK	Stroke input
	GVALUA	Valuator input
	GCHOIC	Choice input
	GPICK	Pick input
	GSTRIN	String input

Description

The INQUIRE INPUT QUEUE OVERFLOW function queries the GKS error state list, and if the input queue has overflowed since the start of the session or since the last call to this function, it returns the identification of the logical input device that caused the overflow. The information is then removed from the GKS error state list.

See Also

AWAIT EVENT
FLUSH DEVICE EVENTS

INQUIRE LEVEL OF GKS

Operating States

GKOP, WSOP, WSAC, SGOP

Syntax

GQLVKS (ERRIND, LEVEL)

Argument	Data Type	Access	Description
ERRIND	Integer	Write	Error status
LEVEL	Integer (constant)	Write	Level of GKS

Constants

Defined Argument	Constant	Description
LEVEL	GLMA	Minimal output, no input
	GLMB	Minimal output, request input
	GLMC	Minimal output, full input
	GL0A	All primitives and attributes, no input
	GL0B	All primitives and attributes, request input
	GL0C	All primitives and attributes, full input
	GL1A	Basic segmentation with full output, no input
	GL1B	Basic segmentation with full output, request input
	GL1C	Basic segmentation with full output, full input
	GL2A	Workstation independent segment storage (WISS), no input
	GL2B	Workstation independent segment storage (WISS), request input
	GL2C	Workstation independent segment storage (WISS), full input

Description

The INQUIRE LEVEL OF GKS function returns the DEC GKS implementation level.

INQUIRE LINETYPE

INQUIRE LINETYPE

Operating States

GKOP, WSOP, WSAC, SGOP

Syntax

GQLN (ERRIND, LTYPE)

Argument	Data Type	Access	Description
ERRIND	Integer	Write	Error indicator
LTYPE	Integer (constant)	Write	Line type

Constants

Defined Argument	Constant	Description
LTYPE	GLSOLI	Solid line
	GLDASH	Dashed line
	GLDOT	Dotted line
	GLDASD	Dashed-dotted line

Description

The INQUIRE LINETYPE function returns the value for the *current polyline line type* entry in the GKS state list as solid, dashed, dotted, dashed-dotted, or any one of the device-dependent types.

Every workstation capable of output (DEC GKS category OUTPUT or OUTIN) defines at least four line types. For more information concerning possible polyline type values, see the *Device Specifics Reference Manual for DEC GKS and DEC PHIGS*.

See Also

SET LINETYPE

INQUIRE LINEWIDTH SCALE FACTOR

Operating States

GKOP, WSOP, WSAC, SGOP

Syntax

GQLWSC (ERRIND, LWIDTH)

Argument	Data Type	Access	Description
ERRIND	Integer	Write	Error indicator
LWIDTH	Real	Write	Line width scale factor

Description

The INQUIRE LINEWIDTH SCALE FACTOR function returns the value for the *current polyline line width scale factor* entry in the GKS state list.

DEC GKS calculates line width as the nominal line width, multiplied by the line width scale factor. The line width scale factor is a real number that is passed to the SET LINEWIDTH SCALE FACTOR function. The workstation maps the value to the nearest available line width defined by the workstation.

See Also

SET LINEWIDTH SCALE FACTOR

INQUIRE LIST OF AVAILABLE GENERALIZED DRAWING PRIMITIVES

INQUIRE LIST OF AVAILABLE GENERALIZED DRAWING PRIMITIVES

Operating States

GKOP, WSOP, WSAC, SGOP

Syntax

GQEGDP (WTYPE, N, ERRIND, NGDP, GDPL)

Argument	Data Type	Access	Description
WTYPE	Integer	Read	Workstation type
N	Integer	Read	Requested list element from the list of available GDPs
ERRIND	Integer	Write	Error indicator
NGDP	Integer	Write	Number of available GDPs
GDPL	Integer	Write	<i>N</i> th element of the list of GDP identifiers

Description

The INQUIRE LIST OF AVAILABLE GENERALIZED DRAWING PRIMITIVES function returns the number of available two-dimensional GDPs and the requested element in the list of the GDP identifiers for the given workstation type.

The FORTRAN binding standard lists this function as INQUIRE LIST element OF AVAILABLE GENERALIZED DRAWING PRIMITIVES.

INQUIRE LIST OF AVAILABLE GENERALIZED DRAWING PRIMITIVES 3

Operating States

GKOP, WSOP, WSAC, SGOP

Syntax

GQEGD3 (WTYPE, N, ERRIND, NGDP, GDPL)

Argument	Data Type	Access	Description
WTYPE	Integer	Read	Workstation type
N	Integer	Read	Requested list element from the list of available GDPs
ERRIND	Integer	Write	Error indicator
NGDP	Integer	Write	Number of available 3D GDPs
GDPL	Integer (constant)	Write	Nth element of list of GDP 3 identifiers

Description

The INQUIRE LIST OF AVAILABLE GENERALIZED DRAWING PRIMITIVES 3 function returns the number of available three-dimensional GDPs and the requested element in the list of the GDP identifiers for the given workstation type.

The FORTRAN binding standard lists this function as INQUIRE LIST element OF AVAILABLE GENERALIZED DRAWING PRIMITIVES 3.

INQUIRE LIST OF AVAILABLE WORKSTATION TYPES

INQUIRE LIST OF AVAILABLE WORKSTATION TYPES

Operating States

GKOP, WSOP, WSAC, SGOP

Syntax

GQEWK (N, ERRIND, NUMBER, WKTYP)

Argument	Data Type	Access	Description
N	Integer	Read	Requested list element from the list of available workstation types
ERRIND	Integer	Write	Error status
NUMBER	Integer	Write	Number of different workstation types
WKTYP	Integer	Write	Nth element from list of available workstation types

Description

The INQUIRE LIST OF AVAILABLE WORKSTATION TYPES function queries the DEC GKS description table and returns the number of different workstation types and the type of the Nth element in the list. You can use a returned type value as a workstation type argument to the OPEN WORKSTATION function, and you can pass it to inquiry functions that take a workstation type argument.

The FORTRAN binding standard lists this function as INQUIRE LIST element OF AVAILABLE WORKSTATION TYPES.

See Also

OPEN WORKSTATION

INQUIRE LIST OF COLOUR INDICES

Operating States

WSOP, WSAC, SGOP

Syntax

GQECI (WKID, N, ERRIND, OL, COLI)

Argument	Data Type	Access	Description
WKID	Integer	Read	Workstation identifier
N	Integer	Read	Requested list element from the list of defined color indexes
ERRIND	Integer	Write	Error indicator
OL	Integer	Write	Number of color table entries
COLI	Integer	Write	<i>N</i> th element of the list of color indexes

Description

The INQUIRE LIST OF COLOUR INDICES function returns the number of defined color index values and the index value of the requested element in the list of color indexes.

The FORTRAN binding standard lists this function as INQUIRE LIST element OF COLOUR INDICES.

See Also

SET COLOUR REPRESENTATION

INQUIRE LIST OF EDGE INDICES

INQUIRE LIST OF EDGE INDICES

Operating States

WSOP, WSAC, SGOP

Syntax

GQEEDI (WKID, N, ERRIND, NEDI, EDI)

Argument	Data Type	Access	Description
WKID	Integer	Read	Workstation identifier
N	Integer	Read	Requested list element from the list of defined edge indexes
ERRIND	Integer	Write	Error indicator
NEDI	Integer	Write	Number of edge indexes
EDI	Integer	Write	<i>N</i> th element of the list of defined edge indexes

Description

The INQUIRE LIST OF EDGE INDICES function returns the number of defined edge index values and the index value of the requested element in the list of defined edge indexes.

The FORTRAN binding standard lists this function as INQUIRE LIST element OF EDGE INDICES.

See Also

SET EDGE REPRESENTATION

INQUIRE LIST OF FILL AREA INDICES

Operating States

WSOP, WSAC, SGOP

Syntax

GQEFAI (WKID, N, ERRIND, OL, FAI)

Argument	Data Type	Access	Description
WKID	Integer	Read	Workstation identifier
N	Integer	Read	Requested list element from the list of defined fill area indexes
ERRIND	Integer	Write	Error indicator
OL	Integer	Write	Number of fill area indexes
FAI	Integer	Write	<i>N</i> th element of the list of defined fill area indexes

Description

The INQUIRE LIST OF FILL AREA INDICES function returns the number of defined fill area index values and the index value of the requested element of the list of defined fill area indexes.

The FORTRAN binding standard lists this function as INQUIRE LIST element OF FILL AREA INDICES.

See Also

SET FILL AREA REPRESENTATION

INQUIRE LIST OF NORMALIZATION TRANSFORMATION NUMBERS

INQUIRE LIST OF NORMALIZATION TRANSFORMATION NUMBERS

Operating States

GKOP, WSOP, WSAC, SGOP

Syntax

GQENTN (N, ERRIND, OL, TNR)

Argument	Data Type	Access	Description
N	Integer	Read	Requested list element from the list of normalized transformation numbers
ERRIND	Integer	Write	Error indicator
OL	Integer	Write	Length of list
TNR	Integer	Write	<i>N</i> th element of the list of normalization transformation numbers

Description

The INQUIRE LIST OF NORMALIZATION TRANSFORMATION NUMBERS function returns the number of defined normalization transformations and the normalization transformation number of the requested element in the list of normalization transformation numbers.

The FORTRAN binding standard lists this function as INQUIRE LIST element OF NORMALIZATION TRANSFORMATION NUMBERS.

See Also

SELECT NORMALIZATION TRANSFORMATION
SET VIEWPORT
SET WINDOW

INQUIRE LIST OF PATTERN INDICES

Operating States

WSOP, WSAC, SGOP

Syntax

GQEPAI (WKID, N, ERRIND, OL, PAI)

Argument	Data Type	Access	Description
WKID	Integer	Read	Workstation identifier
N	Integer	Read	Requested list element from the list of defined pattern indexes
ERRIND	Integer	Write	Error indicator
OL	Integer	Write	Number of pattern indexes
PAI	Integer	Write	<i>N</i> th element of the list of defined pattern indexes

Description

The INQUIRE LIST OF PATTERN INDICES function returns the number of defined pattern index values and the index value of the requested element in the list of defined pattern indexes.

The FORTRAN binding standard lists this function as INQUIRE LIST element OF PATTERN INDICES.

See Also

SET PATTERN REPRESENTATION

INQUIRE LIST OF POLYLINE INDICES

INQUIRE LIST OF POLYLINE INDICES

Operating States

WSOP, WSAC, SGOP

Syntax

GQEPLI (WKID, N, ERRIND, OL, PLI)

Argument	Data Type	Access	Description
WKID	Integer	Read	Workstation identifier
N	Integer	Read	Requested list element from the list of defined polyline indexes
ERRIND	Integer	Write	Error indicator
OL	Integer	Write	Number of defined polyline indexes
PLI	Integer	Write	<i>N</i> th element of the list of defined polyline indexes

Description

The INQUIRE LIST OF POLYLINE INDICES function returns the number defined polyline index values and the polyline index value of the requested element in the list of defined polyline indexes.

The FORTRAN binding standard lists this function as INQUIRE LIST element OF POLYLINE INDICES.

See Also

SET POLYLINE REPRESENTATION

INQUIRE LIST OF POLYMARKER INDICES

Operating States

WSOP, WSAC, SGOP

Syntax

GQEPMI (WKID, N, ERRIND, OL, PMI)

Argument	Data Type	Access	Description
WKID	Integer	Read	Workstation identifier
N	Integer	Read	Requested list element from the list of defined polymarker indexes
ERRIND	Integer	Write	Error indicator
OL	Integer	Write	Number of defined polymarker indexes
PMI	Integer	Write	<i>N</i> th element of the list of defined polymarker indexes

Description

The INQUIRE LIST OF POLYMARKER INDICES function returns the number of defined polymarker index values and the index value of the requested element in the list of defined polymarker indexes.

The FORTRAN binding standard lists this function as INQUIRE LIST element OF POLYMARKER INDICES.

See Also

SET POLYMARKER REPRESENTATION

INQUIRE LIST OF TEXT INDICES

INQUIRE LIST OF TEXT INDICES

Operating States

WSOP, WSAC, SGOP

Syntax

GQETXI (WKID, N, ERRIND, OL, TXI)

Argument	Data Type	Access	Description
WKID	Integer	Read	Workstation identifier
N	Integer	Read	Requested list element from the list of defined text indexes
ERRIND	Integer	Write	Error indicator
OL	Integer	Write	Number of defined text indexes for workstation
TXI	Integer	Write	<i>N</i> th element of the list of defined text index values

Description

The INQUIRE LIST OF TEXT INDICES function returns the number of defined text index values and the text index of the requested element in the list of defined text indexes.

The FORTRAN binding standard lists this function as INQUIRE LIST element OF TEXT INDICES.

See Also

SET TEXT REPRESENTATION

INQUIRE LIST OF VIEW INDICES
Operating States

WSOP, WSAC, SGOP

Syntax

GQEVWI (WKID, N, ERRIND, NVWIX, VIEWI)

Argument	Data Type	Access	Description
WKID	Integer	Read	Workstation identifier
N	Integer	Read	Requested list element from the list of defined view indexes
ERRIND	Integer	Write	Error indicator
NVWIX	Integer	Write	Number of defined view indexes
VIEWI	Integer	Write	<i>N</i> th element of list of defined view indexes

Description

The INQUIRE LIST OF VIEW INDICES function returns the number of defined view index values and the index value of the requested element in the list of defined view indexes.

The FORTRAN binding standard lists this function as INQUIRE LIST element OF VIEW INDICES.

See Also

SET VIEW REPRESENTATION 3

INQUIRE LOCATOR DEVICE STATE

INQUIRE LOCATOR DEVICE STATE

Operating States

WSOP, WSAC, SGOP

Syntax

GQLCS (WKID, DEVNUM, TYPE, MLDR, ERRIND, MODE, ESW, TNR, IPX, IPY, PET, EAREA, LDR, DATREC)

Argument	Data Type	Access	Description
WKID	Integer	Read	Workstation identifier
DEVNUM	Integer	Read	Locator device number
TYPE	Integer (constant)	Read	Type of values requested
MLDR	Integer	Read	Number of elements in output data record
ERRIND	Integer	Write	Error indicator
MODE	Integer (constant)	Write	Operating mode
ESW	Integer (constant)	Write	Echo flag
TNR	Integer	Write	Initial normalization transformation number
IPX, IPY	Real	Write	Initial locator position in WC points
PET	Integer	Write	Prompt and echo type
EAREA(4)	Array of reals	Write	Device coordinate echo area in the order XMIN, XMAX, YMIN, YMAX
LDR	Integer	Write	Number of array elements used in the output data record
DATREC(MLDR)	Character*80	Write	Output data record

Constants

Defined Argument	Constant	Description
TYPE	GSET	Use exact state list values.
	GREALI	Use values approximated by the graphics handler.

INQUIRE LOCATOR DEVICE STATE

Defined Argument	Constant	Description
MODE	GREQU	Request input mode.
	GSAMPL	Sample input mode.
	GEVENT	Event input mode.
ESW	GNECHO	Echo disabled.
	GECHO	Echo enabled.

Description

The INQUIRE LOCATOR DEVICE STATE function returns the current state of the given locator-class logical input device.

The data record returned by this function is the input data record associated with the PET returned in the argument *PET*. Use the UNPACK DATA RECORD function to access this information.

The data record components for each of the possible PETs are listed in the introduction to Chapter 9. Note that this information refers to arguments in the PACK DATA RECORD function. Because the listed arguments have the same names and meanings in both the PACK DATA RECORD and UNPACK DATA RECORD functions, you can associate the data record components easily with either function.

See Also

INITIALIZE LOCATOR

SET LOCATOR MODE

Example 9–1 for a program example using the INQUIRE LOCATOR DEVICE STATE function

INQUIRE LOCATOR DEVICE STATE 3

INQUIRE LOCATOR DEVICE STATE 3

Operating States

WSOP, WSAC, SGOP

Syntax

GQLCS3 (WKID, DEVNUM, TYPE, MLDR, ERRIND, MODE, ESW, ITNR, IVWIX, IPX, IPY, IPZ, PET, EVOL, LDR, DATREC)

Argument	Data Type	Access	Description
WKID	Integer	Read	Workstation identifier
DEVNUM	Integer	Read	Locator device number
TYPE	Integer (constant)	Read	Type of values requested
MLDR	Integer	Read	Number of elements in output data record
ERRIND	Integer	Write	Error indicator
MODE	Integer (constant)	Write	Operating mode
ESW	Integer (constant)	Write	Echo flag
ITNR	Integer	Write	Initial normalization transformation number
IVWIX	Integer	Write	Initial view index
IPX, IPY, IPZ	Real	Write	Initial locator position in WC points
PET	Integer	Write	Prompt and echo type
EVOL(6)	Array of reals	Write	Device coordinate echo volume in the order XMIN, XMAX, YMIN, YMAX, ZMIN, ZMAX
LDR	Integer	Write	Number of array elements used in the output data record
DATREC(MLDR)	Character*80	Write	Output data record

Constants

Defined Argument	Constant	Description
TYPE	GSET	Use exact state list values.
	GREALI	Use values approximated by the graphics handler.

INQUIRE LOCATOR DEVICE STATE 3

Defined Argument	Constant	Description
MODE	GREQU	Request input mode.
	GSAMPL	Sample input mode.
	GEVENT	Event input mode.
ESW	GNECHO	Echo disabled.
	GECHO	Echo enabled.

Description

The INQUIRE LOCATOR DEVICE STATE 3 function returns the current state of the given locator-class logical input device.

The data record returned by this function is the input data record associated with the PET returned in the argument *PET*. Use the UNPACK DATA RECORD function to access this information.

The data record components for each of the possible PETs are listed in the introduction to Chapter 9. Note that this information refers to arguments in the PACK DATA RECORD function. Because the listed arguments have the same names and meanings in both the PACK DATA RECORD and UNPACK DATA RECORD functions, you can associate the data record components easily with either function.

See Also

INITIALIZE LOCATOR 3

SET LOCATOR MODE

Example 9–1 for a program example using the INQUIRE LOCATOR DEVICE STATE function

INQUIRE MARKER SIZE SCALE FACTOR

INQUIRE MARKER SIZE SCALE FACTOR

Operating States

GKOP, WSOP, WSAC, SGOP

Syntax

GQMKSC (ERRIND, MSZSF)

Argument	Data Type	Access	Description
ERRIND	Integer	Write	Error indicator
MSZSF	Real	Write	Marker size scale factor

Description

The INQUIRE MARKER SIZE SCALE FACTOR function returns the value for the *current marker size scale factor* entry in the GKS state list for all polymarker types.

DEC GKS calculates marker size for all types (except the dot marker type) as the nominal marker size multiplied by the marker size scale factor. The marker size scale factor is a real number passed to the SET MARKER SIZE SCALE FACTOR function. The workstation maps the value to the nearest available marker size defined by the workstation. (The dot marker type is always the smallest dot that the workstation can produce.)

See Also

SET MARKER SIZE SCALE FACTOR

INQUIRE MARKER TYPE
Operating States

GKOP, WSOP, WSAC, SGOP

Syntax

GQMK (ERRIND, MTYPE)

Argument	Data Type	Access	Description
ERRIND	Integer	Write	Error indicator
MTYPE	Integer (constant)	Write	Marker type

Constants

Defined Argument	Constant	Description
MTYPE	GPOINT	Dot
	GPLUS	Plus sign
	GAST	Asterisk
	GOMARK	Circle
	GXMARK	Diagonal cross

Description

The INQUIRE MARKER TYPE function returns the value for the *current marker type* entry in the GKS state list as dots, plus signs, asterisks, circles, diagonal crosses, or any of the device-dependent types.

Every workstation capable of output (of DEC GKS category OUTPUT or OUTIN) defines at least five polymarker types. Additionally, there may be many more device-dependent types. For more information concerning the device-dependent polymarker types, see the *Device Specifics Reference Manual for DEC GKS and DEC PHIGS*. For a complete list of available polymarker types, see Appendix B.

See Also

SET MARKER TYPE

INQUIRE MAXIMUM LENGTH OF WORKSTATION STATE TABLES

INQUIRE MAXIMUM LENGTH OF WORKSTATION STATE TABLES

Operating States

GKOP, WSOP, WSAC, SGOP

Syntax

GQLWK (WTYPE, ERRIND, MPLBTE, MPMBTE, MTXBTE, MFABTE, MPAI, MCOLI)

Argument	Data Type	Access	Description
WTYPE	Integer	Read	Workstation type
ERRIND	Integer	Write	Error indicator
MPLBTE	Integer	Write	Maximum number of polyline bundle table entries
MPMBTE	Integer	Write	Maximum number of polymarker bundle table entries
MTXBTE	Integer	Write	Maximum number of text bundle table entries
MFABTE	Integer	Write	Maximum number of fill area bundle table entries
MPAI	Integer	Write	Maximum number of pattern indexes
MCOLI	Integer	Write	Maximum number of color indexes

Description

The INQUIRE MAXIMUM LENGTH OF WORKSTATION STATE TABLES function returns, for a specific workstation type, the maximum number of polyline bundles, polymarker bundles, text bundles, fill area bundles, pattern indexes, and color indexes table entries.

INQUIRE MAXIMUM LENGTH OF WORKSTATION STATE TABLES 3

INQUIRE MAXIMUM LENGTH OF WORKSTATION STATE TABLES 3

Operating States

GKOP, WSOP, WSAC, SGOP

Syntax

GQLWK3 (WTYPE, ERRIND, MPLBTE, MPMBTE, MTXBTE, MFABTE, MPAI, MEDBTE, MCOLI, MVTE)

Argument	Data Type	Access	Description
WTYPE	Integer	Read	Workstation type
ERRIND	Integer	Write	Error indicator
MPLBTE	Integer	Write	Maximum number of polyline bundle table entries
MPMBTE	Integer	Write	Maximum number of polymarker bundle table entries
MTXBTE	Integer	Write	Maximum number of text bundle table entries
MFABTE	Integer	Write	Maximum number of fill area bundle table entries
MPAI	Integer	Write	Maximum number of pattern indexes
MEDBTE	Integer	Write	Maximum number of edge bundle table entries
MCOLI	Integer	Write	Maximum number of color indexes
MVTE	Integer	Write	Maximum number of view table entries

Description

The INQUIRE MAXIMUM LENGTH OF WORKSTATION STATE TABLES 3 function returns, for a specific workstation type, the maximum number of polyline bundles, polymarker bundles, text bundles, fill area bundles, pattern indexes, color indexes, view indexes, and edge indexes table entries.

INQUIRE MAXIMUM NORMALIZATION TRANSFORMATION NUMBER

INQUIRE MAXIMUM NORMALIZATION TRANSFORMATION NUMBER

Operating States

GKOP, WSOP, WSAC, SGOP

Syntax

GQMNTN (ERRIND, MAXTNR)

Argument	Data Type	Access	Description
ERRIND	Integer	Write	Error status
MAXTNR	Integer	Write	Maximum normalization transformation number supported

Description

The INQUIRE MAXIMUM NORMALIZATION TRANSFORMATION NUMBER function returns the maximum normalization transformation number supported by the GKS implementation being used. The maximum number for the DEC GKS software is 256 (numbered 0 to 255). Normalization transformation number 0 is the unity transformation and cannot be changed.

See Also

SELECT NORMALIZATION TRANSFORMATION
SET VIEWPORT INPUT PRIORITY

INQUIRE MORE SIMULTANEOUS EVENTS

Operating States

WSOP, WSAC, SGOP

Syntax

GQSIM (ERRIND, EVENTS)

Argument	Data Type	Access	Description
ERRIND	Integer	Write	Error indicator
EVENTS	Integer (constant)	Write	More simultaneous events flag

Constants

Defined Argument	Constant	Description
EVENTS	GNMORE	No more events
	GMORE	More events

Description

The INQUIRE MORE SIMULTANEOUS EVENTS function queries the GKS state list to see if there are more events on the event input queue that were entered by the user firing a single trigger.

INQUIRE NAME OF OPEN SEGMENT

INQUIRE NAME OF OPEN SEGMENT

Operating States

SGOP

Syntax

GQOPSG (ERRIND, SEGNA)

Argument	Data Type	Access	Description
ERRIND	Integer	Write	Error indicator
SEGNA	Integer	Write	Name of open segment

Description

The INQUIRE NAME OF OPEN SEGMENT function returns the identification number of the currently open segment.

See Also

CREATE SEGMENT

INQUIRE NORMALIZATION TRANSFORMATION

Operating States

GKOP, WSOP, WSAC, SGOP

Syntax

GQNT (NTNR, ERRIND, WINDOW, VIEWPT)

Argument	Data Type	Access	Description
NTNR	Integer	Read	Normalization transformation number
ERRIND	Integer	Write	Error indicator
WINDOW(4)	Array of reals	Write	Window limits in WC points in the order XMIN, XMAX, YMIN, YMAX
VIEWPT(4)	Array of reals	Write	Viewport limits in NDC points in the order XMIN, XMAX, YMIN, YMAX

Description

The INQUIRE NORMALIZATION TRANSFORMATION function returns the boundaries of the normalization window and the normalization viewport associated with the specified normalization transformation number.

The maximum number of normalization transformations for the DEC GKS software is 256 (numbered 0 to 255). Normalization transformation number 0 is the unity transformation and cannot be changed.

See Also

SELECT NORMALIZATION TRANSFORMATION
 SET VIEWPORT
 SET WINDOW

INQUIRE NORMALIZATION TRANSFORMATION 3

INQUIRE NORMALIZATION TRANSFORMATION 3

Operating States

GKOP, WSOP, WSAC, SGOP

Syntax

GQNT3 (CTNR, ERRIND, WNLIM, VPLIM)

Argument	Data Type	Access	Description
CTNR	Integer	Read	Normalization transformation number
ERRIND	Integer	Write	Error indicator
WNLIM(6)	Array of reals	Write	Window limits in WC points in the order XMIN, XMAX, YMIN, YMAX, ZMIN, ZMAX
VPLIM(6)	Array of reals	Write	Viewport limits in NDC points in the order XMIN, XMAX, YMIN, YMAX, ZMIN, ZMAX

Description

The INQUIRE NORMALIZATION TRANSFORMATION 3 function returns the boundaries of the normalization window and the normalization viewport associated with the specified normalization transformation number.

The maximum number of normalization transformations for the DEC GKS software is 256 (numbered 0 to 255). Normalization transformation number 0 is the unity transformation and cannot be changed.

See Also

SELECT NORMALIZATION TRANSFORMATION
SET VIEWPORT 3
SET WINDOW 3

INQUIRE NUMBER OF AVAILABLE LOGICAL INPUT DEVICES

INQUIRE NUMBER OF AVAILABLE LOGICAL INPUT DEVICES

Operating States

GKOP, WSOP, WSAC, SGOP

Syntax

GQLI (WTYPE, ERRIND, NLCD, NSKD, NVLD, NCHD, NPKD, NSTD)

Argument	Data Type	Access	Description
WTYPE	Integer	Read	Workstation type
ERRIND	Integer	Write	Error indicator
NLCD	Integer	Write	Number of locator devices
NSKD	Integer	Write	Number of stroke devices
NVLD	Integer	Write	Number of valuator devices
NCHD	Integer	Write	Number of choice devices
NPKD	Integer	Write	Number of pick devices
NSTD	Integer	Write	Number of string devices

Description

The INQUIRE NUMBER OF AVAILABLE LOGICAL INPUT DEVICES function returns the number of logical input devices in each class for the given workstation type.

INQUIRE NUMBER OF SEGMENT PRIORITIES SUPPORTED

INQUIRE NUMBER OF SEGMENT PRIORITIES SUPPORTED

Operating States

GKOP, WSOP, WSAC, SGOP

Syntax

QQSGP (WTYPE, ERRIND, NSG)

Argument	Data Type	Access	Description
WTYPE	Integer	Read	Workstation type
ERRIND	Integer	Write	Error indicator
NSG	Integer	Write	Number of segment priorities supported

Description

The INQUIRE NUMBER OF SEGMENT PRIORITIES SUPPORTED function returns the number of segment priorities supported for the specified workstation type.

If the returned number of segment priorities supported is 0, the workstation supports an infinite number of segment priorities.

INQUIRE OPERATING STATE VALUE

Operating States

GKCL, GKOP, WSOP, WSAC, SGOP

Syntax

GQOPS (OPSTA)

Argument	Data Type	Access	Description
OPSTA	Integer (constant)	Write	Operating state

Constants

Defined Argument	Constant	Description
OPSTA	GGKCL	GKS is closed.
	GGKOP	GKS is open.
	GWSOP	At least one workstation is open.
	GWSAC	At least one workstation is active.
	GSGOP	A segment is open.

Description

The INQUIRE OPERATING STATE VALUE function returns the DEC GKS operating state.

See Also

OPEN GKS

INQUIRE PATTERN FACILITIES

INQUIRE PATTERN FACILITIES

Operating States

GKOP, WSOP, WSAC, SGOP

Syntax

GQPAF (WTYPE, ERRIND, NPPAI)

Argument	Data Type	Access	Description
WTYPE	Integer	Read	Workstation type
ERRIND	Integer	Write	Error indicator
NPPAI	Integer	Write	Number of predefined pattern indexes

Description

The INQUIRE PATTERN FACILITIES function returns the number of pattern indexes available for the specified workstation type.

INQUIRE PATTERN REFERENCE POINT**Operating States**

GKOP, WSOP, WSAC, SGOP

Syntax

GQPARF (ERRIND, RFX, RFY)

Argument	Data Type	Access	Description
ERRIND	Integer	Write	Error indicator
RFX, RFY	Real	Write	Pattern reference point

Description

The INQUIRE PATTERN REFERENCE POINT function returns the value for the geometric attribute *current pattern reference point* entry in the GKS state list.

This attribute represents the starting point for a pattern used to fill the designated area. DEC GKS uses this value for all subsequent calls to FILL AREA until another value is specified.

INQUIRE PATTERN REPRESENTATION

INQUIRE PATTERN REPRESENTATION

Operating States

WSOP, WSAC, SGOP

Syntax

GQPAR (WKID, PAI, TYPE, DIMX, DIMY, ERRIND, DX, DY, COLIA)

Argument	Data Type	Access	Description
WKID	Integer	Read	Workstation identifier
PAI	Integer	Read	Pattern index
TYPE	Integer (constant)	Read	Type of values to return
DIMX, DIMY	Integer	Read	Maximum pattern array dimensions
ERRIND	Integer	Write	Error indicator
DX, DY	Integer	Write	Pattern array dimensions
COLIA	Array of integers	Write	Pattern array

Constants

Defined Argument	Constant	Description
TYPE	GSET	Use exact state list values.
	GREALI	Use values approximated by the graphics handler.

Description

The INQUIRE PATTERN REPRESENTATION function returns the values associated with the given pattern index value.

See Also

SET PATTERN REPRESENTATION
SET PATTERN SIZE

INQUIRE PATTERN SIZE

Operating States

GKOP, WSOP, WSAC, SGOP

Syntax

GQPA (ERRIND, PWX, PWY, PHX, PHY)

Argument	Data Type	Access	Description
ERRIND	Integer	Write	Error indicator
PWX, PWY	Real	Write	Pattern width vector
PHX, PHY	Real	Write	Pattern height vector

Description

The INQUIRE PATTERN SIZE function returns the value for the *current pattern size* entry in the GKS state list, which is the height and width vectors in world coordinate units.

The pattern size is replicated for use within a fill area. DEC GKS uses this value for all subsequent calls to FILL AREA until you specify another value.

See Also

SET PATTERN SIZE

INQUIRE PICK DEVICE STATE

INQUIRE PICK DEVICE STATE

Operating States

WSOP, WSAC, SGOP

Syntax

GQPKS (WKID, DEVNUM, TYPE, MLDR, ERRIND, MODE, ESW, ISTAT, ISGNA, IPKID, PET, EAREA, LDR, DATREC)

Argument	Data Type	Access	Description
WKID	Integer	Read	Workstation identifier
DEVNUM	Integer	Read	Pick device number
TYPE	Integer (constant)	Read	Type of values requested
MLDR	Integer	Read	Number of elements in output data record
ERRIND	Integer	Write	Error indicator
MODE	Integer (constant)	Write	Operating mode
ESW	Integer (constant)	Write	Echo flag
ISTAT	Integer (constant)	Write	Initial pick status
ISGNA	Integer	Write	Initial segment name
IPKID	Integer	Write	Initial pick identifier
PET	Integer	Write	Prompt and echo type
EAREA(4)	Array of reals	Write	Device coordinate echo area in the order XMIN, XMAX, YMIN, YMAX
LDR	Integer	Write	Number of array elements used in the output data record
DATREC(MLDR)	Character*80	Write	Output data record

Constants

Defined Argument	Constant	Description
TYPE	GSET	Use exact state list values.
	GREALI	Use values approximated by the graphics handler.

INQUIRE PICK DEVICE STATE

Defined Argument	Constant	Description
MODE	GREQU	Request input mode.
	GSAMPL	Sample input mode.
	GEVENT	Event input mode.
ESW	GNECHO	Echo disabled.
	GECHO	Echo enabled.
ISTAT	GOK	Return the initial measure.
	GNPICK	Return no pick.

Description

The INQUIRE PICK DEVICE STATE function returns the current state of the given pick-class logical input device.

The data record returned by this function is the input data record associated with the PET returned in the argument *PET*. Use the UNPACK DATA RECORD function to access this information.

The data record components for each of the possible PETs are listed in the introduction to Chapter 9. Note that this information refers to arguments in the PACK DATA RECORD function. Because the listed arguments have the same names and meanings in both the PACK DATA RECORD and UNPACK DATA RECORD functions, you can associate the data record components easily with either function.

See Also

INITIALIZE PICK

SET PICK MODE

Example 9–2 for a program example using the INQUIRE PICK DEVICE STATE function

INQUIRE PICK DEVICE STATE 3

INQUIRE PICK DEVICE STATE 3

Operating States

WSOP, WSAC, SGOP

Syntax

GQPKS3 (WKID, DEVNUM, TYPE, MLDR, ERRIND, MODE, ESW, ISTAT, ISGNA, IPKID, PET, EVOL, LDR, DATREC)

Argument	Data Type	Access	Description
WKID	Integer	Read	Workstation identifier
DEVNUM	Integer	Read	Pick device number
TYPE	Integer (constant)	Read	Type of values requested
MLDR	Integer	Read	Number of elements in output data record
ERRIND	Integer	Write	Error indicator
MODE	Integer (constant)	Write	Operating mode
ESW	Integer (constant)	Write	Echo flag
ISTAT	Integer (constant)	Write	Initial pick status
ISGNA	Integer	Write	Initial segment name
IPKID	Integer	Write	Initial pick identifier
PET	Integer	Write	Prompt and echo type
EVOL(6)	Array of reals	Write	Device coordinate echo volume in the order XMIN, XMAX, YMIN, YMAX, ZMIN, ZMAX
LDR	Integer	Write	Number of array elements used in the output data record
DATREC(MLDR)	Character*80	Write	Output data record

Constants

Defined Argument	Constant	Description
TYPE	GSET	Use exact state list values.
	GREALI	Use values approximated by the graphics handler.

INQUIRE PICK DEVICE STATE 3

Defined Argument	Constant	Description
MODE	GREQU	Request input mode.
	GSAMPL	Sample input mode.
	GEVENT	Event input mode.
ESW	GNECHO	Echo disabled.
	GECHO	Echo enabled.
ISTAT	GOK	Return the initial measure.
	GNPICK	Return no pick.

Description

The INQUIRE PICK DEVICE STATE 3 function returns the current state of the given pick-class logical input device.

The data record returned by this function is the input data record associated with the PET returned in the argument *PET*. Use the UNPACK DATA RECORD function to access this information.

The data record components for each of the possible PETs are listed in the introduction to Chapter 9. Note that this information refers to arguments in the PACK DATA RECORD function. Because the listed arguments have the same names and meanings in both the PACK DATA RECORD and UNPACK DATA RECORD functions, you can associate the data record components easily with either function.

See Also

INITIALIZE PICK 3

SET PICK MODE

Example 9–2 for a program example using the INQUIRE PICK DEVICE STATE function

INQUIRE PIXEL

INQUIRE PIXEL

Operating States

WSOP, WSAC, SGOP

Syntax

GQPX (WKID, PX, PY, ERRIND, COLI)

Argument	Data Type	Access	Description
WKID	Integer	Read	Workstation identifier
PX, PY	Real	Read	Point in WC units
ERRIND	Integer	Write	Error indicator
COLI	Integer	Write	Color index

Description

The INQUIRE PIXEL function returns the color index of an individual pixel on the display surface.

The specified point is transformed by the current normalization and workstation transformations, and by using a view index of 0, to a pixel on the display surface. The color index associated with this pixel is returned. If the color index of the pixel cannot be ascertained, the value -1 is returned for the pixel.

INQUIRE PIXEL ARRAY

Operating States

WSOP, WSAC, SGOP

Syntax

GQPXA (WKID, PX, PY, DIMX, DIMY, ISC, ISR, DX, DY, ERRIND, INVVAL, COLIA)

Argument	Data Type	Access	Description
WKID	Integer	Read	Workstation identifier
PX, PY	Real	Read	Upper left corner of the requested rectangular region in WC points
DIMX, DIMY	Integer	Read	Dimensions of color index array
ISC, ISR	Integer	Read	Start column and start row in the color index array
DX, DY	Integer	Read	Size of rectangular region in pixels
ERRIND	Integer	Write	Error indicator
INVVAL	Integer (constant)	Write	Flag for invalid color index values
COLIA(DIMX, DIMY)	Array of integers	Write	Color index array

Constants

Defined Argument	Constant	Description
INVVAL	GABSNT	Color array contains no invalid indexes.
	GPRSNT	Color array contains invalid indexes.

Description

The INQUIRE PIXEL ARRAY function returns the color indexes of pixels in a rectangular region on the screen.

The specified point is transformed by the current normalization and workstation transformations, and by using a view index of 0, to a pixel on the display surface. The color indexes of the array of pixels whose upper left-hand corner is the transformed point are returned. If the color index of a particular pixel cannot be ascertained, the value -1 is returned for that pixel.

INQUIRE PIXEL ARRAY

See Also

SET COLOUR REPRESENTATION

INQUIRE PIXEL ARRAY DIMENSIONS

Operating States

WSOP, WSAC, SGOP

Syntax

GQPXAD (WKID, PX, PY, QX, QY, ERRIND, DX, DY)

Argument	Data Type	Access	Description
WKID	Integer	Read	Workstation identifier
PX, PY, QX, QY	Real	Read	Points describing rectangle ((PX, PY) is the upper left point of the rectangle, and (QX,QY) is the lower left point of the rectangle.)
ERRIND	Integer	Write	Error indicator
DX, DY	Integer	Write	X and Y dimensions of pixel array

Description

The INQUIRE PIXEL ARRAY DIMENSIONS function returns the number of pixels in the X and Y axis of a rectangular portion of the display surface.

The two specified points determine the rectangular portion of the display surface. By transforming the two specified points by the current normalization and workstation transformations, and by using a view index of 0, a rectangle is mapped onto the display surface. No clipping is applied. The number of columns and rows of pixels whose positions lie within the rectangle is returned.

INQUIRE POLYLINE COLOUR INDEX

INQUIRE POLYLINE COLOUR INDEX

Operating States

GKOP, WSOP, WSAC, SGOP

Syntax

GQPLCI (ERRIND, COLI)

Argument	Data Type	Access	Description
ERRIND	Integer	Write	Error indicator
COLI	Integer	Write	Polyline color index

Description

The INQUIRE POLYLINE COLOUR INDEX function returns the value for the *current polyline color index* entry in the GKS state list.

See Also

SET POLYLINE COLOUR INDEX
SET COLOUR REPRESENTATION

INQUIRE POLYLINE FACILITIES

Operating States

GKOP, WSOP, WSAC, SGOP

Syntax

GQPLF (WTYPE, N, ERRIND, NLT, LT, NLW, NOMLW, RLWMIN, RLWMAX, NPPLI)

Argument	Data Type	Access	Description
WTYPE	Integer	Read	Workstation type
N	Integer	Read	Requested list element from the list of available line types
ERRIND	Integer	Write	Error indicator
NLT	Integer	Write	Number of available line types
LT	Integer	Write	<i>Nth</i> element of the list of available line types
NLW	Integer	Write	Number of available line widths
NOMLW	Real	Write	Nominal line width
RLWMIN, RLWMAX	Real	Write	Minimum and maximum line widths
NPPLI	Integer	Write	Number of predefined polyline indexes

Description

The INQUIRE POLYLINE FACILITIES function queries the workstation description table and returns the number of line types, the requested element in the list of available line types, the number of line widths, the nominal, minimum, and maximum line widths, and the number of predefined polyline index values.

If the number of line widths returned is 0, the workstation supports a continuous range of line widths.

INQUIRE POLYLINE INDEX

INQUIRE POLYLINE INDEX

Operating States

GKOP, WSOP, WSAC, SGOP

Syntax

GQPLI (ERRIND, PLI)

Argument	Data Type	Access	Description
ERRIND	Integer	Write	Error indicator
PLI	Integer	Write	Polyline index

Description

The INQUIRE POLYLINE INDEX function returns the value for the *current polyline index* entry in the GKS state list.

The polyline bundle table contains entries for the polyline color index, polyline type, and polyline width scale factor attribute values. When calling POLYLINE, DEC GKS uses the bundle table only if the corresponding ASF has been set to BUNDLED.

See Also

SET POLYLINE INDEX

INQUIRE POLYLINE REPRESENTATION

Operating States

WSOP, WSAC, SGOP

Syntax

GQPLR (WKID, PLI, TYPE, ERRIND, LTYPE, LWIDTH, COLI)

Argument	Data Type	Access	Description
WKID	Integer	Read	Workstation identifier
PLI	Integer	Read	Polyline index
TYPE	Integer (constant)	Read	Type of value to return
ERRIND	Integer	Write	Error indicator
LTYPE	Integer (constant)	Write	Polyline type
LWIDTH	Real	Write	Line width scale factor
COLI	Integer	Write	Polyline color index value

Constants

Defined Argument	Constant	Description
TYPE	GSET	Use exact state list values.
	GREALI	Use values approximated by the graphics handler.
LTYPE	GLSOLI	Solid line.
	GLDASH	Dashed line.
	GLDOT	Dotted line.
	GLDASD	Dashed-dotted line.

Description

The INQUIRE POLYLINE REPRESENTATION function returns the values associated with the given polyline index value.

If the specified polyline index is not in the polyline bundle table on the specified workstation, and the specified type of returned values is REALIZED, the representation for polyline index 1 is returned.

See Also

SET POLYLINE REPRESENTATION

INQUIRE POLYMARKER COLOUR INDEX

INQUIRE POLYMARKER COLOUR INDEX

Operating States

GKOP, WSOP, WSAC, SGOP

Syntax

GQPMCI (ERRIND, COLI)

Argument	Data Type	Access	Description
ERRIND	Integer	Write	Error indicator
COLI	Integer	Write	Polymarker color index

Description

The INQUIRE POLYMARKER COLOUR INDEX function queries the GKS state list and returns the value for the *current polymarker color index* entry.

All output polymarkers are shown in the color corresponding to the index value, provided the ASF corresponding to the polymarker color is set to INDIVIDUAL.

See Also

SET COLOUR REPRESENTATION
SET POLYMARKER COLOUR INDEX

INQUIRE POLYMARKER FACILITIES

Operating States

GKOP, WSOP, WSAC, SGOP

Syntax

GQPMF (WTYPE, N, ERRIND, NMT, MT, NMS, NOMMS, RMSMIN, RMSMAX, NPPMI)

Argument	Data Type	Access	Description
WTYPE	Integer	Read	Workstation type
N	Integer	Read	Requested list element from the list of available marker types
ERRIND	Integer	Write	Error indicator
NMT	Integer	Write	Number of available marker types
MT	Integer (constant)	Write	Nth element of the list of available marker types
NMS	Integer	Write	Number of marker sizes
NOMMS	Real	Write	Nominal marker size
RMSMIN, RMSMAX	Real	Write	Minimum and maximum marker sizes
NPPMI	Integer	Write	Number of predefined marker indexes

Constants

Defined Argument	Constant	Description
MT	GPOINT	Dot
	GPLUS	Plus sign
	GAST	Asterisk
	GOMARK	Circle
	GXMARK	Diagonal cross

Description

The INQUIRE POLYMARKER FACILITIES function returns the number of marker types; the requested element in the list of available marker types; the nominal, minimum, and maximum marker sizes; and the number of predefined polymarker index values.

If the returned marker size is 0, the workstation supports a continuous range of marker sizes.

INQUIRE POLYMARKER INDEX

INQUIRE POLYMARKER INDEX

Operating States

GKOP, WSOP, WSAC, SGOP

Syntax

GQPMI (ERRIND, PMI)

Argument	Data Type	Access	Description
ERRIND	Integer	Write	Error indicator
PMI	Integer	Write	Polymarker index

Description

The INQUIRE POLYMARKER INDEX function returns the value for the *current polymarker index* entry in the GKS state list.

The polymarker bundle table contains entries for the polymarker color index, polymarker type, and polymarker size scale factor attribute values. When calling POLYMARKER, DEC GKS uses the bundle table only if the corresponding ASF has been set to BUNDLED.

See Also

SET POLYMARKER INDEX

INQUIRE POLYMARKER REPRESENTATION

Operating States

WSOP, WSAC, SGOP

Syntax

GQPMPR (WKID, PMI, TYPE, ERRIND, MTYPE, MSZSF, COLI)

Argument	Data Type	Access	Description
WKID	Integer	Read	Workstation identifier
PMI	Integer	Read	Polymarker index on workstation
TYPE	Integer (constant)	Read	Type of values to return
ERRIND	Integer	Write	Error indicator
MTYPE	Integer (constant)	Write	Marker type
MSZSF	Real	Write	Marker size scale factor
COLI	Integer	Write	Polymarker color index

Constants

Defined Argument	Constant	Description
TYPE	GSET	Use exact state list values.
	GREALI	Use values approximated by the graphics handler.
MTYPE	GPOINT	Dot.
	GPLUS	Plus sign.
	GAST	Asterisk.
	GOMARK	Circle.
	GXMARK	Diagonal cross.

Description

The INQUIRE POLYMARKER REPRESENTATION function returns the values associated with the given polymarker index value.

If the specified polymarker index is not in the polymarker bundle table on the specified workstation, and the specified type of returned values is REALIZED, the representation for polymarker index 1 is returned.

INQUIRE POLYMARKER REPRESENTATION

See Also

SET POLYMARKER REPRESENTATION

INQUIRE PREDEFINED COLOUR REPRESENTATION

INQUIRE PREDEFINED COLOUR REPRESENTATION

Operating States

GKOP, WSOP, WSAC, SGOP

Syntax

GQPCR (WTYPE, COLI, ERRIND, C1, C2, C3)

Argument	Data Type	Access	Description
WTYPE	Integer	Read	Workstation type
COLI	Integer	Read	Predefined color index
ERRIND	Integer	Write	Error indicator
C1, C2, C3	Real	Write	Color components

Description

The INQUIRE PREDEFINED COLOUR REPRESENTATION function returns the predefined color component values associated with the color model in the workstation state list for the specified color index.

See Also

INQUIRE COLOUR MODEL FACILITIES

INQUIRE PREDEFINED EDGE REPRESENTATION

INQUIRE PREDEFINED EDGE REPRESENTATION

Operating States

GKOP, WSOP, WSAC, SGOP

Syntax

GQPEDR (WTYPE, PEDI, ERRIND, EDFLAG, EDTYPE, EDWSF, COLI)

Argument	Data Type	Access	Description
WTYPE	Integer	Read	Workstation type
PEDI	Integer	Read	Predefined edge index
ERRIND	Integer	Write	Error indicator
EDFLAG	Integer (constant)	Write	Edge flag
EDTYPE	Integer (constant)	Write	Edge type
EDWSF	Real	Write	Edge width scale factor
COLI	Integer	Write	Edge color index

Constants

Defined Argument	Constant	Description
EDFLAG	GOFF	Edge off
	GON	Edge on
EDTYPE	GESOLI	Solid edge
	GEDASH	Dashed edge
	GEDOT	Dotted edge
	GEDASD	Dashed-dotted edge

Description

The INQUIRE PREDEFINED EDGE REPRESENTATION function returns the values for the predefined edge flag, edge type, edge width scale factor, and edge color index associated with the specified edge index.

See Also

SET COLOUR REPRESENTATION

INQUIRE PREDEFINED FILL AREA REPRESENTATION

INQUIRE PREDEFINED FILL AREA REPRESENTATION

Operating States

GKOP, WSOP, WSAC, SGOP

Syntax

GQPFAR (WTYPE, PFAI, ERRIND, INTS, STYLI, COLI)

Argument	Data Type	Access	Description
WTYPE	Integer	Read	Workstation type
PFAI	Integer	Read	Predefined fill area index
ERRIND	Integer	Write	Error indicator
INTS	Integer (constant)	Write	Fill area interior style
STYLI	Integer	Write	Fill area style index
COLI	Integer	Write	Fill area color index

Constants

Defined Argument	Constant	Description
INTS	GHOLLO	Hollow interior
	GSOLID	Solid interior
	GPATTR	Patterned interior
	GHATCH	Hatched interior

Description

The INQUIRE PREDEFINED FILL AREA REPRESENTATION function returns the predefined values for the interior style, style index, and fill area color index associated with a specific fill area index for a given workstation type.

See Also

SET COLOUR REPRESENTATION
SET FILL AREA STYLE INDEX

INQUIRE PREDEFINED PATTERN REPRESENTATION

INQUIRE PREDEFINED PATTERN REPRESENTATION

Operating States

GKOP, WSOP, WSAC, SGOP

Syntax

GQPPAR (WTYPE, PPAI, DIMX, DIMY, ERRIND, DX, DY, COLIA)

Argument	Data Type	Access	Description
WTYPE	Integer	Read	Workstation type
PPAI	Integer	Read	Predefined pattern index
DIMX, DIMY	Integer	Read	Maximum pattern array dimensions
ERRIND	Integer	Write	Error indicator
DX, DY	Integer	Write	Dimensions of the color pattern array
COLIA(DIMX,DIMY)	Array of integers	Write	Color pattern array

Description

The INQUIRE PREDEFINED PATTERN REPRESENTATION function returns a description of the specific pattern by returning the pattern dimensions and the array of color indexes that comprises the pattern.

INQUIRE PREDEFINED POLYLINE REPRESENTATION

INQUIRE PREDEFINED POLYLINE REPRESENTATION

Operating States

GKOP, WSOP, WSAC, SGOP

Syntax

GQPPLR (WTYPE, PLI, ERRIND, LTYPE, LWIDTH, COLI)

Argument	Data Type	Access	Description
WTYPE	Integer	Read	Workstation type
PLI	Integer	Read	Predefined polyline index
ERRIND	Integer	Write	Error indicator
LTYPE	Integer (constant)	Write	Line type
LWIDTH	Real	Write	Line width scale factor
COLI	Integer	Write	Polyline color index

Constants

Defined Argument	Constant	Description
LTYPE	GLSOLI	Solid line
	GLDASH	Dashed line
	GLDOT	Dotted line
	GLDASD	Dashed-dotted line

Description

The INQUIRE PREDEFINED POLYLINE REPRESENTATION function returns the predefined values for the line type, color index, and line width associated with a specific polyline index for a given workstation type.

INQUIRE PREDEFINED POLYMARKER REPRESENTATION

INQUIRE PREDEFINED POLYMARKER REPRESENTATION

Operating States

GKOP, WSOP, WSAC, SGOP

Syntax

GQPPMR (WTYPE, PMI, ERRIND, MSTYPE, MSZSF, COLI)

Argument	Data Type	Access	Description
WTYPE	Integer	Read	Workstation type
PMI	Integer	Read	Predefined polymarker index
ERRIND	Integer	Write	Error indicator
MSTYPE	Integer (constant)	Write	Marker type
MSZSF	Real	Write	Marker size scale factor
COLI	Integer	Write	Polymarker color index

Constants

Defined Argument	Constant	Description
MSTYPE	GPOINT	Dot
	GPLUS	Plus sign
	GAST	Asterisk
	GOMARK	Circle
	GXMARK	Diagonal cross

Description

The INQUIRE PREDEFINED POLYMARKER REPRESENTATION function returns the predefined values for the marker type, color index, and marker size scale factor associated with a specific polymarker index for a given workstation type.

INQUIRE PREDEFINED TEXT REPRESENTATION

INQUIRE PREDEFINED TEXT REPRESENTATION

Operating States

GKOP, WSOP, WSAC, SGOP

Syntax

GQPTXR (WTYPE, PTXI, ERRIND, FONT, PREC, CHXP, CHSP, COLI)

Argument	Data Type	Access	Description
WTYPE	Integer	Read	Workstation type
PTXI	Integer	Read	Text index
ERRIND	Integer	Write	Error indicator
FONT	Integer	Write	Text font number
PREC	Integer (constant)	Write	Text precision
CHXP	Real	Write	Character expansion factor
CHSP	Real	Write	Space between characters
COLI	Integer	Write	Text color index

Constants

Defined Argument	Constant	Description
PREC	GSTRP	String precision. DEC GKS evaluates character height and width attributes only.
	GCHARP	Character precision. DEC GKS evaluates each character for compliance with all other specified text attributes.
	GSTRKP	Stroke precision. DEC GKS looks for exact compliance with all specified text attributes.

Description

The INQUIRE PREDEFINED TEXT REPRESENTATION function returns the predefined values for the text font and precision, character expansion factor, character spacing, and text color index associated with the specific text index for a given workstation type.

INQUIRE PREDEFINED VIEW REPRESENTATION

INQUIRE PREDEFINED VIEW REPRESENTATION

Operating States

GKOP, WSOP, WSAC, SGOP

Syntax

GQPVWR (WTYPE, PVWI, ERRIND, VOM, VMM, VCLIP, XYCLIP, BKCLIP, FRCLIP)

Argument	Data Type	Access	Description
WTYPE	Integer	Read	Workstation type
PVWI	Integer	Read	Predefined view index
ERRIND	Integer	Write	Error indicator
VOM(4,4)	Array of reals	Write	View orientation matrix
VMM(4,4)	Array of reals	Write	View mapping matrix
VCLIP(6)	Array of reals	Write	View clipping limits in the order XMIN, XMAX, YMIN, YMAX, ZMIN, ZMAX
XYCLIP	Integer (constant)	Write	XY clipping indicator
BKCLIP	Integer (constant)	Write	Back clipping indicator
FRCLIP	Integer (constant)	Write	Front clipping indicator

Constants

Defined Argument	Constant	Description
XYCLIP, BKCLIP, FRCLIP	GNCLIP	Clipping disabled
	GCLIP	Clipping enabled

Description

The INQUIRE PREDEFINED VIEW REPRESENTATION function returns the predefined values for the view orientation matrix, view mapping matrix, view clipping limits in NPC points, XY clipping indicator, back clipping indicator, and the front clipping indicator associated with a specific view index for a given workstation type.

INQUIRE SEGMENT ATTRIBUTES

Operating States

WSOP, WSAC, SGOP

Syntax

GQSGA (SGNA, ERRIND, M, VIS, HIL, PRIOR, DET)

Argument	Data Type	Access	Description
SGNA	Integer	Read	Segment name
ERRIND	Integer	Write	Error indicator
M(2,3)	Array of reals	Write	Segment transformation matrix with an array as follows:
			$\begin{bmatrix} M(1, 1) & M(1, 2) & M(1, 3) \\ M(2, 1) & M(2, 2) & M(2, 3) \end{bmatrix}$
VIS	Integer (constant)	Write	Visibility flag
HIL	Integer (constant)	Write	Highlighting flag
PRIOR	Real	Write	Segment priority
DET	Integer (constant)	Write	Detectability flag

Constants

Defined Argument	Constant	Description
VIS	GINVIS	The segment is not visible on the surface.
	GVISI	The segment is visible on the surface.
HIL	GNORML	The segment is not highlighted on the surface.
	GHILIT	The segment is highlighted on the surface.
DET	GUNDET	The segment cannot be picked.
	GDETEC	The segment can be picked.

INQUIRE SEGMENT ATTRIBUTES

Description

The INQUIRE SEGMENT ATTRIBUTES function returns the current attribute values of the specified segment: the two-dimensional segment transformation matrix, visibility, highlighting, priority, and detectability.

See Also

ACCUMULATE TRANSFORMATION MATRIX
CREATE SEGMENT
EVALUATE TRANSFORMATION MATRIX
SET DETECTABILITY
SET HIGHLIGHTING
SET SEGMENT PRIORITY
SET SEGMENT TRANSFORMATION
SET VISIBILITY

INQUIRE SEGMENT ATTRIBUTES 3

Operating States

WSOP, WSAC, SGOP

Syntax

GQSGA3 (SGNA, ERRIND, M, VIS, HIL, PRIOR, DET)

Argument	Data Type	Access	Description
SGNA	Integer	Read	Segment name
ERRIND	Integer	Write	Error indicator
M(3,4)	Array of reals	Write	Segment transformation matrix with an array as follows:
			$\begin{bmatrix} M(1, 1) & M(1, 2) & M(1, 3) & M(1, 4) \\ M(2, 1) & M(2, 2) & M(2, 3) & M(2, 4) \\ M(3, 1) & M(3, 2) & M(3, 3) & M(3, 4) \end{bmatrix}$
VIS	Integer (constant)	Write	Visibility flag
HIL	Integer (constant)	Write	Highlighting flag
PRIOR	Real	Write	Segment priority
DET	Integer (constant)	Write	Detectability flag

Constants

Defined Argument	Constant	Description
VIS	GINVIS	The segment is not visible on the surface.
	GVISI	The segment is visible on the surface.
HIL	GNORML	The segment is not highlighted on the surface.
	GHILIT	The segment is highlighted on the surface.
DET	GUNDET	The segment cannot be picked.
	GDETEC	The segment can be picked.

INQUIRE SEGMENT ATTRIBUTES 3

Description

The INQUIRE SEGMENT ATTRIBUTES 3 function returns the current attribute values of the specified segment: the three-dimensional segment transformation matrix, visibility, highlighting, priority, and detectability.

See Also

ACCUMULATE TRANSFORMATION MATRIX 3
CREATE SEGMENT
EVALUATE TRANSFORMATION MATRIX 3
SET DETECTABILITY
SET HIGHLIGHTING
SET SEGMENT PRIORITY
SET SEGMENT TRANSFORMATION
SET VISIBILITY

INQUIRE SET OF ACTIVE WORKSTATIONS

Operating States

GKOP, WSOP, WSAC, SGOP

Syntax

GQACWK (N, ERRIND, OL, WKID)

Argument	Data Type	Access	Description
N	Integer	Read	Requested set member from the set of active workstations
ERRIND	Integer	Write	Error indicator
OL	Integer	Write	Number of active workstations
WKID	Integer	Write	<i>N</i> th member of the set of active workstations

Description

The INQUIRE SET OF ACTIVE WORKSTATIONS function returns the number of active workstations and the workstation identifier of the requested set member.

The FORTRAN binding standard lists this function as INQUIRE SET member OF ACTIVE WORKSTATIONS.

See Also

ACTIVATE WORKSTATION

INQUIRE SET OF ASSOCIATED WORKSTATIONS

INQUIRE SET OF ASSOCIATED WORKSTATIONS

Operating States

WSOP, WSAC, SGOP

Syntax

GQASWK (SGNA, N, ERRIND, OL, WKID)

Argument	Data Type	Access	Description
SGNA	Integer	Read	Segment name
N	Integer	Read	Requested set member from the set of associated workstations
ERRIND	Integer	Write	Error indicator
OL	Integer	Write	Number of associated workstations
WKID	Integer	Write	Nth member of the set of associated workstations

Description

The INQUIRE SET OF ASSOCIATED WORKSTATIONS function returns the number of workstations associated with the specified segment and the workstation identifier of the requested set member.

The FORTRAN binding standard lists this function as INQUIRE SET member OF ASSOCIATED WORKSTATIONS.

See Also

ASSOCIATE SEGMENT WITH WORKSTATION
CREATE SEGMENT

INQUIRE SET OF OPEN WORKSTATIONS

Operating States

GKOP, WSOP, WSAC, SGOP

Syntax

GQOPWK (N, ERRIND, OL, WKID)

Argument	Data Type	Access	Description
N	Integer	Read	Requested set member from the set of open workstations
ERRIND	Integer	Write	Error indicator
OL	Integer	Write	Number of open workstations
WKID	Integer	Write	<i>N</i> th member of the set of open workstations

Description

The INQUIRE SET OF OPEN WORKSTATIONS function returns the number of open workstations and the workstation identifier of the requested member of the set of open workstations.

The FORTRAN binding standard lists this function as INQUIRE SET member OF OPEN WORKSTATIONS.

See Also

OPEN WORKSTATION

INQUIRE SET OF SEGMENT NAMES IN USE

INQUIRE SET OF SEGMENT NAMES IN USE

Operating States

WSOP, WSAC, SGOP

Syntax

GQSGUS (N, ERRIND, OL, SGNA)

Argument	Data Type	Access	Description
N	Integer	Read	Requested set member from the set of segment names in use
ERRIND	Integer	Write	Error indicator
OL	Integer	Write	Number of existing segments
SGNA	Integer	Write	<i>N</i> th member of the set of segment names in use

Description

The INQUIRE SET OF SEGMENT NAMES IN USE function returns the number of existing segments and the name of the requested member of the set of segment names in use.

The FORTRAN binding standard lists this function as INQUIRE SET member OF SEGMENT NAMES IN USE.

See Also

CREATE SEGMENT

INQUIRE SET OF SEGMENT NAMES ON WORKSTATION

INQUIRE SET OF SEGMENT NAMES ON WORKSTATION

Operating States

WSOP, WSAC, SGOP

Syntax

GQSGWK (WKID, N, ERRIND, OL, SGNA)

Argument	Data Type	Access	Description
WKID	Integer	Read	Workstation identifier
N	Integer	Read	Requested set member from the set of segment names stored on the specified workstation
ERRIND	Integer	Write	Error indicator
OL	Integer	Write	Number of segment names in set
SGNA	Integer	Write	<i>N</i> th member of the set of stored segment names for the specified workstation

Description

The INQUIRE SET OF SEGMENT NAMES ON WORKSTATION function returns the number of segment names stored on the specified workstation and the segment name of the requested member of the set of stored segment names.

The FORTRAN binding standard lists this function as INQUIRE SET member OF SEGMENT NAMES ON WORKSTATION.

See Also

CREATE SEGMENT
RENAME SEGMENT

INQUIRE STRING DEVICE STATE

INQUIRE STRING DEVICE STATE

Operating States

WSOP, WSAC, SGOP

Syntax

GQSTS (WKID, DEVNUM, MLDR, ERRIND, MODE, ESW, LOSTR, ISTR, PET, EAREA, BUFLLEN, INIPOS, LDR, DATREC)

Argument	Data Type	Access	Description
WKID	Integer	Read	Workstation identifier
DEVNUM	Integer	Read	String device number
MLDR	Integer	Read	Number of elements in output data record
ERRIND	Integer	Write	Error indicator
MODE	Integer (constant)	Write	Operating mode
ESW	Integer (constant)	Write	Echo flag
LOSTR	Integer	Write	Returned string size
ISTR	Character*(*)	Write	Initial string
PET	Integer	Write	Prompt and echo type
EAREA(4)	Array of reals	Write	Device coordinate echo area in the order XMIN, XMAX, YMIN, YMAX
BUFLLEN	Integer	Write	Input buffer length
INIPOS	Integer	Write	Initial cursor position
LDR	Integer	Write	Number of array elements used in the output data record
DATREC(MLDR)	Character*80	Write	Output data record

Constants

Defined Argument	Constant	Description
MODE	GREQU	Request input mode
	GSAMPL	Sample input mode
	GEVENT	Event input mode
ESW	GNECHO	Echo disabled
	GECHO	Echo enabled

Description

The INQUIRE STRING DEVICE STATE function returns the current state of the given string-class logical input device.

The data record returned by this function is the input data record associated with the PET returned in the argument *PET*. For FORTRAN, there are no data record components associated with the string-class PETs. All information is returned in the other function arguments. The number of array elements used in the data record is 0.

See Also

INITIALIZE STRING

SET STRING MODE

Example 9-3 for a program example using the INQUIRE STRING DEVICE STATE function

INQUIRE STRING DEVICE STATE (FORTRAN-77 Subset)

INQUIRE STRING DEVICE STATE (FORTRAN-77 Subset)

Operating States

WSOP, WSAC, SGOP

Syntax

GQSTS (WKID, DEVNUM, MLDR, ERRIND, MODE, ESW, LOSTR, ISTR, PET, EAREA, BUFLLEN, INIPOS, LDR, DATREC)

Argument	Data Type	Access	Description
WKID	Integer	Read	Workstation identifier
DEVNUM	Integer	Read	String device number
MLDR	Integer	Read	Number of elements in output data record
ERRIND	Integer	Write	Error indicator
MODE	Integer (constant)	Write	Operating mode
ESW	Integer (constant)	Write	Echo flag
LOSTR	Integer	Write	Returned string size
ISTR	Character*80	Write	Initial string
PET	Integer	Write	Prompt and echo type
EAREA(4)	Array of reals	Write	Device coordinate echo area in the order XMIN, XMAX, YMIN, YMAX
BUFLLEN	Integer	Write	Input buffer length
INIPOS	Integer	Write	Initial cursor position
LDR	Integer	Write	Number of array elements used in the output data record
DATREC(MLDR)	Character*80	Write	Output data record

Constants

Defined Argument	Constant	Description
MODE	GREQU	Request input mode
	GSAMPL	Sample input mode
	GEVENT	Event input mode
ESW	GNECHO	Echo disabled
	GECHO	Echo enabled

INQUIRE STRING DEVICE STATE (FORTRAN-77 Subset)

Description

The INQUIRE STRING DEVICE STATE function returns the current state of the given string-class logical input device.

The data record returned by this function is the input data record associated with the PET returned in the argument *PET*. For FORTRAN, there are no data record components associated with the string-class PETs. All information is returned in the other function arguments. The number of array elements used in the data record is 0.

See Also

INITIALIZE STRING

SET STRING MODE

Example 9-3 for a program example using the INQUIRE STRING DEVICE STATE function

INQUIRE STRING DEVICE STATE 3

INQUIRE STRING DEVICE STATE 3

Operating States

WSOP, WSAC, SGOP

Syntax

GQSTS3 (WKID, DEVNUM, MLDR, ERRIND, MODE, ESW, LOSTR, ISTR, PET, EVOL, BUFLLEN, INIPOS, LDR, DATREC)

Argument	Data Type	Access	Description
WKID	Integer	Read	Workstation identifier
DEVNUM	Integer	Read	String device number
MLDR	Integer	Read	Number of elements in output data record
ERRIND	Integer	Write	Error indicator
MODE	Integer (constant)	Write	Operating mode
ESW	Integer (constant)	Write	Echo flag
LOSTR	Integer	Write	Returned string size
ISTR	Character*(*)	Write	Initial string
PET	Integer	Write	Prompt and echo type
EVOL(6)	Array of reals	Write	Device coordinate echo volume in the order XMIN, XMAX, YMIN, YMAX, ZMIN, ZMAX
BUFLLEN	Integer	Write	Input buffer size
INIPOS	Integer	Write	Initial cursor position
LDR	Integer	Write	Number of array elements used in the output data record
DATREC(MLDR)	Character*80	Write	Output data record

Constants

Defined Argument	Constant	Description
MODE	GREQU	Request input mode
	GSAMPL	Sample input mode
	GEVENT	Event input mode
ESW	GNECHO	Echo disabled
	GECHO	Echo enabled

Description

The INQUIRE STRING DEVICE STATE 3 function returns the current state of the given string-class logical input device.

The data record returned by this function is the input data record associated with the PET returned in the argument *PET*. For FORTRAN, there are no data record components associated with the string-class PETs. All information is returned in the other function arguments. The number of array elements used in the data record is 0.

See Also

INITIALIZE STRING 3

SET STRING MODE

Example 9–3 for a program example using the INQUIRE STRING DEVICE STATE function

INQUIRE STRING DEVICE STATE 3 (FORTRAN-77 Subset)

INQUIRE STRING DEVICE STATE 3 (FORTRAN-77 Subset)

Operating States

WSOP, WSAC, SGOP

Syntax

GQSTS3 (WKID, DEVNUM, MLDR, ERRIND, MODE, ESW, LOSTR, ISTR, PET, EVOL, BUFLLEN, INIPOS, LDR, DATREC)

Argument	Data Type	Access	Description
WKID	Integer	Read	Workstation identifier
DEVNUM	Integer	Read	String device number
MLDR	Integer	Read	Number of elements in output data record
ERRIND	Integer	Write	Error indicator
MODE	Integer (constant)	Write	Operating mode
ESW	Integer (constant)	Write	Echo flag
LOSTR	Integer	Write	Returned string size
ISTR	Character*80	Write	Initial string
PET	Integer	Write	Prompt and echo type
EVOL(6)	Array of reals	Write	Device coordinate echo volume in the order XMIN, XMAX, YMIN, YMAX, ZMIN, ZMAX
BUFLLEN	Integer	Write	Input buffer size
INIPOS	Integer	Write	Initial cursor position
LDR	Integer	Write	Number of array elements used in the output data record
DATREC(MLDR)	Character*80	Write	Output data record

Constants

Defined Argument	Constant	Description
MODE	GREQU	Request input mode
	GSAMPL	Sample input mode
	GEVENT	Event input mode
ESW	GNECHO	Echo disabled
	GECHO	Echo enabled

INQUIRE STRING DEVICE STATE 3 (FORTRAN-77 Subset)

Description

The INQUIRE STRING DEVICE STATE 3 function returns the current state of the given string-class logical input device.

The data record returned by this function is the input data record associated with the PET returned in the argument *PET*. For FORTRAN, there are no data record components associated with the string-class PETs. All information is returned in the other function arguments. The number of array elements used in the data record is 0.

See Also

INITIALIZE STRING 3

SET STRING MODE

Example 9-3 for a program example using the INQUIRE STRING DEVICE STATE function

INQUIRE STROKE DEVICE STATE

INQUIRE STROKE DEVICE STATE

Operating States

WSOP, WSAC, SGOP

Syntax

GQSKS (WKID, DEVNUM, TYPE, N, MLDR, ERRIND, MODE, ESW, ITNR, NP, PXA, PYA, PET, EAREA, BUFLLEN, LDR, DATREC)

Argument	Data Type	Access	Description
WKID	Integer	Read	Workstation identifier
DEVNUM	Integer	Read	Stroke device number
TYPE	Integer (constant)	Read	Type of values requested
N	Integer	Read	Maximum number of points
MLDR	Integer	Read	Number of elements in output data record
ERRIND	Integer	Write	Error indicator
MODE	Integer (constant)	Write	Operating mode
ESW	Integer (constant)	Write	Echo flag
ITNR	Integer	Write	Initial normalization transformation number
NP	Integer	Write	Initial number of points
PXA(N), PYA(N)	Real	Write	Initial stroke position in WC points
PET	Integer	Write	Prompt and echo type
EAREA(4)	Array of reals	Write	Device coordinate echo area in the order XMIN, XMAX, YMIN, YMAX
BUFLLEN	Integer	Write	Input buffer size
LDR	Integer	Write	Number of array elements used in the output data record
DATREC(MLDR)	Character*80	Write	Output data record

Constants

Defined Argument	Constant	Description
TYPE	GSET	Use exact state list values.
	GREALI	Use values approximated by the graphics handler.
MODE	GREQU	Request input mode.
	GSAMPL	Sample input mode.
	GEVENT	Event input mode.
ESW	GNECHO	Echo disabled.
	GECHO	Echo enabled.

Description

The INQUIRE STROKE DEVICE STATE function returns the current state of the given stroke-class logical input device.

The data record returned by this function is the input data record associated with the PET returned in the argument *PET*. Use the UNPACK DATA RECORD function to access this information.

The data record components for each of the possible PETs are listed in the introduction to Chapter 9. Note that this information refers to arguments in the PACK DATA RECORD function. Because the listed arguments have the same names and meanings in both the PACK DATA RECORD and UNPACK DATA RECORD functions, you can associate the data record components easily with either function.

See Also

INITIALIZE STROKE

SET STROKE MODE

Example 9–3 for a program example using an INQUIRE . . . DEVICE STATE function

INQUIRE STROKE DEVICE STATE 3

INQUIRE STROKE DEVICE STATE 3

Operating States

WSOP, WSAC, SGOP

Syntax

GQSKS3 (WKID, DEVNUM, TYPE, N, MLDR, ERRIND, MODE, ESW, ITNR, IVWIX, NP, ISX, ISY, ISZ, PET, EVOL, BUFLLEN, LDR, DATREC)

Argument	Data Type	Access	Description
WKID	Integer	Read	Workstation identifier
DEVNUM	Integer	Read	Stroke device number
TYPE	Integer (constant)	Read	Type of values requested
N	Integer	Read	Dimension of arrays for stroke points
MLDR	Integer	Read	Number of elements in output data record
ERRIND	Integer	Write	Error indicator
MODE	Integer (constant)	Write	Operating mode
ESW	Integer (constant)	Write	Echo flag
ITNR	Integer	Write	Initial normalization transformation number
IVWIX	Integer	Write	Initial view index
NP	Integer	Write	Initial number of points
ISX(N), ISY(N), ISZ(N)	Real	Write	Initial stroke position in WC points
PET	Integer	Write	Prompt and echo type
EVOL(6)	Array of reals	Write	Device coordinate echo volume in the order XMIN, XMAX, YMIN, YMAX, ZMIN, ZMAX
BUFLLEN	Integer	Write	Input buffer size
LDR	Integer	Write	Number of array elements used in the output data record
DATREC(MLDR)	Character*80	Write	Output data record

Constants

Defined Argument	Constant	Description
TYPE	GSET	Use exact state list values.
	GREALI	Use values approximated by the graphics handler.
MODE	GREQU	Request input mode.
	GSAMPL	Sample input mode.
	GEVENT	Event input mode.
ESW	GNECHO	Echo disabled.
	GECHO	Echo enabled.

Description

The INQUIRE STROKE DEVICE STATE 3 function returns the current state of the given stroke-class logical input device.

The data record returned by this function is the input data record associated with the PET returned in the argument *PET*. Use the UNPACK DATA RECORD function to access this information.

The data record components for each of the possible PETs are listed in the introduction to Chapter 9. Note that this information refers to arguments in the PACK DATA RECORD function. Because the listed arguments have the same names and meanings in both the PACK DATA RECORD and UNPACK DATA RECORD functions, you can associate the data record components easily with either function.

See Also

INITIALIZE STROKE 3
 SET STROKE MODE

Example 9–3 for a program example using an INQUIRE . . . DEVICE STATE function

INQUIRE TEXT ALIGNMENT

INQUIRE TEXT ALIGNMENT

Operating States

GKOP, WSOP, WSAC, SGOP

Syntax

GQTXAL (ERRIND, TXALH, TXALV)

Argument	Data Type	Access	Description
ERRIND	Integer	Write	Error indicator
TXALH	Integer (constant)	Write	Horizontal text alignment
TXALV	Integer (constant)	Write	Vertical text alignment

Constants

Defined Argument	Constant	Description
TXALH	GAHNOR	Normal
	GALEFT	Left
	GACENT	Center
	GARITE	Right
TXALV	GAVNOR	Normal
	GATOP	Top
	GACAP	Cap
	GAHALF	Half
	GABASE	Base
	GABOTT	Bottom

Description

The INQUIRE TEXT ALIGNMENT function returns the value for the *current text alignment* entry in the GKS state list.

See Also

SET TEXT ALIGNMENT

INQUIRE TEXT COLOUR INDEX**Operating States**

GKOP, WSOP, WSAC, SGOP

Syntax

GQTXCI (ERRIND, COLI)

Argument	Data Type	Access	Description
ERRIND	Integer	Write	Error indicator
COLI	Integer	Write	Text color index

Description

The INQUIRE TEXT COLOUR INDEX function returns the value for the *current text color index* entry in the GKS state list.

See Also

SET TEXT COLOUR INDEX

INQUIRE TEXT EXTENT

INQUIRE TEXT EXTENT

Operating States

WSOP, WSAC, SGOP

Syntax

GQTX (WKID, PX, PY, STR, ERRIND, CPX, CPY, TXEXPX, TXEXPY)

Argument	Data Type	Access	Description
WKID	Integer	Read	Workstation identifier
PX, PY	Real	Read	Position of text expressed as a WC point
STR	Character*(*)	Read	Character string
ERRIND	Integer	Write	Error indicator
CPX, CPY	Real	Write	Concatenation point in WC units
TXEXPX(4), TXEXPY(4)	Array of reals	Write	Corners of the text extent parallelogram

Description

The INQUIRE TEXT EXTENT function returns the concatenation point and text extent parallelogram for the specified text string.

The text extent is calculated using the currently selected text attributes. The text extent is returned as the four corner points of the enclosing parallelogram. The four points are in counterclockwise order. The concatenation point can be used as the text origin of a subsequent text origin. However, text string concatenation is not a meaningful combination of text path and text alignment.

See Also

SET CHARACTER HEIGHT
SET CHARACTER UP VECTOR
SET TEXT ALIGNMENT
SET TEXT FONT AND PRECISION
TEXT

INQUIRE TEXT EXTENT (FORTRAN-77 Subset)

Operating States

WSOP, WSAC, SGOP

Syntax

GQTXXS (WKID, PX, PY, LSTR, STR, ERRIND, CPX, CPY, TXEXPX, TXEXPY)

Argument	Data Type	Access	Description
WKID	Integer	Read	Workstation identifier
PX, PY	Real	Read	Position of text expressed as a WC point
LSTR	Integer	Read	Length of character string
STR	Character*80	Read	Character string
ERRIND	Integer	Write	Error indicator
CPX, CPY	Real	Write	Concatenation point in WC units
TXEXPX(4), TXEXPY(4)	Array of reals	Write	Corners of the text extent parallelogram

Description

The INQUIRE TEXT EXTENT function returns the concatenation point and text extent parallelogram for the specified text string.

The text extent is calculated using the currently selected text attributes. The text extent is returned as the four corner points of the enclosing parallelogram. The four points are in counterclockwise order. The concatenation point can be used as the text origin of a subsequent text origin. However, text string concatenation is not a meaningful combination of text path and text alignment.

See Also

SET CHARACTER HEIGHT
 SET CHARACTER UP VECTOR
 SET TEXT ALIGNMENT
 SET TEXT FONT AND PRECISION
 TEXT (FORTRAN-77 Subset)

INQUIRE TEXT EXTENT 3

INQUIRE TEXT EXTENT 3

Operating States

WSOP, WSAC, SGOP

Syntax

GQTX3 (WKID, PX, PY, PZ, TDX, TDY, TDZ, STR, ERRIND, CPX, CPY, CPZ, TXX, TXY, TXZ)

Argument	Data Type	Access	Description
WKID	Integer	Read	Workstation identifier
PX, PY, PZ	Real	Read	Position of text expressed as a WC point
TDX(2), TDY(2), TDZ(2)	Array of reals	Read	Direction vectors for text plane in WC points
STR	Character*(*)	Read	Character string
ERRIND	Integer	Write	Error indicator
CPX, CPY, CPZ	Real	Write	Concatenation point in WC units
TXX(4), TXY(4), TXZ(4)	Array of reals	Write	Four corners of the 3D text extent parallelogram

Description

The INQUIRE TEXT EXTENT 3 function returns the concatenation point and text extent parallelogram for the specified text string.

The text extent is calculated using the currently selected text attributes. The text direction vectors are specified as input arguments to the TEXT 3 function. These vectors, together with the text position, define the text plane. The text extent is returned as the four corner points of the enclosing parallelogram. The four points are in counterclockwise order. The concatenation point can be used as the text origin of a subsequent text origin. However, text string concatenation is not a meaningful combination of text path and text alignment.

See Also

SET CHARACTER HEIGHT
SET CHARACTER UP VECTOR
SET TEXT ALIGNMENT
SET TEXT FONT AND PRECISION
TEXT 3

INQUIRE TEXT EXTENT 3 (FORTRAN-77 Subset)

Operating States

WSOP, WSAC, SGOP

Syntax

GQTX3S (WKID, PX, PY, PZ, TDX, TDY, TDZ, NCHS, STR, ERRIND, CPX, CPY, CPZ, TXX, TXY, TXZ)

Argument	Data Type	Access	Description
WKID	Integer	Read	Workstation identifier
PX, PY, PZ	Real	Read	Position of text expressed as a WC point
TDX(2), TDY(2), TDZ(2)	Array of reals	Read	Direction vectors for text plane
NCHS	Integer	Read	Number of characters in the text string
STR	Character*80	Read	Character string
ERRIND	Integer	Write	Error indicator
CPX, CPY, CPZ	Real	Write	Concatenation point in WC units
TXX(4), TXY(4), TXZ(4)	Array of reals	Write	Four corners of the 3D text extent parallelogram

Description

The INQUIRE TEXT EXTENT 3 function returns the concatenation point and text extent parallelogram for the specified text string.

The text extent is calculated using the currently selected text attributes. The text direction vectors are specified as input arguments to the TEXT 3 function. These vectors, together with the text position, define the text plane. The text extent is returned as the four corner points of the enclosing parallelogram. The four points are in counterclockwise order. The concatenation point can be used as the text origin of a subsequent text origin. However, text string concatenation is not a meaningful combination of text path and text alignment.

See Also

SET CHARACTER HEIGHT
 SET CHARACTER UP VECTOR
 SET TEXT ALIGNMENT
 SET TEXT FONT AND PRECISION
 TEXT 3 (FORTRAN-77 Subset)

INQUIRE TEXT FACILITIES

INQUIRE TEXT FACILITIES

Operating States

GKOP, WSOP, WSAC, SGOP

Syntax

GQTXF (WTYPE, N, ERRIND, NFPP, FONT, PREC, NCHH, MINCHH, MAXCHH, NCHX, MINCHX, MAXCHX, NPTXI)

Argument	Data Type	Access	Description
WTYPE	Integer	Read	Workstation type
N	Integer	Read	Requested list element from the list of font-precision pairs
ERRIND	Integer	Write	Error indicator
NFPP	Integer	Write	Number of font-precision pairs
FONT	Integer	Write	Text font from the <i>Nth</i> element of the list of font-precision pairs
PREC	Integer (constant)	Write	Text font from the <i>Nth</i> element of the list of font-precision pairs
NCHH	Integer	Write	Number of available character heights
MINCHH, MAXCHH	Real	Write	Minimum and maximum character heights in device coordinate units
NCHX	Integer	Write	Number of character expansion factors
MINCHX, MAXCHX	Real	Write	Minimum and maximum character expansion factors
NPTXI	Integer	Write	Number of predefined text indexes

Constants

Defined Argument	Constant	Description
PREC	GSTRP	String precision. DEC GKS evaluates character height and width attributes only.
	GCHARP	Character precision. DEC GKS evaluates each character for compliance with all other specified text attributes.

INQUIRE TEXT FACILITIES

Defined Argument	Constant	Description
	GSTRKP	Stroke precision. DEC GKS looks for exact compliance with all specified text attributes.

Description

The INQUIRE TEXT FACILITIES function returns the number of text font and precision pairs, the requested list element in the list of fonts and list of precisions, the number of available character heights, the number of available character expansion factors, the minimum and maximum character expansion factors, and the number of text indexes available for the specified workstation type.

If the number of character heights or character expansion factors is 0, then the workstation supports a continuous range of character heights or character expansion factors.

INQUIRE TEXT FONT AND PRECISION

INQUIRE TEXT FONT AND PRECISION

Operating States

GKOP, WSOP, WSAC, SGOP

Syntax

GQTXFP (ERRIND, FONT, PREC)

Argument	Data Type	Access	Description
ERRIND	Integer	Write	Error indicator
FONT	Integer	Write	Font number
PREC	Integer (constant)	Write	Text precision

Constants

Defined Argument	Constant	Description
PREC	GSTRP	String precision. DEC GKS evaluates character height and width attributes only.
	GCHARP	Character precision. DEC GKS evaluates each character for compliance with all other specified text attributes.
	GSTRKP	Stroke precision. DEC GKS looks for exact compliance with all specified text attributes.

Description

The INQUIRE TEXT FONT AND PRECISION function returns the value for the *current text font and precision* entry in the GKS state list.

This value is used to display subsequent TEXT output primitives, created when the current text font and precision ASF entry in the GKS state list is INDIVIDUAL. This value does not affect the display of subsequent TEXT output primitives, created when the current text font and precision ASF entry in the GKS state list is BUNDLED.

Text font and precision is a single text aspect; a particular text font can be available at some, but not necessarily all, precisions. The text precision value determines the fidelity with which the other text aspects are used. The values of text precision, in order of increasing fidelity, are string, character, and stroke.

INQUIRE TEXT FONT AND PRECISION

See Also

SET TEXT FONT AND PRECISION

INQUIRE TEXT INDEX

INQUIRE TEXT INDEX

Operating States

GKOP, WSOP, WSAC, SGOP

Syntax

GQTXI (ERRIND, TXI)

Argument	Data Type	Access	Description
ERRIND	Integer	Write	Error indicator
TXI	Integer	Write	Text index

Description

The INQUIRE TEXT INDEX function returns the value for the *current text index* entry in the GKS state list.

The text bundle table contains entries for text font and precision, character expansion factor, character spacing, and text color index attribute values. When calling TEXT, DEC GKS uses the bundle table only if the corresponding ASF has been set to BUNDLED.

See Also

SET TEXT INDEX

INQUIRE TEXT PATH
Operating States

GKOP, WSOP, WSAC, SGOP

Syntax

GQTXP (ERRIND, TXP)

Argument	Data Type	Access	Description
ERRIND	Integer	Write	Error indicator
TXP	Integer (constant)	Write	Text path

Constants

Defined Argument	Constant	Description
TXP	GRIGHT	Text string reads from left to right.
	GLEFT	Text string reads from right to left.
	GUP	Text string reads from bottom to top.
	GDOWN	Text string reads from top to bottom.

Description

The INQUIRE TEXT PATH function returns the value for the geometric attribute *current text path* entry in the GKS state list.

See Also

SET TEXT PATH

INQUIRE TEXT REPRESENTATION

INQUIRE TEXT REPRESENTATION

Operating States

WSOP, WSAC, SGOP

Syntax

GQTXR (WKID, TXI, TYPE, ERRIND, FONT, PREC, CHXP, CHSP, COLI)

Argument	Data Type	Access	Description
WKID	Integer	Read	Workstation identifier
TXI	Integer	Read	Text index on workstation
TYPE	Integer (constant)	Read	Type of value to be returned
ERRIND	Integer	Write	Error indicator
FONT	Integer	Write	Hardware or software font number
PREC	Integer (constant)	Write	Text precision
CHXP	Real	Write	Character expansion factor
CHSP	Real	Write	Character spacing
COLI	Integer	Write	Text color index value

Constants

Defined Argument	Constant	Description
TYPE	GSET	Use exact state list values.
	GREALI	Use values approximated by the graphics handler.
PREC	GSTRP	String precision. DEC GKS evaluates character height and width attributes only.
	GCHARP	Character precision. DEC GKS evaluates each character for compliance with all other specified text attributes.
	GSTRKP	Stroke precision. DEC GKS looks for exact compliance with all specified text attributes.

INQUIRE TEXT REPRESENTATION

Description

The INQUIRE TEXT REPRESENTATION function returns the values currently associated with the specified text index value. Although the text font and precision values are set individually, the function returns them as a pair of values.

If the specified text index is not in the text bundle table on the specified workstation, and the specified type of returned values is REALIZED, the representation for text index 1 is used.

See Also

SET TEXT REPRESENTATION

INQUIRE VALUATOR DEVICE STATE

INQUIRE VALUATOR DEVICE STATE

Operating States

WSOP, WSAC, SGOP

Syntax

GQVLS (WKID, DEVNUM, MLDR, ERRIND, MODE, ESW, IVAL, PET, EAREA, LOVAL, HIVAL, LDR, DATREC)

Argument	Data Type	Access	Description
WKID	Integer	Read	Workstation identifier
DEVNUM	Integer	Read	Valuator device number
MLDR	Integer	Read	Number of elements in output data record
ERRIND	Integer	Write	Error indicator
MODE	Integer (constant)	Write	Operating mode
ESW	Integer (constant)	Write	Echo flag
IVAL	Integer	Write	Initial valuator value
PET	Integer	Write	Prompt and echo type
EAREA(4)	Array of reals	Write	Device coordinate echo area in the order XMIN, XMAX, YMIN, YMAX
LOVAL, HIVAL	Real	Write	Minimum and maximum valuator values
LDR	Integer	Write	Number of array elements used in the output data record
DATREC(MLDR)	Character*80	Write	Output data record

Constants

Defined Argument	Constant	Description
MODE	GREQU	Request input mode
	GSAMPL	Sample input mode
	GEVENT	Event input mode
ESW	GNECHO	Echo disabled
	GECHO	Echo enabled

INQUIRE VALUATOR DEVICE STATE

Description

The INQUIRE VALUATOR DEVICE STATE function returns the current state of the given valuator-class logical input device.

The data record returned by this function is the input data record associated with the PET returned in the argument *PET*. For FORTRAN, there are no data record components associated with the valuator-class PETs. All information is returned in the other function arguments. The number of array elements used in the data record is 0.

See Also

INITIALIZE VALUATOR

SET VALUATOR MODE

Example 9–4 for a program example using the INQUIRE VALUATOR DEVICE STATE function

INQUIRE VALUATOR DEVICE STATE 3

INQUIRE VALUATOR DEVICE STATE 3

Operating States

WSOP, WSAC, SGOP

Syntax

GQVLS3 (WKID, DEVNUM, MLDR, ERRIND, MODE, ESW, IVAL, PET, EVOL, LOVAL, HIVAL, LDR, DATREC)

Argument	Data Type	Access	Description
WKID	Integer	Read	Workstation identifier
DEVNUM	Integer	Read	Valuator device number
MLDR	Integer	Read	Number of elements in output data record
ERRIND	Integer	Write	Error indicator
MODE	Integer (constant)	Write	Operating mode
ESW	Integer (constant)	Write	Echo flag
IVAL	Real	Write	Initial valuator value
PET	Integer	Write	Prompt and echo type
EVOL(6)	Array of reals	Write	Device coordinate echo volume in the order XMIN, XMAX, YMIN, YMAX, ZMIN, ZMAX
LOVAL, HIVAL	Real	Write	Minimum and maximum valuator values
LDR	Integer	Write	Number of array elements used in the output data record
DATREC(MLDR)	Character*80	Write	Output data record

Constants

Defined Argument	Constant	Description
MODE	GREQU	Request input mode
	GSAMPL	Sample input mode
	GEVENT	Event input mode
ESW	GNECHO	Echo disabled
	GECHO	Echo enabled

Description

The INQUIRE VALUATOR DEVICE STATE 3 function returns the current state of the given valuator-class logical input device.

The data record returned by this function is the input data record associated with the PET returned in the argument *PET*. For FORTRAN, there are no data record components associated with the valuator-class PETs. All information is returned in the other function arguments. The number of array elements used in the data record is 0.

See Also

INITIALIZE VALUATOR 3

SET VALUATOR MODE

Example 9–4 for a program example using the INQUIRE VALUATOR DEVICE STATE function

INQUIRE VIEW FACILITIES

INQUIRE VIEW FACILITIES

Operating States

GKOP, WSOP, WSAC, SGOP

Syntax

GQVWF (WTYPE, ERRIND, NPVWI)

Argument	Data Type	Access	Description
WTYPE	Integer	Read	Workstation type
ERRIND	Integer	Write	Error indicator
NPVWI	Integer	Write	Number of predefined view indexes

Description

The INQUIRE VIEW FACILITIES function returns the number of predefined view indexes for the specified workstation type.

INQUIRE VIEW REPRESENTATION 3

Operating States

WSOP, WSAC, SGOP

Syntax

CQVWR3 (WKID, VIEWI, CURQ, ERRIND, VWUPD, VWORMT, VWMPMT, VWCPLM, XYCLIP, BKCLIP, FRCLIP)

Argument	Data Type	Access	Description
WKID	Integer	Read	Workstation identifier
VIEWI	Integer	Read	View index
CURQ	Integer (constant)	Read	Current or requested value flag
ERRIND	Integer	Write	Error indicator
VWUPD	Integer (constant)	Write	Viewing transformation update state
VWORMT(4,4)	Array of reals	Write	View orientation matrix
VWMPMT(4,4)	Array of reals	Write	View mapping matrix
VWCPLM(6)	Array of reals	Write	View clipping limits in the order XMIN, XMAX, YMIN, YMAX, ZMIN, ZMAX
XYCLIP, BKCLIP, FRCLIP	Integer (constant)	Write	XY, back and front clipping indicators

Constants

Defined Argument	Constant	Description
CURQ	GCURVL	Return current values.
	GRQSVL	Return requested values.
VWUPD	GNPEND	Action not pending.
	GPEND	Action pending.
XYCLIP, BKCLIP, FRCLIP	GNCLIP	Clipping disabled.
	GCLIP	Clipping enabled.

INQUIRE VIEW REPRESENTATION 3

Description

The INQUIRE VIEW REPRESENTATION 3 function queries the workstation state list and returns the view representation for the given view index. If a viewing transformation change was requested but not yet provided at the time of the call to this function, the viewing transformation update state is PENDING.

See Also

SET VIEW REPRESENTATION 3

INQUIRE WORKSTATION CATEGORY

Operating States

GKOP, WSOP, WSAC, SGOP

Syntax

GQWKCA (WTYPE, ERRIND, WKCAT)

Argument	Data Type	Access	Description
WTYPE	Integer	Read	Workstation type
ERRIND	Integer	Write	Error indicator
WKCAT	Integer (constant)	Write	Workstation category

Constants

Defined Argument	Constant	Description
WKCAT	GOUTPT	Output category
	GINPUT	Input category
	GOUTIN	Output/input category
	GWISS	Workstation independent segment storage
	GMO	Metafile output category
	GMI	Metafile input category

Description

The INQUIRE WORKSTATION CATEGORY function returns the workstation category for the specified workstation type.

INQUIRE WORKSTATION CLASSIFICATION

INQUIRE WORKSTATION CLASSIFICATION

Operating States

GKOP, WSOP, WSAC, SGOP

Syntax

GQWKCL (WTYPE, ERRIND, VRTYPE)

Argument	Data Type	Access	Description
WTYPE	Integer	Read	Workstation type
ERRIND	Integer	Write	Error indicator
VRTYPE	Integer (constant)	Write	Workstation classification

Constants

Defined Argument	Constant	Description
VRTYPE	GVECTR	Vector class
	GRASTR	Raster class
	GOTHWK	Other class

Description

The INQUIRE WORKSTATION CLASSIFICATION function returns the workstation classification for the specified workstation type.

INQUIRE WORKSTATION CONNECTION AND TYPE

Operating States

WSOP, WSAC, SGOP

Syntax

GQWKC (WKID, ERRIND, CONID, WTYPE)

Argument	Data Type	Access	Description
WKID	Integer	Read	Workstation identifier
ERRIND	Integer	Write	Error indicator
CONID	Integer	Write	Connection identifier
WTYPE	Integer	Write	Workstation type

Description

The INQUIRE WORKSTATION CONNECTION AND TYPE function returns the logical name or environment variable associated with the physical device connection from the host to the workstation, and the workstation type.

See Also

OPEN WORKSTATION

Example 4–2 for a program example using the INQUIRE WORKSTATION CONNECTION AND TYPE function

INQUIRE WORKSTATION DEFERRAL AND UPDATE STATES

INQUIRE WORKSTATION DEFERRAL AND UPDATE STATES

Operating States

WSOP, WSAC, SGOP

Syntax

GQWKDU (WKID, ERRIND, DEFMOD, REGMOD, DEMPTY, NFRAME)

Argument	Data Type	Access	Description
WKID	Integer	Read	Workstation identifier
ERRIND	Integer	Write	Error indicator
DEFMOD	Integer (constant)	Write	Deferral mode
REGMOD	Integer (constant)	Write	Implicit regeneration mode
DEMPY	Integer (constant)	Write	Display surface empty flag
NFRAME	Integer (constant)	Write	New frame necessary at update flag

Constants

Defined Argument	Constant	Description
DEFMOD	GASAP	Generate images as soon as possible.
	GBNIG	Generate images before input is requested globally.
	GBNIL	Generate images before input is requested locally.
	GASTI	Generate images sometime. Exact time is not guaranteed.
REGMOD	GSUPPD	Image regeneration is suppressed.
	GALLOW	Image regeneration is allowed.
DEMPY	GNEMPT	Surface is not empty.
	GEMPTY	Surface is empty.
NFRAME	GNO	Do not clear surface at next update.
	GYES	Clear the surface at next update.

INQUIRE WORKSTATION DEFERRAL AND UPDATE STATES

Description

The INQUIRE WORKSTATION DEFERRAL AND UPDATE STATES function returns the current deferral state mode, implicit regeneration mode, and workstation surface status. It also indicates whether a new frame is necessary to update the screen.

See Also

SET DEFERRAL STATE

INQUIRE WORKSTATION MAXIMUM NUMBERS

INQUIRE WORKSTATION MAXIMUM NUMBERS

Operating States

GKOP, WSOP, WSAC, SGOP

Syntax

GQWKM (ERRIND, MAXOPWK, MAXACWK, MXWKAS)

Argument	Data Type	Access	Description
ERRIND	Integer	Write	Error status
MAXOPWK	Integer	Write	Maximum number of simultaneously open workstations
MAXACWK	Integer	Write	Maximum number of simultaneously active workstations
MXWKAS	Integer	Write	Maximum number of workstations associated with segment

Description

The INQUIRE WORKSTATION MAXIMUM NUMBERS function returns the maximum number of open workstations, active workstations, and workstations that can be associated with a segment.

INQUIRE WORKSTATION STATE

Operating States

WSOP, WSAC, SGOP

Syntax

GQWKS (WKID, ERRIND, STATE)

Argument	Data Type	Access	Description
WKID	Integer	Read	Workstation identifier
ERRIND	Integer	Write	Error indicator
STATE	Integer (constant)	Write	Workstation state

Constants

Defined Argument	Constant	Description
STATE	GINACT	Workstation is not active.
	GACTIV	Workstation is active.

Description

The INQUIRE WORKSTATION STATE function identifies the workstation state value as either GINACT or GACTIV.

See Also

ACTIVATE WORKSTATION

INQUIRE WORKSTATION TRANSFORMATION

INQUIRE WORKSTATION TRANSFORMATION

Operating States

WSOP, WSAC, SGOP

Syntax

GQWKT (WKID, ERRIND, TUS, RWINDO, CWINDO, RVIEWP, CVIEWP)

Argument	Data Type	Access	Description
WKID	Integer	Read	Workstation identifier
ERRIND	Integer	Write	Error indicator
TUS	Integer (constant)	Write	Transformation update state
RWINDO(4)	Array of reals	Write	Requested workstation window in NDC points in the order XMIN, XMAX, YMIN, YMAX
CWINDO(4)	Array of reals	Write	Current workstation window in NDC points in the order XMIN, XMAX, YMIN, YMAX
RVIEWP(4)	Array of reals	Write	Requested workstation viewport in device coordinate points in the order XMIN, XMAX, YMIN, YMAX
CVIEWP(4)	Array of reals	Write	Current workstation viewport in device coordinate points in the order XMIN, XMAX, YMIN, YMAX

Constants

Defined Argument	Constant	Description
TUS	GNPEND	Action not pending
	GPEND	Action pending

Description

The INQUIRE WORKSTATION TRANSFORMATION function returns the requested and current workstation windows and workstation viewports and a flag indicating whether a workstation transformation is pending.

If a workstation transformation change was requested but not yet provided at the time of the call to this function, the workstation transformation update state is PENDING.

INQUIRE WORKSTATION TRANSFORMATION

See Also

SET WORKSTATION VIEWPORT
SET WORKSTATION WINDOW

INQUIRE WORKSTATION TRANSFORMATION 3

INQUIRE WORKSTATION TRANSFORMATION 3

Operating States

WSOP, WSAC, SGOP

Syntax

GQWKT3 (WKID, ERRIND, VWUPD, RWKWN, CWKWN, RWKVP, CWKVP)

Argument	Data Type	Access	Description
WKID	Integer	Read	Workstation identifier
ERRIND	Integer	Write	Error indicator
VWUPD	Integer (constant)	Write	Transformation update state
RWKWN(6)	Array of reals	Write	Requested workstation window in NDC points in the order XMIN, XMAX, YMIN, YMAX, ZMIN, ZMAX
CWKWN(6)	Array of reals	Write	Current workstation window in NDC points in the order XMIN, XMAX, YMIN, YMAX, ZMIN, ZMAX
RWKVP(6)	Array of reals	Write	Requested workstation viewport in device coordinate points in the order XMIN, XMAX, YMIN, YMAX, ZMIN, ZMAX
CWKVP(6)	Array of reals	Write	Current workstation viewport in device coordinate points in the order XMIN, XMAX, YMIN, YMAX, ZMIN, ZMAX

Constants

Defined Argument	Constant	Description
VWUPD	GNPEND	Action not pending
	GPEND	Action pending

Description

The INQUIRE WORKSTATION TRANSFORMATION 3 function returns the requested and current workstation windows and workstation viewports, and a flag indicating whether a workstation transformation is pending.

If a workstation transformation change was requested but not yet provided at the time of the call to this function, the workstation transformation update state is PENDING.

See Also

SET WORKSTATION VIEWPORT 3
SET WORKSTATION WINDOW 3

Error-Handling Functions

Insert tabbed divider here. Then discard this sheet.



Error-Handling Functions

The DEC GKS error-handling functions provide a method for you to control the generation of messages to the user, and a method of exit when a DEC GKS function call generates an error.

DEC GKS recognizes a number of error conditions. These error conditions are detected within DEC GKS functions, within procedures called by DEC GKS functions (such as calls to the graphics handler procedures), or within other areas of the application program.

For errors occurring in areas of the application program other than in DEC GKS function calls, either the application program regains control or program execution terminates abnormally. If the application program regains control, it can attempt to properly close DEC GKS or, failing that, attempt an emergency closure by calling the function EMERGENCY CLOSE GKS. If the program terminates abnormally, the results are unpredictable. In the worst case, you lose all graphic information produced before the error.

For errors detected within procedures called by DEC GKS, if the procedure does not generate a fatal error, then the DEC GKS error handlers may be able to process the error and you should be able to save graphic data. Otherwise, the application program regains control, or the application program is forced to call EMERGENCY CLOSE GKS, or in the worst case, you lose all graphic data produced before the error.

For errors detected within DEC GKS functions, DEC GKS performs the following tasks:

1. Sets the DEC GKS error state to ON to prohibit modification of DEC GKS variables
2. Calls ERROR HANDLING and passes the appropriate arguments
3. Performs function-specific error reaction or cleanup
4. Sets the DEC GKS error state to OFF

You can allow DEC GKS to call its own error handler, or you can provide an error handler of your own. An application-supplied, error-handling function can interpret information about the error and store data in a data area for subsequent analysis. Because application-supplied handlers do not have to generate the standard DEC GKS error messages, such a handler can change the format or the text of the messages sent to the user. Also, application-supplied handlers can decide whether to abort a program or to continue despite generated errors, if the errors are not fatal.

Error-Handling Functions

A fatal error occurs within DEC GKS when internal data structures are corrupted, or when accurate and meaningful execution of DEC GKS functions is no longer possible. When a fatal error occurs, DEC GKS executes the current error handler and then terminates execution of the application.

The GKS standard dictates that every error-handling function, whether it be the DEC GKS supplied function or an application-supplied function, accept the following information from DEC GKS upon error generation:

- The GKS error number corresponding to the appropriate error condition (see Appendix A, DEC GKS Error Messages)
- The name of the GKS function that generated the error condition
- The name of the error file specified in the application program in the call to OPEN GKS

To implement an application-supplied, error-handling function, define the ERROR HANDLING function in your application. Use the name *GERHND* and the listed order and type of the three arguments. After you link the application, DEC GKS will use the new error handler when errors occur.

12.1 Function Descriptions

This section describes the DEC GKS error functions in detail.

EMERGENCY CLOSE GKS

Operating States

GKCL, GKOP, WSOP, WSAC, SGOP

Syntax

GECLKS ()

Description

The EMERGENCY CLOSE GKS function attempts to perform a rapid and orderly closure of DEC GKS. Usually, this function is called for error conditions detected outside DEC GKS. If possible, the call to this function closes any open segments, updates all active workstations, deactivates those workstations, closes all open workstations, and then closes DEC GKS.

See Also

Example 12-1 for a program example using the EMERGENCY CLOSE GKS function

ERROR HANDLING

ERROR HANDLING

Operating States

GKCL, GKOP, WSOP, WSAC, SGOP

Syntax

GERHND (ERRNR, FCTID, ERRFIL)

Argument	Data Type	Access	Description
ERRNR	Integer	Read	Error message number
FCTID	Integer	Read	Function identification
ERRFIL	Integer	Read	Error logging file name

Description

The ERROR HANDLING function calls the ERROR LOGGING function and allows continued program execution. By default, DEC GKS calls this function when it encounters an error condition.

To implement an application-supplied error handler, call this function in your application. After you link the application, DEC GKS calls the application error handler when an error occurs.

See Also

EMERGENCY CLOSE GKS

ERROR LOGGING

OPEN GKS

Example 12–1 for a program example using the ERROR HANDLING function

ERROR LOGGING

Operating States

GKCL, GKOP, WSOP, WSAC, SGOP

Syntax

GERLOG (ERRNR, FCTID, ERRFIL)

Argument	Data Type	Access	Description
ERRNR	Integer	Read	Error message number
FCTID	Integer	Read	Function identification
ERRFIL	Integer	Read	Error logging file name

Description

The ERROR LOGGING function writes the standard DEC GKS error message, which includes the number of the error and the text of the message, to the error file and returns to the procedure or function from which it was called.

The error handler function supplied by DEC GKS (ERROR HANDLING) automatically calls the ERROR LOGGING function. An application-supplied error handler can call this function if necessary.

See Also

ERROR HANDLING

Example 12–1 for a program example using the ERROR LOGGING function

12.2 Program Example

12.2 Program Example

Example 12–1 illustrates the use of the error handling functions.

Example 12–1 Creating an Error Handler

```
C This program sets up a new error handler, creates an error,
C then logs the error to an error file. This program writes the
C error message to the file ERROR.DAT, and closes GKS and the workstation.
C
C Key functions:
C
C SET ERROR HANDLER
C ERROR LOGGING
C EMERGENCY CLOSE GKS

      IMPLICIT NONE
      INCLUDE 'gks.f'

      REAL    START_X
      REAL    START_Y
      INTEGER WS_ID

C Open the GKS and workstation environments.

      WS_ID = 1

      OPEN (UNIT=1, FILE='ERROR.DAT', STATUS='NEW')
      CALL GOPKS (1)
      CALL GOPWK (WS_ID, GCONID, GWSDEF)

C Cause an error (the workstation isn't active yet). The message
C will not be displayed on the screen; it will be written to the file
C ERROR.DAT.

      START_X = 0.5
      START_Y = 0.5

      CALL GTX (START_X, START_Y, 'TEST')

C Release the GKS and workstation environments.

      CALL GCLWK (WS_ID)
      CALL GCLKS ()
      END

C User-defined error handler.

      SUBROUTINE GERHND (ERRNR,FCTID,ERRFIL)

      IMPLICIT NONE

      INTEGER ERRNR
      INTEGER FCTID
      CHARACTER*80 ERRFIL

      INTEGER WS_NOT_ACTIVE

C Tell the user the error handler was called .

      WRITE(*) '** Aborting from a severe error.'
      WRITE(*) ' Please check the file error.dat **'
```

(continued on next page)

Example 12–1 (Cont.) Creating an Error Handler

```
C Write the error message to a log file.  
    CALL GERLOG (ERRNR,FCTID,ERRFIL)  
C If ERROR_5, abort.  
    WS_NOT_ACTIVE = 5  
    IF (ERRNR .EQ. WS_NOT_ACTIVE) THEN  
        CALL GECLKS ()  
        CALL EXIT(1)  
    END IF  
    RETURN  
    END
```


Error Messages

Insert tabbed divider here. Then discard this sheet.



A

DEC GKS Error Messages

This appendix lists each of the DEC GKS error messages and the number associated with each message. The FORTRAN calls do *not* return an error status. Only the inquiry functions return an error status through one of the parameters. The available constants for the function completion states are listed in Appendix B. If you want to specify application actions given a particular error status associated with any noninquiry FORTRAN function call, you must write your own error handler.

Some of the DEC GKS specific error messages substitute program information in the message text. In this appendix, the portion of the text to be substituted is shown as ****.

The following sections describe the DEC GKS error messages by category and in numerical order.

A.1 Operating State Errors

Table A-1 lists the errors that result when you call a function that is not permitted in the current operating state. For a list of the functions that you can or cannot call in a given DEC GKS operating state, see Chapter 4.

Table A-1 Operating State Errors

Error	Error Message
0	Successful completion of routine **** User Action: None.
1	GKS not in proper state: GKS shall be in the state GKCL in routine **** User Action: Call the appropriate DEC GKS control function to change the current state. (You must call CLOSE GKS before the current DEC GKS state changes to GKCL.)
2	GKS not in proper state: GKS shall be in the state GKOP in routine **** User Action: Call the appropriate DEC GKS control function to change the current state. (You must call either the function OPEN GKS or CLOSE WORKSTATION before the DEC GKS state changes to GKOP.)
3	GKS not in proper state: GKS shall be in the state WSAC in routine **** User Action: Call the appropriate DEC GKS control function to change the current state. (You must call either the function ACTIVATE WORKSTATION or CLOSE SEGMENT before the DEC GKS state changes to WSAC.)

(continued on next page)

DEC GKS Error Messages

A.1 Operating State Errors

Table A-1 (Cont.) Operating State Errors

Error	Error Message
4	<p>GKS not in proper state: GKS shall be in the state SGOP in routine ****</p> <p>User Action: Call the appropriate DEC GKS control function to change the current state. (You must call the function CREATE SEGMENT before the DEC GKS state changes to SGOP.)</p>
5	<p>GKS not in proper state: GKS shall be either in the state WSAC or in the state SGOP in routine ****</p> <p>User Action: Call the appropriate DEC GKS control function to change the current state. (You must call the function ACTIVATE WORKSTATION before the DEC GKS state changes to WSAC.)</p>
6	<p>GKS not in proper state: GKS shall be in the state WSOP or in the state WSAC in routine ****</p> <p>User Action: Call the appropriate DEC GKS control function to change the current state. (You must call the function OPEN WORKSTATION before the DEC GKS state changes to WSOP.)</p>
7	<p>GKS not in proper state: GKS shall be in one of the states WSOP, WSAC, or SGOP in routine ****</p> <p>User Action: Call the appropriate DEC GKS control function to change the current state. (You must call the function OPEN WORKSTATION before the DEC GKS state changes to WSOP.)</p>
8	<p>GKS not in proper state: GKS shall be in one of the states GKOP, WSOP, WSAC, or SGOP in routine ****</p> <p>User Action: Call the appropriate DEC GKS control function to change the current state. (You must call the function OPEN GKS before the DEC GKS state changes to GKOP.)</p>

A.2 Workstation Errors

Table A-2 lists the errors that result when you call a DEC GKS function with invalid or undefined arguments pertaining to workstations.

Table A-2 Workstation Errors

Error	Error Message
20	<p>Specified workstation identifier is invalid in routine ****</p> <p>User Action: Make sure you have opened a workstation associated with that identifier, that you are not trying to generate output to an inactive workstation, that the arguments are presented in the right order, and if you are using a variable to specify the workstation identifier, that the variable is declared to be an integer.</p>
21	<p>Specified connection identifier is invalid in routine ****</p>

(continued on next page)

Table A-2 (Cont.) Workstation Errors

Error	Error Message
	User Action: Check that the connection identifier is specified correctly. See the sections on specifying the connection identifier in in the programming specifics online manual, which is available on the kit. If you are using the default value, make sure the environment option has been defined.
22	Specified workstation type is invalid in routine **** User Action: Check to make sure you passed either a DEC GKS constant or the corresponding integer value. The workstation type constants are listed in Appendix B. If you are using the default value, make sure the environment option has been defined.
23	Specified workstation type does not exist in routine **** User Action: The implementation of GKS does not support the workstation type associated with the identifier you passed. Pass an identifier associated with a supported device. If you are using the default workstation type, you should use INQUIRE WORKSTATION CONNECTION AND TYPE to check if DEC GKS supports the currently defined workstation type.
24	Specified workstation is open in routine **** User Action: Either remove the function call to OPEN WORKSTATION, or replace the incorrect workstation type argument.
25	Specified workstation is not open in routine **** User Action: Call OPEN WORKSTATION and pass the appropriate workstation identifier.
26	Specified workstation cannot be opened in routine **** User Action: Make sure you specify valid workstation types, bit masks, or environment options, and make sure the information corresponds to a supported, functional physical device.
27	Workstation Independent Segment Storage is not open in routine **** User Action: You tried to copy, associate, or insert a segment from WISS to another workstation. Make sure you have opened WISS in a call to OPEN WORKSTATION, passing GWSWIS as an argument.
28	Workstation Independent Segment Storage is already open in routine **** User Action: Either remove the function call to OPEN WORKSTATION, or replace the incorrect workstation type argument.
29	Specified workstation is active in routine **** User Action: Either remove the function call to ACTIVATE WORKSTATION, or replace the incorrect workstation identifier argument.
30	Specified workstation is not active in routine ****

(continued on next page)

DEC GKS Error Messages

A.2 Workstation Errors

Table A-2 (Cont.) Workstation Errors

Error	Error Message
	User Action: You tried to generate output on an inactive workstation. Call ACTIVATE WORKSTATION, passing the appropriate workstation.
31	Specified workstation is of category MO in routine **** User Action: You attempted to perform an operation that is not permissible on MO workstations. Either remove the function call, change the state of the MO workstation, or check to see if you passed the correct arguments to OPEN WORKSTATION.
32	Specified workstation is not of category MO in routine **** User Action: Open and activate an MO workstation.
33	Specified workstation is of category MI in routine **** User Action: You attempted to perform an operation that is not permissible on MI workstations. Either remove the function call, change the state of the MI workstation, or check to see if you passed the correct arguments to OPEN WORKSTATION.
34	Specified workstation is not of category MI in routine **** User Action: You tried to interpret a file that was not associated with an MI workstation. Open a workstation of category MI.
35	Specified workstation is of category INPUT in routine **** User Action: You attempted to perform an operation that is not permissible on workstations of category INPUT, such as generating output. Either remove the function call, change the workstation type to the needed category, or check to see if you passed the correct arguments to OPEN WORKSTATION.
36	Specified workstation is Workstation Independent Segment Storage in routine **** User Action: You attempted to perform an operation that is not permissible on workstations of category WISS, such as requesting input. Either remove the function call, check that the call uses the correct workstation identifier, or check to see if you passed the correct arguments to OPEN WORKSTATION.
37	Specified workstation is not of category OUTIN in routine **** User Action: Either remove the function call, open and activate an OUTIN workstation, or check to see if you passed the correct arguments to OPEN WORKSTATION.
38	Specified workstation is neither of category INPUT nor of category OUTIN in routine **** User Action: Either remove the function call, change the workstation type to the needed category, or check to see if you passed the correct arguments to OPEN WORKSTATION.
39	Specified workstation is neither of category OUTPUT nor of category OUTIN in routine ****

(continued on next page)

Table A-2 (Cont.) Workstation Errors

Error	Error Message
	User Action: You attempted to perform an operation that is only permissible on workstations of category OUTPUT or OUTIN, such as generating output. Either remove the function call, open and activate a workstation of the correct category, or check to see if you passed the correct arguments to OPEN WORKSTATION.
40	Specified workstation has no pixel store readback capability in routine **** User Action: You called one of the pixel inquiry functions for a device incapable of returning such information. Either remove the function call, or make sure you passed the correct workstation identifier.
41	Specified workstation type is not able to generate the specified generalized drawing primitive in routine **** User Action: Either remove the function call to GENERALIZED DRAWING PRIMITIVE, or make sure you passed the correct GDP identifier.
42	Maximum number of simultaneously open workstations would be exceeded in routine **** User Action: You must remove the function call to OPEN WORKSTATION. You can use INQUIRE WORKSTATION MAXIMUM NUMBERS to determine the maximum number of open workstations that DEC GKS supports.
43	Maximum number of simultaneously active workstations would be exceeded in routine **** User Action: You must remove the function call to ACTIVATE WORKSTATION. You can use INQUIRE WORKSTATION MAXIMUM NUMBERS to determine the maximum number of active workstations that DEC GKS supports.

A.3 Transformation Function Errors

Table A-3 lists the errors that result when you call a DEC GKS transformation function with invalid or undefined arguments.

Table A-3 Transformation Function Errors

Error	Error Message
50	Transformation number is invalid in routine **** User Action: Make sure that either the arguments are specified in the correct order, the transformation number is not negative, or the transformation number is an integer.
51	Rectangle definition is invalid in routine **** User Action: Either the normalization window or viewport is invalid. Make sure you have not reversed the order of the X and Y argument values, that your coordinate values form a valid rectangle, and that your coordinate values are real numbers.
52	Viewport is not within the Normalized Device Coordinate unit square in routine ****

(continued on next page)

DEC GKS Error Messages

A.3 Transformation Function Errors

Table A-3 (Cont.) Transformation Function Errors

Error	Error Message
	User Action: DEC GKS allows unclipped primitives to exceed the NDC unit square ($[1,0,0] \times [0,1,0] \times [0,0,1]$), but does not allow you to define a normalization viewport whose boundaries exceed this square. Redefine the function normalization viewport.
53	Workstation window is not within the Normalized Device Coordinate unit square in routine ***** User Action: Redefine the function normalization viewport to be within the NDC square ($[1,0,0] \times [0,1,0] \times [0,0,1]$).
54	Workstation viewport is not within the display space in routine ***** User Action: Make sure you have not reversed the order of the X, Y, and Z argument values, your coordinate values form a valid rectangle, and your coordinate values are real numbers. You can use the function INQUIRE DISPLAY SPACE SIZE to determine the maximum X, Y, and Z values of the device coordinate plane.

A.4 Attribute Function Errors

Table A-4 lists the errors that result when you call the DEC GKS attribute functions with invalid or undefined arguments.

Table A-4 Attribute Function Errors

Error	Error Message
60	Polyline index is invalid in routine ***** User Action: Make sure the arguments are specified in the correct order and the index is an integer.
61	A representation for the specified polyline index has not been defined on this workstation in routine ***** User Action: Use SET POLYLINE REPRESENTATION to define a representation for the index, or use another predefined index value.
62	A representation for the specified polyline index has not been predefined on this workstation in routine ***** User Action: Use SET POLYLINE REPRESENTATION to define a representation for the index, or use another predefined index value.
63	Specified line type is equal to zero in routine ***** User Action: Make sure the order and the number of the arguments is correct. If you used an inquiry function to obtain a default line type, check the order of the arguments passed to the inquiry function.
64	Specified line type is not supported on this workstation in routine *****

(continued on next page)

Table A-4 (Cont.) Attribute Function Errors

Error	Error Message
	User Action: Change the line type specification. You can use the function INQUIRE POLYLINE FACILITIES to obtain a list of supported line types for a given workstation.
65	Line width scale factor is less than zero in routine **** User Action: Either change the scale factor, or check the order and the number of the specified arguments.
66	Polymarker index is invalid in routine **** User Action: Make sure the arguments are specified in the correct order and the index is an integer.
67	A representation for the specified polymarker index has not been defined on this workstation in routine **** User Action: Use SET POLYMARKER REPRESENTATION to define a representation for a given index, or use another predefined index value.
68	A representation for the specified polymarker index has not been predefined on this workstation in routine **** User Action: Use SET POLYMARKER REPRESENTATION to define a representation for a given index, or use another predefined index value.
69	Specified marker type is equal to zero in routine **** User Action: Make sure the order of the arguments is correct. If you used an inquiry function to obtain a default marker type, check the order of the arguments passed to the inquiry function.
70	Specified marker type is not supported on this workstation in routine **** User Action: Change the marker type specification. You can use the function INQUIRE POLYMARKER FACILITIES to obtain a list of supported line types for a given workstation.
71	Marker size scale factor is less than zero in routine **** User Action: Either change the scale factor, or check the order and the number of the specified arguments.
72	Text index is invalid in routine **** User Action: Make sure the arguments are specified in the correct order and the index is an integer.
73	A representation for the specified text index has not been defined on this workstation in routine **** User Action: Use SET TEXT REPRESENTATION to define a representation for the index value, or use another predefined index value.

(continued on next page)

DEC GKS Error Messages

A.4 Attribute Function Errors

Table A-4 (Cont.) Attribute Function Errors

Error	Error Message
74	<p>A representation for the specified text index has not been predefined on this workstation in routine *****</p> <p>User Action: Use SET TEXT REPRESENTATION to define a representation for the index value, or use another predefined index value.</p>
75	<p>Text font is equal to zero in routine *****</p> <p>User Action: Either change the font number, or check the order and the number of the specified arguments. If you used an inquiry function to obtain a default value, check the order and the number of the arguments passed to the inquiry function.</p>
76	<p>Requested text font is not supported for the specified precision on this workstation in routine *****</p> <p>User Action: Lower the precision or change the font number.</p>
77	<p>Character expansion factor is less than or equal to zero in routine *****</p> <p>User Action: Either change the expansion factor value or check the order and the number of the arguments. If you used an inquiry function to obtain a default value, check the order and the number of the arguments passed to the inquiry function.</p>
78	<p>Character height is less than or equal to zero in routine *****</p> <p>User Action: Either change the height value, or check the order and the number of the arguments. If you used an inquiry function to obtain a default value, check the order and the number of the arguments passed to the inquiry function.</p>
79	<p>Length of character up vector is zero in routine *****</p> <p>User Action: Either change the character up vector, or check the order and the number of the arguments. If you used an inquiry function to obtain a default value, check the order and the number of the arguments passed to the inquiry function.</p>
80	<p>Fill area index is invalid in routine *****</p> <p>User Action: Make sure the arguments are specified in the correct order and the index is an integer.</p>
81	<p>A representation for the specified fill area index has not been defined on this workstation in routine *****</p> <p>User Action: Use SET FILL AREA REPRESENTATION to define a representation for the given index value, or pass another predefined index value.</p>
82	<p>A representation for the specified fill area index has not been predefined on this workstation in routine *****</p> <p>User Action: Use SET FILL AREA REPRESENTATION to define a representation for the given index value, or pass another predefined index value.</p>

(continued on next page)

Table A-4 (Cont.) Attribute Function Errors

Error	Error Message
83	Specified fill area interior style is not supported on this workstation in routine **** User Action: Change the interior style specification. You can use the function INQUIRE FILL AREA FACILITIES to obtain a list of supported interior styles for a given workstation.
84	Style (pattern or hatch) index is equal to zero in routine **** User Action: Either change the style index, or check the order and the number of the specified arguments. If you used an inquiry function to obtain a style index, check the order and the number of the arguments passed to the inquiry function.
85	Specified pattern index is invalid in routine **** User Action: Make sure the arguments are specified in the correct order and the index is an integer.
86	Specified hatch style is not supported on this workstation in routine **** User Action: Either replace the hatch style index, or check the order and the number of the arguments. The inquiry function INQUIRE FILL AREA FACILITIES returns the list of available hatch style indexes.
87	Pattern size value is not positive in routine **** User Action: Either alter the size of the pattern, or check the order and the number of the arguments. If you used an inquiry function to obtain the size of the pattern, check the order and the number of the arguments passed to the inquiry function.
88	A representation for the specified pattern index has not been defined on this workstation in routine **** User Action: Use SET PATTERN REPRESENTATION to define a representation for the pattern index, or pass another predefined index to the function.
89	A representation for the specified pattern index has not been predefined on this workstation in routine **** User Action: Use SET PATTERN REPRESENTATION to define a representation for the pattern index, or pass another predefined index to the function.
90	Interior style PATTERN is not supported on this workstation in routine **** User Action: Specify another interior style to SET FILL AREA INTERIOR STYLE.
91	Dimensions of color array are invalid in routine **** User Action: One or more of the arguments passed to CELL ARRAY are invalid. Make sure the color array is a two-dimensional array, that you have not specified more rows and columns in the cell array that exist from the offset point to the end of the array, and that the cell array contains integers representing colors supported on that workstation.

(continued on next page)

DEC GKS Error Messages

A.4 Attribute Function Errors

Table A-4 (Cont.) Attribute Function Errors

Error	Error Message
92	Color index is less than zero in routine ***** User Action: Either remove the index, or check the order and the number of the arguments. If you used an inquiry function to obtain the color index value, check the order and the number of the arguments passed to the inquiry function.
93	Color index is invalid in routine ***** User Action: Make sure the arguments are specified in the correct order and that the index is an integer.
94	A representation for the specified color index has not been defined on this workstation in routine ***** User Action: Use SET COLOUR REPRESENTATION to define a color representation for the index value, or pass another predefined index value.
95	A representation for the specified color index has not been predefined on this workstation in routine ***** User Action: Use SET COLOUR REPRESENTATION to define a color representation for the index value, or pass another predefined index value.
96	Color index is outside range of the current color model in routine ***** User Action: Check the color model range.
97	Pick identifier is invalid in routine ***** User Action: Either remove the call to SET PICK IDENTIFIER or make sure the pick identifier is an integer. If you obtained the pick identifier from an inquiry function, check the order and the number of the arguments passed to the inquiry function.
98	Specified colour model is not available on the workstation in routine ***** User Action: Call the INQUIRE COLOUR MODEL FACILITIES function for the list of color models supported by your workstation.

A.5 Output Function Errors

Table A-5 lists the errors that result when you call a DEC GKS output function with invalid or undefined arguments.

Table A-5 Output Function Errors

Error	Error Message
100	Number of points is invalid in routine ***** User Action: The number of points specified does not match the number of coordinate points passed. Either alter the specified number of points, or alter the number of coordinate values contained in the arrays passed as arguments.

(continued on next page)

Table A–5 (Cont.) Output Function Errors

Error	Error Message
101	Invalid code in string in routine **** User Action: Your text string contained characters that cannot be printed. Remove the characters.
102	Generalized drawing primitive identifier is invalid in routine **** User Action: Specify another identifier or check to see if the identifier is an integer value.
103	Content of generalized drawing primitive data record is invalid in routine **** User Action: Make sure you passed a correct data record size.
104	At least one active workstation is not able to generate the specified generalized drawing primitive in routine **** User Action: Deactivate the workstations that do not generate the GDPs, or redefine the GDP data record so that all the workstations can generate the primitive.
105	At least one active workstation is not able to generate the specified generalized drawing primitive under the current transformation and clipping rectangle in routine **** User Action: Either redefine the current normalization transformation (creating a different clipping rectangle), or supply different WC points so that the GDP falls within the current clipping rectangle.

A.6 Segment Function Errors

Table A–6 lists the errors that result when you call a DEC GKS segment function with invalid or undefined arguments.

Table A–6 Segment Function Errors

Error	Error Message
120	Specified segment name is invalid in routine **** User Action: Either check the number and the order of the arguments or make sure the segment name is an integer value. If you obtained the segment name from an inquiry function, check the order and the number of the arguments passed to the inquiry function.
121	Specified segment name is already in use in routine **** User Action: Either remove the call to CREATE SEGMENT or check to make sure you specified the correct segment name.
122	Specified segment does not exist in routine ****

(continued on next page)

DEC GKS Error Messages

A.6 Segment Function Errors

Table A-6 (Cont.) Segment Function Errors

Error	Error Message
	User Action: Either check the order and the number of the arguments or make sure you specified an integer value as a segment name. If you used an inquiry function to obtain the segment name, check the order and the number of the arguments passed to the inquiry function.
123	Specified segment does not exist on specified workstation in routine **** User Action: Either remove the function call, or if the segment exists in WISS, associate the segment with the appropriate workstation.
124	Specified segment does not exist on Workstation Independent Segment Storage in routine **** User Action: You attempted to copy, associate, or insert a segment that is not stored in WISS. Either remove the function call or check to see that you specified the correct segment name.
125	Specified segment is open in routine **** User Action: Either remove the call to CREATE SEGMENT or specify another segment name.
126	Segment priority is outside the range [0,1] in routine **** User Action: Change the specified segment priority. If you used an inquiry function to obtain the segment priority value, check the order and the number of the arguments passed to the inquiry function.

A.7 Input Function Errors

Table A-7 lists the errors that result when you call a DEC GKS input function with invalid or undefined arguments.

Table A-7 Input Function Errors

Error	Error Message
140	Specified input device is not present on workstation in routine **** User Action: Make sure you specified the function that applies to the correct logical input device and the correct workstation identifier.
141	Input device is not in REQUEST mode in routine **** User Action: Use one of the SET MODE input functions to set the logical input device to request mode before using this logical input device.
142	Input device is not in SAMPLE mode in routine **** User Action: Use one of the SET MODE input functions to set the logical input device to sample mode before using this logical input device.

(continued on next page)

Table A-7 (Cont.) Input Function Errors

Error	Error Message
143	<p>EVENT and SAMPLE input mode are not available at this level of GKS in routine *****</p> <p>User Action: This error code is required by the standard, but DEC GKS will never generate this error.</p>
144	<p>Specified prompt and echo type is not supported on this workstation in routine *****</p> <p>User Action: Make sure the order of the arguments is correct or change the prompt and echo value. If you obtained the prompt and echo type from an inquiry function, check the order and the number of the arguments passed to the inquiry function.</p>
145	<p>Echo area is outside display space in routine *****</p> <p>User Action: Make sure the specified coordinate points are real values that specify a valid rectangle on the display surface. If you used an inquiry function to obtain the echo area, check the order and the number of the arguments passed to the inquiry function.</p>
146	<p>Contents of input data record are invalid in routine *****</p> <p>User Action: Make sure you specified the correct size of the data record, that the elements of the data record are of the correct data type, and that you have chosen the correct corresponding prompt and echo type. If you used an inquiry function to obtain the data record, check the order and number of the arguments passed to the inquiry function. Also, make sure you have not specified input values that are not accepted by the particular device; you can check the device's capabilities by calling one of the DEFAULT DATA inquiry functions.</p>
147	<p>Input queue has overflowed in routine *****</p> <p>User Action: Check the input queue with greater frequency or flush the input queue.</p>
148	<p>Input queue has not overflowed since GKS was opened or the last invocation of INQUIRE INPUT QUEUE OVERFLOW in routine *****</p> <p>User Action: You called INQUIRE INPUT QUEUE OVERFLOW when the queue was not full, and had not been filled since the beginning of your application. Continue to generate events, if your application still requires input.</p>
149	<p>Input queue has overflowed, but associated workstation has been closed in routine *****</p> <p>User Action: You called INQUIRE INPUT QUEUE OVERFLOW when the queue was full, but since the workstation is closed, information about the overflow is not available. You can set the devices to request mode (removing their prompts from the workstation surface), and then you can either process reports from the queue until empty or you can flush the queue of all reports.</p>
150	<p>No input value of the correct class is in the current event report in routine *****</p> <p>User Action: Make sure you check the input class argument passed to AWAIT EVENT before you try to call the appropriate GET function.</p>
151	<p>Timeout is invalid in routine *****</p>

(continued on next page)

DEC GKS Error Messages

A.7 Input Function Errors

Table A-7 (Cont.) Input Function Errors

Error	Error Message
	User Action: Make sure the timer argument in AWAIT EVENT is a real value between 0.0 and 356,400 seconds, as specified in the format described in the AWAIT EVENT function description in Chapter 9.
152	Initial value is invalid in routine **** User Action: Either check to make sure you specified the correct value, or check the capabilities of the device to see if you requested a value unsupported by the device. If you obtained the value from an inquiry function, check the order and number of arguments specified to the inquiry function.
153	Number of points in the initial stroke is greater than the buffer size in routine **** User Action: Either increase the size of the buffer or reduce the number of points in the initial stroke.
154	Length of initial string is greater than the buffer size in routine **** User Action: Either increase the size of the buffer or decrease the size of the initial string.

A.8 Metafile Function Errors

Table A-8 lists the errors that result when you call a DEC GKS metafile function with invalid or undefined arguments.

Table A-8 Metafile Function Errors

Error	Error Message
160	Item type is not allowed for user items in routine **** User Action: Use an item type greater than or equal to 101, or less than 0 for a user item.
161	Item length is invalid in routine **** User Action: The length of the data item was shorter than necessary for its type. Make sure DEC GKS does not truncate your record when reading the item from a GKSM.
162	No item is left in GKS Metafile input in routine **** User Action: You tried to read past the end of the GKSM. Do not attempt to read items past the item of type 0.
163	Metafile item is invalid in routine **** User Action: Your item data was incorrect. Make sure DEC GKS did not truncate the item while reading from a GKSM and that you specified correct sizes and types. Make sure you are not trying to interpret a user-defined record type. User-defined records have item numbers greater than 100.
164	Item type is not a valid GKS item in routine ****

(continued on next page)

Table A–8 (Cont.) Metafile Function Errors

Error	Error Message
	User Action: You tried to interpret an item of type less than 0 or greater than 100. Make sure DEC GKS did not truncate the item while reading from a GKSM and that you specified correct sizes and types.
165	Content of item data record is invalid for the specified item type in routine **** User Action: There was unexpected or incorrect information in the data record. Make sure you pass the correct storage area.
166	Maximum item data record length is invalid in routine **** User Action: Make sure the data length is not negative.
167	User item cannot be interpreted in routine **** User Action: Do not pass user items to DEC GKS for interpretation.
168	Specified function is not supported in this level of GKS in routine **** User Action: This error code is required by the standard, but DEC GKS will never generate this error.

A.9 Escape Function Errors

Table A–9 lists the errors that result when you call a DEC GKS escape function with invalid or undefined arguments.

Table A–9 Escape Function Errors

Error	Error Message
180	Specified escape function is not supported in routine **** User Action: Check the escape function identifier to make sure that it is a valid integer representing an escape function, and make sure you specified the correct workstation identifier.
181	Specified escape function identifier is invalid in routine **** User Action: Make sure the escape function identifier is a valid integer value.
182	Contents of escape data record are invalid in routine **** User Action: Make sure you specified the correct size of the data record. Also, make sure the elements of the data record are declared to be the correct data type.

A.10 Miscellaneous Errors

Table A–10 lists the DEC GKS miscellaneous errors.

DEC GKS Error Messages

A.10 Miscellaneous Errors

Table A-10 Miscellaneous Errors

Error	Error Message
200	Specified error file is invalid in routine **** User Action: Make sure your specified error handler exists and that it includes the three required parameters in its definition.

A.11 System Errors

Table A-11 lists implementation-dependent system errors.

Table A-11 System Errors

Error	Error Message
300	Storage overflow has occurred in GKS in routine **** User Action: Increase the swap space or the process virtual memory limit.
301	Storage overflow has occurred in segment storage in routine **** User Action: Increase the swap space or the process virtual memory limit.
302	Input/Output error has occurred while reading in routine **** User Action: You specified an illegal metafile for a metafile input workstation. Make sure you specified a valid GKSM or GKS3 metafile, and that you correctly specified the connection identifier.
303	Input/Output error has occurred while writing in routine **** User Action: You specified an illegal metafile for a metafile output workstation. Make sure you specified a valid GKSM or GKS3 metafile, and that you correctly specified the connection identifier.
304	Input/Output error has occurred while sending data to a workstation in routine **** User Action: Submit an SPR.
305	Input/Output error has occurred while receiving data from a workstation in routine **** User Action: Submit an SPR.
306	Input/Output error has occurred during program library management in routine **** User Action: Submit an SPR.
307	Input/Output error has occurred while reading workstation description table in routine **** User Action: Submit an SPR.
308	Arithmetic error has occurred in routine ****

(continued on next page)

Table A-11 (Cont.) System Errors

Error	Error Message
	User Action: You either divided by 0 or caused data overflow. Check the arguments passed in the function call.
400	View up vector and view plane normal are collinear in routine **** User Action: Make sure the view up vector and view plane normal are not collinear.
401	View plane normal is a null vector in routine **** User Action: Make sure the view plane normal is not {0.0, 0.0, 0.0} (null vector).
402	View up vector is a null vector in routine **** User Action: Make sure the view up vector is not {0.0, 0.0, 0.0} (null vector).
403	Projection viewport limits are not within NPC range in routine **** User Action: Make sure the projection viewport limits are in the range 0.0 to 1.0.
404	Projection reference point is between the front and back clipping planes in routine **** User Action: Make sure the projection reference point is not between the front and back clipping planes.
405	Projection reference point is on the view plane in routine **** User Action: Make sure the projection reference point is not on the view plane.
406	Box definition is invalid in routine **** User Action: Make sure the defined box follows the rule $XMIN < XMAX$ and $YMIN < YMAX$.
407	Viewport is not within NDC unit cube in routine **** User Action: Make sure the viewport limits are in the range 0.0 to 1.0.
408	Specified view index is invalid in routine **** User Action: Make sure the view index is a valid index.
409	A representation for the specified view index has not been defined on this workstation in routine **** User Action: Make sure you specified the view index before you call a function that requires the view index argument. Use the INQUIRE LIST OF VIEW INDICES function for a list of the defined view indexes.
410	A representation for the specified view index has not been predefined on this workstation in routine ****

(continued on next page)

DEC GKS Error Messages

A.11 System Errors

Table A-11 (Cont.) System Errors

Error	Error Message
	User Action: Make sure the specified view index is predefined. Use the INQUIRE PREDEFINED VIEW REPRESENTATION function for a list of the predefined view indexes.
411	Workstation window limits are not within the NPC cube in routine **** User Action: Make sure the workstation window limits are in the range 0.0 to 1.0.
412	Back clipping plane is in front of front clipping plane in routine **** User Action: Make sure the back clipping plane is behind the front clipping plane.
413	View clipping limits not within NPC range in routine **** User Action: Make sure the view clipping limits are in the range 0.0 and 1.0.
420	Edge index is invalid in routine **** User Action: Make sure the edge index is greater than or equal to 1.
421	A representation for the specified edge index has not been defined on this workstation in routine **** User Action: Make sure the edge representation is predefined. Use the INQUIRE EDGE FACILITIES function for information on the predefined edge representations.
422	A representation for the specified edge index has not been predefined on this workstation in routine **** User Actions: Make sure the specified edge index has been predefined.
423	Edge type is equal to zero in routine **** User Actions: Make sure the edge type is one of the permitted values. See Appendix B for a list of edge type values.
424	Specified edge type is not supported on this workstation in routine **** User Action: Make sure the edge type is one of the permitted values. See the chapter on your workstation in the <i>Device Specifics Reference Manual for DEC GKS and DEC PHIGS</i> for a list of supported edge types.
425	Edge width scale factor is less than zero in routine **** User Action: Make sure the edge width scale factor is greater than or equal to 0.0.
426	Pattern reference vectors are collinear in routine **** User Action: Make sure the pattern reference vectors are not collinear.
427	Specified HLHSR mode not supported on workstation in routine ****

(continued on next page)

Table A–11 (Cont.) System Errors

Error	Error Message
	User Action: Make sure the specified HLHSR mode is supported by your workstation. See the chapter on your workstation in the <i>Device Specifics Reference Manual for DEC GKS and DEC PHIGS</i> for a list of supported HLHSR modes.
428	Specified HLHSR identifier is invalid in routine **** User Action: Make sure the HLHSR identifier is greater than or equal to 0.
429	Specified HLHSR mode is invalid in routine **** User Action: Make sure the HLHSR mode is greater than or equal to 0.
430	The text direction vectors are collinear in routine **** User Action: Make sure the text direction vectors are not collinear.
431	List of point lists is invalid in routine **** User Action: Make sure the number of lists is greater than 0, and the number of points for a set is greater than or equal to 3.
432	At least one active workstation is not able to generate the specified generalized drawing primitive under the current transformation and clipping volume in routine **** User Action: Do nothing. This is an informational message.

A.12 FORTRAN Binding Errors

Table A–12 lists those error messages that are specific to the FORTRAN binding functions.

Table A–12 FORTRAN Binding Errors

Error	Error Message
2000	Enumeration type out of range—the INTEGER passed as a GKS enumerated type is not within the range of valid values in routine **** User Action: Make sure you properly define the enumerated values.
2001	Output parameter size insufficient—a FORTRAN array or string being passed as an output parameter is too small to contain the returned information in routine **** User Action: Redefine the size of the output string, or check the order or number of arguments passed to the function.
2002	List or set element not available—for a nonempty list or set, a value less than zero or greater than the size of the list or set was passed as the requested list or set element in an inquiry routine ****

(continued on next page)

DEC GKS Error Messages

A.12 FORTRAN Binding Errors

Table A-12 (Cont.) FORTRAN Binding Errors

Error	Error Message
	User Action: Either check the declaration of the value passed, or check the order and number of all arguments passed.

A.13 Implementation-Specific Errors

All the DEC GKS specific errors are negative. These errors are either of severity "error" or "fatal error." Error numbers in the range -90 to -100 are fatal errors. If one of these errors occurs, submit a Software Performance Report (SPR) indicating the error number, corresponding message, and any relevant particulars. For more information concerning SPRs, see the DEC GKS installation guide.

Table A-13 lists the errors that are implementation specific.

Table A-13 Implementation-Specific Errors

Error	Error Message
-2	Requested color map could not be created as specified in routine ***** User Action: The specified color map is too large. Check to make sure you specified the correct color map size and type (either physical or virtual) and that you have not exceeded the limitations of your device.
-3	Invalid data in workstation description file in routine ***** User Action: Make sure the format of your description file is valid for your particular workstation.
-4	Invalid bit mask in workstation type in routine ***** User Action: Check to make sure you specified a bit mask workstation type value that is valid for your workstation, and that you are running your program on the expected type of workstation.
-5	Bad string addresses found writing choice data record in routine ***** User Action: There is an illegal array of string pointers passed to the choice data record in the specified routine. Make sure you properly initialized the arrays containing string addresses and string lengths. Also, make sure you have declared a buffer to hold choice strings, and that your string address array contains addresses of the elements in your choice string array.
-6	Echo area is too narrow for data in routine ***** User Action: The specified input echo area minimum and maximum X values are too close in proximity. Make sure you did not swap X and Y values, and that your specified X values are a greater distance from each other.
-7	Maximum number of representable choices exceeded in routine *****

(continued on next page)

Table A-13 (Cont.) Implementation-Specific Errors

Error	Error Message
	User Action: The number of requested choices is too large for the workstation type. You can use INQUIRE DEFAULT CHOICE DEVICE DATA to obtain the maximum choices available for your workstations, and then break your menu into two smaller menus.
-8	Echo area is too short for data in routine **** User Action: The specified input echo area minimum and maximum Y values are too close in proximity. Make sure you did not swap X and Y values, and that your specified Y values are a greater distance from each other.
-9	Binary format and integer number representation not supported in this implementation of GKS in routine **** User Action: You opened a metafile of an incompatible type. Check the metafile type.
-10	Invalid value specified for ASF in routine **** User Action: Check the array to make sure it has 13 elements and that its elements only contain the value GBUNDL (0) or GINDIV (1).
-11	Invalid value specified for fill area interior style in routine **** User Action: Make sure you passed one of the values GHOLLO (0), GSOLID (1), GPATTR (2), or GHATCH (3).
-12	Invalid value specified for horizontal component of text alignment in routine **** User Action: Make sure you passed one of the values GAHNOR (0), GALEFT (1), GACENT (2), or GARITE (3).
-13	Invalid value specified for vertical component of text alignment in routine **** User Action: Make sure you passed one of the values GAVNOR (0), GATOP (1), GACAP (2), GAHALF (3), GABASE (4), or GABOTT (5).
-14	Invalid value specified for text precision in routine **** User Action: Make sure you passed one of the values GSTRP (0), GCHARP (1), or GSTRKP (2).
-15	Invalid value specified for text path in routine **** User Action: Make sure you passed one of the values GRIGHT (0), GLEFT (1), GUP (2), or GDOWN (3).
-16	Echo switch is invalid in routine **** User Action: Make sure you passed the one of the values GNECHO (0) or GECHO (1). Also, if you used an inquiry function to obtain the echo switch, check to see that the arguments to the inquiry function are specified in the correct order.

(continued on next page)

DEC GKS Error Messages

A.13 Implementation-Specific Errors

Table A-13 (Cont.) Implementation-Specific Errors

Error	Error Message
-17	Inquired device values not set or realized in routine **** User Action: Check the value type argument to make sure it is either GSET (0) or GREALI (1).
-18	The following error occurred when GKS was interpreting an item **** User Action: An error occurred while interpreting a metafile item. DEC GKS follows this error message with another message that signals the appropriate action.
-19	Invalid error status parameter specified in routine **** User Action: You passed an illegal error code to ERROR LOGGING. Make sure the error code passed to ERROR LOGGING is one of the codes described in this appendix.
-20	GKS not in proper state: GKS in the ERROR state in routine **** User Action: You attempted to execute a DEC GKS function other than an error-handling or inquiry function. Remove all calls to DEC GKS functions, other than inquiry and error-handling function calls, from your error-handling code.
-21	Function is not supported in this level of GKS in routine **** User Action: Remove the call to the unsupported function.
-22	Invalid segment transformation in routine **** User Action: Check your calls to EVALUATE TRANSFORMATION MATRIX and to ACCUMULATE TRANSFORMATION MATRIX to make sure you passed valid transformation components. Also, make sure you specified a transformation matrix to SET SEGMENT TRANSFORMATION or to INSERT SEGMENT.
-23	Invalid value specified for clipping flag in routine **** User Action: Make sure you passed either the value GNCLIP (0) or GCLIP (1).
-24	Invalid value specified for viewport priority flag in routine **** User Action: Make sure you passed either the value GHIGHER (0) or GLOWER (1).
-25	Invalid value specified for update workstation flag in routine **** User Action: Make sure you passed either the value GPOSTP (0) or GPERFO(1).
-26	Invalid value specified for deferral mode in routine **** User Action: Make sure you passed one of the values GASAP (0), GBNIG (1), GBNIL (2), or GASTI (3).
-27	Invalid value specified for regeneration mode in routine **** User Action: Make sure you passed either the value GSUPPD (0) or GALLOW (1).

(continued on next page)

Table A-13 (Cont.) Implementation-Specific Errors

Error	Error Message
-28	<p>Invalid value specified for expansion factor in routine ****</p> <p>User Action: Check to make sure you specified a real number value greater than the value 0.0. The value 1.0 causes no expansion.</p>
-29	<p>Invalid data record size for specified prompt and echo type in routine ****</p> <p>User Action: Check to make sure you specified a data record of the correct size as determined by your chosen prompt and echo type.</p>
-30	<p>Cannot load workstation handler: error during image activation in routine ****</p> <p>User Action: DEC GKS could not activate your workstation handler's shareable image. Make sure your workstation handler is a valid, shareable image.</p>
-31	<p>Cannot load graphics handler: invalid DFT in routine ****</p> <p>User Action: Your device function tables are incompatible. You need to build your device function table again using the appropriate macro. For more information, see the <i>Building a Device Handler System for DEC GKS and DEC PHIGS</i>.</p>
-32	<p>Font file for stroke precision text not found or unusable in routine ****</p> <p>User Action: DEC GKS could not activate your workstation handler's shareable image. See the appropriate device-specific chapter in the <i>Device Specifics Reference Manual for DEC GKS and DEC PHIGS</i> to determine if the specified font is supported on your device. If you are not using a DEC GKS supported graphics handler, make sure your handler defines the proper environment options, and they reference a valid file.</p>
-33	<p>Array descriptor is not acceptable in routine ****</p> <p>User Action: An item in the array descriptor is either invalid or inconsistent. Make sure you have passed the array by descriptor and that you fill the descriptor with valid values. If you have, and you use an inquiry function to initialize the array variable, make sure all the arguments are specified to the inquiry function in the correct order. Also, if the array is passed to the CELL ARRAY function, make sure you have declared a two-dimensional array.</p>
-34	<p>String length less than or equal to 0 in routine ****</p> <p>User Action: You specified an invalid character string. Check the declaration, definition, or assignment statements involving the character variable.</p>
-35	<p>Kernel has detected an unexpected error from a device handler in routine ****</p> <p>User Action: The device handler encountered an error. DEC GKS follows this error message with another message that signals the appropriate action.</p>
-36	<p>Cannot load device handler: error during image activation in routine ****</p>

(continued on next page)

DEC GKS Error Messages

A.13 Implementation-Specific Errors

Table A-13 (Cont.) Implementation-Specific Errors

Error	Error Message
	User Action: DEC GKS could not activate your device handler's shareable image. Make sure your device handler is a valid, shareable image. This error message is specific to handlers that affect a device (VAXstations) as opposed to a graphics language (PostScript).
-37	Error in device handler during event flag allocation in routine **** User Action: A graphics handler was unable to acquire all of its needed event flags. The application must release event flags for use by the graphics handler.
-38	Error in device handler, cannot allocate device in routine **** User Action: You used your graphics handler with an invalid physical device. Make sure you use the proper physical device or that you specify the correct workstation type value to OPEN WORKSTATION.
-39	Descriptor is not acceptable in routine **** User Action: Make sure you have passed the variable by descriptor. If you have and you use an inquiry function to initialize the variable, make sure all the arguments are specified to the inquiry function in the correct order.
-40	Illegal device pointer in routine **** User Action: Check your handler code for null pointers or otherwise invalid pointers.
-41	Driver handler WDT is invalid in routine **** User Action: You illegally defined a workstation description table entry. Check your workstation description table definitions for your graphics handler.
-42	Logical name for the list of workstation types, GKS\$LIST_TYPES, could not be translated in routine **** User Action: You improperly defined the logical name. Make sure the translation of GKS\$LIST_TYPES is as expected.
-43	VAXstation Workstation Software is not present, workstation type is invalid in routine **** User Action: Check to make sure either that you specify the correct workstation type when opening a workstation other than a VAXstation, or that you passed a correct workstation type value to one of the workstation description table or state list inquiry functions. If you are working on a MicroVAX, make sure you install the VAXstation Workstation Software.
-44	Error trying to save or restore VT340 color map in routine **** User Action: Submit an SPR.
-45	Unable to decompose fill area or fill area set in routine **** User Action: Simplify the fill area by breaking it up into smaller fill areas until the error does not occur.

(continued on next page)

Table A-13 (Cont.) Implementation-Specific Errors

Error	Error Message
-46	No default connection identifier for specified workstation type in routine **** User Action: Define the environment option to be a valid GKS connection identifier.
-90	Internal GKS error: Bad memory address freed in routine **** User Action: DEC GKS memory data structures were corrupted. Submit an SPR.
-91	Internal GKS error: Invalid function pointer parameter in error handler in routine **** User Action: A DEC GKS internal data structure was corrupted. Submit an SPR.
-92	Internal GKS error: Insufficient virtual memory in routine **** User Action: DEC GKS was unable to allocate enough virtual memory. Check to make sure the problem is not caused by storing too much in segment storage or by defining a very large cell array. If you cannot reduce storage by checking segments and cell arrays, submit an SPR.
-93	Internal GKS error: Prompt and echo type not supported in routine **** User Action: Internal error. Submit an SPR.
-94	Internal GKS error: Corrupted segment memory in routine **** User Action: Internal error. Submit an SPR.
-95	Internal GKS error: Negative size passed to allocate memory in routine **** User Action: An invalid size was passed to the DEC GKS memory allocation routines. If you generate this error using a user-written graphics handler, make sure the value of the local storage area is a valid value.
-96	Internal GKS error: Illegal number of points to device handler for rectangular polygon in routine **** User Action: Internal error. Submit an SPR.
-97	Internal GKS error: Insufficient buffer size for translated logical name in routine **** User Action: Make sure none of the environment options is a string greater than 255 characters. If you cannot locate an error, submit an SPR.
-98	Internal GKS error: Too many translations of logical name in routine **** User Action: You may have recursively defined a logical name. Check the currently defined logical names to see if all are properly defined. If you cannot locate an error, submit an SPR.
-99	Internal GKS error: Unable to reduce number of points in fill area to requested limit in routine ****

(continued on next page)

DEC GKS Error Messages

A.13 Implementation-Specific Errors

Table A-13 (Cont.) Implementation-Specific Errors

Error	Error Message
	User Action: Internal error. Submit an SPR.
-100	Internal GKS error: Device handler received unexpected input access in routine ***** User Action: Internal error. Submit an SPR.
-150	Edge index is less than zero in routine ***** User Action: Make sure the edge index passed in the escape function GESEI is valid.
-151	Edge width scale factor is less than zero ***** User Action: Make sure the edge width scale factor passed in the escape Set Edge Index (GESEW) is valid.
-154	A representation for the specified edge index has not been predefined on this workstation in routine ***** User Action: Check the index of the edge being inquired about to make sure it is predefined.
-155	Display speed is less than zero in routine ***** User Action: Pass a positive real value to the escape Set Speed (GESP).
-156	Loudness is outside range [0,1] in routine ***** User Action: Pass a valid value to the escape Beep (GEB) .
-157	Duration is less than zero in routine ***** User Action: Make sure your duration value is greater than or equal to 0.
-158	GDP primitive is not defined by the supplied data in routine ***** User Action: DEC GKS is unable to form the desired primitive. See the error message listing in the description of the GDP that generated the error (in the <i>Device Specifics Reference Manual for DEC GKS and DEC PHIGS</i>). This listing gives specific information concerning the primitive you attempted to draw.
-159	Arc type is invalid in routine ***** User Action: See the error message listing in the description of the GDP that generated the error (in the <i>Device Specifics Reference Manual for DEC GKS and DEC PHIGS</i>). This listing gives specific information concerning the primitive you attempted to draw.
-160	Insufficient space in escape output data record arrays in routine ***** User Action: You passed addresses of arrays that were too small to contain the data to be written to them. Pass addresses of larger array buffers in the last four components of the escape data record.

(continued on next page)

Table A-13 (Cont.) Implementation-Specific Errors

Error	Error Message
-161	Specified bounding box is too small in routine **** User Action: You specified text attributes that were too large to fill the text in the bounding box (the extent rectangle). Use a larger bounding box, or reduce the text height or the character expansion factor.
-162	Edge index is invalid in routine **** User Action: Make sure you use a valid edge index.
-163	Specified edge type is not supported on this workstation in routine **** User Action: Make sure you use a valid edge type.
-300	Invalid value specified for highlighting in routine **** User Action: Make sure you specify either GNORML (0) or GHILIT (1).
-301	Invalid value specified for visibility in routine **** User Action: Make sure you specify either GINVIS (0) or GVISI (1).
-302	Invalid value specified for detectability in routine **** User Action: Make sure you specify either GUNDET (0) or GDETEC (1).
-303	Input device cannot be activated due to conflict with another input device that is currently active in routine **** User Action: The input device echo area overlaps the input device echo area on the display. Because these two input devices could overwrite each other's echo on the display, they cannot be active at the same time. Typically, on a nonwindowing system output device, a choice, string, or valuator echo area cannot overlap the input echo of a locator, stroke, or pick device. On a windowing system, this error should not occur, because the input echos are drawn in separate windows. You should change the input echo areas and make sure that they do not overlap.
-304	Cannot set input device echo on due to conflict with other input devices active in the same echo area in routine **** User Action: Two or more input devices are already active, but one of them is in NOECHO mode and the application is attempting to put it into ECHO mode. You should change the input echo areas and make sure they do not overlap.
-305	Error parsing GKS\$[wstype]_INPUT_DEVICES logical name due to syntax error in routine **** User Action: Check the definition of the logical name with the syntax description in the Tektronix 41xx chapter in the <i>Device Specifics Reference Manual for DEC GKS and DEC PHIGS</i> .

(continued on next page)

DEC GKS Error Messages

A.13 Implementation-Specific Errors

Table A-13 (Cont.) Implementation-Specific Errors

Error	Error Message
-306	<p>The definition of GKS\$HPGL_THRESHOLD is invalid (contains nonnumeric values in routine) ****</p> <p>User Action: Check the definition of GKS\$HPGL_THRESHOLD and redefine to range 0 to 1023.</p>
-450	<p>Valuator device could not be used—dials are not available on the PCM device in routine ****</p> <p>User Action: The hardware dial box is not available, either because it is not present or because it was not hooked up correctly. Either hook up the dial and button box correctly, or change the value of the environment option "use dials and buttons" so the hardware dials will not be used. See the chapter on PCM button and dials in the <i>Device Specifics Reference Manual for DEC GKS and DEC PHIGS</i>.</p>
-451	<p>Choice device could not be used—buttons are not available on the PCM device in routine ****</p> <p>User Action: The hardware button box is not available, either because it is not present or because it was not hooked up correctly. Either hook up the dial and button box correctly, or change the value of the environment option "use dials and buttons" so the hardware buttons will not be used. See the chapter on PCM button and dials in the <i>Device Specifics Reference Manual for DEC GKS and DEC PHIGS</i>.</p>
-452	<p>PCM initialization failed—hardware dials and buttons not available in routine ****</p> <p>User Action: The PCM server is not running. It has either not been started or the PCM dial and button box is not connected. Either hook up the dial and button box correctly, or change the value of the environment option "use dials and buttons" so the hardware buttons will not be used. See the chapter on PCM button and dials in the <i>Device Specifics Reference Manual for DEC GKS and DEC PHIGS</i>.</p>
-453	<p>PCM configuration failed—valuator or choice device not initialized in routine ****</p> <p>User Action: Either hook up the dial and button box correctly, or change the value of the environment option "use dials and buttons" so the hardware buttons will not be used. See the chapter on PCM button and dials in the <i>Device Specifics Reference Manual for DEC GKS and DEC PHIGS</i>.</p>
-1517	<p>Error trying to open UID database</p> <p>User Action: Verify that the UID files gfx_decw.uid and gfx_decw_xx.uid (where xx is the language code) exist in either the specified system area or user path.</p>
-1518	<p>Error trying to fetch from UID database</p> <p>User Action: Verify that the name used in the DwtFetchWidget or DwtFetchWidgetOverride command matches that in the diagrams (in the <i>Device Specifics Reference Manual for DEC GKS and DEC PHIGS</i>), and has not been altered while editing the UIL files.</p>
-2000	<p>Illegal call to routine **** while GKS is in an error state</p> <p>User Action: Make sure GKS is not called while it is already in an error state. This may happen if you call GKS functions from the GKS error handler provided by the application.</p>

(continued on next page)

Table A-13 (Cont.) Implementation-Specific Errors

Error	Error Message
-2001	Invalid value specified for ASF in routine ***** User Action: Check the array to make sure it has 13 elements and that its elements only contain the value GBUNDL (0) or GINDIV (1).
-2002	Invalid clipping indicator in routine ***** User Action: Make sure you passed either the value GNCLIP (0) or GCLIP (1).
-2003	Invalid normalization transformation priority in routine ***** User Action: Make sure the arguments are specified in the correct order and that the normalization priority value is either GHIGHR (0) or GLOWER (1).
-2004	Invalid view index specified in routine ***** User Action: Make sure the arguments are specified in the correct order and that the specified view index is in the permitted range (greater than or equal to 0 and less than the maximum number of view representations).
-2005	Invalid metafile format found in routine ***** User Action: Make sure the input file you are trying to interpret is a metafile that was generated by DEC GKS.
-2006	Null text direction vector found in routine ***** User Action: Make sure the arguments are specified in the correct order and that the text direction vector is not (0.0, 0.0, 0.0) (null vector).
-2007	Invalid text path specified in routine ***** User Action: Make sure you passed one of the values GRIGHT (0), GLEFT (1), GUP (2), or GDOWN (3).
-2008	Invalid text character alignment specified in routine ***** User Action: Make sure the arguments are specified in the correct order and that the text alignment specified is a permitted value (GAHNOR (0), GALEFT (1), GACENT (2), or GARITE (3) for the horizontal component, and GAVNOR (0), GATOP (1), GACAP (2), GAHALF (3), GABASE (4), or GABOTT (5) for the vertical component).
-2009	Invalid fill area interior style specified in routine ***** User Action: Make sure you passed one of the values GHOLLO (0), GSOLID (1), GPATTR (2), or GHATCH (3).
-2010	Invalid edge flag specified in routine ***** User Action: Make sure the arguments are specified in the correct order and that the specified edge flag is a permitted value (GOFF (0) or GON (1)).

(continued on next page)

DEC GKS Error Messages

A.13 Implementation-Specific Errors

Table A-13 (Cont.) Implementation-Specific Errors

Error	Error Message
-2011	Invalid aspect source flag specified in routine **** User Action: Check the array to make sure it has 13 elements and that its elements only contain the value GBUNDL (0) or GINDIV (1).
-2012	Null pattern reference vector found in routine **** User Action: Make sure the arguments are specified in the correct order and that the specified pattern reference vector is not {0.0, 0.0, 0.0} (null vector).
-2013	Start value for inquiry is negative or too large in routine **** User Action: Make sure the arguments are specified in the correct order and that the start value is in the permitted range (greater than or equal to 0 and less than the maximum number).
-2014	Invalid clear control flag specified in routine **** User Action: Make sure the arguments are specified in the correct order and that the clear control flag is a permitted value (GCONDI (0) or GALWAY (1)).
-2015	Invalid update control flag specified in routine **** User Action: Make sure you passed one of the values GASAP (0), GBNIG (1), GBNIL (2), or GASTI (3).
-2016	Invalid deferral mode specified in routine **** User Action: Make sure the arguments are specified in the correct order and that the specified deferral mode is a permitted value (GSUPPD (0) or GALLOW (1)).
-2017	Invalid implicit regeneration mode specified in routine **** User Action: Make sure the arguments are specified in the correct order and that the specified implicit regeneration mode is a permitted value (GSUPPD (0) or GALLOW (1)).
-2018	Invalid view transformation priority in routine **** User Action: Make sure the arguments are specified in the correct order and that the view transformation priority is a permitted value (GHIGHER (0) or GLOWER (1)).
-2019	Invalid text precision specified in routine **** User Action: Make sure the arguments are specified in the correct order and that the specified text precision is a permitted value (GSTRP (0), GCHARP (1), or GSTRKP (2)).
-2020	Invalid coordinate switch argument in routine **** User Action: Make sure the arguments are in the correct order and that the coordinate switch argument is a permitted value (GWC (0) or GNDC (1)).

(continued on next page)

Table A-13 (Cont.) Implementation-Specific Errors

Error	Error Message
-2021	Invalid projection type argument in routine **** User Action: Make sure the arguments are specified in the correct order and that the projection type argument is either GPARL (0) or GPERS(1).
-2022	Front plane is too near back plane for this viewport in routine **** User Action: Make the distance between the front and back planes is larger.
-2023	Equal window z limits not allowed with unequal viewport z limits in routine **** User Action: Either make the window Z limits unequal, or make the viewport Z limits equal.
-2024	Invalid input device mode specified in routine **** User Action: Make sure the arguments are specified in the correct order and that the specified input device mode is one of the following values: GREQU (0), GSAMPL (1), or GEVENT (2).
-2025	Invalid input device echo state specified in routine **** User Action: Make sure the arguments are specified in the correct order and that the input device echo state is either GNECHO (0) or GECHO (1).
-2026	Invalid polyline/fill area control flag in data record in routine **** User Action: Make sure the arguments are specified in the correct order and that the polyline /fill area control flag is a permitted value (GPLINE (0) or GFILLA (1)).
-2027	Invalid attribute control flag in data record in routine **** User Action: Make sure the arguments are specified in the correct order and that the attribute control flag in the data record is a permitted value (GCURNT (0) or GSPEC (1)).
-2028	Invalid data record size for specified prompt and echo type in routine **** User Action: Check the specified data record size against the data record required for the specified PET.
-2029	Invalid normalization transformation number specified in routine **** User Action: Make sure the arguments are specified in the correct order and that the specified normalization transformation number is a permitted value.
-2030	NDC position(s) outside NDC unit cube in routine **** User Action: Make sure the specified NDC values are in the range 0.0 to 1.0.
-2031	Invalid input device class specified in routine ****

(continued on next page)

DEC GKS Error Messages

A.13 Implementation-Specific Errors

Table A-13 (Cont.) Implementation-Specific Errors

Error	Error Message
	User Action: Make sure the arguments are specified in the correct order and that the specified input device class is one of the following values: GNCLAS (0), GLOCAT (1), GSTROK (2), GVALUA (3), GCHOIC (4), GPICK (5), GSTRIN (6), or GVIEW (7).
-2032	Invalid visibility flag specified in routine **** User Action: Make sure the arguments are specified in the correct order and that the specified visibility flag is either GINVIS (0) or GVISI (1).
-2033	Invalid highlighting flag specified in routine **** User Action: Make sure the arguments are specified in the correct order and that the specified highlighting flag is either GNORML (0) or GHILIT (1).
-2034	Invalid detectability flag specified in routine **** User Action: Make sure the arguments are specified in the correct order and that the specified detectability flag is either GUNDET (0) or GDETEC (1).
-2036	Invalid matrix specified, matrix has a zero fourth row in routine **** User Action: Make sure the specified matrix does not have a zero fourth row.
-2037	Warning, the specified view matrix is not invertible in routine **** User Action: Specify a matrix that is invertible.
-2038	Warning, a 3D item cannot be written into GKSM (2D metafile output) in routine **** User Action: Avoid using three-dimensional functions when writing to a GKSM two-dimensional metafile.
-2039	Warning, the normalization transformation is 3D and cannot be used with GKSM (2D metafile output) in routine **** User Action: Avoid using three-dimensional functions when writing to a GKSM two-dimensional metafile.
-2040	Warning, the segment transformation is 3D and cannot be used with GKSM (2D metafile output) in routine **** User Action: Avoid using three-dimensional functions when writing to a GKSM two-dimensional metafile.
-2041	Warning, 3D inquiries are invalid with GKSM (2D metafile output) in routine **** User Action: Avoid using three-dimensional functions when writing to a GKSM two-dimensional metafile.
-2500	Insufficient buffer size for translated logical name in routine ****

(continued on next page)

Table A-13 (Cont.) Implementation-Specific Errors

Error	Error Message
	User Action: Make sure none of the environment options equals a string greater than 255 characters. If you cannot locate an error, submit an SPR.
-2501	Too many translations of logical name in routine **** User Action: You may have recursively defined a logical name. Check the currently defined logical names to see if all are properly defined. If you cannot locate an error, submit an SPR.
-2502	System error during logical name translation in routine **** User Action: Make sure the environment option has a valid assignment string or other environment option. The maximum length of assignment is 255 characters.
-2503	Error closing the error logging file in routine **** User Action: Make sure there is enough free disk space available.
-2504	Image activation error during load of workstation handler in routine **** User Action: Make sure your workstation handler is a valid, shareable image. Reinstall DEC GKS.
-2505	Invalid WFT detected during load of workstation handler in routine **** User Action: You need to build your device function table again using the appropriate macro. For more information, see the <i>Building a Device Handler System for DEC GKS and DEC PHIGS</i> .
-2506	Invalid string descriptor found in routine **** User Action: Make sure you specify a valid string descriptor. See the file gks_descrip.h in the system library.
-2507	Insufficient buffer space to convert string descriptor in routine **** User Action: Submit an SPR.
-2508	Image activation error during load of device handler in routine **** User Action: Make sure your device handler is a valid, shareable image. This error message is specific to handlers that affect a device (VAXstations) as opposed to a graphics language (PostScript).
-2509	Invalid DFT detected during load of device handler in routine **** User Action: You need to build your device function table again using the appropriate macro. For more information, see the <i>Building a Device Handler System for DEC GKS and DEC PHIGS</i> .
-2510	System error during logical name creation in routine **** User Action: GKS is not able to create a process table logical. See your system manager.

(continued on next page)

DEC GKS Error Messages

A.13 Implementation-Specific Errors

Table A-13 (Cont.) Implementation-Specific Errors

Error	Error Message
-2511	Invalid array descriptor in routine **** User Action: Make sure you have passed the array by descriptor and that you fill the descriptor with valid values. If you have, and you use an inquiry function to initialize the array variable, make sure that all the arguments are specified to the inquiry function in the correct order. Also, if the array is passed to the CELL ARRAY function, make sure you have declared a two-dimensional array.
-2512	Invalid representation of integer in translated string in routine **** User Action: Make sure the environment options that require an integer define a valid integer (not another data type).
-2513	Error translating workstation type list in routine **** User Action: Make sure the translation of GKS\$LIST_TYPES is as expected.
-2514	Invalid device handler WDT detected in routine **** User Action: Check your workstation description table definitions for your graphics handler.
-2515	Device handler WDT has at least one bundle table of zero size in routine **** User Action: Check your workstation description table. All bundle tables must be defined in the workstation description table.
-2517	Invalid function name descriptor found in routine **** User Action: Submit an SPR.
-2518	Create operation failed on GKS error file in routine **** User Action: Check the validity of the error file specification and check the available disk space.
-2519	Open operation failed on GKS error file in routine **** User Action: Check the validity of the error file specification and check the available disk space.
-2520	Close operation failed on GKS error file in routine **** User Action: Check the validity of the error file specification and check the available disk space.
-2521	Delete operation failed on empty GKS error file in routine **** User Action: Check the validity of the error file specification, file references, file protection, and file owner.

(continued on next page)

Table A-13 (Cont.) Implementation-Specific Errors

Error	Error Message
-2522	Font file not found or unusable in routine **** User Action: See the appropriate device-specific chapter in the <i>Device Specifics Reference Manual for DEC GKS and DEC PHIGS</i> to determine if the specified font is supported on your device. If you are not using a DEC GKS supported graphics handler, make sure your handler defines the proper logical names, and that the logicals reference a valid file.
-2523	String length less than or equal to 0 in routine **** User Action: Check the declaration, definition, or assignment statements involving the character variable.
-2525	Invalid function identifier found in routine **** User Action: Submit an SPR.
-2528	Internal error converting time to binary form in routine **** User Action: Submit an SPR.
-2529	System error setting interval timer in routine **** User Action: Submit an SPR.
-2530	System error waiting for event flag in routine **** User Action: Submit an SPR.
-2531	System error canceling interval timer in routine **** User Action: Submit an SPR.
-2532	System error setting event flag in routine **** User Action: Submit an SPR.
-2533	System error getting event flag in routine **** User Action: Submit an SPR.
-2534	System error freeing event flag in routine **** User Action: Submit an SPR.
-2535	System error setting timer event handler in routine **** User Action: Submit an SPR.
-2536	System error resetting timer event handler in routine **** User Action: Submit an SPR.

(continued on next page)

DEC GKS Error Messages

A.13 Implementation-Specific Errors

Table A-13 (Cont.) Implementation-Specific Errors

Error	Error Message
-2537	Nonfatal input manager error detected in routine **** User Action: Submit an SPR.
-2538	Insufficient buffer space available in routine **** User Action: Submit an SPR.
-2539	Device handler error detected in routine **** User Action: DEC GKS follows this error message with another message that signals the appropriate action.

A.14 Fatal Errors

Table A-14 lists the fatal DEC GKS and system errors.

Table A-14 Fatal Errors

Error	Error Message
-3000	Internal data error detected in routine **** User Action: Submit an SPR.
-3001	Error looping condition forced emergency close in routine **** User Action: Submit an SPR.
-3002	Fatal error condition forced emergency close in routine **** User Action: Submit an SPR.
-3003	Illegal call to native interface, bypassing FORTRAN binding, detected in routine **** User Action: Submit an SPR.
-3004	Illegal call to native interface, bypassing C binding, detected in routine **** User Action: Submit an SPR.
-3500	Memory zone allocation error, insufficient virtual memory in routine **** User Action: Increase the system page or swap file.
-3501	Memory zone deallocation error in routine **** User Action: Submit an SPR.

(continued on next page)

Table A-14 (Cont.) Fatal Errors

Error	Error Message
-3502	Memory allocation error, insufficient virtual memory in routine **** User Action: Increase the system page or swap file.
-3503	Memory deallocation error in routine **** User Action: Submit an SPR.
-3504	Memory reallocation error in routine **** User Action: Submit an SPR.
-3505	Float array allocation error, insufficient virtual memory in routine **** User Action: Submit an SPR.
-3506	Float array deallocation error in routine **** User Action: Submit an SPR.
-3508	Workstation handler error in routine **** User Action: Submit an SPR.
-3509	Fatal input manager error detected in routine **** User Action: Submit an SPR.
-3510	Inadequate size for the internal array containing the list of prompt and echo types in routine **** User Action: Submit an SPR.
IVP	DEC GKS Installation Verification Procedure failed User Action: See the installation guide for information on what to do if the IVP fails.

Constants

Insert tabbed divider here. Then discard this sheet.



B

Constants

This appendix lists the DEC GKS constants defined for the FORTRAN binding language interface, their values, and a short description of each. Using constants in your DEC GKS programs makes your code easier to read.

To use constants in your program, you must include a definitions file in your code. The language definition file for the FORTRAN binding is gks.f.

Table B-1 lists the FORTRAN constants names, their values, and descriptions.

Table B-1 Constants

Constant	Value	Description
Arc Types:		
GATOPN	1	Arc type open
GATPIE	2	Arc type pie
GATCHRD	3	Arc type chord
Aspect Source Flags:		
GBUNDL	0	Bundled
GINDIV	1	Individual
Attribute Control Flags:		
GCURNT	0	Input data record current values
GSPEC	1	Input data record specified values
Choice Input Prompt Flags:		
GPROFF	0	Prompt off
GPRON	1	Prompt on
Choice Status Types:		
GNONE	0	No input obtained
GOK	1	Input obtained
GNCHOI	3	Input is NOCHOICE

(continued on next page)

Constants

Table B-1 (Cont.) Constants

Constant	Value	Description
Clear Screen States:		
GCONDI	0	Clear conditionally
GALWAY	1	Clear always
Clipping Flags:		
GNCLIP	0	Clipping off
GCLIP	1	Clipping on
Color Availability Flags:		
GMONOC	0	Monochrome
GCOLOR	1	Color
Coordinate Switch:		
GWC	0	World coordinates
GNDC	1	Normalized device coordinates
Current and Request Values:		
GCURVL	0	Current value
GRQSVL	1	Request value
Default Connection Identifier:		
GCONID	0	Default connection identifier
Deferral Modes:		
GASAP	0	As soon as possible
GBNIG	1	Before the next global interaction
GBNIL	2	Before the next local interaction
GASTI	3	At some time
Detectability Flags:		
GUNDET	0	Set to undetectable
GDETEC	1	Set to detectable
Device Coordinate Units:		
GMETRE	0	Meters
GOTHU	1	Other units

(continued on next page)

Table B-1 (Cont.) Constants

Constant	Value	Description
Display Surface States:		
GNEMPT	0	Display surface not empty
GEMPTY	1	Display surface empty
Dynamic Modification States:		
GIRG	0	Implicit regeneration necessary
GIMM	1	Immediate
Echo States:		
GNECHO	0	Echo disabled
GECHO	1	Echo enabled
Edge Flags:		
GOFF	0	Edge flag off
GON	1	Edge flag on
Edge Types:		
GESOL	1	Solid
GEDASH	2	Dashed
GEDOT	3	Dotted
GEDASD	4	Dashed-dotted
GEDS2D	-1	Double-dashed dotted
GEDS3D	-2	Triple-dashed dotted
GELGDS	-3	Long-dashed
GELSDS	-4	Long-short-dashed
GESPDS	-5	Spaced-dashed
GESPDT	-6	Spaced-dotted
GEDBDT	-7	Double dotted
GETPDT	-8	Triple dotted
Error-Handling Modes:		
GERROF	0	Error handling off
GERRON	1	Error handling on
Escape Function Identifiers:		
GESP	-100	Set display speed
GEP	-101	Generate hardcopy of workstation surface

(continued on next page)

Constants

Table B–1 (Cont.) Constants

Constant	Value	Description
Escape Function Identifiers:		
GEB	–103	Beep
GEPOPW	–106	Pop workstation
GEP SHW	–107	Push workstation
GESEHM	–108	Set error handling mode
GESVE	–109	Set viewport event
GEAWC	–110	Associated workstation type and connection identifier
GESCL	–111	Software clipping
GESWM	–150	Set writing mode
GESLC	–151	Set line cap style
GESLJ	–152	Set line join style
GESEC	–153	Set edge control flag
GESET	–154	Set edge type
GESEW	–155	Set edge width scale factor
GESECI	–156	Set edge color index
GESEI	–157	Set edge index
GESEA	–158	Set aspect source flag (ASF)
GEBTB	–160	Begin transformation block
GEETB	–161	End transformation block
GES SHM	–162	Set segment highlighting method
GESHM	–163	Set highlighting method
GBTB3	–164	Begin transformation block 3
GESER	–200	Set edge representation
GESWT	–202	Set window title
GESRS	–203	Set reset string
GESCS	–204	Set cancel string
GESES	–205	Set enter string
GESIB	–206	Set icon bitmaps
GEIWM	–251	Inquire current writing mode
GEILC	–252	Inquire current line cap style
GEILJ	–253	Inquire current line join style
GEIEA	–254	Inquire current edge attributes
GEIVD	–255	Inquire viewport data
GEIS	–300	Inquire current display speed
GEILEI	–302	Inquire list of edge indexes
GEISE	–303	Inquire segment extent
GEIWID	–304	Inquire window identifiers

(continued on next page)

Table B-1 (Cont.) Constants

Constant	Value	Description
Escape Function Identifiers:		
GEISHM	-305	Inquire segment highlighting method
GEIHM	-306	Inquire highlighting method
GEIPBI	-307	Inquire pasteboard identifier
GEIMBI	-308	Inquire menu bar identifier
GEISHI	-309	Inquire shell identifier
GEILE	-350	Inquire list of available escapes
GEIDS	-351	Inquire default display speed
GEILCJ	-352	Inquire line cap join facilities
GEIEF	-354	Inquire edge facilities
GEIPER	-355	Inquire predefined edge representation
GEIMEB	-356	Inquire maximum number of edge bundles
GEILH	-358	Inquire list of highlighting methods
GIER	-359	Inquire edge representation
GEMNW	-400	Evaluate NDC mapping of a WC point
GEMDN	-401	Evaluate DC mapping of an NDC point
GEMWN	-402	Evaluate WC mapping of an NDC point
GEMND	-403	Evaluate NDC mapping of a DC point
GEIGEX	-404	Inquire extent of GDP
GECON	-440	Set connection identifier string
GESDB	-500	Set double buffering
GESBPM	-501	Set background pixmaps
GEIDBM	-502	Inquire double buffer pixmap
GEIBGM	-503	Inquire background pixmap
Fill Area Control Flags:		
GPLINE	0	Data record polyline control flag
GFILLA	1	Data record fill area control flag
Fill Area Interior Styles:		
GHOLLO	0	Hollow interior
GSOLID	1	Solid interior
GPATTR	2	Patterned interior
GHATCH	3	Hatched interior

(continued on next page)

Constants

Table B–1 (Cont.) Constants

Constant	Value	Description
GDP Bundle Types:		
GPLATT	0	GDP polyline bundle
GPMATT	1	GDP polymarker bundle
GTXATT	2	GDP text bundle
GFAATT	3	GDP fill area bundle
GEDATT	4	GDP edge bundle
GDP Graphics Primitives:		
GGDISP	–100	Disjoint polyline
GGCCP	–101	Center and point on circle
GGC3P	–102	Three points on circle
GGCCR	–103	Center and radius of circle
GGC2PR	–104	Two points and radius of circle
GGAC2P	–106	Center and two points of an arc
GGA3P	–107	Three points of arc
GGACVR	–108	Center and two vector radius of arc
GGA2PR	–109	Two points and radius of the arc
GGACPA	–110	Center point and angle for arc
GGECA	–111	Center and axes of ellipse
GGEFP	–113	Foci and point of ellipse
GGEACA	–114	Center, two vectors of elliptic arc
GGEACP	–115	Center, two vectors of elliptic arc
GGEAFP	–116	Foci, two points on elliptic arc
GGR2P	–125	Rectangle by two points
GGFAS	–332	Fill area set
GGFCCP	–333	Fill circle using center, point
GGFC3P	–334	Fill circle using three points
GGFCCR	–335	Fill circle using center and radius
GGFCPR	–336	Fill circle using two points and radius
GGFACP	–338	Fill arc using center and two points of the arc
GGFA3P	–339	Fill arc using three points
GGFACV	–340	Fill arc using two vectors and radius
GGFAPR	–341	Fill arc using two points and radius of the arc
GGFACA	–342	Fill arc using center, point, angle
GGFECA	–343	Fill ellipse using center, axes
GGFEFP	–345	Fill ellipse using foci, point

(continued on next page)

Table B-1 (Cont.) Constants

Constant	Value	Description
GDP Graphics Primitives:		
GGFEAC	-346	Fill elliptic arc using center, axes, two vectors
GGFEAF	-348	Fill elliptic arc using foci, two points
GGFR2P	-349	Fill rectangle using two points
GGIA	-400	Packed cell array GDP
GKS Level Types:		
GLMA	-3	Minimal output, no input
GLMB	-2	Minimal output, request input
GLMC	-1	Minimal output, full input
GL0A	0	All primitives and attributes, no input
GL0B	1	All primitives and attributes, request input
GL0C	2	All primitives and attributes, full input
GL1A	3	Basic segmentation with full output, no input
GL1B	4	Basic segmentation with full output, request input
GL1C	5	Basic segmentation with full output, full input
GL2A	6	Workstation independent segment storage (WISS), no input
GL2B	7	Workstation independent segment storage (WISS), request input
GL2C	8	Workstation independent segment storage (WISS), full input
GKS Operating States:		
GGKCL	0	GKS closed
GGKOP	1	GKS open
GWSOP	2	At least one workstation open
GWSAC	3	At least one workstation active
GSGOP	4	At least one segment open
HLHSR Identifiers:		
GHRIN	0	No HLHSR processing
GHRIP	1	Painter's algorithm

(continued on next page)

Constants

Table B–1 (Cont.) Constants

Constant	Value	Description
HLHSR Modes:		
GHRMN	0	No HLHSR processing
GHRMPT	1	Painter's algorithm
Highlighting Flags:		
GNORML	0	Primitives not highlighted
GHILIT	1	Primitives highlighted
Highlighting Methods:		
GHMDEF	0	Default highlighting
GHMCOM	1	Complement mode highlighting
GHMCOL	2	Color highlighting
GHMLIN	3	Line highlighting
GHMFIL	4	Fill area highlighting
GHMDDUA	5	Line and fill area highlighting
Implicit Regeneration States:		
GSUPPD	0	Implicit regeneration suppressed
GALLOW	1	Implicit regeneration allowed
Input Classes:		
GNCLAS	0	No input class
GLOCAT	1	Locator input class
GSTROK	2	Stroke input class
GVALUA	3	Valuator input class
GCHOIC	4	Choice input class
GPICK	5	Pick input class
GSTRIN	6	String input class
GVIEW	7	Viewport input class
Input Mode Types:		
GREQU	0	Request mode
GSAMPL	1	Sample mode
GEVENT	2	Event mode

(continued on next page)

Table B-1 (Cont.) Constants

Constant	Value	Description
Input Priority States:		
GHIGHR	0	Relative input priority higher
GLOWER	1	Relative input priority lower
Invalid Index Flags:		
GABSNT	0	Invalid indexes absent
GPRSNT	1	Invalid indexes present
Last Event Flags:		
GNMORE	0	No more simultaneous events
GMORE	1	More simultaneous events
Line Cap Styles:		
GLCBUT	2	Line cap type butted
GLCRND	3	Line cap type rounded
GLCSQR	4	Line cap type square
Line Join Styles:		
GLJMTR	2	Line join type mitre
GLJRND	3	Line join type round
GLJBVL	4	Line join type bevel
Line Types (DEC GKS Implementation Specific):		
GLDS2D	-1	Line type double-dashed dotted
GLDS3D	-2	Line type triple-dashed dotted
GLLGDS	-3	Line type long-dashed
GLLSDS	-4	Line type long-short-dashed
GLSPDS	-5	Line type spaced-dashed
GLSPDT	-6	Line type spaced-dotted
GLDBDT	-7	Line type double-dotted
GLTPDT	-8	Line type triple-dotted
Line Types (Standard):		
GLSOLI	1	Solid
GLDASH	2	Dashed
GLDOT	3	Dotted
GLDASD	4	Dash-dotted

(continued on next page)

Constants

Table B-1 (Cont.) Constants

Constant	Value	Description
Marker Types (DEC GKS Implementation Specific):		
GMSCIR	-1	Marker type solid circle
GMTRU	-2	Marker type hollow up triangle
GMSTRU	-3	Marker type solid up triangle
GMTRD	-4	Marker type hollow down triangle
GMSTRD	-5	Marker type solid down triangle
GMSQ	-6	Marker type hollow square
GMSQ	-7	Marker type solid square
GBT	-8	Marker type hollow bow tie
GMBT	-9	Marker type solid bow tie
GMHG	-10	Marker type hollow hourglass
GMSHG	-11	Marker type solid hourglass
GMDIA	-12	Marker type hollow diamond
GMSDIA	-13	Marker type solid diamond
Marker Types (Standard):		
GPOINT	1	Dot
GPLUS	2	Plus
GAST	3	Asterisk
GOMARK	4	Circle
GXMARK	5	Diagonal cross
New Frame Action States:		
GNO	0	No new frame action on update
GYES	1	New frame action on update
Pick Status Types:		
GNONE	0	No input obtained
GOK	1	Input obtained
GNPICK	2	Input is NOPICK
Projection Types:		
GPARL	0	Parallel projection type
GPERS	1	Perspective projection type

(continued on next page)

Table B–1 (Cont.) Constants

Constant	Value	Description
Regeneration Flag States:		
GPOSTP	0	Implicit regeneration postponed
GPERFO	1	Implicit regeneration performed
Relative input priority:		
GHIGHER	0	Relative input priority higher
GLOWER	1	Relative input priority lower
Request Status Types:		
GNONE	0	No input obtained
GOK	1	Input obtained
Returned Type Values:		
GSET	0	Type of returned value set
GREALI	1	Type of returned value realized
Simultaneous Event Flags:		
GNMORE	0	No more simultaneous events
GMORE	1	More simultaneous events
Text Horizontal Alignment Types:		
GAHNOR	0	Normal
GALEFT	1	Left
GACENT	2	Center
GARITE	3	Right
Text Path Types:		
GRIGHT	0	Right
GLEFT	1	Left
GUP	2	Up
GDOWN	3	Down
Text Precision Types:		
GSTRP	0	String
GCHARP	1	Character
GSTRKP	2	Stroke

(continued on next page)

Constants

Table B-1 (Cont.) Constants

Constant	Value	Description
Text Vertical Alignment Types:		
GAVNOR	0	Normal
GATOP	1	Top
GACAP	2	Cap
GAHALF	3	Half
GABASE	4	Base
GABOTT	5	Bottom
Visibility Flags:		
GINVIS	0	Set to invisible
GVISI	1	Set to visible
Update States:		
GNPEND	0	Not pending
GPEND	1	Pending
Workstation Category Types:		
GOUTPT	0	Output
GINPUT	1	Input
GOUTIN	2	Out/In
GWISS	3	Workstation independent segment storage (WISS)
GMO	4	Metafile output
GMI	5	Metafile input
Workstation Class Types:		
GVECTR	0	Vector
GRASTR	1	Raster
GOTHWK	2	Other device
Workstation Color Availability States:		
GMONOC	0	Monochrome
GCOLOR	1	Color
Workstation States:		
GINACT	0	Inactive
GACTIV	1	Active

(continued on next page)

Table B-1 (Cont.) Constants

Constant	Value	Description
Workstation Types:		
GWSDEF	0	Default workstation type
GMOUTP	2	DEC GKS output metafile
GMINPT	3	DEC GKS input metafile
GWSWIS	5	Workstation independent segment storage (WISS)
GCGMO	7	CGM output metafile
GVTO	10	Digital VT125 (output only)
GV125C	11	Digital VT125 with color option
GV125B	12	Digital VT125 (monochrome)
GV240C	13	Digital VT240 with color option
GV240B	14	Digital VT240 (monochrome)
GLCP01	15	Digital LCG01 printer
GLCG01	15	Digital LCG01 printer
GV330	16	Digital VT330 (monochrome)
GV340	17	Digital VT340 (color)
GLA34	31	Digital LA34 with graphics option
GLA100	31	Digital LA100
GLA50	32	Digital LA50 with 2:1 aspect ratio
GLA210	34	Digital LA210
GLA75	35	Digital LA75
GLN03P	38	Digital LN03 PLUS
GVSI	42	Digital VAXstation I
GVSII	41	Digital VAXstation II
GVSGPX	41	Digital VAXstation II/GPX
GV2000	41	Digital VAXstation 2000
GV3200	41	Digital VAXstation 3200
GV3500	41	Digital VAXstation 3500
GLVPA	51	Digital LVP16 color plotter (8 1/2 by 11 inches)
GLVPB	52	Digital LVP16 color plotter (11 by 17 inches)
GHP747	51	Hewlett-Packard HP7475
GHP755	53	Hewlett-Packard HP7550 pen plotter
GHP758	54	Hewlett-Packard HP7580 pen plotter
GLGMPS	55	LaserGraphics film recorder
GH7585	56	Hewlett-Packard HP7585 pen plotter
GPTSC	61	PostScript graphics handler
GTEKO	70	Tektronix 4014 (output only)

(continued on next page)

Constants

Table B-1 (Cont.) Constants

Constant	Value	Description
Workstation Types:		
GT4014	72	Tektronix 4014
GT410	80	Tektronix 4107 (output only)
GT4107	82	Tektronix 4107
GT2070	83	Tektronix 2070
GT4207	84	Tektronix 4207
GT1280	85	Tektronix 1280
GT4128	86	Tektronix 4128
GT1290	87	Tektronix 1290
GT4129	88	Tektronix 4129
GVS50	87	Tektronix GVS50
GVS500	88	Tektronix GVS500
GLJ250	91	Digital LJ250 90 DPI
GLJ25X	92	Digital LJ250 180 DPI
GDECWO	210	DECwindows output
GDECW	211	DECwindows
GDECWD	212	DECwindows drawable
GDECWW	213	DECwindows drawable
GPEXO	220	PEX output
GPEX	221	PEX
GPEXD	222	PEX drawable
GPEXW	223	PEX drawable
GMOTFO	230	Motif output
GMOTIF	231	Motif
GMOTFD	232	Motif drawable
GMOTFW	233	Motif drawable
GMPEXO	240	OSF/Motif® output
GMPEX	241	OSF/Motif
GMPEXD	242	OSF/Motif drawable
GMPEXW	243	OSF/Motif drawable
GDDIF	250	DDIF output
Writing Modes:		
GWMCMT	2	Complement writing mode
GWMERS	3	Erase writing mode
GWMOVY	4	Overlay writing mode

Program Example

Insert tabbed divider here. Then discard this sheet.



Program Example

This appendix provides information on binding-specific code examples contained in this manual.

Table C-1 lists the code examples available throughout the binding. The code examples are listed alphabetically by function.

Table C-1 Binding-Specific Code Examples

Function	Example
ACCUMULATE TRANSFORMATION MATRIX	Example 7-1
ACTIVATE WORKSTATION	Example 4-1
ASSOCIATE SEGMENT WITH WORKSTATION	Example 8-1
AWAIT EVENT	Example 9-1
CELL ARRAY	Example 5-1
CLEAR WORKSTATION	Example 4-1
CLOSE GKS	Example 4-1
CLOSE SEGMENT	Example 8-1
CLOSE WORKSTATION	Example 4-1
COPY SEGMENT TO WORKSTATION	Example 8-1
CREATE SEGMENT	Example 8-1
DEACTIVATE WORKSTATION	Example 4-1
EMERGENCY CLOSE GKS	Example 12-1
ERROR HANDLING	Example 12-1
ERROR LOGGING	Example 12-1
ESCAPE	Example 4-2
EVALUATE TRANSFORMATION MATRIX	Example 7-2
FILL AREA	Example 6-1
GENERALIZED DRAWING PRIMITIVE	Example 5-2
GET LOCATOR	Example 9-1
INITIALIZE LOCATOR	Example 9-1
INITIALIZE PICK	Example 9-2
INITIALIZE STRING	Example 9-3
INITIALIZE VALUATOR	Example 9-4
INQUIRE DISPLAY SPACE SIZE	Example 7-4

(continued on next page)

Program Example

Table C-1 (Cont.) Binding-Specific Code Examples

Function	Example
INQUIRE DYNAMIC MODIFICATION OF WORKSTATION ATTRIBUTES	Example 7-4
INQUIRE LOCATOR DEVICE STATE	Example 9-1
INQUIRE PICK DEVICE STATE	Example 9-2
INQUIRE STRING DEVICE STATE	Example 9-3
INQUIRE VALUATOR DEVICE STATE	Example 9-4
INQUIRE WORKSTATION CONNECTION AND TYPE	Example 4-2
INSERT SEGMENT	Example 8-2
MESSAGE	Example 9-1
OPEN GKS	Example 4-1
OPEN WORKSTATION	Example 4-1
POLYLINE	Example 6-3
POLYMARKER	Example 6-4
REDRAW ALL SEGMENTS ON WORKSTATION	Example 8-1
REQUEST STRING	Example 9-3
SAMPLE PICK	Example 9-2
SAMPLE VALUATOR	Example 9-4
SELECT NORMALIZATION TRANSFORMATION	Example 7-3
SET ASPECT SOURCE FLAGS	Example 6-2
SET CHARACTER HEIGHT	Example 6-4
SET CLIPPING INDICATOR	Example 7-3
SET COLOUR REPRESENTATION	Example 6-1
SET DEFERRAL STATE	Example 5-1
SET DETECTABILITY	Example 9-2
SET FILL AREA COLOUR INDEX	Example 6-1
SET FILL AREA INDEX	Example 6-2
SET FILL AREA INTERIOR STYLE	Example 6-1
SET FILL AREA REPRESENTATION	Example 6-2
SET HIGHLIGHTING	Example 8-3
SET LINETYPE	Example 6-3
SET LOCATOR MODE	Example 9-1
SET MARKER TYPE	Example 6-4
SET PICK IDENTIFIER	Example 9-2
SET PICK MODE	Example 9-2
SET POLYMARKER COLOUR INDEX	Example 6-4
SET SEGMENT TRANSFORMATION	Example 7-1
SET TEXT ALIGNMENT	Example 6-4
SET TEXT PATH	Example 6-4

(continued on next page)

Table C-1 (Cont.) Binding-Specific Code Examples

Function	Example
SET VALUATOR MODE	Example 9-4
SET VIEWPORT	Example 7-3
SET WINDOW	Example 7-3
SET WORKSTATION VIEWPORT	Example 7-4
TEXT	Example 6-4
UPDATE WORKSTATION	Example 4-1

Function Identifiers

Insert tabbed divider here. Then discard this sheet.



D

Function Identifiers

Table D–1 lists the defined FORTRAN binding function names in the first column and their corresponding error constants in the second column. These are the same as the function names given in the language binding description, except the initial character G is replaced by E. The function names are arranged alphabetically.

Table D–1 FORTRAN Binding Function Identifiers

Function Name	Error Constant
ACCUMULATE TRANSFORMATION MATRIX	EACTM
ACCUMULATE TRANSFORMATION MATRIX 3	EACTM3
ACTIVATE WORKSTATION	EACWK
ASSOCIATE SEGMENT WITH WORKSTATION	EASGWK
AWAIT EVENT	EWAIT
CELL ARRAY	ECA
CELL ARRAY 3	ECA3
CLEAR WORKSTATION	ECLRWK
CLOSE GKS	ECLKS
CLOSE SEGMENT	ECLSG
CLOSE WORKSTATION	ECLWK
COPY SEGMENT TO WORKSTATION	ECSGWK
CREATE SEGMENT	ECRSG
DEACTIVATE WORKSTATION	EDAWK
DELETE SEGMENT	EDSG
DELETE SEGMENT FROM WORKSTATION	EDSGWK
EMERGENCY CLOSE GKS	EECLKS
ERROR HANDLING	EERHND
ERROR LOGGING	EERLOG
ESCAPE	EESC
EVALUATE TRANSFORMATION MATRIX	EEVTM
EVALUATE TRANSFORMATION MATRIX 3	EEVTM3
EVALUATE VIEW MAPPING MATRIX 3	EEVMM3
EVALUATE VIEW ORIENTATION MATRIX 3	EEVOM3
FILL AREA	EFA

(continued on next page)

Function Identifiers

Table D–1 (Cont.) FORTRAN Binding Function Identifiers

Function Name	Error Constant
FILL AREA 3	EFA3
FILL AREA SET	EFAS
FILL AREA SET 3	EFAS3
FLUSH DEVICE EVENTS	EFLUSH
GENERALIZED DRAWING PRIMITIVE	EGDP
GENERALIZED DRAWING PRIMITIVE 3	EGDP3
GET CHOICE	EGTCH
GET ITEM TYPE FROM GKSM	EGTITM
GET LOCATOR	EGTLC
GET LOCATOR 3	EGTLC3
GET PICK	EGTPK
GET STRING	EGTST
GET STROKE	EGTSK
GET STROKE 3	EGTSK3
GET VALUATOR	EGTVL
INITIALIZE CHOICE	EINCH
INITIALIZE CHOICE 3	EINCH3
INITIALIZE LOCATOR	EINLC
INITIALIZE LOCATOR 3	EINLC3
INITIALIZE PICK	EINPK
INITIALIZE PICK 3	EINPK3
INITIALIZE STRING	EINST
INITIALIZE STRING 3	EINST3
INITIALIZE STROKE	EINSK
INITIALIZE STROKE 3	EINSK3
INITIALIZE VALUATOR	EINVL
INITIALIZE VALUATOR 3	EINVL3
INQUIRE ASPECT SOURCE FLAGS	EQASF
INQUIRE ASPECT SOURCE FLAGS 3	EQASF3
INQUIRE CHARACTER BASE VECTOR	EQCHB
INQUIRE CHARACTER EXPANSION FACTOR	EQCHXP
INQUIRE CHARACTER HEIGHT	EQCHH
INQUIRE CHARACTER SPACING	EQCHSP
INQUIRE CHARACTER UP VECTOR	EQCHUP
INQUIRE CHARACTER WIDTH	EQCHW
INQUIRE CHOICE DEVICE STATE	EQCHS
INQUIRE CHOICE DEVICE STATE 3	EQCHS3
INQUIRE CLIPPING	EQCLIP

(continued on next page)

Table D-1 (Cont.) FORTRAN Binding Function Identifiers

Function Name	Error Constant
INQUIRE CLIPPING 3	EQCLP3
INQUIRE COLOUR FACILITIES	EQCF
INQUIRE COLOUR MODEL	EQCMD
INQUIRE COLOUR MODEL FACILITIES	EQCMDF
INQUIRE COLOUR REPRESENTATION	EQCR
INQUIRE CURRENT EDGE COLOUR INDEX	EQEDCI
INQUIRE CURRENT EDGE FLAG	EQEDFG
INQUIRE CURRENT EDGE INDEX	EQEDI
INQUIRE CURRENT EDGETYPE	EQEDT
INQUIRE CURRENT EDGEWIDTH SCALE	EQEWSC
INQUIRE CURRENT HLHSR IDENTIFIER VALUE	EQHRIV
INQUIRE CURRENT NORMALIZATION TRANSFORMATION NUMBER	EQCNTN
INQUIRE CURRENT PATTERN REFERENCE POINT AND VECTORS	EQPRPV
INQUIRE CURRENT PICK IDENTIFIER VALUE	EQPKID
INQUIRE CURRENT VIEW INDEX	EQVWI
INQUIRE DEFAULT CHOICE DEVICE DATA	EQDCH
INQUIRE DEFAULT CHOICE DEVICE DATA 3	EQDCH3
INQUIRE DEFAULT DEFERRAL STATE VALUES	EQDDS
INQUIRE DEFAULT LOCATOR DEVICE DATA	EQDLC
INQUIRE DEFAULT LOCATOR DEVICE DATA 3	EQDLC3
INQUIRE DEFAULT PICK DEVICE DATA	EQDPK
INQUIRE DEFAULT PICK DEVICE DATA 3	EQDPK3
INQUIRE DEFAULT STRING DEVICE DATA	EQDST
INQUIRE DEFAULT STRING DEVICE DATA 3	EQDST3
INQUIRE DEFAULT STROKE DEVICE DATA	EQDSK
INQUIRE DEFAULT STROKE DEVICE DATA 3	EQDSK3
INQUIRE DEFAULT VALUATOR DEVICE DATA	EQDVL
INQUIRE DEFAULT VALUATOR DEVICE DATA 3	EQDVL3
INQUIRE DISPLAY SPACE SIZE	EQDSP
INQUIRE DISPLAY SPACE SIZE 3	EQDSP3
INQUIRE DYNAMIC MODIFICATION OF SEGMENT ATTRIBUTES	EQDSGA
INQUIRE DYNAMIC MODIFICATION OF WORKSTATION ATTRIBUTES	EQDWKA
INQUIRE DYNAMIC MODIFICATION OF WORKSTATION ATTRIBUTES 3	EQDWA3
INQUIRE EDGE FACILITIES	EQEDF

(continued on next page)

Function Identifiers

Table D–1 (Cont.) FORTRAN Binding Function Identifiers

Function Name	Error Constant
INQUIRE EDGE REPRESENTATION	EQEDR
INQUIRE FILL AREA COLOUR INDEX	EQFACI
INQUIRE FILL AREA FACILITIES	EQFAF
INQUIRE FILL AREA INDEX	EQFAI
INQUIRE FILL AREA INTERIOR STYLE	EQFAIS
INQUIRE FILL AREA REPRESENTATION	EQFAR
INQUIRE FILL AREA STYLE INDEX	EQFASI
INQUIRE GENERALIZED DRAWING PRIMITIVE	EQGDP
INQUIRE GENERALIZED DRAWING PRIMITIVE 3	EQGDP3
INQUIRE HLHSR FACILITIES	EQHRF
INQUIRE HLHSR MODE	EQHRM
INQUIRE INPUT QUEUE OVERFLOW	EQIQOV
INQUIRE LEVEL OF GKS	EQLVKS
INQUIRE LINETYPE	EQLN
INQUIRE LINEWIDTH SCALE FACTOR	EQLWSC
INQUIRE LIST OF AVAILABLE GENERALIZED DRAWING PRIMITIVES	EQEGDP
INQUIRE LIST OF AVAILABLE GENERALIZED DRAWING PRIMITIVES 3	EQEGD3
INQUIRE LIST OF AVAILABLE WORKSTATION TYPES	EQEWK
INQUIRE LIST OF COLOUR INDICES	EQECI
INQUIRE LIST OF EDGE INDICES	EQEEDI
INQUIRE LIST OF FILL AREA INDICES	EQEFAI
INQUIRE LIST OF NORMALIZATION TRANSFORMATION NUMBERS	EQENTN
INQUIRE LIST OF PATTERN INDICES	EQEPAI
INQUIRE LIST OF POLYLINE INDICES	EQEPLI
INQUIRE LIST OF POLYMARKER INDICES	EQEPMI
INQUIRE LIST OF TEXT INDICES	EQETXI
INQUIRE LIST OF VIEW INDICES	EQEVWI
INQUIRE LOCATOR DEVICE STATE	EQLCS
INQUIRE LOCATOR DEVICE STATE 3	EQLCS3
INQUIRE MARKER SIZE SCALE FACTOR	EQMKSC
INQUIRE MARKER TYPE	EQMK
INQUIRE MAXIMUM LENGTH OF WORKSTATION STATE TABLES	EQLWK
INQUIRE MAXIMUM LENGTH OF WORKSTATION STATE TABLES 3	EQLWK3
INQUIRE MAXIMUM NORMALIZATION TRANSFORMATION NUMBER	EQMNTN

(continued on next page)

Table D-1 (Cont.) FORTRAN Binding Function Identifiers

Function Name	Error Constant
INQUIRE MORE SIMULTANEOUS EVENTS	EQSIM
INQUIRE NAME OF OPEN SEGMENT	EQOPSG
INQUIRE NORMALIZATION TRANSFORMATION	EQNT
INQUIRE NORMALIZATION TRANSFORMATION 3	EQNT3
INQUIRE NUMBER OF AVAILABLE LOGICAL INPUT DEVICES	EQLI
INQUIRE NUMBER OF SEGMENT PRIORITIES SUPPORTED	EQSGP
INQUIRE OPERATING STATE VALUE	EQOPS
INQUIRE PATTERN FACILITIES	EQPAF
INQUIRE PATTERN REFERENCE POINT	EQPARF
INQUIRE PATTERN REPRESENTATION	EQPAR
INQUIRE PATTERN SIZE	EQPA
INQUIRE PICK DEVICE STATE	EQPKS
INQUIRE PICK DEVICE STATE 3	EQPKS3
INQUIRE PIXEL	EQPX
INQUIRE PIXEL ARRAY	EQPXA
INQUIRE PIXEL ARRAY DIMENSIONS	EQPXAD
INQUIRE POLYLINE COLOUR INDEX	EQPLCI
INQUIRE POLYLINE FACILITIES	EQPLF
INQUIRE POLYLINE INDEX	EQPLI
INQUIRE POLYLINE REPRESENTATION	EQPLR
INQUIRE POLYMARKER COLOUR INDEX	EQPMCI
INQUIRE POLYMARKER FACILITIES	EQPMF
INQUIRE POLYMARKER INDEX	EQPMI
INQUIRE POLYMARKER REPRESENTATION	EQPMR
INQUIRE PREDEFINED COLOUR REPRESENTATION	EQPCR
INQUIRE PREDEFINED EDGE REPRESENTATION	EQPEDR
INQUIRE PREDEFINED FILL AREA REPRESENTATION	EQPFAR
INQUIRE PREDEFINED PATTERN REPRESENTATION	EQPPAR
INQUIRE PREDEFINED POLYLINE REPRESENTATION	EQPPLR
INQUIRE PREDEFINED POLYMARKER REPRESENTATION	EQPPMR
INQUIRE PREDEFINED TEXT REPRESENTATION	EQPTXR
INQUIRE PREDEFINED VIEW REPRESENTATION	EQPVWR
INQUIRE SEGMENT ATTRIBUTES	EQSGA
INQUIRE SEGMENT ATTRIBUTES 3	EQSGA3
INQUIRE SET OF ACTIVE WORKSTATIONS	EQACWK
INQUIRE SET OF ASSOCIATED WORKSTATIONS	EQASWK
INQUIRE SET OF OPEN WORKSTATIONS	EQOPWK

(continued on next page)

Function Identifiers

Table D–1 (Cont.) FORTRAN Binding Function Identifiers

Function Name	Error Constant
INQUIRE SET OF SEGMENT NAMES IN USE	EQSGUS
INQUIRE SET OF SEGMENT NAMES ON WORKSTATION	EQSGWK
INQUIRE STRING DEVICE STATE	EQSTS
INQUIRE STRING DEVICE STATE 3	EQSTS3
INQUIRE STROKE DEVICE STATE	EQSKS
INQUIRE STROKE DEVICE STATE 3	EQSKS3
INQUIRE TEXT ALIGNMENT	EQTXAL
INQUIRE TEXT COLOUR INDEX	EQTXCI
INQUIRE TEXT EXTENT	EQTXX
INQUIRE TEXT EXTENT (FORTRAN–77 Subset)	EQTXXS
INQUIRE TEXT EXTENT 3	EQTX3
INQUIRE TEXT EXTENT 3 (FORTRAN–77 Subset)	EQTX3S
INQUIRE TEXT FACILITIES	EQTXF
INQUIRE TEXT FONT AND PRECISION	EQTXFP
INQUIRE TEXT INDEX	EQTXI
INQUIRE TEXT PATH	EQTXP
INQUIRE TEXT REPRESENTATION	EQTXR
INQUIRE VALUATOR DEVICE STATE	EQVLS
INQUIRE VALUATOR DEVICE STATE 3	EQVLS3
INQUIRE VIEW FACILITIES	EQVWF
INQUIRE VIEW REPRESENTATION 3	EQVWR
INQUIRE WORKSTATION CATEGORY	EQWKCA
INQUIRE WORKSTATION CLASSIFICATION	EQWKCL
INQUIRE WORKSTATION CONNECTION AND TYPE	EQWKC
INQUIRE WORKSTATION DEFERRAL AND UPDATE STATES	EQWKDU
INQUIRE WORKSTATION MAXIMUM NUMBERS	EQWKM
INQUIRE WORKSTATION STATE	EQWKS
INQUIRE WORKSTATION TRANSFORMATION	EQWKT
INQUIRE WORKSTATION TRANSFORMATION 3	EQWKT3
INSERT SEGMENT	EINSG
INSERT SEGMENT 3	EINSG3
INTERPRET ITEM	EIITM
MESSAGE	EMSG
MESSAGE (FORTRAN–77 Subset)	EMSGS
OPEN GKS	EOPKS
OPEN WORKSTATION	EOPWK
PACK DATA RECORD	EPREC
POLYLINE	EPL

(continued on next page)

Table D-1 (Cont.) FORTRAN Binding Function Identifiers

Function Name	Error Constant
POLYLINE 3	EPL3
POLYMARKER	EPM
POLYMARKER 3	EPM3
READ ITEM FROM GKSM	ERDITM
REDRAW ALL SEGMENTS ON WORKSTATION	ERSGWK
RENAME SEGMENT	ERENSG
REQUEST CHOICE	ERQCH
REQUEST LOCATOR	ERQLC
REQUEST LOCATOR 3	ERQLC3
REQUEST PICK	ERQPK
REQUEST STRING	ERQST
REQUEST STROKE	ERQSK
REQUEST STROKE 3	ERQSK3
REQUEST VALUATOR	ERQVL
SAMPLE CHOICE	ESMCH
SAMPLE LOCATOR	ESMLC
SAMPLE LOCATOR 3	ESMLC3
SAMPLE PICK	ESMPK
SAMPLE STRING	ESMST
SAMPLE STROKE	ESMSK
SAMPLE STROKE 3	ESMSK3
SAMPLE VALUATOR	ESMVL
SELECT NORMALIZATION TRANSFORMATION	ESELNT
SET ASPECT SOURCE FLAGS	ESASF
SET ASPECT SOURCE FLAGS 3	ESASF3
SET CHARACTER EXPANSION FACTOR	ESCHXP
SET CHARACTER HEIGHT	ESCHH
SET CHARACTER SPACING	ESCHSP
SET CHARACTER UP VECTOR	ESCHUP
SET CHOICE MODE	ESCHM
SET CLIPPING INDICATOR	ESCLIP
SET COLOUR MODEL	ESCMD
SET COLOUR REPRESENTATION	ESCR
SET DEFERRAL STATE	ESDS
SET DETECTABILITY	ESDTEC
SET EDGE COLOUR INDEX	ESEDCI
SET EDGE FLAG	ESEDFG
SET EDGE INDEX	ESEDI

(continued on next page)

Function Identifiers

Table D–1 (Cont.) FORTRAN Binding Function Identifiers

Function Name	Error Constant
SET EDGE REPRESENTATION	ESEDR
SET EDGETYPE	ESEDT
SET EDGEWIDTH SCALE FACTOR	ESEWSC
SET FILL AREA COLOUR INDEX	ESFACI
SET FILL AREA INDEX	ESFAI
SET FILL AREA INTERIOR STYLE	ESFAIS
SET FILL AREA REPRESENTATION	ESFAR
SET FILL AREA STYLE INDEX	ESFASI
SET HIGHLIGHTING	ESHLIT
SET HLHSR IDENTIFIER	ESHRID
SET HLHSR MODE	ESHRM
SET LINETYPE	ESLN
SET LINEWIDTH SCALE FACTOR	ESLWSC
SET LOCATOR MODE	ESLCM
SET MARKER SIZE SCALE FACTOR	ESMKSC
SET MARKER TYPE	ESMK
SET PATTERN REFERENCE POINT	ESPARF
SET PATTERN REFERENCE POINT AND VECTORS	ESPRPV
SET PATTERN REPRESENTATION	ESPAR
SET PATTERN SIZE	ESPA
SET PICK IDENTIFIER	ESPKID
SET PICK MODE	ESPKM
SET POLYLINE COLOUR INDEX	ESPLCI
SET POLYLINE INDEX	ESPLI
SET POLYLINE REPRESENTATION	ESPLR
SET POLYMARKER COLOUR INDEX	ESPMCI
SET POLYMARKER INDEX	ESPMI
SET POLYMARKER REPRESENTATION	ESPMR
SET SEGMENT PRIORITY	ESSGP
SET SEGMENT TRANSFORMATION	ESSGT
SET SEGMENT TRANSFORMATION 3	ESSGT3
SET STRING MODE	ESSTM
SET STROKE MODE	ESSKM
SET TEXT ALIGNMENT	ESTXAL
SET TEXT COLOUR INDEX	ESTXCI
SET TEXT FONT AND PRECISION	ESTXFP
SET TEXT INDEX	ESTXI
SET TEXT PATH	ESTXP

(continued on next page)

Table D-1 (Cont.) FORTRAN Binding Function Identifiers

Function Name	Error Constant
SET TEXT REPRESENTATION	ESTXR
SET VALUATOR MODE	ESVLM
SET VIEW INDEX	ESVWI
SET VIEW REPRESENTATION 3	ESVWR3
SET VIEW TRANSFORMATION INPUT PRIORITY	ESVTIP
SET VIEWPORT	ESVP
SET VIEWPORT 3	ESV3
SET VIEWPORT INPUT PRIORITY	ESVPIP
SET VISIBILITY	ESVIS
SET WINDOW	ESWN
SET WINDOW 3	ESW3
SET WORKSTATION VIEWPORT	ESWKVP
SET WORKSTATION VIEWPORT 3	ESWKV3
SET WORKSTATION WINDOW	ESWKWN
SET WORKSTATION WINDOW 3	ESWKW3
TEXT	ETX
TEXT (FORTRAN-77 Subset)	ETXS
TEXT 3	ETX3
TEXT 3 (FORTRAN-77 Subset)	ETXS3
UNPACK DATA RECORD	EUREC
UPDATE WORKSTATION	EUWK
WRITE ITEM TO GKSM	EWITM

Initial Attribute Values

Insert tabbed divider here. Then discard this sheet.



Initial Attribute Values

This appendix lists the initial values of all output attributes and normalization transformation settings according to the following categories:

- Polyline attributes
- Polymarker attributes
- Text attributes
- Fill area attributes
- Fill area set attributes
- Segment attributes
- Normalization transformation settings

E.1 Initial Polyline Attributes

This section lists the initial values for the polyline attributes.

Attribute	Initial Value	Description
Polyline index	1	Polyline bundle number 1
Line type	GLSOLI	Solid line
Line width	1.0	Minimum width
Line color index	1	Workstation dependent value
Line type ASF	GINDIV	Use current line type
Line width ASF	GINDIV	Use current line width
Line color index ASF	GINDIV	Use current line color index

Initial Attribute Values

E.2 Initial Polymarker Attributes

E.2 Initial Polymarker Attributes

This section lists the initial values for the polymarker attributes.

Attribute	Initial Value	Description
Polymarker index	1	Polymarker bundle number 1
Marker type	GAST	Asterisk for marker
Marker size	1.0	Nominal size
Color index	1	Workstation dependent value
Marker type ASF	GINDIV	Use current marker type
Marker size ASF	GINDIV	Use current marker size
Marker color index ASF	GINDIV	Use current marker color index

E.3 Initial Text Attributes

This section lists the initial values for the text attributes.

Attribute	Initial Value	Description
Text index	1	Text bundle number 1
Text font and precision	1 GSTRP	Hardware font 1, string precision
Character expansion factor	1.0	Width-to-height ratio from font file
Character spacing	0.0	Adjacent character bodies
Color index	1	Workstation dependent value
Text font and precision ASF	GINDIV	Use current font and precision
Character expansion factor ASF	GINDIV	Use current width and height ratio
Character spacing ASF	GINDIV	Use current character space
Text color index ASF	GINDIV	Use current text color index
Character height	0.01	Capital letters at 0.01 WC units
Character up vector	(0, 1)	Up vector parallel to y-axis in WC units
Text path	GRIGHT	Right angle clockwise from up vector
Text alignment	GAHNOR GAVNOR	Natural alignment with respect to text path

E.4 Initial Fill Area Attributes

This section lists the initial values for the fill area attributes.

Attribute	Initial Value	Description
Fill area index	1	Fill area bundle number 1
Interior style	GHOLLO	Boundary of polygonal area
Style index	1	Workstation-dependent pattern or hatch style
Color index	1	Workstation-dependent value
Interior style ASF	GINDIV	Use current interior style
Style index ASF	GINDIV	Use current pattern or hatch style
Color index ASF	GINDIV	Use current fill area color index
Pattern size	1.0,1.0	Unit square in WC units
Pattern reference point	(0.0, 0.0, 0.0)	Pattern starting point in WC units

E.5 Initial Fill Area Set Attributes

This section lists the initial values for the fill area set attributes.

Attribute	Initial Value	Description
Fill area index	1	Fill area bundle number 1
Interior style	GHOLLO	Boundary of polygonal area
Fill area style index	1	Workstation-dependent pattern or hatch style
Fill area color index	1	Workstation-dependent value
Fill area interior style ASF	GINDIV	Use current interior style
Fill area style index ASF	GINDIV	Use current pattern or hatch style
Fill area color index ASF	GINDIV	Use current fill area color index
Pattern size	1.0,1.0	Unit square in WC units
Pattern reference point	(0.0, 0.0, 0.0)	Pattern starting point in WC units
Edge index	1	Use edge bundle number 1
Edge flag	GOFF	Edge not drawn
Edge type	GLSOLI	Solid edge
Edge width scale factor	1.0	Minimum width
Edge color index	1	Workstation-dependent value
Edge flag ASF	GINDIV	Use current edge flag

Initial Attribute Values

E.5 Initial Fill Area Set Attributes

Attribute	Initial Value	Description
Edge type ASF	GINDIV	Use current edge type
Edge width scale factor ASF	GINDIV	Use current edge width scale factor
Edge color index ASF	GINDIV	Use current edge color index

E.6 Initial Segment Attributes

This section lists the initial values for the segment attributes.

Attribute	Initial Value	Description
Transformation number	0	The identity transformation presents the segment as stored in NDC space.
Visibility	GVISI	The segment is visible.
Highlighting	GNORML	The segment is not highlighted.
Segment priority	0.0	The segment has the lowest priority.
Detectability	GUNDET	The segment is undetectable.

The default segment transformation is called the identity transformation. The identity transformation uses a 3×4 matrix as follows:

$$\begin{bmatrix} 1.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 1.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 1.0 & 0.0 \end{bmatrix}$$

E.7 Initial Normalization Transformation Settings

The initial normalization transformation number is the value 0.

The initial viewport input priority is in sequential order from the values 0 to 255, with transformation number 0 the highest and 255 the lowest.

The default normalization window and viewport limits are cubic, begin with a lower left corner point of (0.0, 0.0, 0.0), and extend to the value (1.0, 1.0, 1.0) on the X, Y, and Z axes.

Initially, clipping is enabled (GCLIP) at the normalization viewport limit.

Differences in GKS

Implementations

Insert tabbed divider here. Then discard this sheet.



Differences in GKS Implementations

The GKS standard omits a lot details about how to implement certain functionality. This allows different GKS implementations the freedom to adapt to different environments and requirements.

The allowed differences in GKS implementations can be divided into two groups:

- Global differences
- Workstation-dependent differences

This appendix lists the global differences allowed by the standard, and the corresponding DEC GKS implementation. Workstation-dependent differences are listed in the *Device Specifics Reference Manual for DEC GKS and DEC PHIGS*.

F.1 Global Differences

The global differences relate to the implementation as a whole. The allowable global differences are listed in Table F–1, along with the DEC GKS-specific values. The differences are grouped into three main categories: functional scope, capacity, and miscellaneous.

Table F–1 Global Differences

Description	DEC GKS Value
Functional Scope	
GKS level	DEC GKS is level 2c.
Capacity	
Number of available workstation types	There are 63 workstation types, not including the default type. The workstation types include MO (metafile output), MI (metafile input), and WISS (workstation-independent segment storage).
List of available workstation types	See the <i>Device Specifics Reference Manual for DEC GKS and DEC PHIGS</i> for the list of supported workstation types.
Maximum number of simultaneously open workstations	The maximum is 256.
Maximum number of simultaneously active workstations	The maximum is 256.

(continued on next page)

Differences in GKS Implementations

F.1 Global Differences

Table F–1 (Cont.) Global Differences

Description	DEC GKS Value
Capacity	
Maximum number of workstations associated with a segment	The maximum is 256.
Maximum normalization transformation number	The maximum is 255.
Number of simultaneously definable segments (per workstation)	The maximum number is 128.
Maximum size of input queue	The maximum size is 256 characters.
Number of fonts available	DEC GKS has 25 device-independent fonts, in addition to device-dependent fonts. See the <i>Device Specifics Reference Manual for DEC GKS and DEC PHIGS</i> for information on fonts.
Number of GDPs	DEC GKS has 31 GDPs. See the <i>Device Specifics Reference Manual for DEC GKS and DEC PHIGS</i> for more information on the supported GDPs.
Number of escapes	DEC GKS has 61 escape functions. See the <i>Device Specifics Reference Manual for DEC GKS and DEC PHIGS</i> for more information on the supported escapes.
Miscellaneous	
Initial setting of ASFs	DEC GKS initially sets all the ASFs to INDIVIDUAL.
EMERGENCY CLOSE GKS behavior	The EMERGENCY CLOSE GKS function closes the segment, if it is open, deactivates any active workstations, closes all open workstations, then closes GKS.
Actions performed on parameters of inquiry functions if the information is unavailable	The information returned in a DEC GKS inquiry function is only valid if the value of the error indicator is 0.
Metafile format used by each workstation type of category MO	See the <i>DEC GKS User's Guide</i> for a detailed description of the metafile format.
Font definitions	See the <i>Device Specifics Reference Manual for DEC GKS and DEC PHIGS</i> for complete information on fonts supported by DEC GKS.

Index

A

Access type, *Part 1*, 1–6
ACCUMULATE TRANSFORMATION MATRIX 3
 function, *Part 1*, 7–13
ACCUMULATE TRANSFORMATION MATRIX
 function, *Part 1*, 7–10
 example, *Part 1*, 7–38
Accumulating
 segment transformations, *Part 1*, 8–9
Action pending states
 list of, *Part 2*, B–12
ACTIVATE WORKSTATION function, *Part 1*, 4–9
 example, *Part 1*, 4–30
Activating workstations, *Part 1*, 4–5
Alignment
 text, *Part 1*, 6–51
Angles
 See also Segments
 rotation, *Part 1*, 8–8
ANSI
 CGM standard, *Part 1*, 10–1
 GKS standard, *Part 1*, 1–1
Appearance
 attributes, *Part 1*, 6–1
Arc types
 list of, *Part 2*, B–1
Arguments
 characteristics of, *Part 1*, 1–5
 descriptions, *Part 1*, 1–5
 inquiry error status, *Part 2*, 11–3
 inquiry value type argument, *Part 2*, 11–4
Arrays
 color index, *Part 1*, 5–4, 5–6
ASAP, *Part 1*, 1–7
ASFs, *Part 1*, 6–4
Aspect ratio
 See also Transformations
Aspect source flags
 list of, *Part 2*, B–1
ASSOCIATE SEGMENT WITH WORKSTATION
 function, *Part 1*, 8–11
 example, *Part 1*, 8–29
Association
 See also Segments
 segments, *Part 1*, 8–3
 windows and viewports, *Part 1*, 7–5

Asynchronous input, *Part 1*, 9–16
 See also Input
Attribute control flags
 list of, *Part 2*, B–1
Attribute functions, *Part 1*, 6–1 to 6–72
 introduction to, *Part 1*, 6–1 to 6–5
Attributes, *Part 1*, 1–3
 attribute source flags, *Part 1*, 6–4
 bound to primitives, *Part 1*, 6–1
 bundled, *Part 1*, 6–3
 fill area, *Part 1*, 6–2
 fill area set, *Part 1*, 6–3
 GDPs, *Part 1*, 6–3
 geometric and nongeometric, *Part 1*, 6–1
 implicit regenerations, *Part 1*, 6–4
 segments, *Part 1*, 8–3
 individual, *Part 1*, 6–3
 initial values, *Part 2*, E–1 to E–4
 input prompt and echo types, *Part 1*, 9–5
 list of errors, *Part 2*, A–6 to A–10
 metafiles, *Part 1*, 10–2
 pick identification, *Part 1*, 8–2
 polyline, *Part 1*, 6–2
 polymarker, *Part 1*, 6–2
 segments, *Part 1*, 8–5
 text, *Part 1*, 6–2
Attribute source flags, *Part 1*, 6–4
Audit metafiles, *Part 1*, 10–1
AWAIT EVENT function, *Part 1*, 9–19, 9–24
 example, *Part 1*, 9–100
Axes, *Part 1*, 7–1
 See also Coordinates
 See also Segments
 segment fixed point, *Part 1*, 8–7

B

Background
 color, *Part 1*, 6–5
Binding
 attributes to primitives, *Part 1*, 6–1
 FORTRAN
 list of constants, *Part 2*, B–1 to B–14, D–1
 to D–9
Boundaries, *Part 1*, 7–8
 See Windows or Viewports

Break input, *Part 1*, 9–18
Buffers
 See also Data records
 See also Input
 input data record, *Part 1*, 9–8
 string input, *Part 1*, 9–4
 stroke input, *Part 1*, 9–4
Bundles, *Part 1*, 6–3
 See also Attributes
 color, *Part 1*, 6–15, 6–16
 edge, *Part 1*, 6–21
 fill area, *Part 1*, 6–26, 6–29
 pattern styles, *Part 1*, 6–40
 polyline, *Part 1*, 6–44, 6–45
 polymarkers, *Part 1*, 6–48, 6–49
 text, *Part 1*, 6–56, 6–59

C

Calls
 error handler, *Part 2*, 12–1
 function
 reproducing, *Part 1*, 10–2
Categories
 See also Workstations
 of functions, *Part 1*, 1–1
 workstations, *Part 1*, 4–2
 list of, *Part 1*, 4–2
CELL ARRAY 3 function, *Part 1*, 5–6
CELL ARRAY function, *Part 1*, 5–4
 example, *Part 1*, 5–27
Cell arrays, *Part 1*, 5–4, 5–6
CGM metafiles
 ANSI standard, *Part 1*, 10–1
 creating, *Part 1*, 10–3 to 10–4
Change vectors
 input, *Part 1*, 9–5
 segment translation, *Part 1*, 8–7
Characters
 height, *Part 1*, 6–12
 strings, *Part 1*, 5–20, 5–21, 5–23, 5–25
Choice
 See also Input
 input class, *Part 1*, 9–4
 specifying NOCHOICE input, *Part 1*, 9–18,
 9–19
Choice input prompt flags
 list of, *Part 2*, B–1
Choice status types
 list of, *Part 2*, B–1
Classes
 See also Input
 See also Logical input devices
 choice, *Part 1*, 9–4
 locator, *Part 1*, 9–4
 pick, *Part 1*, 9–4
 string, *Part 1*, 9–4

Classes (cont'd)
 stroke, *Part 1*, 9–4
 valuator, *Part 1*, 9–4
Cleanup
 error handling, *Part 2*, 12–1
Clearing
 See also Workstations
 workstation surface, *Part 1*, 4–10
 implicit regeneration, *Part 1*, 4–7
Clear screen states
 list of, *Part 2*, B–2
CLEAR WORKSTATION function, *Part 1*, 4–10
 example, *Part 1*, 4–30
Clipping, *Part 1*, 7–4, 7–7
 See also Transformations
 disable, *Part 1*, 7–4
 enable, *Part 1*, 7–4
 segments, *Part 1*, 8–9
 text precision, *Part 1*, 6–54
Clipping flag
 initial value, *Part 2*, E–4
Clipping flags
 list of, *Part 2*, B–2
CLOSE GKS function, *Part 1*, 4–11
 example, *Part 1*, 4–30
CLOSE SEGMENT function, *Part 1*, 8–12
 example, *Part 1*, 8–29
CLOSE WORKSTATION function, *Part 1*, 4–12
 example, *Part 1*, 4–30
Closing
 See also GKS
 See also Workstations
 GKS, *Part 1*, 4–5
 error handling, *Part 2*, 12–1
 segments, *Part 1*, 4–5
 workstations, *Part 1*, 4–5
Color
 See also Attributes
 background, *Part 1*, 6–5
 fill area, *Part 1*, 6–25
 foreground, *Part 1*, 6–5
 indexes
 2D arrays, *Part 1*, 5–4
 3D arrays, *Part 1*, 5–6
 markers, *Part 1*, 6–47
 model, *Part 1*, 6–15
 polyline, *Part 1*, 6–43
 representation, *Part 1*, 6–16
 text, *Part 1*, 6–53
Color availability flags
 list of, *Part 2*, B–2
Compiling
 ULTRIX programs, *Part 1*, 3–1
 VMS programs, *Part 1*, 2–2
Completion states, *Part 2*, A–1

Components

- See also Rotation
- See also Scale
- See also Translation
- segment transformations, *Part 1*, 8–7

Composition

- See also Transformations
- picture, *Part 1*, 1–3, 7–1

Conditions

- error, *Part 2*, 12–1, A–1 to A–37

Configuration files, *Part 1*, 3–8

- customizing
 - system level, *Part 1*, 3–8
 - user level, *Part 1*, 3–8

Connection identifiers

- metafiles, *Part 1*, 10–1, 10–4
- specifying on ULTRIX, *Part 1*, 3–4
- specifying on VMS, *Part 1*, 2–2

Constants

- arc types, *Part 2*, B–1
- aspect source flags, *Part 2*, B–1
- attribute control flags, *Part 2*, B–1
- choice input prompt flags, *Part 2*, B–1
- choice status types, *Part 2*, B–1
- clear screen states, *Part 2*, B–2
- clipping flags, *Part 2*, B–2
- color availability flags, *Part 2*, B–2
- coordinate switch, *Part 2*, B–2
- current and request values, *Part 2*, B–2
- default connection identifier, *Part 2*, B–2
- deferral modes, *Part 2*, B–2
- detectability flags, *Part 2*, B–2
- device coordinate units, *Part 2*, B–2
- display surface states, *Part 2*, B–3
- dynamic modification states, *Part 2*, B–3
- echo states, *Part 2*, B–3
- edge flags, *Part 2*, B–3
- edge types, *Part 2*, B–3
- error-handling modes, *Part 2*, B–3
- escape function identifiers, *Part 2*, B–3 to B–5
- fill area control flags, *Part 2*, B–5
- fill area interior styles, *Part 2*, B–5
- FORTTRAN binding, *Part 2*, B–1
- GDP bundle types, *Part 2*, B–5
- GDP graphics primitives, *Part 2*, B–6 to B–7
- GKS level types, *Part 2*, B–7
- GKS operating states, *Part 2*, B–7
- highlighting flags, *Part 2*, B–8
- highlighting methods, *Part 2*, B–8
- HLHSR identifiers, *Part 2*, B–7
- HLHSR modes, *Part 2*, B–7
- implicit regeneration states, *Part 2*, B–8
- input classes, *Part 2*, B–8
- input mode types, *Part 2*, B–8
- input priority states, *Part 2*, B–8
- invalid index flags, *Part 2*, B–9
- last event flags, *Part 2*, B–9

Constants (cont'd)

- line cap styles, *Part 2*, B–9
- line join styles, *Part 2*, B–9
- line types, *Part 2*, B–9
- line types (standard), *Part 2*, B–9
- listing of, *Part 2*, B–1 to B–14, D–1 to D–9
- marker types, *Part 2*, B–10
- marker types (standard), *Part 2*, B–10
- new frame action states, *Part 2*, B–10
- pick status types, *Part 2*, B–10
- projection types, *Part 2*, B–10
- regeneration flag states, *Part 2*, B–10
- relative input priority, *Part 2*, B–11
- request status types, *Part 2*, B–11
- requirements, *Part 1*, 2–1, 3–1
- returned type values, *Part 2*, B–11
- simultaneous event flags, *Part 2*, B–11
- text horizontal alignment types, *Part 2*, B–11
- text path types, *Part 2*, B–11
- text precision types, *Part 2*, B–11
- text vertical alignment types, *Part 2*, B–12
- update states, *Part 2*, B–12
- visibility flags, *Part 2*, B–12
- workstation category types, *Part 2*, B–12
- workstation class types, *Part 2*, B–12
- workstation color availability states, *Part 2*, B–12
- workstation states, *Part 2*, B–12
- workstation types, *Part 2*, B–13 to B–14
- writing modes, *Part 2*, B–14

Control

- error handling, *Part 2*, 12–1
- workstation surface, *Part 1*, 4–6

Control functions, *Part 1*, 4–1 to 4–42

- introduction to, *Part 1*, 4–1 to 4–8
- metafiles, *Part 1*, 10–2

Coordinates

- See also Transformations
- input change vectors, *Part 1*, 9–5
- locator and stroke input, *Part 1*, 9–4
- maximum device, *Part 1*, 7–8
- systems, *Part 1*, 7–1
 - used for output, *Part 1*, 5–2
- viewport input priority, *Part 1*, 7–6, 9–21

Coordinate switch

- list of, *Part 2*, B–2

Copying segments, *Part 1*, 8–3

COPY SEGMENT TO WORKSTATION function, *Part 1*, 8–13

- example, *Part 1*, 8–29

CREATE SEGMENT function, *Part 1*, 8–14

- example, *Part 1*, 8–29

Creating

- metafiles, *Part 1*, 10–1
- segments, *Part 1*, 4–5, 8–1

Current

- See also Transformations
- metafile item, *Part 1*, 10–4

Current (cont'd)
 windows and viewports, *Part 1*, 7–8
Current event report entry, *Part 1*, 9–20
 See also Event mode
 See also Input
 Current and request values
 list of, *Part 2*, B–2
 Cycling
 disabled input echo, *Part 1*, 9–17
 logical input device control, *Part 1*, 9–17

D

Data
 user defined
 metafiles, *Part 1*, 10–2
 Data records
 See also Escapes
 See also Input
 input, *Part 1*, 9–8, 9–9
 determining size of, *Part 1*, 9–8
 packing, *Part 1*, 9–8
 prompt and echo types, *Part 1*, 9–5 to 9–16
 sizes, *Part 1*, 9–22
 standard, *Part 1*, 9–8
 using inquiry functions, *Part 1*, 9–22
 metafile
 item, *Part 1*, 10–2
 Data structures
 See also GKS
 Data types
 arguments, *Part 1*, 1–6
 DEACTIVATE WORKSTATION function, *Part 1*,
 4–13
 example, *Part 1*, 4–30
 Deactivating
 See also Workstations
 workstations, *Part 1*, 4–5
 Default connection identifier
 list of, *Part 2*, B–2
 Defaults
 See also Attributes
 See also Transformations
 colors, *Part 1*, 6–5
 GKS error handler, *Part 2*, 12–4
 identity segment transformation, *Part 1*, 8–7
 normalization window, *Part 1*, 7–3
 unity transformation, *Part 1*, 7–4
 Deferral
 See also Implicit regenerations
 DECwindows, *Part 1*, 1–7
 output, *Part 1*, 4–6, 5–3
 Deferral modes
 list of, *Part 2*, B–2
 Definition file, *Part 1*, 2–1

Definition files
 including, *Part 1*, 2–1, 3–1
 Degrees
 See also GDPs
 See also Segments
 translating to radians, *Part 1*, 8–8
 DELETE SEGMENT FROM WORKSTATION
 function, *Part 1*, 8–16
 DELETE SEGMENT function, *Part 1*, 8–15
 Deleting segments, *Part 1*, 8–2
 Descriptions
 functions, *Part 1*, 1–4
 Description tables, *Part 1*, 4–1
 GKS, *Part 2*, 11–1
 workstation, *Part 2*, 11–1
 Detectability flags
 list of, *Part 2*, B–2
 Detecting
 errors, *Part 2*, 12–1
 segments, *Part 1*, 8–5
 Device
 transformations, *Part 1*, 7–7 to 7–8
 Device coordinates, *Part 1*, 7–1
 See also Transformations
 See also Workstations
 Device coordinate units
 list of, *Part 2*, B–2
 Device dependent
 bundled attributes, *Part 1*, 6–3
 Device independent
 attributes, *Part 1*, 6–1
 Device-independent programming
 input, *Part 1*, 9–22
 Device number, *Part 1*, 9–1
 Devices
 See also Workstations
 logical input, *Part 1*, 9–1 to 9–3
 manipulation
 GESC function, *Part 1*, 4–14
 maximum coordinate values, *Part 1*, 7–8
 physical input, *Part 1*, 9–1
 Display
 See also Workstations
 surface, *Part 1*, 7–1
 surface control, *Part 1*, 7–1
 CLEAR WORKSTATION function, *Part 1*,
 4–10
 REDRAW ALL SEGMENTS ON
 WORKSTATION function, *Part 1*,
 4–25
 SET DEFERRAL STATE function, *Part 1*,
 4–26
 UPDATE WORKSTATION function, *Part*
1, 4–28
 Display surface states
 list of, *Part 2*, B–3

Dynamic modification
 See also Implicit regenerations
 attributes, *Part 1, 4-7*
 workstation transformations, *Part 1, 4-7*
Dynamic modification states
 list of, *Part 2, B-3*

E

Echo
 See also Input
 cycling and disabled echo, *Part 1, 9-17*
 input values, *Part 1, 9-2, 9-3, 9-16*
 prompt and echo types, *Part 1, 9-5 to 9-16*
Echo area, *Part 1, 9-2*
Echo states
 list of, *Part 2, B-3*
Edge
 index, *Part 1, 6-20*
 representation, *Part 1, 6-21*
 type, *Part 1, 6-23*
 width scale factor, *Part 1, 6-24*
Edge flags
 list of, *Part 2, B-3*
Edge types
 list of, *Part 2, B-3*
Emergency
 closure of GKS, *Part 2, 12-1*
EMERGENCY CLOSE GKS function, *Part 2, 12-3*
 example, *Part 2, 12-6*
Ending
 GKS program, *Part 1, 4-11*
Entries
 See also GKS
 bundle table, *Part 1, 6-3*
Environment
 GKS, *Part 1, 4-1*
 initializing, *Part 1, 4-22*
 workstation, *Part 1, 4-1*
Environment variables, *Part 1, 3-4*
 default file, *Part 1, 3-5*
 defining
 at csh, *Part 1, 3-4*
 at sh, *Part 1, 3-4*
 in file, *Part 1, 3-4*
 GKSsaf, *Part 1, 3-6*
 GKSconid, *Part 1, 3-4, 3-6*
 .GKSdefaults, *Part 1, 3-5*
 GKSdefmode, *Part 1, 3-6*
 GKSerrfile, *Part 1, 3-6, 3-7*
 GKSerror, *Part 1, 3-6*
 GKSirg, *Part 1, 3-6*
 GKSmetafile_type, *Part 1, 3-6*
 GKSndc_clip, *Part 1, 3-7*
 GKSstroke_font1, *Part 1, 3-7*
 GKSswstype, *Part 1, 3-7*
 search order, *Part 1, 3-5*

Environment variables (cont'd)
 system defaults file, *Part 1, 3-4*
 types, *Part 1, 3-6*
 general, *Part 1, 3-6*
 user defaults file, *Part 1, 3-4*
Error codes
 defined, *Part 1, 2-4, 3-7*
 ULTRIX, *Part 1, 3-7*
 VMS, *Part 1, 2-4*
Error files
 default, *Part 1, 2-5*
 defined, *Part 1, 3-7*
 ULTRIX, *Part 1, 3-7*
 VMS, *Part 1, 2-5*
Error handling, *Part 1, 2-4, 2-5, 3-7; Part 2, D-1*
 GKS, *Part 1, 1-4*
ERROR HANDLING function, *Part 2, 12-4*
Error-handling functions, *Part 2, 12-1 to 12-7*
 introduction to, *Part 2, 12-1 to 12-2*
Error-handling modes
 list of, *Part 2, B-3*
ERROR LOGGING function, *Part 2, 12-5*
 example, *Part 2, 12-6*
Errors
 file, *Part 2, 12-2*
 inquiry error status argument, *Part 2, 11-3*
 logging, *Part 1, 4-4; Part 2, 12-5*
 messages, *Part 2, A-1 to A-37*
 attributes, *Part 2, A-6 to A-10*
 escapes, *Part 2, A-15*
 fatal, *Part 2, A-36 to A-37*
 FORTRAN binding, *Part 2, A-19 to A-20*
 implementation-specific, *Part 2, A-20 to A-36*
 input, *Part 2, A-12 to A-14*
 metafiles, *Part 2, A-14 to A-15*
 miscellaneous, *Part 2, A-15 to A-16*
 operating state, *Part 2, A-1 to A-2*
 output, *Part 2, A-10 to A-11*
 segments, *Part 2, A-11 to A-12*
 system, *Part 2, A-16 to A-19*
 transformations, *Part 2, A-5 to A-6*
 workstation, *Part 2, A-2 to A-5*
 state list, *Part 1, 4-22*
 states, *Part 2, 12-1*
Error status files
 list of, *Part 1, 3-1*
ESCAPE function, *Part 1, 4-14*
 example, *Part 1, 4-32*
Escape function identifiers
 list of, *Part 2, B-3*
Escapes
 GESC, *Part 1, 4-14*
 list of errors, *Part 2, A-15*
 Set DEC GKS connection identifier string, *Part 1, 4-24*

EVALUATE TRANSFORMATION MATRIX 3
 function, *Part 1*, 7–17

EVALUATE TRANSFORMATION MATRIX
 function, *Part 1*, 7–15
 example, *Part 1*, 7–43

EVALUATE VIEW MAPPING MATRIX 3 function,
Part 1, 7–19

EVALUATE VIEW ORIENTATION MATRIX 3
 function, *Part 1*, 7–22

Event functions, *Part 1*, 9–19

Event input queue, *Part 1*, 9–19
 overflow, *Part 1*, 9–20

Event mode, *Part 1*, 9–19 to 9–21
 See also Input
 cycling devices, *Part 1*, 9–17

Examples
 list of functions, *Part 2*, C–1
 table of, *Part 2*, C–1

Executing
 ULTRIX programs, *Part 1*, 3–1
 VMS programs, *Part 1*, 2–2

Expansion
 See also Scale
 See also Segments
 segments, *Part 1*, 8–7
 text, *Part 1*, 6–11

Extent rectangle
 See also Attributes
 See also Segments
 See also Text
 segments
 highlighting, *Part 1*, 8–6

F

Fatal errors, *Part 2*, 12–1
 list of, *Part 2*, A–36 to A–37

File
 definition, *Part 1*, 2–1

Files
 error, *Part 2*, 12–2
 error status
 list of, *Part 1*, 3–1

File specifications
 metafiles, *Part 1*, 10–1

FILL AREA 3 function, *Part 1*, 5–9

Fill area control flags
 list of, *Part 2*, B–5

FILL AREA function, *Part 1*, 5–8
 example, *Part 1*, 6–61

Fill area interior styles
 list of, *Part 2*, B–5

Fill areas, *Part 1*, 5–8
 See also Attributes
 attributes
 SET FILL AREA COLOUR INDEX
 function, *Part 1*, 6–25

Fill areas
 attributes (cont'd)
 SET FILL AREA INDEX function, *Part 1*,
 6–26
 SET FILL AREA STYLE INDEX function,
Part 1, 6–31
 SET PATTERN REFERENCE POINT
 function, *Part 1*, 6–38
 SET PATTERN SIZE function, *Part 1*,
 6–41
 bundles, *Part 1*, 6–26
 3D, *Part 1*, 5–9
 initial attributes, *Part 2*, E–3
 interior styles, *Part 1*, 6–27
 representation, *Part 1*, 6–29
 See also GFAS3 function, *Part 1*, 5–10
 See also GFAS function, *Part 1*, 5–11
 style indexes, *Part 1*, 6–31

Fill area set, *Part 1*, 5–10, 5–11

FILL AREA SET 3 function, *Part 1*, 5–11

FILL AREA SET function, *Part 1*, 5–10

Fill area sets
 initial attributes, *Part 2*, E–3 to E–4

Fixed points
 See also Rotation
 See also Scale
 See also Segments
 segment transformations, *Part 1*, 8–7

Flags
 See also Attributes
 ASF, *Part 1*, 6–7, 6–9
 aspect source, *Part 1*, 6–4
 edge flag, *Part 1*, 6–19

Flush
 event queue, *Part 1*, 9–20

FLUSH DEVICE EVENTS, *Part 1*, 9–20, 9–21

FLUSH DEVICE EVENTS function, *Part 1*, 9–26

Fonts
 establishing, *Part 1*, 6–54

Foreground color, *Part 1*, 6–5

Format
 function descriptions, *Part 1*, 1–4
 metafiles, *Part 1*, 10–2

FORTTRAN binding
 errors
 list of, *Part 2*, A–19 to A–20
 list of constants, *Part 2*, B–1 to B–14, D–1 to
 D–9
 list of function names, *Part 2*, D–1 to D–9

FORTTRAN binding files
 VMS, *Part 1*, 2–1, 3–1

Function
 attribute, *Part 1*, 6–6
 constants, *Part 1*, 1–6
 control, *Part 1*, 4–8
 description, *Part 1*, 1–6
 escape, *Part 1*, 4–14
 header, *Part 1*, 1–4

Function (cont'd)

- metafile, *Part 1*, 4–14
- operating states, *Part 1*, 1–5
- output, *Part 1*, 5–3
- presentation, *Part 1*, 1–4 to 1–8
- Program Examples sections, *Part 1*, 1–6
- See Also sections, *Part 1*, 1–6
- segment, *Part 1*, 8–10
- syntax, *Part 1*, 1–5

Functional standards

- See also GKS

Function names

- listing of, *Part 2*, D–1 to D–9

Functions

- See also GKS
- DEC GKS categories, *Part 1*, 1–1
- error-handling, *Part 2*, 12–1 to 12–2
- input, *Part 1*, 9–23
- transformation, *Part 1*, 7–9

G

- GACTM3 function, *Part 1*, 7–13
- GACTM function, *Part 1*, 7–10
- GACWK function, *Part 1*, 4–9
- GASGWK function, *Part 1*, 8–11
- GCA3 function, *Part 1*, 5–6
- GCA function, *Part 1*, 5–4
- GCLKS function, *Part 1*, 4–11
- GCLRWK function, *Part 1*, 4–10
- GCLSG function, *Part 1*, 8–12
- GCLWK function, *Part 1*, 4–12
- GCRSG function, *Part 1*, 8–14
- GCSGWK function, *Part 1*, 8–13
- GDAWK function, *Part 1*, 4–13
- GDP bundle types
 - list of, *Part 2*, B–5
- GDP graphics primitives
 - list of, *Part 2*, B–6
- GDPs, *Part 1*, 5–14
 - attributes, *Part 1*, 6–3
- GDSG function, *Part 1*, 8–15
- GDSGWK function, *Part 1*, 8–16
- GECLKS function, *Part 2*, 12–3
- GENERALIZED DRAWING PRIMITIVE 3
 - function, *Part 1*, 5–14
- GENERALIZED DRAWING PRIMITIVE function,
 - Part 1*, 5–12
 - example, *Part 1*, 5–29
- Generalized drawing primitives
 - See GDPs
- Generation
 - See also Output
 - output, *Part 1*, 5–1
 - attributes, *Part 1*, 6–1
 - pictures, *Part 1*, 7–1
- Geometric attributes, *Part 1*, 6–1
- GERHND function, *Part 2*, 12–4
- GERLOG function, *Part 2*, 12–5
- GESC function, *Part 1*, 4–14
- GET CHOICE function, *Part 1*, 9–27
- GET functions, *Part 1*, 9–19
- GET ITEM TYPE FROM GKSM function, *Part 1*, 10–6
- GET ITEM TYPE FROM METAFILE function
 - See GET ITEM TYPE FROM GKSM function
- GET LOCATOR 3 function, *Part 1*, 9–29
- GET LOCATOR function, *Part 1*, 9–28
 - example, *Part 1*, 9–100
- GET PICK function, *Part 1*, 9–30
- GET STRING (FORTRAN–77 Subset) function,
 - Part 1*, 9–32
- GET STRING function, *Part 1*, 9–31
- GET STROKE 3 function, *Part 1*, 9–35
- GET STROKE function, *Part 1*, 9–33
- GET VALUATOR function, *Part 1*, 9–37
- GEVMM3 function, *Part 1*, 7–19
- GEVOM3 function, *Part 1*, 7–22
- GEVTM3 function, *Part 1*, 7–17
- GEVTM function, *Part 1*, 7–15
- GFA3 function, *Part 1*, 5–9
- GFA function, *Part 1*, 5–8
- GFAS3 function, *Part 1*, 5–11
- GFAS function, *Part 1*, 5–10
- GFLUSH function, *Part 1*, 9–26
- GGDP3 function, *Part 1*, 5–14
- GGDP function, *Part 1*, 5–12
- GGTCH function, *Part 1*, 9–27
- GGTITM function, *Part 1*, 10–6
- GGTLC3 function, *Part 1*, 9–29
- GGTLC function, *Part 1*, 9–28
- GGTPK function, *Part 1*, 9–30
- GGTSK3 function, *Part 1*, 9–35
- GGTSK function, *Part 1*, 9–33
- GGTST function, *Part 1*, 9–31, 9–32
- GGTVL function, *Part 1*, 9–37
- GIITM function, *Part 1*, 10–7
- GINCH3 function, *Part 1*, 9–40
- GINCH function, *Part 1*, 9–38
- GINLC3 function, *Part 1*, 9–44
- GINLC function, *Part 1*, 9–42
- GINPK3 function, *Part 1*, 9–48
- GINPK function, *Part 1*, 9–46
- GINSG3 function, *Part 1*, 8–19
- GINSG function, *Part 1*, 8–17
- GINSK3 function, *Part 1*, 9–60
- GINSK function, *Part 1*, 9–58
- GINST3 function, *Part 1*, 9–54, 9–56
- GINST function, *Part 1*, 9–50, 9–52
- GINVL3 function, *Part 1*, 9–64
- GINVL function, *Part 1*, 9–62
- GKS
 - ANSI and ISO standards, *Part 1*, 1–1
 - categories of functions, *Part 1*, 1–1

GKS (cont'd)

- closing, *Part 1*, 4–5
- description table, *Part 1*, 4–1
- environment, *Part 1*, 4–1, 4–22
- error handling, *Part 1*, 1–4; *Part 2*, 12–1
- input
 - levels of, *Part 1*, 1–4
- introduction to, *Part 1*, 1–1 to 1–4
- kernel, *Part 1*, 4–1
- levels, *Part 1*, 1–4
- metafile standard, *Part 1*, 10–1
- opening, *Part 1*, 4–4
- operating state
 - errors, *Part 2*, A–1 to A–2
- output
 - levels of, *Part 1*, 1–4
- GKS\$ASF, *Part 1*, 2–3
- GKS\$CONID, *Part 1*, 2–3
- GKS\$DEF_MODE, *Part 1*, 2–3
- GKS\$ERRFILE, *Part 1*, 2–4
- GKS\$ERROR, *Part 1*, 2–4
- GKS\$IRG, *Part 1*, 2–4
- GKS\$METAFILE_TYPE, *Part 1*, 2–4
- GKS\$NDC_CLIP, *Part 1*, 2–4
- GKS\$STROKE_FONT1, *Part 1*, 2–4
- GKS\$WSTYPE, *Part 1*, 2–4
- GKS3D
 - environment, *Part 1*, 4–11
- GKS3 metafiles
 - creating, *Part 1*, 10–1 to 10–3
- GKSasf, *Part 1*, 3–6
- gksconfig.c, *Part 1*, 3–8
- GKSconid, *Part 1*, 3–6
- GKSdefmode, *Part 1*, 3–6
- GKSerrfile, *Part 1*, 3–6
- GKSError, *Part 1*, 3–6
- GKSirg, *Part 1*, 3–6
- GKS level types
 - list of, *Part 2*, B–7
- GKSmetafile_type, *Part 1*, 3–6
- GKSM metafiles, *Part 1*, 10–1
 - creating, *Part 1*, 10–1 to 10–3
- GKSndc_clip, *Part 1*, 3–7
- GKS operating states
 - list of, *Part 2*, B–7
- GKSstroke_font1, *Part 1*, 3–7
- GKSswtype, *Part 1*, 3–7
- gks_decw_config.c, *Part 1*, 3–8
- GMSG function, *Part 1*, 4–20
- GMSGs function, *Part 1*, 4–21
- GOPKS function, *Part 1*, 4–22
- GOPWK function, *Part 1*, 4–23
- GPL3 function, *Part 1*, 5–17
- GPL function, *Part 1*, 5–16
- GPM3 function, *Part 1*, 5–19
- GPM function, *Part 1*, 5–18
- GPREC function, *Part 1*, 9–66, 9–67
- GQACWK function, *Part 2*, 11–135
- GQASF3 function, *Part 2*, 11–7
- GQASF function, *Part 2*, 11–5
- GQASWK function, *Part 2*, 11–136
- GQCF function, *Part 2*, 11–21
- GQCHB function, *Part 2*, 11–9
- GQCHH function, *Part 2*, 11–11
- GQCHS3 function, *Part 2*, 11–17
- GQCHS function, *Part 2*, 11–15
- GQCHSP function, *Part 2*, 11–12
- GQCHUP function, *Part 2*, 11–13
- GQCHW function, *Part 2*, 11–14
- GQCHXP function, *Part 2*, 11–10
- GQCLIP function, *Part 2*, 11–19
- GQCLP3 function, *Part 2*, 11–20
- GQCMDf function, *Part 2*, 11–23
- GQCMD function, *Part 2*, 11–22
- GQCNTN function, *Part 2*, 11–31
- GQCR function, *Part 2*, 11–24
- GQDCH3 function, *Part 2*, 11–36
- GQDCH function, *Part 2*, 11–35
- GQDDS function, *Part 2*, 11–37
- GQDLC3 function, *Part 2*, 11–39
- GQDLC function, *Part 2*, 11–38
- GQDPK3 function, *Part 2*, 11–41
- GQDPK function, *Part 2*, 11–40
- GQDSGA function, *Part 2*, 11–50
- GQDSK3 function, *Part 2*, 11–45
- GQDSK function, *Part 2*, 11–44
- GQDSP3 function, *Part 2*, 11–49
- GQDSP function, *Part 2*, 11–48
- GQDST3 function, *Part 2*, 11–43
- GQDST function, *Part 2*, 11–42
- GQDVL3 function, *Part 2*, 11–47
- GQDVL function, *Part 2*, 11–46
- GQDWA3 function, *Part 2*, 11–54
- GQDWKA function, *Part 2*, 11–52
- GQECI function, *Part 2*, 11–77
- GQEDCI function, *Part 2*, 11–25
- GQEDF function, *Part 2*, 11–57
- GQEDFG function, *Part 2*, 11–26
- GQEDI function, *Part 2*, 11–27
- GQEDR function, *Part 2*, 11–58
- GQEDT function, *Part 2*, 11–28
- GQEEDI function, *Part 2*, 11–78
- GQEFAI function, *Part 2*, 11–79
- GQEGD3 function, *Part 2*, 11–75
- GQEGDP function, *Part 2*, 11–74
- GQENTN function, *Part 2*, 11–80
- GQEPAI function, *Part 2*, 11–81
- GQEPLI function, *Part 2*, 11–82
- GQEPMI function, *Part 2*, 11–83
- GQETXI function, *Part 2*, 11–84
- GQEVWI function, *Part 2*, 11–85
- GQEWK function, *Part 2*, 11–76

GQEWSC function, *Part 2*, 11–29
 GQFACI function, *Part 2*, 11–59
 GQFAF function, *Part 2*, 11–60
 GQFAI function, *Part 2*, 11–62
 GQFAIS function, *Part 2*, 11–63
 GQFAR function, *Part 2*, 11–64
 GQFASI function, *Part 2*, 11–65
 GQGDP3 function, *Part 2*, 11–67
 GQGDP function, *Part 2*, 11–66
 GQHRF function, *Part 2*, 11–68
 GQHHRIV function, *Part 2*, 11–30
 GQHRM function, *Part 2*, 11–69
 GQIQOV function, *Part 2*, 11–70
 GQLCS3 function, *Part 2*, 11–88
 GQLCS function, *Part 2*, 11–86
 GQLI function, *Part 2*, 11–99
 GQLN function, *Part 2*, 11–72
 GQLVKS function, *Part 2*, 11–71
 GQLWK3 function, *Part 2*, 11–93
 GQLWK function, *Part 2*, 11–92
 GQLWSC function, *Part 2*, 11–73
 GQMK function, *Part 2*, 11–91
 GQMKSC function, *Part 2*, 11–90
 GQMNTN function, *Part 2*, 11–94
 GQNT3 function, *Part 2*, 11–98
 GQNT function, *Part 2*, 11–97
 GQOPS function, *Part 2*, 11–101
 GQOPSG function, *Part 2*, 11–96
 GQOPWK function, *Part 2*, 11–137
 GQPAF function, *Part 2*, 11–102
 GQPA function, *Part 2*, 11–105
 GQPARF function, *Part 2*, 11–103
 GQPAR function, *Part 2*, 11–104
 GQPCR function, *Part 2*, 11–123
 GQPEDR function, *Part 2*, 11–124
 GQPFAR function, *Part 2*, 11–125
 GQPKID function, *Part 2*, 11–33
 GQPKS3 function, *Part 2*, 11–108
 GQPKS function, *Part 2*, 11–106
 GQPLCI function, *Part 2*, 11–114
 GQPLF function, *Part 2*, 11–115
 GQPLI function, *Part 2*, 11–116
 GQPLR function, *Part 2*, 11–117
 GQPMCI function, *Part 2*, 11–118
 GQPMF function, *Part 2*, 11–119
 GQPMI function, *Part 2*, 11–120
 GQPMR function, *Part 2*, 11–121
 GQPPAR function, *Part 2*, 11–126
 GQPPLR function, *Part 2*, 11–127
 GQPPMR function, *Part 2*, 11–128
 GQPRPV function, *Part 2*, 11–32
 GQPTXR function, *Part 2*, 11–129
 GQPVWR function, *Part 2*, 11–130
 GQPXAD function, *Part 2*, 11–113
 GQPXA function, *Part 2*, 11–111
 GQPX function, *Part 2*, 11–110
 GQSGA3 function, *Part 2*, 11–133
 GQSGA function, *Part 2*, 11–131
 GQSGP function, *Part 2*, 11–100
 GQSGUS function, *Part 2*, 11–138
 GQSGWK function, *Part 2*, 11–139
 GQSIM function, *Part 2*, 11–95
 GQSKS3 function, *Part 2*, 11–150
 GQSKS function, *Part 2*, 11–148
 GQSTS3 function, *Part 2*, 11–144, 11–146
 GQSTS function, *Part 2*, 11–140, 11–142
 GQTX3 function, *Part 2*, 11–156
 GQTX3S function, *Part 2*, 11–157
 GQTXAL function, *Part 2*, 11–152
 GQTXCI function, *Part 2*, 11–153
 GQTXF function, *Part 2*, 11–158
 GQTXFP function, *Part 2*, 11–160
 GQTXI function, *Part 2*, 11–162
 GQTXP function, *Part 2*, 11–163
 GQTXR function, *Part 2*, 11–164
 GQTXS function, *Part 2*, 11–154
 GQTXXS function, *Part 2*, 11–155
 GQVLS3 function, *Part 2*, 11–168
 GQVLS function, *Part 2*, 11–166
 GQVWF function, *Part 2*, 11–170
 GQVWI function, *Part 2*, 11–34
 GQVWR function, *Part 2*, 11–171
 GQWKCA function, *Part 2*, 11–173
 GQWKC function, *Part 2*, 11–175
 GQWKCL function, *Part 2*, 11–174
 GQWKDU function, *Part 2*, 11–176
 GQWKM function, *Part 2*, 11–178
 GQWKS function, *Part 2*, 11–179
 GQWKT3 function, *Part 2*, 11–182
 GQWKT function, *Part 2*, 11–180
 Graphics handlers, *Part 1*, 4–1
 See also Devices
 See also Workstations
 input, *Part 1*, 9–5
 nominal sizes, *Part 1*, 6–1
 GRDITM function, *Part 1*, 10–8
 GRENSG function, *Part 1*, 8–21
 GRQCH function, *Part 1*, 9–68
 GRQLC3 function, *Part 1*, 9–70
 GRQLC function, *Part 1*, 9–69
 GRQPK function, *Part 1*, 9–71
 GRQSK3 function, *Part 1*, 9–78
 GRQSK function, *Part 1*, 9–76
 GRQST function, *Part 1*, 9–72, 9–74
 GRQVL function, *Part 1*, 9–80
 GRSWK function, *Part 1*, 4–25
 GSASF3 function, *Part 1*, 6–9
 GSASF function, *Part 1*, 6–7
 GSCHH function, *Part 1*, 6–12
 GSCHM function, *Part 1*, 9–92
 GSCHSP function, *Part 1*, 6–13

GSCHUP function, *Part 1*, 6–14
 GSCHXP function, *Part 1*, 6–11
 GSCLIP function, *Part 1*, 7–25
 GSCMD function, *Part 1*, 6–15
 GSCR function, *Part 1*, 6–16
 GSDS function, *Part 1*, 4–26
 GSDTEC function, *Part 1*, 8–22
 GSEDCI function, *Part 1*, 6–18
 GSEDFG function, *Part 1*, 6–19
 GSEDI function, *Part 1*, 6–20
 GSEDR function, *Part 1*, 6–21
 GSEDT function, *Part 1*, 6–23
 GSELNT function, *Part 1*, 7–24
 GSEWSC function, *Part 1*, 6–24
 GSFACI function, *Part 1*, 6–25
 GSFAI function, *Part 1*, 6–26
 GSFAIS function, *Part 1*, 6–27
 GSFAR function, *Part 1*, 6–29
 GSFASI function, *Part 1*, 6–31
 GSHLIT function, *Part 1*, 8–23
 GSHRID function, *Part 1*, 6–32
 GSHRM function, *Part 1*, 6–33
 GSLCM function, *Part 1*, 9–93
 GSLN function, *Part 1*, 6–34
 GSLWSC function, *Part 1*, 6–35
 GSMCH function, *Part 1*, 9–81
 GSMK function, *Part 1*, 6–37
 GSMKSC function, *Part 1*, 6–36
 GSMLC3 function, *Part 1*, 9–83
 GSMLC function, *Part 1*, 9–82
 GSMPK function, *Part 1*, 9–84
 GSMSK3 function, *Part 1*, 9–89
 GSMSK function, *Part 1*, 9–87
 GSMST function, *Part 1*, 9–85, 9–86
 GSMVL function, *Part 1*, 9–91
 GSPA function, *Part 1*, 6–41
 GSPARF function, *Part 1*, 6–38
 GSPAR function, *Part 1*, 6–40
 GSPKID function, *Part 1*, 6–42
 GSPKM function, *Part 1*, 9–94
 GSPLCI function, *Part 1*, 6–43
 GSPLI function, *Part 1*, 6–44
 GSPLR function, *Part 1*, 6–45
 GSPMCI function, *Part 1*, 6–47
 GSPMI function, *Part 1*, 6–48
 GSPMR function, *Part 1*, 6–49
 GSPRPV function, *Part 1*, 6–39
 GSSGP function, *Part 1*, 8–24
 GSSGT3 function, *Part 1*, 8–26
 GSSGT function, *Part 1*, 8–25
 GSSKM function, *Part 1*, 9–96
 GSSTM function, *Part 1*, 9–95
 GSTXAL function, *Part 1*, 6–51
 GSTXCI function, *Part 1*, 6–53
 GSTXFP function, *Part 1*, 6–54
 GSTXI function, *Part 1*, 6–56

GSTXP function, *Part 1*, 6–57
 GSTXR function, *Part 1*, 6–59
 GSV3 function, *Part 1*, 7–30
 GSVIS function, *Part 1*, 8–28
 GSVLM function, *Part 1*, 9–97
 GSVP function, *Part 1*, 7–29
 GSVPIP function, *Part 1*, 7–31
 GSVTIP function, *Part 1*, 7–28
 GSVWI function, *Part 1*, 7–26
 GSVWR3 function, *Part 1*, 7–27
 GSW3 function, *Part 1*, 7–33
 GSWKV3 function, *Part 1*, 7–35
 GSWKVP function, *Part 1*, 7–34
 GSWKW3 function, *Part 1*, 7–37
 GSWKWN function, *Part 1*, 7–36
 GSWN function, *Part 1*, 7–32
 GTX3 function, *Part 1*, 5–23
 GTX3S function, *Part 1*, 5–25
 GTX function, *Part 1*, 5–20
 GTXS function, *Part 1*, 5–21
 GUREC function, *Part 1*, 9–98, 9–99
 GUWK function, *Part 1*, 4–28
 GWAIT function, *Part 1*, 9–24
 GWITM function, *Part 1*, 10–10

H

Handlers

See also Devices
 See also Workstations
 See Graphics handlers
 errors, *Part 2*, 12–1
 set and realized values, *Part 2*, 11–4

Hardware fonts, *Part 1*, 6–54

See also Fonts

Hatches, *Part 1*, 6–27

See also Fill areas
 fill areas, *Part 1*, 5–8, 5–9
 style index values, *Part 1*, 6–31

Height

See also Attributes
 See also Transformations
 text, *Part 1*, 6–12

Highlighting

segments, *Part 1*, 8–6

Highlighting flags

list of, *Part 2*, B–8

Highlighting methods

list of, *Part 2*, B–8

HLHSR identifiers

list of, *Part 2*, B–7

HLHSR modes

list of, *Part 2*, B–7

Hollow

fill area interior style, *Part 1*, 6–27
 fill areas, *Part 1*, 5–8, 5–9

Identifiers

pick, *Part 1*, 8–2, 9–4

Identity

segment transformation, *Part 1*, 8–8

transformation, *Part 1*, 7–7

Implementation-specific errors

list of, *Part 2*, A–20 to A–36

Implicit regenerations, *Part 1*, 4–7

See also Deferral

attribute changes, *Part 1*, 6–4

segments, *Part 1*, 8–3

workstation transformations, *Part 1*, 7–8

Implicit regeneration states

list of, *Part 2*, B–8

Include

definition files, *Part 1*, 2–1, 3–1

INCLUDE statement

all languages, *Part 1*, 2–1, 3–1

Index

See also Attributes

See also Bundles

color, *Part 1*, 6–15, 6–16

2D arrays, *Part 1*, 5–4

3D arrays, *Part 1*, 5–6

edge, *Part 1*, 6–21

edge color, *Part 1*, 6–18

fill area, *Part 1*, 6–26, 6–29

styles, *Part 1*, 6–31

into bundle tables, *Part 1*, 6–3

pattern styles, *Part 1*, 6–40

polyline, *Part 1*, 6–45

polymarkers, *Part 1*, 6–49

text, *Part 1*, 6–56, 6–59

Individual attributes, *Part 1*, 6–3

Initialize

See also GKS

See also Workstations

GKS environment, *Part 1*, 4–22

workstation environment, *Part 1*, 4–23

INITIALIZE CHOICE 3 function, *Part 1*, 9–40

INITIALIZE CHOICE function, *Part 1*, 9–38

INITIALIZE functions, *Part 1*, 9–3

packing input data records, *Part 1*, 9–8

INITIALIZE LOCATOR 3 function, *Part 1*, 9–44

INITIALIZE LOCATOR function, *Part 1*, 9–42

example, *Part 1*, 9–100

INITIALIZE PICK 3 function, *Part 1*, 9–48

INITIALIZE PICK function, *Part 1*, 9–46

example, *Part 1*, 9–103

INITIALIZE STRING (FORTRAN–77 Subset)

function, *Part 1*, 9–52

INITIALIZE STRING 3 (FORTRAN–77 Subset)

function, *Part 1*, 9–56

INITIALIZE STRING 3 function, *Part 1*, 9–54

INITIALIZE STRING function, *Part 1*, 9–50

example, *Part 1*, 9–108

INITIALIZE STROKE 3 function, *Part 1*, 9–60

INITIALIZE STROKE function, *Part 1*, 9–58

INITIALIZE VALUATOR 3 function, *Part 1*, 9–64

INITIALIZE VALUATOR function, *Part 1*, 9–62

example, *Part 1*, 9–111

Initializing a logical input device, *Part 1*, 9–3

Initializing input, *Part 1*, 9–16

Initial string

input, *Part 1*, 9–4

Input

asynchronous, *Part 1*, 9–16

breaking, *Part 1*, 9–18

classes, *Part 1*, 9–1, 9–4

current values, *Part 1*, 9–22

cycling device control, *Part 1*, 9–17

data record

sizes, *Part 1*, 9–22

standard, *Part 1*, 9–8

using inquiry functions, *Part 1*, 9–22

default values, *Part 1*, 9–22

device-independent programming, *Part 1*, 9–22

event mode, *Part 1*, 9–19 to 9–21

flushing the queue, *Part 1*, 9–20

event queue, *Part 1*, 9–19

event queue overflow, *Part 1*, 9–20

initializing, *Part 1*, 9–16

inquiry function use, *Part 1*, 9–22

list of errors, *Part 2*, A–12 to A–14

menus, *Part 1*, 9–4

metafiles, *Part 1*, 10–1, 10–2

operating modes, *Part 1*, 9–2, 9–3, 9–16 to 9–21

pick

visibility, *Part 1*, 8–10

pick identification, *Part 1*, 8–2

request mode, *Part 1*, 9–17 to 9–18

sample mode, *Part 1*, 9–18 to 9–19

segment detectability, *Part 1*, 8–5

segments, *Part 1*, 8–2

specifying no input, *Part 1*, 9–18

synchronous, *Part 1*, 9–16

text, *Part 1*, 9–4

triggers, *Part 1*, 9–3, 9–18

viewport priority, *Part 1*, 7–6, 9–21

workstation categories, *Part 1*, 4–2

Input classes

list of, *Part 2*, B–8

Input data records

sizes, *Part 1*, 9–22

standard list, *Part 1*, 9–9 to 9–16

Input functions, *Part 1*, 9–1 to 9–115

introduction to, *Part 1*, 9–1 to 9–21

Input mode types

list of, *Part 2*, B–8

Input operating modes, *Part 1*, 9–16
 Input priority
 initial value, *Part 2*, E–4
 Input priority states
 list of, *Part 2*, B–8
 INQUIRE ASPECT SOURCE FLAGS 3 function,
 Part 2, 11–7
 INQUIRE ASPECT SOURCE FLAGS function,
 Part 2, 11–5
 INQUIRE CHARACTER BASE VECTOR function,
 Part 2, 11–9
 INQUIRE CHARACTER EXPANSION FACTOR
 function, *Part 2*, 11–10
 INQUIRE CHARACTER HEIGHT function, *Part*
 2, 11–11
 INQUIRE CHARACTER SPACING function, *Part*
 2, 11–12
 INQUIRE CHARACTER UP VECTOR function,
 Part 2, 11–13
 INQUIRE CHARACTER WIDTH function, *Part 2*,
 11–14
 INQUIRE CHOICE DEVICE STATE 3 function,
 Part 2, 11–17
 INQUIRE CHOICE DEVICE STATE function,
 Part 2, 11–15
 INQUIRE CLIPPING 3 function, *Part 2*, 11–20
 INQUIRE CLIPPING function, *Part 2*, 11–19
 INQUIRE COLOUR FACILITIES function, *Part*
 2, 11–21
 INQUIRE COLOUR MODEL FACILITIES
 function, *Part 2*, 11–23
 INQUIRE COLOUR MODEL function, *Part 2*,
 11–22
 INQUIRE COLOUR REPRESENTATION function,
 Part 2, 11–24
 INQUIRE CURRENT EDGE COLOUR INDEX
 function, *Part 2*, 11–25
 INQUIRE CURRENT EDGE FLAG function, *Part*
 2, 11–26
 INQUIRE CURRENT EDGE INDEX function,
 Part 2, 11–27
 INQUIRE CURRENT EDGETYPE function, *Part*
 2, 11–28
 INQUIRE CURRENT EDGEWIDTH SCALE
 FACTOR function, *Part 2*, 11–29
 INQUIRE CURRENT NORMALIZATION
 TRANSFORMATION NUMBER function,
 Part 2, 11–31
 INQUIRE CURRENT PATTERN REFERENCE
 POINT AND VECTORS function, *Part 2*,
 11–32
 INQUIRE CURRENT PICK IDENTIFIER VALUE
 function, *Part 2*, 11–33
 INQUIRE CURRENT VIEW INDEX function,
 Part 2, 11–34
 INQUIRE DEFAULT CHOICE DEVICE DATA 3
 function, *Part 2*, 11–36
 INQUIRE DEFAULT CHOICE DEVICE DATA
 function, *Part 2*, 11–35
 INQUIRE DEFAULT DEFERRAL STATE
 VALUES function, *Part 2*, 11–37
 INQUIRE DEFAULT LOCATOR DEVICE DATA 3
 function, *Part 2*, 11–39
 INQUIRE DEFAULT LOCATOR DEVICE DATA
 function, *Part 2*, 11–38
 INQUIRE DEFAULT PICK DEVICE DATA 3
 function, *Part 2*, 11–41
 INQUIRE DEFAULT PICK DEVICE DATA
 function, *Part 2*, 11–40
 INQUIRE DEFAULT STRING DEVICE DATA 3
 function, *Part 2*, 11–43
 INQUIRE DEFAULT STRING DEVICE DATA
 function, *Part 2*, 11–42
 INQUIRE DEFAULT STROKE DEVICE DATA 3
 function, *Part 2*, 11–45
 INQUIRE DEFAULT STROKE DEVICE DATA
 function, *Part 2*, 11–44
 INQUIRE DEFAULT VALUATOR DEVICE DATA
 3 function, *Part 2*, 11–47
 INQUIRE DEFAULT VALUATOR DEVICE DATA
 function, *Part 2*, 11–46
 INQUIRE DISPLAY SPACE SIZE 3 function, *Part*
 2, 11–49
 INQUIRE DISPLAY SPACE SIZE function, *Part*
 2, 11–48
 example, *Part 1*, 7–52
 INQUIRE DYNAMIC MODIFICATION OF
 SEGMENT ATTRIBUTES function, *Part 2*,
 11–50
 INQUIRE DYNAMIC MODIFICATION OF
 WORKSTATION ATTRIBUTES 3 function,
 Part 2, 11–54
 INQUIRE DYNAMIC MODIFICATION OF
 WORKSTATION ATTRIBUTES function,
 Part 2, 11–52
 example, *Part 1*, 7–52
 INQUIRE EDGE FACILITIES function, *Part 2*,
 11–57
 INQUIRE EDGE REPRESENTATION function,
 Part 2, 11–58
 INQUIRE FILL AREA COLOUR INDEX function,
 Part 2, 11–59
 INQUIRE FILL AREA FACILITIES function,
 Part 2, 11–60
 INQUIRE FILL AREA INDEX function, *Part 2*,
 11–62
 INQUIRE FILL AREA INTERIOR STYLE
 function, *Part 2*, 11–63
 INQUIRE FILL AREA REPRESENTATION
 function, *Part 2*, 11–64

INQUIRE FILL AREA STYLE INDEX function,
Part 2, 11–65

INQUIRE GENERALIZED DRAWING
 PRIMITIVE 3 function, *Part 2*, 11–67

INQUIRE GENERALIZED DRAWING
 PRIMITIVE function, *Part 2*, 11–66

INQUIRE HLHSR FACILITIES function, *Part 2*,
 11–68

INQUIRE HLHSR IDENTIFIER VALUE
 See INQUIRE CURRENT HLHSR
 IDENTIFIER VALUE function

INQUIRE HLHSR IDENTIFIER VALUE function,
Part 2, 11–30

INQUIRE HLHSR MODE function, *Part 2*, 11–69

INQUIRE INPUT QUEUE OVERFLOW function,
Part 1, 9–20; *Part 2*, 11–70

INQUIRE LEVEL OF GKS function, *Part 2*,
 11–71

INQUIRE LINE COLOUR INDEX
 See INQUIRE POLYLINE COLOUR INDEX
 function

INQUIRE LINE FACILITIES
 See INQUIRE POLYLINE FACILITIES function

INQUIRE LINE INDEX
 See INQUIRE POLYLINE INDEX function

INQUIRE LINE REPRESENTATION
 See INQUIRE POLYLINE REPRESENTATION
 function

INQUIRE LINETYPE function, *Part 2*, 11–72

INQUIRE LINewidth SCALE FACTOR function,
Part 2, 11–73

INQUIRE LIST element OF AVAILABLE
 GENERALIZED DRAWING PRIMITIVES
 See INQUIRE LIST OF AVAILABLE
 GENERALIZED DRAWING PRIMITIVES
 function

INQUIRE LIST element OF AVAILABLE
 GENERALIZED DRAWING PRIMITIVES 3
 See INQUIRE LIST OF AVAILABLE
 GENERALIZED DRAWING PRIMITIVES 3
 function

INQUIRE LIST element OF AVAILABLE
 WORKSTATION TYPES
 See INQUIRE LIST OF AVAILABLE
 WORKSTATION TYPES function

INQUIRE LIST element OF COLOUR INDICES
 See INQUIRE LIST OF COLOUR INDICES
 function

INQUIRE LIST element OF EDGE INDICES
 See INQUIRE LIST OF EDGE INDICES
 function

INQUIRE LIST element OF FILL AREA INDICES
 See INQUIRE LIST OF FILL AREA INDICES
 function

INQUIRE LIST element OF LINE INDICES
 See INQUIRE LIST OF POLYLINE INDICES
 function

INQUIRE LIST element OF MARKER INDICES
 See INQUIRE LIST OF POLYMARKER
 INDICES function

INQUIRE LIST element OF NORMALIZATION
 TRANSFORMATION NUMBERS
 See INQUIRE LIST OF NORMALIZATION
 TRANSFORMATION NUMBERS function

INQUIRE LIST element OF PATTERN INDICES
 See INQUIRE LIST OF PATTERN INDICES
 function

INQUIRE LIST element OF POLYLINE INDICES
 See INQUIRE LIST OF POLYLINE INDICES
 function

INQUIRE LIST element OF POLYMARKER
 INDICES
 See INQUIRE LIST OF POLYMARKER
 INDICES function

INQUIRE LIST element OF TEXT INDICES
 See INQUIRE LIST OF TEXT INDICES
 function

INQUIRE LIST element OF VIEW INDICES
 See INQUIRE LIST OF VIEW INDICES
 function

INQUIRE LIST OF AVAILABLE GENERALIZED
 DRAWING PRIMITIVES 3 function, *Part 2*,
 11–75

INQUIRE LIST OF AVAILABLE GENERALIZED
 DRAWING PRIMITIVES function, *Part 2*,
 11–74

INQUIRE LIST OF AVAILABLE WORKSTATION
 TYPES function, *Part 2*, 11–76

INQUIRE LIST OF COLOUR INDICES function,
Part 2, 11–77

INQUIRE LIST OF EDGE INDICES function,
Part 2, 11–78

INQUIRE LIST OF FILL AREA INDICES
 function, *Part 2*, 11–79

INQUIRE LIST OF LINE INDICES
 See INQUIRE LIST OF POLYLINE INDICES
 function

INQUIRE LIST OF MARKER INDICES
 See INQUIRE LIST OF POLYMARKER
 INDICES function

INQUIRE LIST OF NORMALIZATION
 TRANSFORMATION NUMBERS function,
Part 2, 11–80

INQUIRE LIST OF PATTERN INDICES function,
Part 2, 11–81

INQUIRE LIST OF POLYLINE INDICES function,
Part 2, 11–82

INQUIRE LIST OF POLYMARKER INDICES
 function, *Part 2*, 11–83
 INQUIRE LIST OF TEXT INDICES function,
 Part 2, 11–84
 INQUIRE LIST OF VIEW INDICES function,
 Part 2, 11–85
 INQUIRE LOCATOR DEVICE STATE 3 function,
 Part 2, 11–88
 INQUIRE LOCATOR DEVICE STATE function,
 Part 2, 11–86
 example, *Part 1*, 9–100
 INQUIRE MARKER COLOUR INDEX
 See INQUIRE POLYMARKER COLOUR
 INDEX function
 INQUIRE MARKER FACILITIES
 See INQUIRE POLYMARKER FACILITIES
 function
 INQUIRE MARKER INDEX
 See INQUIRE POLYMARKER INDEX function
 INQUIRE MARKER REPRESENTATION
 See INQUIRE POLYMARKER
 REPRESENTATION function
 INQUIRE MARKER SIZE SCALE FACTOR
 function, *Part 2*, 11–90
 INQUIRE MARKER TYPE function, *Part 2*,
 11–91
 INQUIRE MAXIMUM LENGTH OF
 WORKSTATION STATE TABLES 3
 function, *Part 2*, 11–93
 INQUIRE MAXIMUM LENGTH OF
 WORKSTATION STATE TABLES
 function, *Part 2*, 11–92
 INQUIRE MAXIMUM NORMALIZATION
 TRANSFORMATION NUMBER function,
 Part 2, 11–94
 INQUIRE MORE SIMULTANEOUS EVENTS
 function, *Part 2*, 11–95
 INQUIRE NAME OF OPEN SEGMENT function,
 Part 2, 11–96
 INQUIRE NORMALIZATION
 TRANSFORMATION 3 function, *Part*
 2, 11–98
 INQUIRE NORMALIZATION
 TRANSFORMATION function, *Part 2*,
 11–97
 INQUIRE NUMBER OF AVAILABLE LOGICAL
 INPUT DEVICES function, *Part 2*, 11–99
 INQUIRE NUMBER OF SEGMENT PRIORITIES
 SUPPORTED function, *Part 2*, 11–100
 INQUIRE OPERATING STATE VALUE function,
 Part 2, 11–101
 INQUIRE PATTERN FACILITIES function, *Part*
 2, 11–102
 INQUIRE PATTERN REFERENCE POINT
 function, *Part 2*, 11–103
 INQUIRE PATTERN REPRESENTATION
 function, *Part 2*, 11–104
 INQUIRE PATTERN SIZE function, *Part 2*,
 11–105
 INQUIRE PICK DEVICE STATE 3 function, *Part*
 2, 11–108
 INQUIRE PICK DEVICE STATE function, *Part 2*,
 11–106
 example, *Part 1*, 9–103
 INQUIRE PIXEL ARRAY DIMENSIONS function,
 Part 2, 11–113
 INQUIRE PIXEL ARRAY function, *Part 2*, 11–111
 INQUIRE PIXEL function, *Part 2*, 11–110
 INQUIRE POLYLINE COLOUR INDEX function,
 Part 2, 11–114
 INQUIRE POLYLINE FACILITIES function, *Part*
 2, 11–115
 INQUIRE POLYLINE INDEX function, *Part 2*,
 11–116
 INQUIRE POLYLINE REPRESENTATION
 function, *Part 2*, 11–117
 INQUIRE POLYMARKER COLOUR INDEX
 function, *Part 2*, 11–118
 INQUIRE POLYMARKER FACILITIES function,
 Part 2, 11–119
 INQUIRE POLYMARKER INDEX function, *Part*
 2, 11–120
 INQUIRE POLYMARKER REPRESENTATION
 function, *Part 2*, 11–121
 INQUIRE PREDEFINED COLOUR
 REPRESENTATION function, *Part 2*,
 11–123
 INQUIRE PREDEFINED EDGE
 REPRESENTATION function, *Part 2*,
 11–124
 INQUIRE PREDEFINED FILL AREA
 REPRESENTATION function, *Part 2*, 11–125
 INQUIRE PREDEFINED LINE
 REPRESENTATION
 See INQUIRE PREDEFINED POLYLINE
 REPRESENTATION function
 INQUIRE PREDEFINED MARKER
 REPRESENTATION
 See INQUIRE PREDEFINED POLYMARKER
 REPRESENTATION function
 INQUIRE PREDEFINED PATTERN
 REPRESENTATION function, *Part 2*,
 11–126
 INQUIRE PREDEFINED POLYLINE
 REPRESENTATION function, *Part 2*, 11–127
 INQUIRE PREDEFINED POLYMARKER
 REPRESENTATION function, *Part 2*, 11–128
 INQUIRE PREDEFINED TEXT
 REPRESENTATION function, *Part 2*,
 11–129

INQUIRE PREDEFINED VIEW
 REPRESENTATION function, *Part 2*,
 11–130

INQUIRE SEGMENT ATTRIBUTES 3 function,
 Part 2, 11–133

INQUIRE SEGMENT ATTRIBUTES function,
 Part 2, 11–131

INQUIRE SET member OF ACTIVE
 WORKSTATIONS
 See INQUIRE SET OF ACTIVE
 WORKSTATIONS function

INQUIRE SET member OF ASSOCIATED
 WORKSTATIONS
 See INQUIRE SET OF ASSOCIATED
 WORKSTATIONS function

INQUIRE SET member OF OPEN
 WORKSTATIONS
 See INQUIRE SET OF OPEN
 WORKSTATIONS function

INQUIRE SET member OF SEGMENT NAMES
 IN USE
 See INQUIRE SET OF SEGMENT NAMES IN
 USE function

INQUIRE SET member OF SEGMENT NAMES
 ON WORKSTATION
 See INQUIRE SET OF SEGMENT NAMES ON
 WORKSTATION function

INQUIRE SET OF ACTIVE WORKSTATIONS
 function, *Part 2*, 11–135

INQUIRE SET OF ASSOCIATED
 WORKSTATIONS function, *Part 2*,
 11–136

INQUIRE SET OF OPEN WORKSTATIONS
 function, *Part 2*, 11–137

INQUIRE SET OF SEGMENT NAMES IN USE
 function, *Part 2*, 11–138

INQUIRE SET OF SEGMENT NAMES ON
 WORKSTATION function, *Part 2*, 11–139

INQUIRE STRING DEVICE STATE (FORTRAN–
 77 Subset) function, *Part 2*, 11–142

INQUIRE STRING DEVICE STATE 3
 (FORTRAN–77 Subset) function, *Part 2*,
 11–146

INQUIRE STRING DEVICE STATE 3 function,
 Part 2, 11–144

INQUIRE STRING DEVICE STATE function,
 Part 2, 11–140
 example, *Part 1*, 9–108

INQUIRE STROKE DEVICE STATE 3 function,
 Part 2, 11–150

INQUIRE STROKE DEVICE STATE function,
 Part 2, 11–148

INQUIRE TEXT ALIGNMENT function, *Part 2*,
 11–152

INQUIRE TEXT BASE VECTOR
 See INQUIRE CHARACTER BASE VECTOR
 function

INQUIRE TEXT COLOUR INDEX function, *Part*
 2, 11–153

INQUIRE TEXT EXPANSION FACTOR
 See INQUIRE CHARACTER EXPANSION
 FACTOR function

INQUIRE TEXT EXTENT (FORTRAN–77 Subset)
 function, *Part 2*, 11–155

INQUIRE TEXT EXTENT 3 (FORTRAN–77
 Subset) function, *Part 2*, 11–157

INQUIRE TEXT EXTENT 3 function, *Part 2*,
 11–156

INQUIRE TEXT EXTENT function, *Part 2*,
 11–154

INQUIRE TEXT FACILITIES function, *Part 2*,
 11–158

INQUIRE TEXT FONT AND PRECISION
 function, *Part 2*, 11–160

INQUIRE TEXT HEIGHT
 See INQUIRE CHARACTER HEIGHT function

INQUIRE TEXT INDEX function, *Part 2*, 11–162

INQUIRE TEXT PATH function, *Part 2*, 11–163

INQUIRE TEXT REPRESENTATION function,
 Part 2, 11–164

INQUIRE TEXT SPACING
 See INQUIRE CHARACTER SPACING function

INQUIRE TEXT UP VECTOR
 See INQUIRE CHARACTER UP VECTOR
 function

INQUIRE TEXT WIDTH
 See INQUIRE CHARACTER WIDTH function

INQUIRE VALUATOR DEVICE STATE 3 function,
 Part 2, 11–168

INQUIRE VALUATOR DEVICE STATE function,
 Part 2, 11–166
 example, *Part 1*, 9–111

INQUIRE VIEW FACILITIES function, *Part 2*,
 11–170

INQUIRE VIEW REPRESENTATION 3 function,
 Part 2, 11–171

INQUIRE WORKSTATION CATEGORY function,
 Part 2, 11–173

INQUIRE WORKSTATION CLASSIFICATION
 function, *Part 2*, 11–174

INQUIRE WORKSTATION CONNECTION AND
 TYPE function, *Part 2*, 11–175
 example, *Part 1*, 4–32

INQUIRE WORKSTATION DEFERRAL AND
 UPDATE STATES function, *Part 2*, 11–176

INQUIRE WORKSTATION MAXIMUM
 NUMBERS function, *Part 2*, 11–178

INQUIRE WORKSTATION STATE function, *Part*
 2, 11–179

INQUIRE WORKSTATION TRANSFORMATION
 3 function, *Part 2*, 11–182

INQUIRE WORKSTATION TRANSFORMATION

function, *Part 2*, 11–180

Inquiry functions, *Part 2*, 11–4 to 11–183

input use, *Part 1*, 9–22

introduction to, *Part 2*, 11–1 to 11–4

Inserting segments, *Part 1*, 8–3

INSERT SEGMENT 3 function, *Part 1*, 8–19

INSERT SEGMENT function, *Part 1*, 8–17

example, *Part 1*, 8–34

Interface

prompt and echo types, *Part 1*, 9–5 to 9–16

Interior styles

See also Attributes

See also Hatches

See also Patterns

of fill areas, *Part 1*, 6–27

Interpret

metafiles, *Part 1*, 10–1

INTERPRET ITEM function, *Part 1*, 10–7

Invalid index flags

list of, *Part 2*, B–9

Items

metafile header, *Part 1*, 10–2

K

Kernel

GKS, *Part 1*, 4–1

L

Last event flags

list of, *Part 2*, B–9

Lengths

See also Data records

See also Input

input data record, *Part 1*, 9–8

metafile data record, *Part 1*, 10–4

Levels

of GKS, *Part 1*, 1–4

Line cap styles

list of, *Part 2*, B–9

Line join styles

list of, *Part 2*, B–9

Lines

See also Attributes

See also Output

generating, *Part 1*, 5–16, 5–17

types, *Part 1*, 1–3, 6–34

width, *Part 1*, 6–35

Line types

list of, *Part 2*, B–9

Line types (standard)

list of, *Part 2*, B–9

Linking, *Part 1*, 2–2, 3–1

DEC FORTRAN programs, *Part 1*, 3–2

FORTRAN, *Part 1*, 3–3

reducing time, *Part 1*, 3–8

Lists

See also GKS

See also Input

See also Workstations

GKS state, *Part 2*, 11–1

segment state, *Part 2*, 11–1

viewport input priority, *Part 1*, 7–6, 9–21

workstation state, *Part 2*, 11–1

Locator

input class, *Part 1*, 9–4

viewport input priority, *Part 1*, 7–6, 9–21

Logging

errors, *Part 2*, 12–5

Logical device number

See Device number

Logical input devices, *Part 1*, 9–1 to 9–3

See also Input

activating, *Part 1*, 9–2, 9–3

classes, *Part 1*, 9–1

controlling the appearance of, *Part 1*, 9–2

deactivating, *Part 1*, 9–2, 9–3

device number, *Part 1*, 9–1

initializing, *Part 1*, 9–3

triggering, *Part 1*, 9–3

workstation identifier, *Part 1*, 9–1

Logical names, *Part 1*, 2–3

defining

at DCL level, *Part 1*, 2–3

general, *Part 1*, 2–3

GKS\$ASF, *Part 1*, 2–3

GKS\$CONID, *Part 1*, 2–3

GKS\$DEF_MODE, *Part 1*, 2–3

GKS\$ERRFILE, *Part 1*, 2–4

GKS\$ERROR, *Part 1*, 2–4

GKS\$IRG, *Part 1*, 2–4

GKS\$METAFILE_TYPE, *Part 1*, 2–4

GKS\$NDC_CLIP, *Part 1*, 2–4

GKS\$STROKE_FONT1, *Part 1*, 2–4

GKS\$WSTYPE, *Part 1*, 2–4

search order, *Part 1*, 2–3

types, *Part 1*, 2–3

VMS

default, *Part 1*, 2–2

GKS\$CONID, *Part 1*, 2–2

GKS\$ERRFILE, *Part 1*, 2–5

GKS\$WSTYPE, *Part 1*, 2–2

M

Mapping

See also Transformations

color indexes, *Part 1*, 5–4, 5–6

device transformations, *Part 1*, 7–7

Markers, *Part 1*, 5–18, 5–19

See also Attributes

See also Output

size, *Part 1*, 6–36

Markers (cont'd)

types, *Part 1*, 6–37

Marker types

list of, *Part 2*, B–10

Marker types (standard)

list of, *Part 2*, B–10

Matrix

See also Rotation

See also Scale

See also Translation

segment transformation, *Part 1*, 8–8

Matrixes

view mapping, *Part 1*, 7–7

view orientation, *Part 1*, 7–7

Measure

See also Logical input devices

cycling input device control, *Part 1*, 9–17

Menus

See also Choice

input, *Part 1*, 9–4

MESSAGE (FORTRAN–77 Subset) function, *Part 1*, 4–21

MESSAGE function, *Part 1*, 4–20

example, *Part 1*, 9–100

Messages

See also Errors

error, *Part 2*, A–1 to A–37

logging errors, *Part 2*, 12–5

produced by error handler, *Part 2*, 12–2

sent to workstations, *Part 1*, 4–20, 4–21

Metafile functions, *Part 1*, 1–4, 10–5 to 10–10

introduction to, *Part 1*, 10–1 to 10–5

Metafiles, *Part 1*, 10–1

creating, *Part 1*, 10–1

creating CGM metafiles, *Part 1*, 10–3

current item, *Part 1*, 10–4

item header, *Part 1*, 10–2

list of errors, *Part 2*, A–14 to A–15

reading, *Part 1*, 10–4 to 10–5

reproducing pictures, *Part 1*, 10–1

reserved data numbers, *Part 1*, 10–5

structure, *Part 1*, 10–2

user-defined data, *Part 1*, 10–5

workstation categories, *Part 1*, 4–2

Mirror images

2D cell arrays, *Part 1*, 5–4

3D cell arrays, *Part 1*, 5–6

Mode

See also Input

control

SET CHOICE MODE function, *Part 1*, 9–92

SET LOCATOR MODE function, *Part 1*, 9–93

SET PICK MODE function, *Part 1*, 9–94

SET STRING MODE function, *Part 1*, 9–95

Mode

control (cont'd)

SET STROKE MODE function, *Part 1*, 9–96

SET VALUATOR MODE function, *Part 1*, 9–97

event, *Part 1*, 9–2, 9–3, 9–19

AWAIT EVENT function, *Part 1*, 9–24

FLUSH DEVICE EVENTS function, *Part 1*, 9–26

GET CHOICE function, *Part 1*, 9–27

GET LOCATOR 3 function, *Part 1*, 9–29

GET LOCATOR function, *Part 1*, 9–28

GET PICK function, *Part 1*, 9–30

GET STRING function, *Part 1*, 9–31

GET STROKE 3 function, *Part 1*, 9–35

GET STROKE function, *Part 1*, 9–33

GET VALUATOR function, *Part 1*, 9–37

input operating, *Part 1*, 9–2, 9–3, 9–16

request, *Part 1*, 9–2, 9–3, 9–17

REQUEST CHOICE function, *Part 1*, 9–68

REQUEST LOCATOR 3 function, *Part 1*, 9–70

REQUEST LOCATOR function, *Part 1*, 9–69

REQUEST PICK function, *Part 1*, 9–71

REQUEST STRING function, *Part 1*, 9–72

REQUEST STROKE 3 function, *Part 1*, 9–78

REQUEST STROKE function, *Part 1*, 9–76

REQUEST VALUATOR function, *Part 1*, 9–80

sample, *Part 1*, 9–2, 9–3, 9–18

SAMPLE CHOICE function, *Part 1*, 9–81

SAMPLE LOCATOR 3 function, *Part 1*, 9–83

SAMPLE LOCATOR function, *Part 1*, 9–82

SAMPLE PICK function, *Part 1*, 9–84

SAMPLE STRING function, *Part 1*, 9–85

SAMPLE STROKE 3 function, *Part 1*, 9–89

SAMPLE STROKE function, *Part 1*, 9–87

SAMPLE VALUATOR function, *Part 1*, 9–91

Model

color, *Part 1*, 6–15

Multiple transformations

See also Segments

See also Transformations

N

Names

error messages, *Part 2*, 12–2

segment, *Part 1*, 8–1

NDC

See also Transformations

See Normalized device coordinates

NDC (cont'd)
 fixed points, *Part 1*, 8-7
New frame necessary at update entry, *Part 1*, 8-4
 New frame action states
 list of, *Part 2*, B-10
 Nominal sizes, *Part 1*, 6-1
 Nongeometric attributes, *Part 1*, 6-1
 See also Attributes
 Normalization
 clipping, *Part 1*, 7-4
 overlapping viewports, *Part 1*, 7-6
 transformations, *Part 1*, 1-3
 maximum number, *Part 1*, 7-5
 viewports, *Part 1*, 7-4
 windows, *Part 1*, 7-3
 Normalization transformations
 See also Transformations
 See Transformations
 Normalized device coordinates, *Part 1*, 7-1
 Normalized projection coordinates, *Part 1*, 7-1,
 7-7
 NPC
 See Normalized projection coordinates
 Numbers
 See also Errors
 See also Input
 error, *Part 2*, A-1
 error messages
 handling, *Part 2*, 12-2

O

OFF
 error state, *Part 2*, 12-1
 Offset
 cell array, *Part 1*, 5-4
 ON
 error state, *Part 2*, 12-1
 One-to-one
 See also Mapping
 OPEN GKS function, *Part 1*, 4-22
 example, *Part 1*, 4-30
 Opening
 GKS, *Part 1*, 4-4
 GKSM metafile workstations, *Part 1*, 10-1
 segments, *Part 1*, 4-5, 8-1
 workstations, *Part 1*, 4-4
 Opening a workstation, *Part 1*, 2-2, 3-3 to 3-4
 OPEN WORKSTATION function, *Part 1*, 4-23
 example, *Part 1*, 4-30
 Operating modes
 input, *Part 1*, 9-2, 9-3, 9-16 to 9-21
 Operating states, *Part 1*, 4-3
 list of errors, *Part 2*, A-1 to A-2
 using output, *Part 1*, 5-1

Operating system
 ULTRIX, *Part 1*, 3-1
 Order
 See also Transformations
 viewport input priority, *Part 1*, 7-6
 Origin
 See also Transformations
 world coordinate system, *Part 1*, 7-3
 Output
 See also Attributes
 altering the primitive, *Part 1*, 5-2
 attribute functions
 See Attribute functions, *Part 1*, 6-1
 attributes, *Part 1*, 1-3, 5-2
 bound attributes, *Part 1*, 6-1
 default windows and viewports, *Part 1*, 5-2
 deferral, *Part 1*, 4-6, 5-3
 DECwindows, *Part 1*, 1-7
 list of errors, *Part 2*, A-10 to A-11
 list of primitives, *Part 1*, 1-2
 lost during transformations, *Part 1*, 7-8
 metafiles, *Part 1*, 10-1, 10-2
 pick identification, *Part 1*, 8-2
 pictures, *Part 1*, 7-1
 segments, *Part 1*, 8-1
 valid operating states, *Part 1*, 5-1
 workstation categories, *Part 1*, 4-2, 5-1
 Output attributes
 See Attributes
 Output functions, *Part 1*, 5-1 to 5-31
 introduction to, *Part 1*, 5-1 to 5-3
 Overflow
 event input queue, *Part 1*, 9-20
 Overlapping
 See also Transformations
 segments, *Part 1*, 8-6
 viewports, *Part 1*, 7-6, 9-21

P

PACK DATA RECORD (FORTRAN-77 Subset)
 function, *Part 1*, 9-67
 PACK DATA RECORD function, *Part 1*, 9-66
 Packing input data records, *Part 1*, 9-8
 Parallel projection, *Part 1*, 7-7
 Passing mechanisms
 arguments, *Part 1*, 1-6
 Pasteboard
 See also Transformations
 normalization viewport, *Part 1*, 7-5
 Path
 See also Text
 text, *Part 1*, 6-57
 Patterns, *Part 1*, 6-27
 See also Attributes
 fill areas, *Part 1*, 5-8, 5-9
 reference points, *Part 1*, 6-38

Patterns (cont'd)

- reference point vector, *Part 1*, 6–39
- representation, *Part 1*, 6–40
- specifying size, *Part 1*, 6–41
- style index values, *Part 1*, 6–31

Pending

- See also Implicit regenerations
- bundle changes, *Part 1*, 4–7
- output generation, *Part 1*, 4–6
- segment attribute changes, *Part 1*, 4–7
- workstation transformations, *Part 1*, 4–7

Perspective projection, *Part 1*, 7–7

Physical input devices, *Part 1*, 9–1

pi, *Part 1*, 8–8

Pick

- See also Input
- See also Segments
- identifier, *Part 1*, 8–2, 9–4
- input class, *Part 1*, 9–4
- segment detectability, *Part 1*, 8–5
- specifying NOPICK input, *Part 1*, 9–18, 9–19
- visibility, *Part 1*, 8–10

Pick status types

- list of, *Part 2*, B–10

Pictures

- See also Output
- See also Transformations
- composition, *Part 1*, 1–3, 7–1
- reproducing
 - metafiles, *Part 1*, 10–1

Pipeline

- See also Segments

Plotting

- See also Transformations
- pictures, *Part 1*, 7–1

Pointers

- See also Bundles
- into bundle tables, *Part 1*, 6–3

Points

- See also Transformations
- coordinate, *Part 1*, 7–1
- pattern reference, *Part 1*, 6–38, 6–39
- segments
 - fixed points, *Part 1*, 8–7
- viewport input priority, *Part 1*, 7–6

Polygons

- See also Attributes
- See also Output
- fill areas, *Part 1*, 5–8, 5–9
- using GPL, *Part 1*, 5–16
- using GPL3, *Part 1*, 5–17

Polyline

- See also Attributes
- See also Output
- bundles, *Part 1*, 6–44
- line type, *Part 1*, 1–3
- representation, *Part 1*, 6–45

Polyline (cont'd)

- type, *Part 1*, 6–34
- POLYLINE 3 function, *Part 1*, 5–17
- POLYLINE function, *Part 1*, 5–16
 - example, *Part 1*, 6–67

Polylines

- initial attributes, *Part 2*, E–1

Polymarker

- See also Output

- See also Transformations

attributes

SET MARKER SIZE SCALE FACTOR

- function, *Part 1*, 6–36

SET MARKER TYPE function, *Part 1*, 6–37

SET POLYMARKER COLOUR INDEX function, *Part 1*, 6–47

SET POLYMARKER INDEX function, *Part 1*, 6–48

- bundle table, *Part 1*, 6–48

- representation, *Part 1*, 6–49

POLYMARKER 3 function, *Part 1*, 5–19

POLYMARKER function, *Part 1*, 5–18

- example, *Part 1*, 6–69

Polymarkers

- initial attributes, *Part 2*, E–2

Positioning

- primitives, *Part 1*, 7–5

Precision

- text, *Part 1*, 6–54

Presentation

- See also Transformations

- pictures, *Part 1*, 7–7

Primitives

- See also Attributes

- See also Output

- attributes, *Part 1*, 6–1

- bound attributes, *Part 1*, 6–1

- clipping segments, *Part 1*, 8–9

- highlighting, *Part 1*, 8–6

- input prompt and echo types, *Part 1*, 9–5

- list, *Part 1*, 1–2

- lost during regeneration, *Part 1*, 4–7

- lost during transformations, *Part 1*, 7–8

- output, *Part 1*, 5–1 to 5–3

- pick identification, *Part 1*, 8–2

- reproducing

- metafiles, *Part 1*, 10–1

- segment detectability, *Part 1*, 8–5

- segments, *Part 1*, 8–1

- transformation, *Part 1*, 7–3

Priority

- See also Input

- segments, *Part 1*, 8–6

- viewport input, *Part 1*, 7–6, 9–21

Program example, *Part 2*, C-1 to C-3
Programming
 See also GKS
 device-independent input, *Part 1*, 9-22
 error handling, *Part 2*, 12-1
Programs
 execution of, *Part 1*, 2-2, 3-1
 pausing, *Part 1*, 1-7
Projections
 parallel, *Part 1*, 7-7
 perspective, *Part 1*, 7-7
Projection types
 list of, *Part 2*, B-10
Prompt and echo types, *Part 1*, 9-2, 9-5 to 9-16
 See also Input
 standard data records, *Part 1*, 9-8
Proportionate
 See also Transformations

Q

Queue
 event input, *Part 1*, 9-19

R

Radians
 translating to degrees, *Part 1*, 8-8
Ranges
 See also Transformations
 windows and viewports, *Part 1*, 7-3
Ratio
 See also Transformations
Reading a metafile, *Part 1*, 10-4
READ ITEM FROM GKSM function, *Part 1*, 10-8
READ ITEM FROM METAFILE function
 See READ ITEM FROM GKSM function
Realized values, *Part 2*, 11-4
Real numbers
 input, *Part 1*, 9-4
Records
 See also Escapes
 See also GDPs
 See also Input
 input, *Part 1*, 9-8, 9-9
 prompt and echo types, *Part 1*, 9-5 to 9-16
 standard, *Part 1*, 9-8
Rectangles
 See also Attributes
 See also Transformations
 clipping, *Part 1*, 7-4
 segments, *Part 1*, 8-9
REDRAW ALL SEGMENTS ON WORKSTATION
 function, *Part 1*, 4-25
 example, *Part 1*, 8-29

Regeneration flag states
 list of, *Part 2*, B-10
Regenerations
 segments, *Part 1*, 8-3
 workstation surface, *Part 1*, 4-7
 workstation transformations, *Part 1*, 7-8
Relative input priority
 list of, *Part 2*, B-11
Releasing
 DEC GKS buffers, *Part 1*, 4-11
RENAME SEGMENT function, *Part 1*, 8-21
Renaming
 segments, *Part 1*, 8-1
Reports
 current event on input queue, *Part 1*, 9-19
Representation
 See also Attributes
 bundle table entries, *Part 1*, 6-3
 color, *Part 1*, 6-16
 edge, *Part 1*, 6-21
 fill area, *Part 1*, 6-29
 functions, *Part 1*, 6-4
 implicit regenerations, *Part 1*, 6-4
 pattern, *Part 1*, 6-40
 polyline, *Part 1*, 6-45
 polymarker, *Part 1*, 6-49
 text, *Part 1*, 6-59
Reproducing
 metafiles, *Part 1*, 10-1
REQUEST CHOICE function, *Part 1*, 9-68
REQUEST functions, *Part 1*, 9-2, 9-3, 9-17
REQUEST LOCATOR 3 function, *Part 1*, 9-70
REQUEST LOCATOR function, *Part 1*, 9-69
Request mode, *Part 1*, 9-17 to 9-18
 See also Input
 breaking, *Part 1*, 9-18
REQUEST PICK function, *Part 1*, 9-71
Request status types
 list of, *Part 2*, B-11
REQUEST STRING (FORTRAN-77 Subset)
 function, *Part 1*, 9-74
REQUEST STRING function, *Part 1*, 9-72
 example, *Part 1*, 9-108
REQUEST STROKE 3 function, *Part 1*, 9-78
REQUEST STROKE function, *Part 1*, 9-76
REQUEST VALUATOR function, *Part 1*, 9-80
Returned type values
 list of, *Part 2*, B-11
Reverse video
 highlighting segments, *Part 1*, 8-6
Rotation
 fixed points, *Part 1*, 8-7
 segments, *Part 1*, 8-7
RUN DCL command, *Part 1*, 2-2

S

SAMPLE CHOICE function, *Part 1*, 9–81
SAMPLE functions, *Part 1*, 9–18
SAMPLE LOCATOR 3 function, *Part 1*, 9–83
SAMPLE LOCATOR function, *Part 1*, 9–82
Sample mode, *Part 1*, 9–18 to 9–19
SAMPLE PICK function, *Part 1*, 9–84
 example, *Part 1*, 9–103
SAMPLE STRING (FORTRAN–77 Subset)
 function, *Part 1*, 9–86
SAMPLE STRING function, *Part 1*, 9–85
SAMPLE STROKE 3 function, *Part 1*, 9–89
SAMPLE STROKE function, *Part 1*, 9–87
SAMPLE VALUATOR function, *Part 1*, 9–91
 example, *Part 1*, 9–111
Scale
 See also Segments
 edge width factor, *Part 1*, 6–24
 fixed points, *Part 1*, 8–7
 segments, *Part 1*, 8–7
 valuator input, *Part 1*, 9–4
Scale factors, *Part 1*, 6–1
Scratch pad
 See also Transformations
 normalization window, *Part 1*, 7–5
Segment functions, *Part 1*, 8–1 to 8–42
 introduction to, *Part 1*, 8–1 to 8–10
Segments
 accumulated transformations, *Part 1*, 8–9
 associating, *Part 1*, 8–3
 attributes, *Part 1*, 8–5
 SET DETECTABILITY function, *Part 1*,
 8–22
 SET HIGHLIGHTING function, *Part 1*,
 8–23
 SET SEGMENT PRIORITY function, *Part*
 1, 8–24
 SET VISIBILITY function, *Part 1*, 8–28
 clipping, *Part 1*, 8–9
 closing, *Part 1*, 4–5
 copying, *Part 1*, 8–3
 creating, *Part 1*, 4–5, 8–1
 deleting, *Part 1*, 8–2
 detectability, *Part 1*, 8–5
 highlighting, *Part 1*, 8–6
 initial attributes, *Part 2*, E–4
 input, *Part 1*, 8–2
 inserting, *Part 1*, 8–3
 list of errors, *Part 2*, A–11 to A–12
 metafiles, *Part 1*, 10–2
 names, *Part 1*, 8–1
 opening, *Part 1*, 4–5, 8–1
 order of transformation, *Part 1*, 8–9
 overlapping, *Part 1*, 8–6
 priority, *Part 1*, 8–6
 redrawn, *Part 1*, 4–7

Segments (cont'd)

renaming, *Part 1*, 8–1
rotating, *Part 1*, 8–7
scaling, *Part 1*, 8–7
selecting a transformation, *Part 1*, 8–8
state list, *Part 1*, 8–1
storage, *Part 1*, 8–2
surface update, *Part 1*, 8–3
transformation, *Part 1*, 7–10, 7–13, 7–15, 7–17,
 8–7 to 8–9
ACCUMULATE TRANSFORMATION
 MATRIX 3 function, *Part 1*, 7–13
ACCUMULATE TRANSFORMATION
 MATRIX function, *Part 1*, 7–10
EVALUATE TRANSFORMATION MATRIX
 3 function, *Part 1*, 7–17
EVALUATE TRANSFORMATION MATRIX
 function, *Part 1*, 7–15
EVALUATE VIEW MAPPING MATRIX 3
 function, *Part 1*, 7–19
EVALUATE VIEW ORIENTATION
 MATRIX 3 function, *Part 1*, 7–22
SET SEGMENT TRANSFORMATION 3
 function, *Part 1*, 8–26
SET SEGMENT TRANSFORMATION
 function, *Part 1*, 8–25
transformation matrix, *Part 1*, 8–8
translating, *Part 1*, 8–7
visibility, *Part 1*, 8–10
WDSS, *Part 1*, 8–2
WISS, *Part 1*, 8–3
SELECT NORMALIZATION TRANSFORMATION
 function, *Part 1*, 7–24
 example, *Part 1*, 7–48
SET ASPECT SOURCE FLAGS 3 function, *Part*
 1, 6–9
SET ASPECT SOURCE FLAGS function, *Part 1*,
 6–7
 example, *Part 1*, 6–64
SET CHARACTER EXPANSION FACTOR
 function, *Part 1*, 6–11
SET CHARACTER HEIGHT function, *Part 1*,
 6–12
 example, *Part 1*, 6–69
SET CHARACTER SPACING function, *Part 1*,
 6–13
SET CHARACTER UP VECTOR function, *Part 1*,
 6–14
SET CHOICE MODE function, *Part 1*, 9–92
SET CLIPPING INDICATOR function, *Part 1*,
 7–25
 example, *Part 1*, 7–48
SET COLOUR MODEL function, *Part 1*, 6–15
SET COLOUR REPRESENTATION function,
 Part 1, 6–16
 example, *Part 1*, 6–61

Set DEC GKS Connection Identifier String escape, *Part 1*, 4–24

SET DEFERRAL STATE function, *Part 1*, 4–26
example, *Part 1*, 5–27

SET DETECTABILITY function, *Part 1*, 8–22
example, *Part 1*, 9–103

SET EDGE COLOUR INDEX function, *Part 1*, 6–18

SET EDGE FLAG function, *Part 1*, 6–19

SET EDGE INDEX function, *Part 1*, 6–20

SET EDGE REPRESENTATION function, *Part 1*, 6–21

SET EDGETYPE function, *Part 1*, 6–23

SET EDGEWIDTH SCALE FACTOR function, *Part 1*, 6–24

SET FILL AREA COLOUR INDEX function, *Part 1*, 6–25
example, *Part 1*, 6–61

SET FILL AREA INDEX function, *Part 1*, 6–26
example, *Part 1*, 6–64

SET FILL AREA INTERIOR STYLE function, *Part 1*, 6–27
example, *Part 1*, 6–61

SET FILL AREA REPRESENTATION function, *Part 1*, 6–29
example, *Part 1*, 6–64

SET FILL AREA STYLE INDEX function, *Part 1*, 6–31

SET HIGHLIGHTING function, *Part 1*, 8–23
example, *Part 1*, 8–39

SET HLHSR IDENTIFIER function, *Part 1*, 6–32

SET HLHSR MODE function, *Part 1*, 6–33

SET LINE COLOUR INDEX
See SET POLYLINE COLOUR INDEX function

SET LINE INDEX
See SET POLYLINE INDEX function

SET LINE REPRESENTATION
See SET POLYLINE REPRESENTATION function

SET LINETYPE function, *Part 1*, 6–34
example, *Part 1*, 6–67

SET LINEWIDTH SCALE FACTOR function, *Part 1*, 6–35

SET LOCATOR MODE function, *Part 1*, 9–93
example, *Part 1*, 9–100

SET MARKER COLOUR INDEX
See SET POLYMARKER COLOUR INDEX function

SET MARKER INDEX
See SET POLYMARKER INDEX function

SET MARKER REPRESENTATION
See SET POLYMARKER REPRESENTATION function

SET MARKER SIZE SCALE FACTOR function, *Part 1*, 6–36

SET MARKER TYPE function, *Part 1*, 6–37
example, *Part 1*, 6–69

SET MODE functions, *Part 1*, 9–2, 9–3, 9–16, 9–17, 9–18

SET PATTERN REFERENCE POINT AND VECTORS function, *Part 1*, 6–39

SET PATTERN REFERENCE POINT function, *Part 1*, 6–38

SET PATTERN REPRESENTATION function, *Part 1*, 6–40

SET PATTERN SIZE function, *Part 1*, 6–41

SET PICK IDENTIFIER function, *Part 1*, 6–42
example, *Part 1*, 9–103

SET PICK MODE function, *Part 1*, 9–94
example, *Part 1*, 9–103

SET POLYLINE COLOUR INDEX function, *Part 1*, 6–43

SET POLYLINE INDEX function, *Part 1*, 6–44

SET POLYLINE REPRESENTATION function, *Part 1*, 6–45

SET POLYLINE TYPE
See SET LINETYPE function

SET POLYLINE WIDTH SCALE FACTOR
See SET LINEWIDTH SCALE FACTOR function

SET POLYMARKER COLOUR INDEX function, *Part 1*, 6–47
example, *Part 1*, 6–69

SET POLYMARKER INDEX function, *Part 1*, 6–48

SET POLYMARKER REPRESENTATION function, *Part 1*, 6–49

SET POLYMARKER SIZE SCALE FACTOR
See SET MARKER SIZE SCALE FACTOR function

SET POLYMARKER TYPE
See SET MARKER TYPE function

SET SEGMENT PRIORITY function, *Part 1*, 8–24

SET SEGMENT TRANSFORMATION 3 function, *Part 1*, 8–26

SET SEGMENT TRANSFORMATION function, *Part 1*, 8–25
example, *Part 1*, 7–38

SET STRING MODE function, *Part 1*, 9–95

SET STROKE MODE function, *Part 1*, 9–96

SET TEXT ALIGNMENT function, *Part 1*, 6–51
example, *Part 1*, 6–69

SET TEXT COLOUR INDEX function, *Part 1*, 6–53

SET TEXT EXPANSION FACTOR function
See SET CHARACTER EXPANSION FACTOR function

SET TEXT FONT AND PRECISION function, *Part 1*, 6–54

SET TEXT HEIGHT function
See SET CHARACTER HEIGHT function

SET TEXT INDEX function, *Part 1*, 6–56
 SET TEXT PATH function, *Part 1*, 6–57
 example, *Part 1*, 6–69
 SET TEXT REPRESENTATION function, *Part 1*,
 6–59
 SET TEXT SPACING function
 See SET CHARACTER SPACING function
 SET TEXT UP VECTOR function
 See SET CHARACTER UP VECTOR function
 Settings
 See also Attributes
 See also Transformations
 attribute values, *Part 1*, 6–1
 pattern sizes, *Part 1*, 6–41
 segment transformations, *Part 1*, 8–8
 SET VALUATOR MODE function, *Part 1*, 9–97
 example, *Part 1*, 9–111
 Set values, *Part 2*, 11–4
 SET VIEW INDEX function, *Part 1*, 7–26
 SET VIEWPORT 3 function, *Part 1*, 7–30
 SET VIEWPORT function, *Part 1*, 7–29
 example, *Part 1*, 7–48
 SET VIEWPORT INPUT PRIORITY function,
 Part 1, 7–31
 SET VIEW REPRESENTATION 3 function, *Part*
 1, 7–20, 7–27
 SET VIEW TRANSFORMATION INPUT
 PRIORITY function, *Part 1*, 7–28
 SET VISIBILITY function, *Part 1*, 8–28
 SET WINDOW 3 function, *Part 1*, 7–33
 SET WINDOW function, *Part 1*, 7–32
 example, *Part 1*, 7–48
 SET WORKSTATION VIEWPORT 3 function,
 Part 1, 7–35
 SET WORKSTATION VIEWPORT function, *Part*
 1, 7–34
 example, *Part 1*, 7–52
 SET WORKSTATION WINDOW 3 function, *Part*
 1, 7–37
 SET WORKSTATION WINDOW function, *Part 1*,
 7–36
 Shift segments, *Part 1*, 8–7
 Shrink segments, *Part 1*, 8–7
 Simultaneous event flags
 list of, *Part 2*, B–11
 Sizes
 input data record, *Part 1*, 9–22
 markers, *Part 1*, 6–36
 patterns, *Part 1*, 6–41
 segments, *Part 1*, 8–8
 Software fonts, *Part 1*, 6–54
 Solid
 See also Attributes
 fill area interior style, *Part 1*, 6–27
 fill areas, *Part 1*, 5–8, 5–9
 Spacing
 text, *Part 1*, 6–13
 Standards
 See also ANSI
 See also GKS
 Input data records, *Part 1*, 9–9 to 9–16
 metafiles, *Part 1*, 10–1
 State lists
 GKS, *Part 1*, 4–3, 8–1; *Part 2*, 11–1
 attributes, *Part 1*, 6–1
 initializing, *Part 1*, 4–22
 segment, *Part 1*, 4–3, 8–1
 segments, *Part 2*, 11–1
 surface control entries, *Part 1*, 4–8
 workstation, *Part 1*, 4–3; *Part 2*, 11–1
 attributes, *Part 1*, 6–3
 initialization of, *Part 1*, 4–23
 Statements
 include, *Part 1*, 2–1, 3–1
 States
 error, *Part 2*, 12–1
 operating, *Part 1*, 4–3
 Status
 inquiry error status argument, *Part 2*, 11–3
 Storage
 metafiles, *Part 1*, 1–4, 10–1
 segments, *Part 1*, 8–2
 Strings
 See also Text
 input class, *Part 1*, 9–4
 Stroke
 input class, *Part 1*, 9–4
 viewport input priority, *Part 1*, 9–21
 viewport priority, *Part 1*, 7–6
 Structure
 metafiles, *Part 1*, 10–2
 Styles
 See also Attributes
 fill areas, *Part 1*, 6–31
 Surface
 See also Implicit regenerations
 control, *Part 1*, 4–6
 foreground and background colors, *Part 1*, 6–5
 implicit regenerations
 attribute changes, *Part 1*, 6–4
 regeneration, *Part 1*, 4–7
 state list entries, *Part 1*, 4–8
 update
 segments, *Part 1*, 8–3
 Symbols
 polymarkers, *Part 1*, 5–18, 5–19
 Synchronous input, *Part 1*, 9–16
 See also Input
 Syntax
 format, *Part 1*, 1–5

System errors
list of, *Part 2*, A-16 to A-19

T

Tables

See also Attributes
See also Bundles
attribute bundle, *Part 1*, 6-3
color index, *Part 1*, 6-15, 6-16
edge bundle index, *Part 1*, 6-21
fill area bundle index, *Part 1*, 6-29
GKS description, *Part 2*, 11-1
pattern style bundle index, *Part 1*, 6-40
polyline bundle index, *Part 1*, 6-45
polymarker bundle index, *Part 1*, 6-49
text bundle index, *Part 1*, 6-59
workstation description, *Part 2*, 11-1

Terminate

GKS environment, *Part 1*, 4-11
workstation environment, *Part 1*, 4-12

Terminating

error handling, *Part 2*, 12-1
request input, *Part 1*, 9-17

Text

See also Attributes
See also TEXT 3 function
See also TEXT function
alignment, *Part 1*, 6-51
attributes
SET CHARACTER EXPANSION FACTOR
function, *Part 1*, 6-11
SET CHARACTER HEIGHT function,
Part 1, 6-12
SET CHARACTER SPACING function,
Part 1, 6-13
SET CHARACTER UP VECTOR function,
Part 1, 6-14
SET TEXT ALIGNMENT function, *Part 1*,
6-51
SET TEXT FONT AND PRECISION
function, *Part 1*, 6-54
SET TEXT INDEX function, *Part 1*, 6-56
SET TEXT PATH function, *Part 1*, 6-57
bundles, *Part 1*, 6-56
character width, *Part 1*, 6-11
expansion factor, *Part 1*, 6-11
fonts, *Part 1*, 6-54
height, *Part 1*, 6-12
initial attributes, *Part 2*, E-2
input, *Part 1*, 9-4
path, *Part 1*, 6-57
precision, *Part 1*, 6-54
representation, *Part 1*, 6-59
spacing, *Part 1*, 6-13
up-vector, *Part 1*, 6-14

TEXT (FORTRAN-77 Subset) function, *Part 1*,
5-21
TEXT 3 (FORTRAN-77 Subset) function, *Part 1*,
5-25
TEXT 3 function, *Part 1*, 5-23, 5-25
TEXT function, *Part 1*, 5-20, 5-21
example, *Part 1*, 6-69
Text horizontal alignment types
list of, *Part 2*, B-11
Text path types
list of, *Part 2*, B-11
Text precision types
list of, *Part 2*, B-11
Text vertical alignment types
list of, *Part 2*, B-12
Toggling
logical input device control, *Part 1*, 9-17
Transformation functions, *Part 1*, 7-1 to 7-56
introduction to, *Part 1*, 7-1 to 7-8
Transformations
device, *Part 1*, 7-7
identity, *Part 1*, 7-7
identity (segment), *Part 1*, 8-8
implicit regenerations, *Part 1*, 7-8
input change vectors, *Part 1*, 9-5
list of errors, *Part 2*, A-5 to A-6
metafiles, *Part 1*, 10-2
normalization, *Part 1*, 1-3, 6-12, 7-3 to 7-6
clipping, *Part 1*, 7-4
initial attributes, *Part 2*, E-4
maximum number, *Part 1*, 7-5
overlapping viewports, *Part 1*, 7-6
normalization viewports, *Part 1*, 7-4
normalization windows, *Part 1*, 7-3
overlapping viewports, *Part 1*, 9-21
segments, *Part 1*, 8-7 to 8-9
accumulating, *Part 1*, 8-9
fixed points, *Part 1*, 8-7
matrix, *Part 1*, 8-8
unity, *Part 1*, 7-4
used for output, *Part 1*, 5-2
view, *Part 1*, 7-7
viewport input priority, *Part 1*, 9-21
workstation, *Part 1*, 1-3
Translations
segments, *Part 1*, 8-7
viewport input priority, *Part 1*, 7-6
Transporting
metafiles, *Part 1*, 10-1
Transposing
pictures, *Part 1*, 7-4
Triggers
input, *Part 1*, 9-3, 9-18
Type
edge, *Part 1*, 6-23
Types
inquiry value type argument, *Part 2*, 11-4
lines, *Part 1*, 6-34

Types (cont'd)

- markers, *Part 1*, 6–37
- prompt and echo, *Part 1*, 9–5 to 9–16
- workstation
 - metafile, *Part 1*, 10–1
- workstations, *Part 1*, 4–2

U

ULTRIX linking

- DEC FORTRAN programs, *Part 1*, 3–2
- FORTRAN, *Part 1*, 3–3

ULTRIX operating system, *Part 1*, 3–1 to 3–8

Unity transformation, *Part 1*, 7–4

UNPACK DATA RECORD (FORTRAN–77 Subset)

- function, *Part 1*, 9–99

UNPACK DATA RECORD function, *Part 1*, 9–98

Update

- See also Implicit regenerations
- attribute changes, *Part 1*, 6–4
- regenerating the surface, *Part 1*, 4–7
- releasing deferred output, *Part 1*, 4–6
- surface
 - segments, *Part 1*, 8–3
 - the workstation surface, *Part 1*, 4–6

Update states

- list of, *Part 2*, B–12

UPDATE WORKSTATION function, *Part 1*, 4–28

- example, *Part 1*, 4–30

Up-vector

- text, *Part 1*, 6–14

User defined

- error handler, *Part 2*, 12–1

V

Valuator

- input class, *Part 1*, 9–4

Values

- attribute, *Part 1*, 6–1
- initial attribute, *Part 2*, E–1 to E–4
- maximum device coordinates, *Part 1*, 7–8
- of constants, *Part 2*, B–1 to B–14, D–1 to D–9

Vectors

- See also GDPs
- See also Segments
- pattern reference point, *Part 1*, 6–39
- text up-vector, *Part 1*, 6–14
- translation point, *Part 1*, 8–7

View

- reference plane, *Part 1*, 7–23
- transformations, *Part 1*, 7–7
- volume, *Part 1*, 7–20

View mapping matrix, *Part 1*, 7–7

View orientation matrix, *Part 1*, 7–7

View plane, *Part 1*, 7–7

Viewports

- See also Transformations
- input priority, *Part 1*, 7–6, 9–21
- normalization, *Part 1*, 7–4
 - initial value, *Part 2*, E–4
- overlapping, *Part 1*, 7–6, 9–21
- workstation, *Part 1*, 7–7

View reference coordinates, *Part 1*, 7–1

- VRC, *Part 1*, 7–7

View table, *Part 1*, 7–7

Visibility flags

- list of, *Part 2*, B–12

Visibility segments, *Part 1*, 8–10

Visual interface

- See also Input

- input prompt and echo types, *Part 1*, 9–5 to 9–16

VMS logical names

- GKS\$CONID, *Part 1*, 2–2
- GKS\$ERRFILE, *Part 1*, 2–5
- GKS\$WSTYPE, *Part 1*, 2–2

VRC

- See View reference coordinates

W

WDSS, *Part 1*, 8–2

- See also Segments

Width

- See also Attributes
- See also Transformations
- character, *Part 1*, 6–11
- lines, *Part 1*, 6–35

Windows

- See also Transformations
- normalization
 - initial value, *Part 2*, E–4
 - workstation, *Part 1*, 7–7

WISS, *Part 1*, 4–2, 8–3

Workstation category types

- list of, *Part 2*, B–12

Workstation class types

- list of, *Part 2*, B–12

Workstation color availability states

- list of, *Part 2*, B–12

Workstation identifier, *Part 1*, 9–1

Workstations

- activating, *Part 1*, 4–5
- attributes, *Part 1*, 6–1
- clearing the surface, *Part 1*, 4–10
- closing, *Part 1*, 4–5
- deactivating, *Part 1*, 4–5
- definition of, *Part 1*, 4–2
- description tables, *Part 1*, 4–1
- device coordinates, *Part 1*, 7–1
- device manipulation
 - GESC, *Part 1*, 4–14

Workstations (cont'd)

- device number, *Part 1*, 9-1
- environment, *Part 1*, 4-1
 - initialization of, *Part 1*, 4-23
 - terminating, *Part 1*, 4-12
- foreground and background colors, *Part 1*, 6-5
- identifiers
 - input, *Part 1*, 9-1
- implicit regenerations
 - transformations, *Part 1*, 7-8
- list of errors, *Part 2*, A-2 to A-5
- maximum device coordinates, *Part 1*, 7-8
- nominal sizes, *Part 1*, 6-1
- opening, *Part 1*, 4-4
- sending messages to, *Part 1*, 4-20, 4-21
- state list
 - attributes, *Part 1*, 6-3
 - color model, *Part 1*, 6-15
 - color table, *Part 1*, 6-16
 - edge bundle table, *Part 1*, 6-21
 - fill area bundle table, *Part 1*, 6-29
 - pattern style bundle table, *Part 1*, 6-40
 - polyline bundle table, *Part 1*, 6-45
 - polymarker bundle table, *Part 1*, 6-49
 - text bundle table, *Part 1*, 6-59
- stored segments, *Part 1*, 8-1

- surface, *Part 1*, 7-1
- surface control, *Part 1*, 4-6
- surface regeneration, *Part 1*, 4-7
- transformations, *Part 1*, 1-3, 7-7 to 7-8
- types, *Part 1*, 4-2
 - metafile, *Part 1*, 10-1
- update
 - segments, *Part 1*, 8-3

Workstation states

- list of, *Part 2*, B-12

Workstation type

- default, *Part 1*, 2-2, 3-4
- defined, *Part 1*, 2-2, 3-4
- specifying on ULTRIX, *Part 1*, 3-4
- specifying on VMS, *Part 1*, 2-2

Workstation types

- list of, *Part 2*, B-13

World coordinates, *Part 1*, 7-1

See also Transformations

- fixed points, *Part 1*, 8-7

- origin, *Part 1*, 7-3

WRITE ITEM TO GKSM function, *Part 1*, 10-10

Writing modes

- list of, *Part 2*, B-14

Writing to metafiles, *Part 1*, 10-2