American National Standard
for Information Systems -

Computer Graphics -
GKS - MUMPS Language Binding

0 Introduction

The Graphical Kernel System (GKS) is registered as ANSI X3.124-1985. As explained in the Scope and Field of Applications of X3.124, that American National Standard is specified in a language independent manner and needs to be embedded in language dependent layers (language bindings) for use with particular programming languages.

The purpose of this document is to define a standard binding for the MUMPS programming language, ANSI X11.1-1990.

1 Scope and field of application

The Graphical Kernel System (GKS) ANSI X3.124-1985 specifies a language independent nucleus of a graphics system. For integration into a programming language, GKS is embedded in a language dependent layer obeying the particular conventions of that language. This document specifies such a language dependent layer for the MUMPS language, ANSI X11.1-1990.

2 References

ANSI X3.124-1985 American National Standard for Information Systems — Computer Graphics — Graphics Kernel System (GKS) Functional Description.

ANSI X3.124.1-1985 American National Standard for Information Systems — Computer Graphics — Graphics Kernel System (GKS) FORTRAN Binding

ANSI X11.1-1990 American National Standard for Information Systems — Programming Language — MUMPS.

3 Principles

3.1 MUMPS syntax

3.1.1 Metalanguage forms

As defined by X11/92-46 "Device Handling" (MDC Type-A document):

```
                        *     *  !  *
                              *
                        *     *     *
... [ ? intexpr ]       *
                        *     *  #  *
                              *
format            ::=   *
                              *
                        *           ?
intexpr                       *
                        *
                              *
                        *
/controlmnemonic  [  (  L  expr  )  ]
      *

                        *   ?   *   *
alpha *
controlmnemonic   ::=   *       *   *
     * ...
                        * alpha *   *
digit *
```

The GKS-applicable form of the useargument syntax is:

```
useargument       ::=   expr   :  [
deviceparameters ] : mnemonicspace

mnemonicspace     ::=   expr   V
mnemonicspacename

mnemonicspacename ::=   * GKS *
```

(Note that this is the syntax that applies to GKS — other forms of the useargument are used for other device types)

The following metalanguage elements will be used to describe the GKS parameters:

```
      gksnam      ::=   expr V strlit

      gksfile     ::=   expr V strlit

      gksstr      ::=   expr V strlit

      gksnum      ::=   numexpr

      gksint      ::=   intexpr

      gksenum     ::=   expr V strlit

      gksdat      ::=   expr V strlit

      gkspnt      ::=   L gksint

      gksplst     ::=   L gkspnt
```

Certain GKS controlmnemonics return a value or collection of values. It is permitted to issue these controlmnemonics with either the WRITE or the READ statement. If a READ statement is used, the argument list in that statement and/or subsequent statements must be ordered to correctly accept the returned values. If a WRITE statement is used, the values returned may be read from the buffer by a single, or series of, READ statements. These READs must be correctly ordered to match the returned values. However, there may be intermediate calculations utilizing some of the returned values before reading the remaining values in the list.

Reading the return list of values may be terminated without error by issuing another GKS controlmnemonic. In this case, all return values not yet assigned to a variable will be lost to the application program.

Certain GKS controlmnemonics result in a collection of values being returned -- the number of values actually returned is specified by some of the initial values returned. The following metalanguage syntax will represent this construct:

v1 ::= gksint	number of segment names (*n*)
v2 ::= M gksnam	*n* values; each a segment name

v1 in this definition would represent the number of values to be specified by v2, and would be performed by the following example:

```
   Read V1 For II=1:1:V1 Read V2(II)
```

3.1.2 Application program environment

In MUMPS routines, devices will be opened using the GKS mnemonicspace parameter. GKS directives are sent to GKS devices using controlmnemonics combined with the READ and WRITE commands.

3.1.3 Side effects

Device-related svns (such as $DEVICE, $KEY, $X, $Y) are not modified by GKS controlmnemonics or by their return value(s).

3.2 MUMPS controlmnemonics and GKS functions

Each GKS function is listed in the following section, grouped together by function type. Each function has an associated controlmnemonic which is the means by which it is invoked, a level, which indicates which versions of GKS will support the function, and the GKS function name which is the standard GKS description for the function.

See the index for the list of GKS functions sorted by function name or by controlmnemonic.

Each GKS function is presented with an example of how to use it in MUMPS, these lines begin with '»'. In addition, GKS functions which return values include an example on receiving those values, these lines begin with '«'.

4 GKS function bindings

4.1 GKS management functions

control- mnemonic	Level	GKS function name

4.1.1 GOPKS ma Open GKS

2 parameters

p1 ::= gksfile	error file
p2 ::= gksint	amount of memory units for buffer area

```
» W /GOPKS("error.fil",10240)
```

4.1.2 GCLKS ma Close GKS

No parameters

```
» W /GCLKS
```

4.1.3 GOPWK ma Open workstation

3 parameters

p1 ::= gksnam workstation ID

p2 ::= gksnam connection ID

p3 ::= gksenum workstation type:

```
* OUTPUT *
* INPUT  *
* OUTIN  *
* WISS   *
* MO     *
* MI     *
```

```
» W /GOPWK("WORK1",255,"OUTPUT")
```

4.1.4 GCLWK ma Close Workstation

1 parameter

p1 ::= gksnam workstation ID

```
» W /GCLWK("WORK1")
```

4.1.5 GACWK ma Activate Workstation

1 parameter

p1 ::= gksnam workstation ID

```
» W /GACWK("WORK1")
```

4.1.6 GDAWK ma Deactivate workstation

1 parameter

p1 ::= gksnam workstation ID

```
» W /GDAWK("WORK1")
```

4.1.7 GCLRWK ma Clear workstation

2 parameters

p1 ::= gksnam workstation ID

p2 ::= gksenum control flag:

```
* CONDITIONALLY *
* ALWAYS        *
```

```
» W /GCLRWK("WORK1","ALWAYS")
```

4.1.8 GRSGWK 1a Redraw all segments on workstation

1 parameter

p1 ::= gksnam workstation ID

» W /GRSGWK("WORK1")

4.1.9 GUWK ma Update workstation

2 parameters

p1 ::= gksnam workstation ID

p2 ::= gksenum control flag:

```
* PERFORM  *
* POSTPONE *
```

» W /GUWK("WORK1","PERFORM")

4.1.10 GSDS 1a Set default state

3 parameters

p1 ::= gksnam workstation ID

p2 ::= gksenum deferral mode:

```
* ASAP *
* BNIG *
* BNIL *
* ASTI *
```

p3 ::= gksenum implicit regeneration mode:

```
* SUPPRESSED *
* ALLOWED    *
```

» W /GSDS("WORK1","ASAP","ALLOWED")

4.1.11 GMSG 1a Message

2 parameters

p1 ::= gksnam workstation ID

p2 ::= gksstr message text

» W /GMSG("WORK1","MESSAGE")

4.1.12 GESC ma Escape

1 parameter

p1 ::= gksdat input data

» W /GESC("special request:12,13")

1 output value

v1 ::= gksdat output data

« R RESULT

4.2 Output functions

	control- mnemonic	Level	GKS function name
4.2.1	GPL	ma	Polyline

2 parameters

p1 ::= gksint number of points (2..n)

p2 ::= gksplst n points (WC)

» W /GPL(2,"1,1,100,100")

4.2.2	GPM	ma	Polymarker

2 parameters

p1 ::= gksint number of points (1..n)

p2 ::= gksplst n points (WC)

» W /GPM(2."1,1,100,100")

4.2.3	GTX	ma	Text

2 parameters

p1 ::= gkspnt point for text position (WC)

p2 ::= gksstr text string

» W /GTX("100,100","End of Line")

4.2.4	GFA	ma	Fill area

2 parameters

p1 ::= gksint number of points (3..n)

p2 ::= gksplst n points (WC)

» W /GFA(3,"1,1,100,100,100,1")

4.2.5 GCA 0a Cell array
4 parameters

p1 ::= gksplst 2 points defining rectangle (WC)

p2 ::= gksint Cell width (1..x)

p3 ::= gksint Cell height (1..y)

p4 ::= L gksint x*y list of integer values describing colors

» W /GCA("1,1,100,100",2,2,"1,1,1,1")

4.2.6 GGDP 0a Generalized drawing primative
4 parameters

p1 ::= gksint number of points (0..n)

p2 ::= gksplst n points (WC)

p3 ::= gksnam GDP identifier

p4 ::= gksdat GDP data record

» W /GDP(2,"1,1,100,100","SPIRAL","width:5,length:5")

4.3 Output attributes

	control- mnemonic	Level	GKS function name
4.3.1	GSPLI 1 parameter	0a	Set polyline index
	p1 ::= gksint		polyline index (1..n)
	» W /GSPLI(1)		
4.3.2	GSLN 1 parameter	ma	Set linetype
	p1 ::= gksint		line type
	» W /GSLN(5)		
4.3.3	GSLWSC 1 parameter	0a	Set linewidth scale factor
	p1 ::= gksnum		line width scale factor
	» W /GSLWSC(1.5)		
4.3.4	GSPLCI 1 parameter	ma	Set polyline colour index
	p1 ::= gksint		polyline colour index (0..n)
	» W /GSPLCI(12)		
4.3.5	GSPMI 1 parameter	0a	Set polymarker index
	p1 ::= gskint		polymarker index (1..n)
	» W /GSPMI(2)		
4.3.6	GSMK 1 parameter	0a	Set marker type
	p1 ::= gksint		marker type (-n..-1,1..n)
	» W /GSMK(3)		

4.3.7 GSMKSC 0a Set marker size scale factor
1 parameter

p1 ::= gksnum marker size scale factor

» W /GSMKSC(1.5)

4.3.8 GSPMCI ma Set polymarker colour index
1 parameter

p1 ::= gksint polymarker colour index (0..n)

» W /GSPMCI(9)

4.3.9 GSTXI 0a Set text index
1 parameter

p1 ::= gksint text index (1..n)

» W /GSTXI(4)

4.3.10 GSTXFP 0a Set text font and precision
2 parameters

p1 ::= gksint text font (-n..-1,1..n)

p2 ::= gksenum precision:

```
* STRING *
* CHAR   *
* STROKE *
```

» W /GSTXFP(1,"STROKE")

4.3.11 GSCHXP 0a Set character expansion factor
1 parameter

p1 ::= gksnum character expansion factor

» W /GSCHXP(1.5)

4.3.12 GSCHSP 0a Set character spacing
1 parameter

p1 ::= gksnum character spacing

» W /GSCHSP(1.5)

4.3.13 GSTXCI ma Set text colour index
1 parameter

p1 ::= gksint text colour index (0..n)

» W /GSTXCI(5)

4.3.14 GSCHH ma Set character height
1 parameter

p1 ::= gksnum character height (WC)

» W /GSCHH(2.5)

4.3.15 GSCHUP ma Set character up vector
1 parameter

p1 ::= L gksnum 2 value character up vector (WC)

» W /GSCHUP("1.5,9.5")

4.3.16 GSTXP 0a Set text path
1 parameter

p1 ::= gksenum text path:

```
* RIGHT *
* LEFT  *
* UP    *
* DOWN  *
```

» W /GSTXP("RIGHT")

4.3.17 GSTXAL ma Set text alignment

2 parameters

p1 ::= gksenum horizontal text alignment:

```
* NORMAL *
* LEFT   *
* CENTRE *
* RIGHT  *
```

p2 ::= gksenum vertical text alignment:

```
* NORMAL *
* TOP    *
* CAP    *
* HALF   *
* BASE   *
* BOTTOM *
```

» W /GSTXAL("CENTER","TOP")

4.3.18 GSFAI 0a Set fill area index

1 parameter

p1 ::= gksint fill area index (1..n)

» W /GSFAI(1)

4.3.19 GSFAIS ma Set fill area interior style

1 parameter

p1 ::= gksenum fill area interior style:

```
* HOLLOW  *
* SOLID   *
* PATTERN *
* HATCH   *
```

» W /GSFAIS("HATCH")

4.3.20 GSFASI 0a Set fill area style index

1 parameter

p1 ::= gksint fill area style index (-n..-1,1..n)

» W /GSFASI(1)

4.3.21 GSFACI ma Set fill area colour index
1 parameter

p1 ::= gksint fill area colour index (0..n)

» W /GSFACI(1)

4.3.22 GSPA 0a Set pattern size
1 parameter

p1 ::= L gksnum 2 value pattern size (WC)

» W /GSPA("1.1,3.1")

4.3.23 GSPARF 0a Set pattern reference point
1 parameter

p1 ::= gkspnt reference point (WC)

» W /GSPARF("100,100")

```
4.3.24   GSASF          0a       Set aspect source flags
         13 parameters

         p1  ::= gksenum         ASF linetype
                                        * INDIVIDUAL *
                                        * BUNDLED    *

         p2  ::= gksenum         ASF linewidth scale factor
                                        * INDIVIDUAL *
                                        * BUNDLED    *

         p3  ::= gksenum         ASF polyline colour index
                                        * INDIVIDUAL *
                                        * BUNDLED    *

         p4  ::= gksenum         ASF marker type
                                        * INDIVIDUAL *
                                        * BUNDLED    *

         p5  ::= gksenum         ASF marker size scale factor
                                        * INDIVIDUAL *
                                        * BUNDLED    *

         p6  ::= gksenum         ASF polymarker colour index
                                        * INDIVIDUAL *
                                        * BUNDLED    *

         p7  ::= gksenum         ASF text font and precision
                                        * INDIVIDUAL *
                                        * BUNDLED    *

         p8  ::= gksenum         ASF character expansion factor
                                        * INDIVIDUAL *
                                        * BUNDLED    *

         p9  ::= gksenum         ASF character spacing
                                        * INDIVIDUAL *
                                        * BUNDLED    *

         p10 ::= gksenum         ASF text colour index
                                        * INDIVIDUAL *
                                        * BUNDLED    *

         p11 ::= gksenum         ASF fill area interior style
                                        * INDIVIDUAL *
                                        * BUNDLED    *

         p12 ::= gksenum         ASF fill area style index
                                        * INDIVIDUAL *
                                        * BUNDLED    *

         p13 ::= gksenum         ASF fill area colour
                                        * INDIVIDUAL *
```

* BUNDLED *

A null parameter leaves the ASF unchanged. The default ASF for all parameters is INDIVIDUAL.

Please note the ANSI standard for the default ASF is INDIVIDUAL while the ISO default is BUNDLED. A GKS application program should be very cautious in leaving the ASF to the default settings.

```
» W /GSASF("INDIVIDUAL","","","","","","","","","","","","")
```

4.3.25 GSPKID 1b Set pick identifications
1 parameter

p1 ::= gksnam pick identifier

```
» W /GSPKID("PICK1")
```

4.4 Workstation attributes

	control- mnemonic	Level	GKS function name
4.4.1	GSPLR 5 parameters	1a	Set polyline representation

p1 ::= gksnam workstation ID

p2 ::= gksint polyline index (1..n)

p3 ::= gksint line type (-n..-1,1..n)

p4 ::= gksnum linewidth scale factor

p5 ::= gksint polyline colour index (0..n)

```
» W /GSPLR("WORK1",1,2,1.5,9)
```

4.4.2	GSPMR 5 parameters	1a	Set polymarker representation

p1 ::= gksnam workstation ID

p2 ::= gksint polymarker index (1..n)

p3 ::= gksint marker type (-n..-1,1..n)

p4 ::= gksnum marker size scale factor

p5 ::= gksint polymarker colour index (0..n)

```
» W /GSPMR("WORK1",1,2,1.5,9)
```

4.4.3 GSTXR 1a Set text representation
7 parameters

p1 ::= gksnam workstation ID

p2 ::= gksint text index (1..n)

p3 ::= gksint text font (-n..-1,1..n)

p4 ::= gksenum text precision:

```
* STRING *
* CHAR   *
* STROKE *
```

p5 ::= gksnum character expansion factor

p6 ::= gksnum character spacing

p7 ::= gksint text colour index (0..n)

» W /GSTXR("WORK1",1,2,"STROKE",1.5,.75,1)

4.4.4 GSFAR 1a Set fill area representation
5 parameters

p1 ::= gksnam workstation ID

p2 ::= gksint fill area index (1..n)

p3 ::= gksenum fill area interior style:

```
* HOLLOW  *
* SOLID   *
* PATTERN *
* HATCH   *
```

p4 ::= gksint fill area style index (-n..-1,1..n)

p5 ::= gksint fill area colour index (0..n)

» W /GSFAR("WORK1",1,"HOLLOW",2,1)

4.4.5 GSPAR 1a Set pattern representation
5 parameters

p1 ::= gksnam workstation ID

p2 ::= gksint pattern index (1..n)

p3 ::= gksint pattern node DX (1..n)

p4 ::= gksint pattern node DY (1..n)

p5 ::= L gksint integer array from node (1,1) through (DX,DY) pointing each cell into the colour table for the workstation. (0..n)

```
» W /GSPAR("WORK1",3,2,2,"1,1,2,2")
```

4.4.6 GSCR 0a Set colour representation
3 parameters

p1 ::= gksnam workstation ID

p2 ::= gksint colour index (0..n)

p3 ::= L gksnum 3 value colour intensity (red,green,blue)

```
» W /GSCR("WORK1",1,"1,0,0")
```

4.5 Normalization transformation

	control- mnemonic	Level	GKS function name
4.5.1	GSWN 2 parameters	ma	Set window

p1 ::= gksint transformation number (1..n)

p2 ::= L gksnum 4 value window limits (Xmin, Xmax, Ymin, Ymax) (WC)

```
» W /GSWN(1,"1.5,9.5,1.5,9.5")
```

4.5.2	GSVP 2 parameters	ma	Set viewport

p1 ::= gksint transformation number (1..n)

p2 ::= L gksnum 4 value viewport limits (Xmin, Xmax, Ymin, Ymax) (NDC)

```
» W /GSVP(1,"0.5,100,0.5,50")
```

4.5.3	GSVPIP 3 parameters	mb	Set viewport input priority

p1 ::= gksint transformation number (0..n)

p2 ::= gksint reference transformation number (0..n)

p3 ::= gksenum relative priority:

```
* HIGHER *
* LOWER  *
```

```
» W /GSVPIP(3,2,"LOWER")
```

4.5.4	GSELNT 1 parameter	ma	Select normalization transformation

p1 ::= gksint transformation number (0..n)

```
» W /GSELNT(3)
```

4.5.5 GSCLIP ma Set clipping indicator
1 parameter

p1 ::= gksenum clipping indicator:

```
* CLIP   *
* NOCLIP *
```

» W /GSCLIP("CLIP")

4.6 Workstation transformation

control-
mnemonic Level GKS function name

4.6.1 GSWKWN ma Set workstation window
2 parameters

p1 ::= gksnam workstation ID

p2 ::= L gksnum 4 value workstation window limit (Xmin, Xmax, Ymin, Ymax) (NDC)

» W /GSWKWN("WORK1","0,.9,0,.9")

4.6.2 GSWKVP ma Set workstation viewport
2 parameters

p1 ::= gksnam workstation ID

p2 ::= L gksnum 4 value workstation viewport limit (Xmin, Xmax, Ymin, Ymax) (DC)

» W /GSWKVP("WORK1","0.1,1.1,0.1,1.1")

4.7 Segment manipulation functions

control-
mnemonic **Level** **GKS function name**

4.7.1 GCRSG 1a Create segment
1 parameter

p1 ::= gksnam segment name

» W /GCRSG("SEG1")

4.7.2 GCLSG 1a Close segment
No parameters

» W /GCLSG

4.7.3 GRENSG 1a Rename segment
2 parameters

p1 ::= gksnam old segment name

p2 ::= gksnam new segment name

» W /GRENSG("SEG1","SEG-A")

4.7.4 GDSG 1a Delete segment
1 parameter

p1 ::= gksnam segment name

» W /GDSG("SEG1")

4.7.5 GDSGWK 1a Delete segment from workstation
2 parameters

p1 ::= gksnam workstation ID

p2 ::= gksnam segment name

» W /GDSGWK("WORK1","SEG1")

4.7.6 GASGWK 2a Associate segment with workstation
2 parameters

p1 ::= gksnam workstation ID

p2 ::= gksnam segment name

» W /GASGWK("WORK1","SEG1")

4.7.7 GCSGWK 2a Copy segment to workstation
2 parameters

p1 ::= gksnam workstation ID

p2 ::= gksnam segment name

» W /GCSGWK("WORK1","SEG1")

4.7.8 GINSG 2a Insert segment
2 parameters

p1 ::= gksnam segment name

p2 ::= L gksnum 2x3 value transformation matrix

» W /GINSG("SEG1","9.9,0,9.9,0,1.3,1.3")

4.8 Segment attributes

control-
mnemonic **Level** **GKS function name**

4.8.1 GSSGT 1a Set segment transformation
2 parameters

p1 ::= gksnam segment name

p2 ::= L gksnum 2x3 value transformation matrix

» W /GSSGT("SEG1","9.9,0,9.9,0,1.3,1.3")

4.8.2 GSVIS 1a Set visibility
2 parameters

p1 ::= gksnam segment name

p2 ::= gksenum visibility:

```
* VISIBLE   *
* INVISIBLE *
```

» W /GSVIS("SEG1","INVISIBLE")

4.8.3 GSHLIT 1a Set highlighting
2 parameters

p1 ::= gksnam segment name

p2 ::= gksenum highlighting:

```
* NORMAL      *
* HIGHLIGHTED *
```

» W /GSHLIT("SEG1","HIGHLIGHTED")

4.8.4 GSSGP 1a Set segment priority
2 parameters

p1 ::= gksnam segment name

p2 ::= gksnum segment priority [0,1]

» W /GSSGP("SEG1",.5)

4.8.4 GSDTEC 1b Set detectability

2 parameters

p1 ::= gksnam segment name

p2 ::= gksenum detectability:

```
* UNDETECTABLE *
* DETECTABLE   *
```

```
» W /GSDTEC("SEG1","UNDETECTABLE")
```

4.9 Initialization of input devices

control-
mnemonic Level GKS function name

4.9.1 GINLC mb Initialize locator
7 parameters

p1 ::= gksnam workstation ID

p2 ::= gksint locator device number (1..n)

p3 ::= gksint initial normalization transform number (0..n)

p4 ::= gkspnt initial locator position (WC)

p5 ::= gksint prompt and echo type (-n..-1,1..n)

p6 ::= L gksnum 4 values defining echo area (Xmin, Xmax, Ymin, Ymax) (DC)

p7 ::= gksdat locator data record

```
» W /GINLC("WORK1",100,3,"1,1",1,"1,0,1,0","LOCATOR DATA")
```

4.9.2 GINSK mb Initialize stroke
8 parameters

p1 ::= gksnam workstation ID

p2 ::= gksint stroke device number (1..n)

p3 ::= gksint initial normalization transform number (0..n)

p4 ::= gksint number of points in initial stroke (0..n)

p5 ::= L gkspnt n points in initial stroke (WC)

p6 ::= gksint prompt and echo type (-n..-1,1..n)

p7 ::= L gksnum 4 values defining echo area (Xmin, Xmax, Ymin, Ymax) (DC)

p8 ::= gksdat stroke data record

```
» W /GINSK("WORK1",2,1,2,"1,1,100,100",1,"1,0,1,0","STROKE")
```

4.9.3 GINVL mb Initialize valuation

6 parameters

p1 ::= gksnam workstation ID

p2 ::= gksint valuator device number (1..n)

p3 ::= gksnum initial value

p4 ::= gksint prompt and echo type (-n..-1,1..n)

p5 ::= L gksnum 4 values defining echo area (Xmin, Xmax, Ymin, Ymax) (DC)

p6 ::= gksdat valuator data record

```
» W /GINVL("WORK1",99,12.5,1,"1,0,1,0","VALUATOR")
```

4.9.4 GINCH mb Initialize choice

7 parameters

p1 ::= gksnam workstation ID

p2 ::= gksint choice device number (1..n)

p3 ::= gksenum initial status:

```
* OK       *
* NOCHOICE *
```

p4 ::= gksint initial choice number (1..n)

p5 ::= gksint prompt and echo type (-n..-1,1..n)

p6 ::= L gksnum 4 values defining echo area (Xmin, Xmax, Ymin, Ymax) (DC)

p7 ::= gksdat choice data record

```
» W /GINCH("WORK1",1,"OK",1,1,"1,0,1,0","CHOICE DATA")
```

4.9.5 GINPK 1b Initialize pick
8 parameters

p1 ::= gksnam workstation ID

p2 ::= gksint pick device number (1..n)

p3 ::= gksenum initial status:

```
* OK     *
* NOPICK *
```

p4 ::= gksnam initial segment name

p5 ::= gksnam initial pick ID

p6 ::= gksint prompt and echo type (-n..-1,1..n)

p7 ::= L gksnum 4 values defining echo area (Xmin, Xmax, Ymin, Ymax) (DC)

p8 ::= gksdat pick data record

» W /GINPK("WORK1",88,"OK","SEG1","PICK1",1,"1,0,1,0","PICK")

4.9.6 GINST mb Initialize string
6 paramters

p1 ::= gksnam workstation ID

p2 ::= gksint string device number (1..n)

p3 ::= gksstr initial string

p4 ::= gksint prompt and echo type (-n..-1,1..n)

p5 ::= L gksnum 4 values defining echo area (Xmin, Xmax, Ymin, Ymax) (DC)

p6 ::= gksdat string data record

» W /GINST("WORK1",1,"TEST STRING",1,"1,0,1,0","STRING")

4.10 Setting input device mode

control-
mnemonic **Level** **GKS function name**

4.10.1 GSLCM mb Set locator mode
4 parameters

p1 ::= gksnam workstation ID

p2 ::= gksint locator device number (1..n)

p3 ::= gksenum operating mode:

```
* REQUEST *
* SAMPLE  *
* EVENT   *
```

p4 ::= gksenum echo switch:

```
* ECHO   *
* NOECHO *
```

» W /GSLCM("WORK1",1,"SAMPLE","ECHO")

4.10.2 GSSKM mb Set stroke mode
4 parameters

p1 ::= gksnam workstation ID

p2 ::= gksint stroke device number (1..n)

p3 ::= gksenum operating mode:

```
* REQUEST *
* SAMPLE  *
* EVENT   *
```

p4 ::= gksenum echo switch:

```
* ECHO   *
* NOECHO *
```

» W /GSSKM("WORK1",1,"SAMPLE","ECHO")

4.10.3 GSVLM mb Set valuator mode
4 parameters

p1 ::= gksnam workstation ID

p2 ::= gksint valuator device number (1..n)

p3 ::= gksenum operating mode:

```
* REQUEST *
* SAMPLE  *
* EVENT   *
```

p4 ::= gksenum echo switch:

```
* ECHO   *
* NOECHO *
```

» W /GSVLM("WORK1",1,"SAMPLE","ECHO")

4.10.4 GSCHM mb Set choice mode
4 parameters

p1 ::= gksnam workstation ID

p2 ::= gksint choice device number (1..n)

p3 ::= gksenum operating mode:

```
* REQUEST *
* SAMPLE  *
* EVENT   *
```

p4 ::= gksenum echo switch:

```
* ECHO   *
* NOECHO *
```

» W /GSCHM("WORK1",1,"SAMPLE","ECHO")

4.10.5 GSPKM 1b Set pick mode

4 parameters

p1 ::= gksnam workstation ID

p2 ::= gksint pick device number (1..n)

p3 ::= gksenum operating mode:

```
* REQUEST *
* SAMPLE  *
* EVENT   *
```

p4 ::= gksenum echo switch:

```
* ECHO   *
* NOECHO *
```

» W /GSPKM("WORK1",1,"SAMPLE","ECHO")

4.10.6 GSSTM mb Set string mode

4 parameters

p1 ::= gksnam workstation ID

p2 ::= gksint string device number (1..n)

p3 ::= gksenum operating mode:

```
* REQUEST *
* SAMPLE  *
* EVENT   *
```

p4 ::= gksenum echo switch:

```
* ECHO   *
* NOECHO *
```

» W /GSSTM("WORK1",1,"SAMPLE","ECHO")

4.11 Request input function

control- mnemonic	Level	GKS function name

4.11.1 GRQLC mb Request locator
2 parameters

p1 ::= gksnam workstation ID

p2 ::= gksint locator device number (1..n)

» W /GRQLC("WORK1",1)

3 output values

v1 ::= gksenum status:

* OK *
* NONE *

v2 ::= gksint normalization transform number (0..n)

v3 ::= gkspnt locator position (x,y) (WC)

« R Status,Transfrm,Point

4.11.2 GRQSK mb Request stroke
2 parameters

p1 ::= gksnam workstation ID

p2 ::= gksint stroke device number (1..n)

» W /GRQSK("WORK1",1)

4 output values

v1 ::= gksenum status:

* OK *
* NONE *

v2 ::= gksint normalization transform number (0..n)

v3 ::= gksint number of points (0..n)

v4 ::= M gkspnt n points of stroke (WC)

« R Status,Transfrm,NumPoint F II=1:1:NumPoint R Point(II)

4.11.3 GRQVL mb Request valuator

2 parameters

p1 ::= gksnam workstation ID

p2 ::= gksint valuator device number (1..n)

» W /GRQVL("WORK1",1)

2 output values

v1 ::= gksenum status:

```
* OK   *
* NONE *
```

v2 ::= gksnum value

« R Status,Value

4.11.4 GRQCH mb Request choice

2 parameters

p1 ::= gksnam workstation ID

p2 ::= gksint choice device number (1..n)

» W /GRQCH("WORK1",1)

2 output values

v1 ::= gksenum status:

```
* OK       *
* NOCHOICE *
* NONE     *
```

v2 ::= gksint choice number (1..n)

« R Status,Choice

4.11.5 GRQPK 1b Request pick

2 parameters

p1 ::= gksnam workstation ID

p2 ::= gksint pick device number (1..n)

» W /GRQPK("WORK1",1)

3 output values

v1 ::= gksenum status:

```
* OK     *
* NOPICK *
* NONE   *
```

v2 ::= gksnam segment name

v3 ::= gksnam pick ID

« R Status,Segment,Pick

4.11.6 GRQST mb Request string

2 parameters

p1 ::= gksnam workstation ID

p2 ::= gksint string device number (1..n)

» W /GRQST("WORK1",1)

2 output values

v1 ::= gksenum status:

```
* OK   *
* NONE *
```

v2 ::= gksstr character string

« R Status,String

4.12 Sample input function

control-
mnemonic Level GKS function name

4.12.1 GSMLC mc Sample locator
2 parameters

p1 ::= gksnam workstation ID

p2 ::= gksint locator device number (1..n)

» W /GSMLC("WORK1",1)

2 values returned

v1 ::= gksint normalization transform number (0..n)

v2 ::= gkspnt locator position (WC)

« R Transfrm,Position

4.12.2 GSMSK mc Sample stroke
2 parameters

p1 ::= gksnam workstation ID

p2 ::= gksint stroke device number (1..n)

» W /GSMSK("WORK1",1)

3 values returned

v1 ::= gksint normalization transform number (0..n)

v2 ::= gksint number of points (0..n)

v3 ::= M gkspnt n points of stroke (WC)

« R Transfrm,Num F ii=1:1:Num R Point(ii)

4.12.3 GSMVL mc Sample valuator
2 parameters

p1 ::= gksnam workstation ID

p2 ::= gksint valuator device number (1..n)

» W /GSMVL("WORK1",1)

1 value returned

v1 ::= gksnum valuator value

« R Value

4.12.4 GSMCH mc Sample choice
2 parameters

p1 ::= gksnam workstation ID

p2 ::= gksint choice device number (1..n)

» W /GSMCH("WORK1",1)

2 values returned

v1 ::= gksenum status:

```
* OK       *
* NOCHOICE *
```

v2 ::= gksint choice number (1..n)

« R Status,Choice

4.12.5 GSMPK 1c Sample pick

2 parameters

p1 ::= gksnam workstation ID

p2 ::= gksint pick device number (1..n)

» W /GSMPK("WORK1",1)

3 values returned

v1 ::= gksenum status:

```
* OK     *
* NOPICK *
```

v2 ::= gksnam segment name

v3 ::= gksnam pick identifier

« R Status,Segmnt,Pick

4.12.6 GSMST mc Sample string

2 parameters

p1 ::= gksnam workstation ID

p2 ::= gksint string device number (1..n)

» W /GSMST("WORK1",1)

1 value returned

v1 ::= gksstr character string

« R String

4.13 Event input function

control-
mnemonic Level GKS function name

4.13.1 GWAIT mc Await event
1 parameter

p1 ::= gksnum timeout (seconds)

» W /GWAIT(0.1)

3 values returned

v1 ::= gksnam workstation ID

v2 ::= gksenum input class:

```
* NONE     *
* LOCATOR  *
* STROKE   *
* VALUATOR *
* CHOICE   *
* PICK     *
* STRING   *
```

v3 ::= gksint logical input device number (1..n)

« R Work,Class,Device

4.13.2 GFLUSH mc Flush device events
3 parameters

p1 ::= gksnam workstation ID

p2 ::= gksenum input class:

```
* LOCATOR  *
* STROKE   *
* VALUATOR *
* CHOICE   *
* PICK     *
* STRING   *
```

p3 ::= gksint logical input device number (1..n)

» W /GFLUSH("WORK1","STROKE",1)

4.13.3 GGTLC mc Get locator
No parameters

» W /GGTLC

2 values returned

v1 ::= gksint normalization transform number (0..n)

v2 ::= gkspnt locator position (WC)

« R Transfrm,Locator

4.13.4 GGTSK mc Get stroke
No parameters

» W /GGTSK

3 values returned

v1 ::= gksint normalization transform number (0..n)

v2 ::= gksint number of points (0..n)

v3 ::= M gkspnt n points of stroke (WC)

« R Transfrm,Num F ii=1:1:Num R Point(ii)

4.13.5 GGTVL mc Get valuator
No parameters

» W /GGTVL

1 values returned

v1 ::= gksnum value

« R Value

4.13.6 GGTCH mc Get choice

No parameters

» W /GGTCH

2 values returned

v1 ::= gksenum status:

```
* OK       *
* NOCHOICE *
```

v2 ::= gksint choice number (1..n)

« R Status,Choice

4.13.7 GGTPK 1c Get pick

No parameters

» W /GGTPK

3 values returned

v1 ::= gksenum status:

```
* OK     *
* NOPICK *
```

v2 ::= gksnam segment name

v3 ::= gksnam pick identifier

« R Status,Segment,Pick

4.13.8 GGTST mc Get string

No parameters

» W /GGTST

1 values returned

v1 ::= gksstr character string

« R String

4.14 Metafile functions

control-
mnemonic Level GKS function name

4.14.1 GWITM 0a Write item to GKSM
4 parameters

p1 ::= gksnam workstation ID

p2 ::= gksint item type

p3 ::= gksint item data record length (0..n)

p4 ::= gksdat item data record

» W /GWITM("WORK1",1,10,"This is it")

4.14.2 GGTITM 0a Get item type from GKSM
1 parameter

p1 ::= gksnam workstation ID

» W /GGTITM("WORK1")

2 values returned

v1 ::= gksint item type

v2 ::= gksint item data record length (0..n)

« R Item,ItemLen

4.14.3 GRDITM 0a Read item from GKSM
2 parameters

p1 ::= gksnam workstation ID

p2 ::= gksint maximum item data record length (0..n)

» W /GRDITM("WORK1",10)

1 value returned

v1 ::= gksdat item data record

» R Item

4.14.4 GIITM 0a Interpret item
3 parameters

p1 ::= gksint item type

p2 ::= gksint item data record length (0..n)

p3 ::= gksdat item data record

» W /GIITM(1,10,"This is it")

4.15 Inquiry function for operating state value

control-
mnemonic   **Level**   **GKS function name**

4.15.1 GQOPS 0a Inquire operating state value
No parameters

» W /GQOPS

1 value returned

v1 ::= gksenum operating state value:

```
* GKCL *
* GKOP *
* WSOP *
* WSAC *
* SGOP *
```

« R Value

4.16 Inquiry function for GKS description table

control-
mnemonic Level GKS function name

4.16.1 GQLVKS ma Inquire level of GKS
No parameters

» W /GQLVK

2 values returned

v1 ::= gksint error indicator

v2 ::= gksenum level of GKS:

```
* ma *
* mb *
* mc *
* 0a *
* 0b *
* 0c *
* 1a *
* 1b *
* 1c *
* 2a *
* 2b *
* 2c *
```

« R Error,Level

4.16.2 GQEWK 0a Inquire list of available workstation types
No parameters

» W /GQEWK

3 values returned

v1 ::= gksint error indicator

v2 ::= gksint number of workstation types (1..n)

v3 ::= M gksenum n values for workstation types

```
* OUTPUT *
* INPUT  *
* OUTIN  *
* WISS   *
* MO     *
* MI     *
```

« R Error,NumWork F ii=1:1:NumWork R Work(ii)

4.16.3 GQWKM 1a Inquire workstation maximum numbers
No parameters

» W /GQWKM

4 values returned

v1 ::= gksint error indicator

v2 ::= gksint maximum number of simultaneous open workstations (1..n)

v3 ::= gksint maximum number of simultaneous active workstations (1..n)

v4 ::= gksint maximum number of workstations associated with segment (1..n)

« R Error,MaxOpen,MaxActiv,MaxWSeg

4.16.4 GQMNTN 0a Inquire maximum normalization transformation number
No parameters

» W /GQMNTN

2 values returned

v1 ::= gksint error indicator

v2 ::= gksint maximum normalization transform number (1..n)

« R Error,MaxTran

4.17 Inquiry functions for GKS state list

control-
mnemonic Level GKS function name

4.17.1 GQOPWK 0a Inquire set of open workstations
No parameters

» W /GQOPWK

3 values returned

v1 ::= gksint error indicator

v2 ::= gksint number of open workstations (0..n)

v3 ::= M gksnam n values of open workstations

« R Error,Num F ii=1:1:Num R Work(ii)

4.17.2 GQACWK 1a Inquire set of active workstations
No parameters

» W /GQACWK

3 values returned

v1 ::= gksint error indicator

v2 ::= gksint number of actice workstations (0..n)

v3 ::= M gksnam n values of active workstations

« R Error,Num F ii=1:1:Num R Work(ii)

4.17.3 GQCPAV ma Inquire current primitive attribute values
No parameters

» W /GQCPAV

15 values returned

v1 ::= gksint error indicator

v2 ::= gksint current polyline index (1..n)

v3 ::= gksint current polymarker index (1..n)

v4 ::= gksint current text index (1..n)

v5 ::= gksnum current character height (WC)

v6 ::= L gksnum 2 value current character up vector (WC)

v7 ::= gksnum current character width (WC)

v8 ::= L gksnum 2 value current character base vector (WC)

v9 ::= gksenum current text path:

```
* RIGHT *
* LEFT  *
* UP    *
* DOWN  *
```

v10 ::= gksenum current horizontal text alignment:

```
* NORMAL *
* LEFT   *
* CENTRE *
* RIGHT  *
```

v11 ::= gksenum current vertical text alignment:

```
* NORMAL *
* TOP    *
* CAP    *
* HALF   *
* BASE   *
* BOTTOM *
```

v12 ::= gksint current fill area index (1..N)

v13 ::= L gksnum 2 value current pattern width vector (WC)

v14 ::= L gksnum 2 value current pattern height vector (WC)

v15 ::= gkspnt current pattern reference point (WC)

« R Error,PolyLine,PolyMark,Text,TextHt,TextUp,TextWid,TextBase
« R TextPath,TextHor,TextVer,FillArea,PatWid,PatHght,PatPnt

4.17.4 GQPKID 1b Inquire current pick identifier
No parameters

» W /GQPKID

2 values returned

v1 ::= gksint error indicator

v2 ::= gksnam current pick identifier

« R Error,Pick

4.17.5 GQCIAV ma Inquire current individual attribute values

No parameters

» W /GQCIAV

16 values returned

v1 ::= gksint error indicator

v2 ::= gksint current linetype (-n..-1,1..n)

v3 ::= gksnum current linewidth scale factor

v4 ::= gksint current polyline colour index (0..n)

v5 ::= gksint current marker type (-n..-1,1..n)

v6 ::= gksnum current marker size scale factor

v7 ::= gksint polymarker colour index (0..n)

v8 ::= gksint current text font (-n..-1,1..n)

v9 ::= gksenum current text precision:

```
* STRING *
* CHAR   *
* STROKE *
```

v10 ::= gksnum current character expansion factor

v11 ::= gksnum current character spacing

v12 ::= gksint current text colour index (0..n)

v13 ::= gksenum current fill area interior style:

```
* HOLLOW  *
* SOLID   *
* PATTERN *
* HATCH   *
```

v14 ::= gksint current fill area style index (-n..-1,1..n)

v15 ::= gksint current fill area colour index (0..n)

v16 ::= L gksenum 13 value current list of aspect source flags:

```
* BUNDLED    *
* INDIVIDUAL *
```

« R Error,LinType,LinWidSF,LinColr
« R MrkType,MrkSizSF,MrkColr
« R TextFont,TextPrec,CharExp,CharSpc,TextColr
« R Fill,FillStyl,FillColr,Aspect

4.17.6 GQCNTN ma Inquire current normalization transformation number

No parameters

» W /GQASF

2 values returned

v1 ::= gksint error indicator

v2 ::= gksint current normalization transform number (0..n)

« R Error,Trans

4.17.7 GQENTN 0a Inquire list of normalization transformation numbers

No parameters

» W /GQENTN

2 values returned

v1 ::= gksint error indicator

v2 ::= L gksint transformation number

« R Error,Tranlst

4.17.8 GQNT ma Inquire normalization transformation

1 parameter

p1 ::= gksint normalization transform number

» W /GQNT(1)

3 values returned

v1 ::= gksint error indicator

v2 ::= L gksnum 4 value window limit (WC)

v3 ::= L gksnum 4 value viewport limit (NDC)

« R Error,Window,ViewPort

4.17.9 GQCLIP ma Inquire clipping
No parameters

» W /GQCLIP

3 values returned

v1 ::= gksint error indicator

v2 ::= gksenum clipping indicator:

```
* CLIP   *
* NOCLIP *
```

v3 ::= L gksnum 4 value clipping rectangle (NDC)

« R Error,Clip,ClipRect

4.17.10 GQOPSG 1a Inquire name of open segment
No parameters

» W /GQOPSG

2 values returned

v1 ::= gksint error indicator

v2 ::= gksnam name of open segment

« R Error,OpenSeg

4.17.11 GQSGUS 1a Inquire set of segment names in use
No parameters

» W /GQSGUS

3 values returned

v1 ::= gksint error indicator

v2 ::= gksint number of segment names (0..n)

v3 ::= M gksnam n values of segment names in use

« R Error,Num F ii=1:1:Num R Segment(ii)

4.17.12 GQSIM mc Inquire more simultaneous events

No parameters

» W /GQSIM

2 values returned

v1 ::= gksint error indicator

v2 ::= gksenum more simultaneous events:

```
* NOMORE *
* MORE   *
```

« R Error, More

4.18 Inquiry function for workstation state list

control-
mnemonic **Level** **GKS function name**

4.18.1 GQWKC ma Inquire workstation connection and type
1 parameter

p1 ::= gksnam workstation ID

» W /GQWKC("WORK1")

3 values returned

v1 ::= gksint error indicator

v2 ::= gksnam connection identifier

v3 ::= gksenum workstation type:

```
* OUTPUT *
* INPUT  *
* OUTIN  *
* WISS   *
* MO     *
* MI     *
```

« R Error,Ident,WorkType

4.18.2 GQWKS 0a Inquire workstation state
1 parameter

p1 ::= gksnam workstation ID

» W /GQWKS("WORK1")

2 values returned

v1 ::= gksint error indicator

v2 ::= gksenum workstation state:

```
* INACTIVE *
* ACTIVE   *
```

« R Error,State

4.18.3 GQWKDU 0a Inquire workstation deferral and update states

```
 1 parameter

p1 ::= gksnam          workstation ID

» W /GQWKDU("WORK1")

 5 values returned

v1 ::= gksint          error indicator

v2 ::= gksenum         deferral mode:
                                           * ASAP *
                                           * BNIG *
                                           * BNIL *
                                           * ASTI *

v3 ::= gksenum         implicit regeneration mode:
                                           * SUPPRESSED *
                                           * ALLOWED    *

v4 ::= gksenum         display surface empty:
                                           * EMPTY    *
                                           * NOTEMPTY *

v5 ::= gksenum         new frame action necessary at update:
                                           * NO  *
                                           * YES *

« R Error,DefMode,RegMode,DispSurf,NewFrm
```

4.18.4 GQEPLI 1a Inquire list of polyline indices

```
 1 parameter

p1 ::= gksnam          workstation ID

» W /GQEPLI("WORK1")

 3 values returned

v1 ::= gksint          error indicator

v2 ::= gksint          number of polyline bundle table entries (5..n)

v3 ::= M gksint        n values of defined polyline indices (1..n)

« R Error,NumLine F ii=1:1:NumLine R Lines(ii)
```

4.18.5 GQPLR 1a Inquire polyline representation

3 parameters

p1 ::= gksnam workstation ID

p2 ::= gksint polyline index (1..n)

p3 ::= gksenum type of returned value:
* SET *
* REALIZED *

» W /GQPLR("WORK1",1,"SET")

4 values returned

v1 ::= gksint error indicator

v2 ::= gksint linetype (-n..-1,1..n)

v3 ::= gksnum linewidth scale factor

v4 ::= gksint polyline colour index (0..n)

« R Error,LinType,LinScale,LineColr

4.18.6 GQEPMI 1a Inquire list of polymarker indices

1 parameter

p1 ::= gksnam workstation ID

» W /GQEPMI("WORK1")

3 values returned

v1 ::= gksint error indicator

v2 ::= gksint number of polymarker bundle table entries (5..n)

v3 ::= M gksint n values of defined polymarker indices (1..n)

« R Error,NumMark F ii=1:1:NumMark R Marks(ii)

4.18.7 GQPMR 1a Inquire polymarker representation

3 parameters

p1 ::= gksnam workstation ID

p2 ::= gksint polymarker index (1..n)

p3 ::= gksenum type of returned value:
* SET *
* REALIZED *

» W /GQPMR("WORK1",1,"SET")

4 values returned

v1 ::= gksint error indicator

v2 ::= gksint marker type (-n..-1,1..n)

v3 ::= gksnum marker size scale factor

v4 ::= gksint polymarker colour index (0..n)

« R Error,MarkType,MarkScal,MarkColr

4.18.8 GQETXI 1a Inquire list of text indices

1 parameter

p1 ::= gksnam workstation ID

» W /GQETXI("WORK1")

3 values returned

v1 ::= gksint error indicator

v2 ::= gksint number of text bundle table entries (6..n)

v3 ::= M gksint n values of defined text indices (1..n)

« R Error,NumText F ii=1:1:NumText R Text(ii)

4.18.9 GQTXR 1a Inquire text representation

3 parameters

p1 ::= gksnam workstation ID

p2 ::= gksint text index (1..n)

p3 ::= gksenum type of returned value:

* SET *
* REALIZED *

```
» W /GQTXR("WORK1",1,"SET")
```

6 values returned

v1 ::= gksint error indicator

v2 ::= gksint text font (-n..-1,1..n)

v3 ::= gksenum text precision:

* STROKE *
* CHAR *
* STRING *

v4 ::= gksnum character expansion factor

v5 ::= gksnum character spacing

v6 ::= gksint text colour index (0..n)

```
« R Error,Font,Precis,ExpFac,Space,Colour
```

4.18.10 GQTXX ma INQUIRE TEXT EXTENT

3 parameters

p1 ::= gksnam workstation ID

p2 ::= gkspnt text position (WC)

p3 ::= gksstr character string

```
» W /GQTXX("WORK1","1,1","NOW IS THE TIME")
```

3 values returned

v1 ::= gksint error indicator

v2 ::= gkspnt concatenation point (WC)

v3 ::= L gkspnt 4 values; text extent rectangle (WC)

```
« R Error,Point,Extent
```

4.18.11 GQEFAI 1a Inquire list of fill area indices
1 parameter

p1 ::= gksnam workstation ID

```
» W /GQEFAI("WORK1")
```

3 values returned

v1 ::= gksint error indicator

v2 ::= gksint number of fill area bundle table entries (5..n)

v3 ::= M gksint n values of defined fill area indices (1..n)

```
« R Error,NumFill F ii=1:1:NumFill R Fill(ii)
```

4.18.12 GQFAR 1a Inquire fill area representation
3 parameters

p1 ::= gksnam workstation ID

p2 ::= gksint fill area index (1..n)

p3 ::= gksenum type of returned value:
* SET *
* REALIZED *

```
» W /GQFAR("WORK1",1,"SET")
```

4 values returned

v1 ::= gksint error indicator

v2 ::= gksenum fill area style:
* HOLLOW *
* SOLID *
* PATTERN *
* HATCH *

v3 ::= gksint fill area style index (-n..-1,1..n)

v4 ::= gksint fill area colour index (0..n)

```
« R Error,Style,StyleI,ColorI
```

4.18.13 GQEPAI 1a Inquire list of pattern indices

1 parameter

p1 ::= gksnam workstation ID

» W /GQEPAI("WORK1")

3 values returned

v1 ::= gksint error indicator

v2 ::= gksint number of pattern table entries (0..n)

v3 ::= M gksint n values of pattern indices (1..n)

« R Error,NumPat F ii=1:1:NumPat R Pattern(ii)

4.18.14 GQPAR 1a Inquire pattern representation

3 parameters

p1 ::= gksnam workstation ID

p2 ::= gksint fill area index (1..n)

p3 ::= gksenum type of returned value:
* SET *
* REALIZED *

» W /GQPAR("WORK1",1,"REALIZED")

3 values returned

v1 ::= gksint error indicator

v2 ::= L gksint 2 value pattern array dimension (n,m) (1..n)

v3 ::= L gksint n*m values of pattern array (0..n)

« R Error,Size,Array

4.18.15 GQECI ma Inquire list of colour indices
1 parameter

p1 ::= gksnam workstation ID

» W /GQECI("WORK1")

3 values returned

v1 ::= gksint error indicator

v2 ::= gksint number of colour table entries (2..n)

v3 ::= M gksint n colour indices (0..n)

« R Error,NumColor F II=1:1:NumColor R Color(II)

4.18.16 GQCR ma Inquire colour representation
3 parameters

p1 ::= gksnam workstation ID

p2 ::= gksint colour index (0..n)

p3 ::= gksenum type of returned value:
* SET *
* REALIZED *

» W /GQCR("WORK1",1,"REALIZED")

2 values returned

v1 ::= gksint error indicator

v2 ::= L gksnum 3 value colour intensity (Red,Green,Blue)

« R Error,RGB

4.18.17 GQWKT ma Inquire workstation transformation
1 parameter

p1 ::= gksnam workstation ID

» W /GQWKT("WORK1")

6 values returned

v1 ::= gksint error indicator

v2 ::= gksenum workstation transformation update state:
* NOTPENDING *
* PENDING *

v3 ::= L gksnum 4 value requested workstation window (NDC)

v4 ::= L gksnum 4 value current workstation window (NDC)

v5 ::= L gksnum 4 value requested workstation viewport (DC)

v6 ::= L gksnum 4 value current workstation viewport (DC)

« R Error,State,ReqWind,CurWind,ReqView,CurView

4.18.18 GQSGWK 1a Inquire set of segment names on workstation
1 parameter

p1 ::= gksnam workstation ID

» W /GQSGWK("WORK1")

3 values returned

v1 ::= gksint error indicator

v2 ::= gksint number of segment names (0..n)

v3 ::= M gksnam n stored segment names for this workstation

« R Error,NumSeg F ii=1:1:NumSeg R Segment(ii)

4.18.19 GQLCS mb Inquire locator device state
3 parameters

p1 ::= gksnam workstation ID

p2 ::= gksint locator device number (1..n)

p3 ::= gksenum type of returned value:
* SET *
* REALIZED *

» W /GQLCS("WORK1",1,"SET")

8 values returned

v1 ::= gksint error indicator

v2 ::= gksenum operating mode:
* REQUEST *
* SAMPLE *
* EVENT *

v3 ::= gksenum echo switch:
* ECHO *
* NOECHO *

v4 ::= gksint initial normalization transform number (0..n)

v5 ::= gkspnt initial locator position (WC)

v6 ::= gksint prompt and echo type (-n..-1,1..n)

v7 ::= L gksnum 4 value echo area (DC)

v8 ::= gksdat locator data record

« R Error,OpMode,Echo,Trans,Point,Prompt,EchoArea,LocDat

4.18.20 GQSKS mb Inquire stroke device state
3 parameters

p1 ::= gksnam workstation ID

p2 ::= gksint stroke device number (1..n)

p3 ::= gksenum type of returned value:
* SET *
* REALIZED *

» W /GQSKS("WORK1",1,"SET")

9 values returned

v1 ::= gksint error indicator

v2 ::= gksenum operating mode:
* REQUEST *
* SAMPLE *
* EVENT *

v3 ::= gksenum echo switch:
* ECHO *
* NOECHO *

v4 ::= gksint initial normalization transform number (0..n)

v5 ::= gksint initial number of points (0..n)

v6 ::= M gkspnt n initial points in stroke (WC)

v7 ::= gksint prompt and echo type (-n..-1,1..n)

v8 ::= L gksnum 4 value echo area (DC)

v9 ::= gksdat stroke data record

« R Error,OpMode,Echo,Trans,NumPnt F ii=1:1:NumPnt R Point(ii)
« R Prompt,EchoArea,StrDat

4.18.21 GQVLS mb Inquire valuator device state
2 parameters

p1 ::= gksnam workstation ID

p2 ::= gksint valuator device number (1..n)

» W /GQVLS("WORK1",1)

7 values returned

v1 ::= gksint error indicator

v2 ::= gksenum operating mode:
* REQUEST *
* SAMPLE *
* EVENT *

v3 ::= gksenum echo switch:
* ECHO *
* NOECHO *

v4 ::= gksnum initial value

v5 ::= gksint prompt and echo type (-n..-1,1..n)

v6 ::= L gksnum 4 value echo area (DC)

v7 ::= gksdat valuator data record

« R Error,OpMode,Echo,Value,EchoArea,ValDat

4.18.22 GQCHS mb Inquire choice device state

2 parameters

p1 ::= gksnam workstation ID

p2 ::= gksint choice device number (1..n)

» W /GQCHS("WORK1",1)

8 values returned

v1 ::= gksint error indicator

v2 ::= gksenum operating mode:

* REQUEST *
* SAMPLE *
* EVENT *

v3 ::= gksenum echo switch:

* ECHO *
* NOECHO *

v4 ::= gksenum initial status:

* OK *
* NOCHOICE *

v5 ::= gksint initial choice number (1..n)

v6 ::= gksint prompt and echo type (-n..-1,1..n)

v7 ::= L gksnum 4 value echo area (DC)

v8 ::= gksdat choice data record

« R Error,OpMode,Echo,InStatus,InChoice,Prompt,EchoArea,ChDat

4.18.23 GQPKS 1b Inquire pick device state

3 parameters

p1 ::= gksnam workstation ID

p2 ::= gksint pick device number (1..n)

p3 ::= gksenum type of returned value:

* SET *
* REALIZED *

```
» W /GQPKS("WORK1",1,"SET")
```

9 values returned

v1 ::= gksint error indicator

v2 ::= gksenum operating mode:

* REQUEST *
* SAMPLE *
* EVENT *

v3 ::= gksenum echo switch:

* ECHO *
* NOECHO *

v4 ::= gksenum initial status:

* PICK *
* NOPICK *

v5 ::= gksnam initial segment

v6 ::= gksnam initial pick segment

v7 ::= gksint prompt and echo type (-n..-1,1..n)

v8 ::= L gksnum 4 value echo area (DC)

v9 ::= gksdat pick data record

```
« R Error,OpMode,Echo,InStatus,InitSeg,InitPick
« R Prompt,EchoArea,PickDat
```

4.18.24 GQSTS mb Inquire string device state

2 parameters

p1 ::= gksnam workstation ID

p2 ::= gksint string device number (1..n)

» W /GQSTS("WORK1",1)

7 values returned

v1 ::= gksint error indicator

v2 ::= gksenum operating mode:
* REQUEST *
* SAMPLE *
* EVENT *

v3 ::= gksenum echo switch:
* ECHO *
* NOECHO *

v4 ::= gksstr initial string

v5 ::= gksint prompt and echo type (-n..-1,1..n)

v6 ::= L gksnum 4 value echo area (DC)

v7 ::= gksdat string data record

« R Error,OpMode,Echo,Value,EchoArea,ValDat

4.19 Inquiry functions for workstation description table

control-
mnemonic Level GKS function name

4.19.1 GQWKCA 0a Inquire workstation catagory
1 parameter

p1 ::= gksnam workstation type

» W /GQWKCA("WORK1")

2 values returned

v1 ::= gksint error indicator

v2 ::= gksenum workstation category:

```
* OUTPUT *
* INPUT  *
* OUTIN  *
* WISS   *
* MO     *
* MI     *
```

« R Error,WorkCat

4.19.2 GQWKCL 0a Inquire workstation classification
1 parameter

p1 ::= gksnam workstation type

» W /GQWKCL("WORK1")

2 values returned

v1 ::= gksint error indicator

v2 ::= gksenum classification:

```
* VECTOR *
* RASTER *
* OTHER  *
```

« R Error,Class

4.19.3 GQDSP ma Inquire display space size
1 parameter

p1 ::= gksnam workstation type

» W /GQDSP("WORK1")

4 values returned

v1 ::= gksint error indicator

v2 ::= gksenum device coordinate units:
* METRES *
* OTHER *

v3 ::= L gksnum 2 value display size in device coordinate (DC) units

v4 ::= L gksint 2 value display size in raster units (1..n)

« R Error,Units,SizeU,SizeR

4.19.4 GQDWKA 1a Inquire dynamic modification of workstation attributes
1 parameter

p1 ::= gksnam workstation type

» W /GQDWKA("WORK1")

8 values returned

v1 ::= gksint error indicator

v2 ::= gksenum representation changeable: polyline bundle

v3 ::= gksenum representation changeable: polymarker bundle

v4 ::= gksenum representation changeable: text bundle

v5 ::= gksenum representation changeable: fill area bundle

v6 ::= gksenum representation changeable: pattern

v7 ::= gksenum representation changeable: colour

v8 ::= gksenum transformation changeable: workstation, changeable values:
* IRG *
* IMM *

« R Error,PolyL,PolyM,Text,Fill,Pattrn,Color,Trans

4.19.5 GQDDS 1a Inquire default deferral state values
1 parameter

p1 ::= gksnam workstation type

» W /GQDDS("WORK1")

3 values returned

v1 ::= gksint error indicator

v2 ::= gksenum default value for deferral mode:
* ASAP *
* BNIG *
* BNIL *
* ASTI *

v3 ::= gksenum default value for implicit regeneration mode:
* SUPPRESSED *
* ALLOWED *

« R Error,Defer,Regen

4.19.6 GQPLF ma Inquire polyline facilities
1 parameter

p1 ::= gksnam workstation type

» W /GQPLF("WORK1")

8 values returned

v1 ::= gksint error indicator

v2 ::= gksint number of available linetypes (4..n)

v3 ::= M gksint n available linetypes (-n..-1,1..n)

v4 ::= gksint number of available linewidths (0..n)

v5 ::= gksnum nominal linewidth (DC)

v6 ::= gksnum minimum linewidth (DC)

v7 ::= gksnum maximum linewidth (DC)

v8 ::= gksint number of predefined polyline indices (0,5..n)

« R Error,Num F ii=1:1:Num R LineType(ii)
« R WidNum,WidNom,WidMin,WidMax,Lineindx

4.19.7 GQPPLR 0a Inquire predefined polyline representation
2 parameters

p1 ::= gksnam workstation type

p2 ::= gksint predefined polyline index (1..n)

» W /GQPPLR("WORK1",1)

4 values returned

v1 ::= gksint error indicator

v2 ::= gksint linetype (-n..-1,1..n)

v3 ::= gksnum linewidth scale factor

v4 ::= gksint polyline colour index (0..n)

« R Error,LineType,WidScale,LinColor

4.19.8 GQPMF ma Inquire polymarker facilities
1 parameter

p1 ::= gksnam workstation type

» W /GQPMF("WORK1")

8 values returned

v1 ::= gksint error indicator

v2 ::= gksint number of available marker types (5..n)

v3 ::= M gksint n available marker types (-n..-1,1..n)

v4 ::= gksint number of available marker sizes (0..n)

v5 ::= gksnum nominal marker size (DC)

v6 ::= gksnum minimum marker size (DC)

v7 ::= gksnum maximum maximum size (DC)

v8 ::= gksint number of predefined polymarker indices (0,5..n)

« R Error,Num F ii=1:1:Num R MarkType(ii)
« R SizNum,SizNom,SizMin,SizMax,Markindx

4.19.9 GQPPMR 0a Inquire predefined polymarker representation
2 parameters

p1 ::= gksnam workstation type

p2 ::= gksint predefined polymarker index (1..n)

» W /GQPPMR("WORK1",1)

4 values returned

v1 ::= gksint error indicator

v2 ::= gksint marker type (-n..-1,1..n)

v3 ::= gksnum marker size scale factor

v4 ::= gksint polymarker colour index (0..n)

« R Error,MarkType,SizScale,MarkIndx

4.19.10 GQTXF ma Inquire text facilities
1 parameter

p1 ::= gksnam workstation type

» W /GQTXF("WORK1")

10 values returned

v1 ::= gksint error indicator

v2 ::= gksint number of text font and precision pairs (1..n)

v3 ::= M gksint : gksenum
n values, text font : precision:

```
* STRING *
* CHAR   *
* STROKE *
```

v4 ::= gksint number of available character heights (0..n)

v5 ::= gksnum minimum character height (DC)

v6 ::= gksnum maximum character height (DC)

v7 ::= gksint number of available character expansion factors (0..n)

v8 ::= gksnum minimum character expansion factor

v9 ::= gksnum maximum character expansion factor

v10 ::= gksint number of predefined text indices (0,2..n)

« R Error,NumFont F ii=1:1:NumFont R FontPrec(ii)
« R NumHght,MinHght,MaxHght,NumExp,MinExp,MaxExp,NumText

4.19.11 GQPTXR 0a Inquire predefined text representation
2 parameters

p1 ::= gksnam workstation type

p2 ::= gksint predefined text index (1..n)

» W /GQPTXR("WORK1",1)

6 values returned

v1 ::= gksint error indicator

v2 ::= gksint text font (-n..-1,1..n)

v3 ::= gksenum precision:

```
* STRING *
* CHAR   *
* STROKE *
```

v4 ::= gksnum character expansion factor

v5 ::= gksnum character spacing

v6 ::= gksint text colour index (0..n)

« R Error,Font,Prec,Expan,Space,Color

4.19.12 GQFAF ma Inquire fill area facilities
1 parameter

p1 ::= gksnam workstation type

» W /GQFAF("WORK1")

6 values returned

v1 ::= gksint error indicator

v2 ::= gksint number of available fill area interior styles (1..n)

v3 ::= M gksenum n value list of fill area interior styles:

```
* HOLLOW  *
* SOLID   *
* PATTERN *
* HATCH   *
```

v4 ::= gksint number of available hatch styles (0..n)

v5 ::= M gksint n value list of hatch styles (-n..-1,1..n)

v6 ::= gksint number of predefined fill area indices (0,5..n)

« R Error,NumFillA F ii=1:1:NumFillA R FillA(ii)
« R NumHatch F ii=1:1:NumHatch R Hatch(ii)
« R NumIndex

4.19.13 GQPFAR 0a Inquire predefined fill area representation
2 parameters

p1 ::= gksnam workstation type

p2 ::= gksint predefined fill area index (1..n)

» W /GQPFAR("WORK1",1)

4 values returned

v1 ::= gksint error indicator

v2 ::= gksenum fill area interior style:

```
* HOLLOW  *
* SOLID   *
* PATTERN *
* HATCH   *
```

v3 ::= gksint fill area style index (-n..-1,1..n)

v4 ::= gksint fill area colour index (0..n)

« R Error,Style,Index,Color

4.19.14 GQPAF 0a Inquire pattern facility
1 parameter

p1 ::= gksnam workstation type

» W /GQPAF("WORK1")

2 values returned

v1 ::= gksint error indicator

v2 ::= gksint number of predefined pattern indices (0..n)

« R Error,NumPat

4.19.15 GQPPAR 0a Inquire predefined pattern representation

2 parameters

p1 ::= gksnam workstation type

p2 ::= gksint predefined pattern index (1..n)

```
» W /GQPPAR("WORK1",1)
```

3 values returned

v1 ::= gksint error indicator

v2 ::= L gksint 2 value array dimensions (n,m) (1..n)

v3 ::= M gksint n*m pattern array (0..n)

```
« R Error,DimXY F ii=1:1:DimXY*$P(DimXY,",",2) R Pattern(ii)
```

4.19.16 GQCF ma Inquire colour facilities

1 parameter

p1 ::= gksnam workstation type

```
» W /GQCF("WORK1")
```

4 values returned

v1 ::= gksint error indicator

v2 ::= gksint number of available colours or intensities (0,2..n)

v3 ::= gksenum colour available:

```
* COLOUR     *
* MONOCHROME *
```

v4 ::= gksint number of predefined colour indices (2..n)

```
« R Error,NColors,Color,NumIndx
```

4.19.17 GQPCR 0a Inquire predefined colour representation
2 parameters

p1 ::= gksnam workstation type

p2 ::= gksint predefined colour index (0..n)

» W /GQPCR("WORK1",1)

2 values returned

v1 ::= gksint error indicator

v2 ::= L gksnum 3 value colour intensity (red,green,blue)

« R Error,RGB

4.19.18 GQEGDP 0a Inquire list of available generalized drawing primitives
1 parameter

p1 ::= gksnam workstation type

» W /GQEGDP("WORK1")

3 values returned

v1 ::= gksint error indicator

v2 ::= gksint number of available generalized drawing primitives (0..n)

v3 ::= M gksnam n GDP identifiers

« R Error,Num F ii=1:1:Num R GDP(ii)

4.19.19 GQGPD 0a Inquire generalized drawing primitive
2 parameters

p1 ::= gksnam workstation type

p2 ::= gksnam GDP identifier

» W /GQGPD("WORK1","FAST")

3 values returned

v1 ::= gksint error indicator

v2 ::= gksint number of sets of attributes used (0..n)

v3 ::= M gksenum list of sets of attributes used:

```
* POLYLINE   *
* POLYMARKER *
* TEXT       *
* FILL AREA  *
```

« R Error,NumAttr F II=1:1:NumAttr R Attrib(II)

4.19.20 GQLWK ma Inquire maximum length of workstation state tables
1 parameter

p1 ::= gksnam workstation type

» W /GQLWK("WORK1")

7 values returned

v1 ::= gksint error indicator

v2 ::= gksint maximum number of polyline bundle table entries (0,5..n)

v3 ::= gksint maximum number of polymarker bundle table entries (0,5..n)

v4 ::= gksint maximum number of text bundle table entries (0,2..n)

v5 ::= gksint maximum number of fill area bundle table entries (0,5..n)

v6 ::= gksint maximum number of pattern indices (0..n)

v7 ::= gksint maximum number of colour indices (0..n)

« R Error,MaxLine,MaxMark,MaxText,MaxFill,MaxPat,MaxColr

4.19.21 GQSGP 1a Inquire number of segment priorities supported
1 parameter

p1 ::= gksnam workstation type

» W /GQSGP("WORK1")

2 values returned

v1 ::= gksint error indicator

v2 ::= gksint number of segment priorities supported (0..n)

« R Error,NumSegP

4.19.22 GQDSGA 1a Inquire dynamic modification segment attributes
1 parameter

p1 ::= gksnam workstation type

» W /GQDSGA("WORK1")

8 values returned

v1 ::= gksint error indicator

v2 ::= gksenum segment transformation changeable:
* IRG *
* IMM *

v3 ::= gksenum visibility changeable from 'visible' to 'invisible':
* IRG *
* IMM *

v4 ::= gksenum visibility changeable from 'invisible' to 'visible':
* IRG *
* IMM *

v5 ::= gksenum highlighting changeable:
* IRG *
* IMM *

v6 ::= gksenum segment priority changeable:
* IRG *
* IMM *

v7 ::= gksenum adding primitives to the open segment:
* IRG *
* IMM *

v8 ::= gksenum segment deletion immediately visible; choices:
* IRG *
* IMM *

« R Error,Trans,VisIn,InVis,Highl,Prio,AddPrim,DelVis

4.19.23 GQLI mb Inquire number of available logical input devices

1 parameter

p1 ::= gksnam workstation type

```
» W /GQLI("WORK1")
```

7 values returned

v1 ::= gksint error indicator

v2 ::= gksint number of locator devices (0..n)

v3 ::= gksint number of stroke devices (0..n)

v4 ::= gksint number of valuator devices (0..n)

v5 ::= gksint number of choice devices (0..n)

v6 ::= gksint number of pick devices (0..n)

v7 ::= gksint number of string devices (0..n)

```
« R Error,NumLoc,NumStrk,NumVal,NumCh,NumPick,NumStr
```

4.19.24 GQDLC mb Inquire default locator device data

2 parameters

p1 ::= gksnam workstation type

p2 ::= gksint logical input device number (1..n)

```
» W /GQDCL("WORK1",1)
```

6 values returned

v1 ::= gksint error indicator

v2 ::= gkspnt default initial locator position (WC)

v3 ::= gksint number of prompt and echo types (1..n)

v4 ::= M gksint n value list of available prompt and echo types (-n..-1,1..n)

v5 ::= L gksnum 4 values defining default echo area (DC)

v6 ::= gksdat default locator data record

```
« R Error,LocPnt,NumPrmpt F ii=1:1:NumPrmpt R Prompt(ii)
« R EchoArea,LocDat
```

4.19.25 GQDSK mb Inquire default stroke device data

2 parameters

p1 ::= gksnam workstation type

p2 ::= gksint logical input device number

```
» W /GQDSK("WORK1",1)
```

6 values returned

v1 ::= gksint error indicator

v2 ::= gksint maximum input buffer size (64..n)

v3 ::= gksint number of available prompt and echo types (1..n)

v4 ::= M gksint n value list of available prompt and echo types (-n..-1,1..n)

v5 ::= L gksnum 4 value default echo area (DC)

v6 ::= gksdat default stroke data record

```
« R Error,BufSiz,NumPrmp F ii=1:1:NumPrmp R Prompt(ii)
« R EchoArea,StrokDat
```

4.19.26 GQDVL mb Inquire default valuator device data

2 parameters

p1 ::= gksnam workstation type

p2 ::= gksint logical input device number (1..n)

```
» W /GQDVL("WORK1",1)
```

6 values returned

v1 ::= gksint error indicator

v2 ::= gksnum default initial value

v3 ::= gksint number of available prompt and echo types (1..n)

v4 ::= M gksint n value list of available prompt and echo types (-n..-1,1..n)

v5 ::= L gksnum 4 value default echo area (DC)

v6 ::= gksdat default valuator data record

```
« R Error,Value,NumPrmp F ii=1:1:NumPrmp R Prompt(ii)
« R EchoArea,ValueDat
```

4.19.27 GQDCH mb Inquire default choice device data

2 parameters

p1 ::= gksnam workstation type

p2 ::= gksint logical input device number (1..n)

```
» W /GQDCH("WORK1",1)
```

6 values returned

v1 ::= gksint error indicator

v2 ::= gksint maximum number of choice alternatives (1..n)

v3 ::= gksint number of available prompt and echo types (1..n)

v4 ::= M gksint n value list of available prompt and echo types (-n..-1,1..n)

v5 ::= L gksnum 4 value default echo area (DC)

v6 ::= gksdat default choice data record

```
« R Error,MaxChoic,NumChoic F ii=1:1:NumChoic R Choice(ii)
« R EchoArea,ChoicDat
```

4.19.28 GQDPK 1b Inquire default pick device data

2 parameters

p1 ::= gksnam workstation type

p2 ::= gksint logical input device number (1..n)

```
» W /GQDPK("WORK1",1)
```

5 values returned

v1 ::= gksint error indicator

v2 ::= gksint number of available prompt and echo types (1..n)

v3 ::= M gksint n value list of available prompt and echo types (-n..-1,1..n)

v4 ::= L gksnum 4 value default echo area (DC)

v5 ::= gksdat default pick data record

```
« R Error,NumPrmp F ii=1:1:NumPrmp R Prompt(ii)
« R EchoArea,PickDat
```

4.19.29 GQDST mb Inquire default string device data
2 parameters

p1 ::= gksnam workstation type

p2 ::= gksint logical input device number (1..n)

» W /GQDST("WORK1",1)

6 values returned

v1 ::= gksint error indicator

v2 ::= gksint maximum string buffer size (72..n)

v3 ::= gksint number of available prompt and echo types (1..n)

v4 ::= M gksint n value list of available prompt and echo types (-n..-1,1..n)

v5 ::= L gksnum 4 value default echo area (DC)

v6 ::= gksdat default string data record

« R Error,Strmax,Num F ii=1:1:Num R Prompt(ii)
« R Echo,Data

4.20 Inquiry functions for segment state list

control-
mnemonic Level GKS function name

4.20.1 GQASWK 1a Inquire set of associated workstations
1 parameter

p1 ::= gksnam segment name

» W /GQASWK("SEG1")

3 values returned

v1 ::= gksint error indicator

v2 ::= gksint number of associated workstations (1..n)

v3 ::= M gksnam n associated workstation ID's

« R Error,Num F ii=1:1:Num R Work(ii)

4.20.2 GQSGA 1a Inquire segment attributes
1 parameter

p1 ::= gksnam segment name

» W /GQSGA("SEG1")

6 values returned

v1 ::= gksint error indicator

v2 ::= L gksnum 2x3 value segment transform matrix

v3 ::= gksenum visibility:

```
* VISIBLE   *
* INVISIBLE *
```

v4 ::= gksenum highlighting:

```
* NORMAL      *
* HIGHLIGHTED *
```

v5 ::= gksnum segment priority [0,1]

v6 ::= gksenum detectability:

```
* UNDETECTABLE *
* DETECTABLE   *
```

« R Error,Matrix,Visib,High,Prio,Detect

4.21 Pixel inquiry

control- mnemonic	Level	GKS function name

4.21.1 GQPXAD 0a Inquire pixel array dimensions
3 parameters

p1 ::= gksnam workstation ID

p2 ::= gkspnt point P (WC)

p3 ::= gkspnt point Q (WC)

» W /GQPXAD("WORK1","1,1","100,100")

2 values returned

v1 ::= gksint error indicator

v2 ::= L gksint 2 values; dimension of pixel array (1..n)

« R Error,WidLen

4.21.2 GQPXA 0a Inquire pixel array
3 parameters

p1 ::= gksnam workstation ID

p2 ::= gkspnt point P (WC)

p3 ::= L gksint 2 values; dimension of colour index array (n,m) (1..n)

» W /GQPXA("WORK1","1,1","10,10")

3 values returned

v1 ::= gksint error indicator

v2 ::= gksenum presence of invalid values:
* ABSENT *
* PRESENT *

v3 ::= M gksint n*m colour index array (-1..n)

« R Error,Invalid F ii=1:1:10*10 R Color(ii)

4.21.3 GQPX 0a Inquire pixel

2 parameters

p1 ::= gksnam workstation ID

p2 ::= gkspnt pixel point (WC)

» W /GQPX("WORK1","100,100")

2 values returned

v1 ::= gksint error indicator

v2 ::= gksint colour index (-1..n)

« R Error,Color

4.22 Inquiry function for GKS error state list

control- mnemonic	Level	GKS function name

4.22.1 GQIQOV mc Inquire input queue overflow

No parameters

» W /GQIQOV

4 values returned

v1 ::= gksint error indicator

v2 ::= gksstr workstation ID

v3 ::= gksenum input class:

```
* LOCATOR  *
* STROKE   *
* VALUATOR *
* CHOICE   *
* PICK     *
* STRING   *
```

v4 ::= gksint logical input device number (1..n)

« R Error,WorkSta,InClass,InDev

4.23 Utility functions

control-mnemonic Level GKS function name

4.23.1 GEVTM 1a Evaluate transformation matrix
5 parameters

p1 ::= gkspnt fixed point (WC/NDC)

p2 ::= L gksnum 2 value shift vector (WC/NDC)

p3 ::= gksnum rotation angle in radians (positive if counterclockwise)

p4 ::= L gksnum 2 value scale factor

p5 ::= gksenum coordinate switch:

```
* WC  *
* NDC *
```

» W /GevTM("1,1","1.1,0.5","1.223","2.0,1.5","WC")

1 value returned

v1 ::= L gksnum 2x3 segment transform matrix

« R Matrix

4.23.2 GACTM 1a Accumulate transformation matrix
6 parameters

p1 ::= gksstr segment transformation matrix

p2 ::= gkspnt fixed point (WC/NDC)

p3 ::= L gksnum 2 value shift vector (WC/NDC)

p4 ::= gksnum rotation angle in radians (positive if counterclockwise)

p5 ::= L gksnum 2 value scale factor

p6 ::= gksenum coordinate switch:

```
* WC  *
* NDC *
```

» W /GACTM(1,"1,1","1.1,0.5","1.223","2.0,1.5","WC")

1 value returned

v1 ::= L gksnum 2x3 segment transform matrix

« R Matrix

4.24 Error handling

<u>**control-**</u>
<u>**mnemonic**</u> **Level** **GKS function name**

4.24.1 GECLGKS 0a Emergency close GKS
No parameters

» W /GECLGKS

4.24.2 GERHND 0a Error handling
3 parameters

<u>p1</u> ::= <u>gksint</u> error number

<u>p2</u> ::= <u>gksnam</u> identification of GKS procedure called by the application program which caused the error detection

<u>p3</u> ::= <u>gksfile</u> error file

» W /GERHND(123,"AAA","ERROR.FIL")

4.24.3 GERLOG 0a Error logging
3 parameters

<u>p1</u> ::= <u>gksint</u> error number

<u>p2</u> ::= <u>gksnam</u> identification of GKS procedure called by the application program which caused the error detection

<u>p3</u> ::= <u>gksfile</u> error file

» W /GERLOG(321,"AAA","ERROR.FIL")