INTERNATIONAL STANDARD

ISO 8651-2

First edition
1988-02-01

INTERNATIONAL ORGANIZATION FOR STANDARDIZATION
ORGANISATION INTERNATIONALE DE NORMALISATION
МЕЖДУНАРОДНАЯ ОРГАНИЗАЦИЯ ПО СТАНДАРТИЗАЦИИ

Information processing systems — Computer graphics — Graphical Kernel System (GKS) language bindings —

Part 2 : Pascal

Systèmes de traitement de l'information — Infographie — Système graphique de base (GKS) — Interface langage

Partie 2 : Pascal

Reference number
ISO 8651-2 : 1988 (E

Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work.

Draft International Standards adopted by the technical committees are circulated to the member bodies for approval before their acceptance as International Standards by the ISO Council. They are approved in accordance with ISO procedures requiring at least 75 % approval by the member bodies voting.

International Standard ISO 8651-2 was prepared by Technical Committee ISO/TC 97, *Information processing systems*.

Users should note that all International Standards undergo revision from time to time and that any reference made herein to any other International Standard implies its latest edition, unless otherwise stated.

Printed in Switzerland

Contents

Page

Annexes

This page intentionally left blank

Information processing systems — Computer graphics — Graphical Kernel System (GKS) language bindings —

Part 2 : Pascal

0 Introduction

The Graphical Kernel System (GKS), the functional description of which is given in ISO 7942, 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 part of ISO 8651 is to define a standard binding for the Pascal computer programming language.

1 Scope and field of application

ISO 7942 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 part of ISO 8651 specifies such a language-dependent layer for the Pascal language.

2 References

ISO 7942, *Information processing systems - Computer graphics - Graphical Kernel System (GKS) functional description.*

ISO 7185, *Programming languages - Pascal.*

ISO 2382-13, *Data processing - Vocabulary - Part 13: Computer Graphics.*

3 The Pascal language binding of GKS

3.1 Specification

The GKS language binding interface for ISO Pascal (ISO 7185) shall be as described in clauses 3, 4, 5, and 6.

3.2 Mapping of GKS function names to Pascal procedure names

The function names of GKS are all mapped to Pascal procedures which begin with the letter "G". Words and phrases used in the GKS function names are often abbreviated in the Pascal representation. There is a set of such abbreviations given in table 1 and the resulting Pascal procedure names are listed in tables 2, 3, and 4. For example, the abbreviation for the GKS function DELETE SEGMENT FROM WORKSTATION is GDelSegWs. "Del", "Seg", "Ws" are the abbreviations for DELETE, SEGMENT and WORKSTATION. Conjunctives such as "from", "and", "of" and "to" are mapped to null strings, as are a number of other words used in the GKS abstract names. For example, INQUIRE MAXIMUM LENGTH OF WORKSTATION STATE TABLES is mapped to GInqMaxWsSt. Here LENGTH and TABLES are represented by null strings.

3.3 The many-one nature of the Pascal interface

There is not a strict one-to-one correspondence between GKS abstract functions and Pascal procedures. A method employing variant records is used to represent several logically related GKS abstract functions by one Pascal procedure. The first parameter of such a procedure is always an enumerated type which is the tag field of a variant record which is itself a parameter of the Pascal procedure. This technique is used across two classes of abstract functions - those relating to the setting and inquiring of output primitive representations, and those relating to the setting and inquiring of the input classes. Where this method is used, the rules for deriving the Pascal name of the GKS abstract function are

a) Output Primitive Representations

   1) The GKS words Polyline, Polymarker, Text, and Fill Area are replaced by "Prim" (which is the abbreviation for PRIMITIVE).

   2) The first parameter of the function is an enumerated type (GEPrim) which has one of the values GVPolyline, GVPolymarker, GVText, GVFillArea.

b) Input Classes

   1) The GKS words Locator, Stroke, Valuator, Choice, Pick, and String are replaced by "Input".

   2) The first parameter of the function is an enumerated type (GEInputClass) which has one of the values GVLocator, GVStroke, GVValuator, GVChoice, GVPick, GVString.

3.4 The one-one nature of the Pascal interface

The Pascal interface to GKS described in 3.3 reflects the GKS major dimensions of Output Primitive Representations and Input Classes. However, the possibility exists that on small systems such an interface might cause difficulties, especially with respect to implementation of the Input Classes. Therefore, the Pascal Binding also adopts a mandatory representation which uses a one-one mapping for the setting of primitive representations and input classes.

3.5 The one-many nature of the Pascal interface

The GKS abstract functions INQUIRE CURRENT PRIMITIVE ATTRIBUTE VALUES and INQUIRE CURRENT INDIVIDUAL ATTRIBUTE VALUES are represented by the method described in 3.4. In addition, to allow for the possible frequent use of only some of the information, these

functions have also been split into a number of Pascal procedures. Both representations are mandatory.

3.6 Implementation of the interfaces

Since any of the methods referred to in 3.3, 3.4 and 3.5 can be implemented easily in terms of another, the additional interfaces do not present a great burden for the implementor, nor does it cause an additional burden for application programs. Implementors are encouraged to use one method in the core of their implementation. In any event all sets of procedures shall be provided.

Table 1 - Abbreviations ordered alphabetically

GKS word	Abbreviation
ACCUMULATE	Accum
ALIGNMENT	Align
ALL	NULL
AND	NULL
ASPECT SOURCE FLAGS	ASF
ASSOCIATE	Assoc
ATTRIBUTE	Attr
ATTRIBUTES	Attr
AVAILABLE	NULL
CHARACTER	Char
CLASSIFICATION	Class
CLIPPING	Clip
COLOUR	Colr
CONNECTION	Conn
CURRENT	Cur
DEFAULT	Def
DEFERRAL	Defer
DELETE	Del
DETECTABILITY	Det
DIMENSIONS	Dim
DYNAMIC	Dyn
EVALUATE	Eval
EXPANSION	Expan
FACILITIES	Facil
FACTOR	NULL
FILL AREA	Fill
FROM	NULL
GENERALIZED DRAWING PRIMITIVE	GDP
GRAPHICAL KERNEL SYSTEM	GKS
GKSM	NULL
HIGHLIGHTING	Highlight
IDENTIFIER	Id
IN	NULL
INDEX	Ind
INDICATOR	NULL
INDICES	Ind
INDIVIDUAL	Indiv
INITIALISE	Init
INPUT	NULL
INQUIRE	Inq
INTERIOR	Int

Table 1 - Abbreviations ordered alphabetically

GKS word	Abbreviation
LENGTH	NULL
LIST	NULL
LOGICAL	NULL
MATRIX	NULL
MAXIMUM	Max
MODIFICATION	Mod
NAME	NULL
NORMALIZATION	Norm
NUMBER	Num
NUMBERS	Num
OF	NULL
ON	NULL
OPERATING	Op
POLYLINE	Line
POLYMARKER	Marker
PRECISION	Prec
PREDEFINED	Pred
PRIMITIVE	Prim
QUEUE	NULL
REFERENCE	Ref
REPRESENTATION	Rep
REQUEST	Req
SEGMENT	Seg
SEGMENTS	Seg
SET	NULL
SIMULTANEOUS	NULL
SPACE	NULL
STATE	St
SUPPORTED	NULL
TABLES	NULL
TO	NULL
TRANSFORMATION	Tran
UPDATE	Upd
USE	NULL
VALUE	NULL
VALUES	NULL
VISIBILITY	Vis
WITH	NULL
WORKSTATION	Ws

NOTE - NULL represents the null string

The Pascal language binding of GKS

Table 2 - GKS function names and Pascal names ordered by Pascal name

GKS Function Name	Level	Pascal Name
ACCUMULATE TRANSFORMATION MATRIX	L1a	GAccumTran
ACTIVATE WORKSTATION	L0a	GActivateWs
ASSOCIATE SEGMENT WITH WORKSTATION	L2a	GAssocSegWs
AWAIT EVENT	L0c	GAwaitEvent
CELL ARRAY	L0a	GCellArray
CLEAR WORKSTATION	L0a	GClearWs
CLOSE GKS	L0a	GCloseGKS
CLOSE SEGMENT	L1a	GCloseSeg
CLOSE WORKSTATION	L0a	GCloseWs
COPY SEGMENT TO WORKSTATION	L2a	GCopySegWs
CREATE SEGMENT	L1a	GCreateSeg
DEACTIVATE WORKSTATION	L0a	GDeactivateWs
DELETE SEGMENT	L1a	GDelSeg
DELETE SEGMENT FROM WORKSTATION	L1a	GDelSegWs
EMERGENCY CLOSE GKS	L0a	GEmergencyCloseGKS
ERROR HANDLING	L0a	GErrorHandling
ERROR LOGGING	L0a	GErrorLogging
ESCAPE	L0a	GEscape
ESCAPE	L0a	GEscapeGeneralized
EVALUATE TRANSFORMATION MATRIX	L1a	GEvalTran
FILL AREA	L0a	GFill
FLUSH DEVICE EVENTS	L0c	GFlushDeviceEvents
GENERALIZED DRAWING PRIMITIVE (GDP)	L0a	GGDP
GENERALIZED DRAWING PRIMITIVE (GDP)	L0a	GGDPGeneralized
GET CHOICE	L0c	GGetChoice
GET CHOICE	L0c	GGetInput(Choice
GET LOCATOR	L0c	GGetInput(Locator
GET PICK	L1c	GGetInput(Pick
GET STRING	L0c	GGetInput(String
GET STROKE	L0c	GGetInput(Stroke
GET VALUATOR	L0c	GGetInput(Valuator
GET ITEM TYPE FROM GKSM	L0a	GGetItemType
GET LOCATOR	L0c	GGetLocator
GET PICK	L1c	GGetPick
GET STRING	L0c	GGetString
GET STROKE	L0c	GGetStroke
GET VALUATOR	L0c	GGetValuator
INITIALISE CHOICE	L0b	GInitChoice
INITIALISE CHOICE	L0b	GInitInput(Choice
INITIALISE LOCATOR	L0b	GInitInput(Locator
INITIALISE PICK	L1b	GInitInput(Pick
INITIALISE STRING	L0b	GInitInput(String
INITIALISE STROKE	L0b	GInitInput(Stroke
INITIALISE VALUATOR	L0b	GInitInput(Valuator
INITIALISE LOCATOR	L0b	GInitLocator
INITIALISE PICK	L1b	GInitPick
INITIALISE STRING	L0b	GInitString
INITIALISE STROKE	L0b	GInitStroke
INITIALISE VALUATOR	L0b	GInitValuator

Table 2 - Names ordered by Pascal name **The Pascal language binding of GKS**

GKS Function Name	Level	Pascal Name
INQUIRE ASPECT SOURCE FLAGS	L0a	GInqASF
INQUIRE SET OF ACTIVE WORKSTATIONS	L1a	GInqActiveWs
INQUIRE SET OF ASSOCIATED WORKSTATIONS	L1a	GInqAssocWs
INQUIRE CHARACTER BASE VECTOR	L0a	GInqCharBaseVector
INQUIRE CHARACTER EXPANSION FACTOR	L0a	GInqCharExpan
INQUIRE CHARACTER HEIGHT	L0a	GInqCharHeight
INQUIRE CHARACTER SPACING	L0a	GInqCharSpacing
INQUIRE CHARACTER UP VECTOR	L0a	GInqCharUpVector
INQUIRE CHARACTER WIDTH	L0a	GInqCharWidth
INQUIRE CHOICE DEVICE STATE	L0b	GInqChoiceDeviceSt
INQUIRE CLIPPING	L0a	GInqClip
INQUIRE COLOUR FACILITIES	L0a	GInqColrFacil
INQUIRE COLOUR REPRESENTATION	L0a	GInqColrRep
INQUIRE CURRENT INDIVIDUAL ATTRIBUTE VALUES	L0a	GInqCurIndivAttr
INQUIRE CURRENT NORMALIZATION TRANSFORMATION NUMBER	L0a	GInqCurNormTranNum
INQUIRE CURRENT PICK IDENTIFIER	L1b	GInqCurPickId
INQUIRE CURRENT PRIMITIVE ATTRIBUTE VALUES	L0a	GInqCurPrimAttr
INQUIRE DEFAULT CHOICE DEVICE DATA	L0b	GInqDefChoiceDeviceData
INQUIRE DEFAULT DEFERRAL STATE VALUES	L1a	GInqDefDeferSt
INQUIRE DEFAULT CHOICE DEVICE DATA	L0b	GInqDefInputDeviceData(Choice
INQUIRE DEFAULT LOCATOR DEVICE DATA	L0b	GInqDefInputDeviceData(Locator
INQUIRE DEFAULT PICK DEVICE DATA	L1b	GInqDefInputDeviceData(Pick
INQUIRE DEFAULT STRING DEVICE DATA	L0b	GInqDefInputDeviceData(String
INQUIRE DEFAULT STROKE DEVICE DATA	L0b	GInqDefInputDeviceData(Stroke
INQUIRE DEFAULT VALUATOR DEVICE DATA	L0b	GInqDefInputDeviceData(Valuator
INQUIRE DEFAULT LOCATOR DEVICE DATA	L0b	GInqDefLocatorDeviceData
INQUIRE DEFAULT PICK DEVICE DATA	L1b	GInqDefPickDeviceData
INQUIRE DEFAULT STRING DEVICE DATA	L0b	GInqDefStringDeviceData
INQUIRE DEFAULT STROKE DEVICE DATA	L0b	GInqDefStrokeDeviceData
INQUIRE DEFAULT VALUATOR DEVICE DATA	L0b	GInqDefValuatorDeviceData
INQUIRE DISPLAY SPACE SIZE	L0a	GInqDisplaySize
INQUIRE DYNAMIC MODIFICATION OF SEGMENT ATTRIBUTES	L1a	GInqDynModSegAttr
INQUIRE DYNAMIC MODIFICATION OF WORKSTATION ATTRIBUTES	L1a	GInqDynModWsAttr
INQUIRE FILL AREA COLOUR INDEX	L0a	GInqFillColrInd
INQUIRE FILL AREA FACILITIES	L0a	GInqFillFacil
INQUIRE FILL AREA INDEX	L0a	GInqFillInd
INQUIRE FILL AREA INTERIOR STYLE	L0a	GInqFillIntStyle
INQUIRE FILL AREA REPRESENTATION	L1a	GInqFillRep
INQUIRE FILL AREA STYLE INDEX	L0a	GInqFillStyleInd
INQUIRE GENERALIZED DRAWING PRIMITIVE	L0a	GInqGDP
INQUIRE CHOICE DEVICE STATE	L0b	GInqInputDeviceSt(Choice
INQUIRE LOCATOR DEVICE STATE	L0b	GInqInputDeviceSt(Locator
INQUIRE PICK DEVICE STATE	L1b	GInqInputDeviceSt(Pick
INQUIRE STRING DEVICE STATE	L0b	GInqInputDeviceSt(String
INQUIRE STROKE DEVICE STATE	L0b	GInqInputDeviceSt(Stroke
INQUIRE VALUATOR DEVICE STATE	L0b	GInqInputDeviceSt(Valuator
INQUIRE INPUT QUEUE OVERFLOW	L0c	GInqInputOverflow
INQUIRE LEVEL OF GKS	L0a	GInqLevelGKS
INQUIRE POLYLINE COLOUR INDEX	L0a	GInqLineColrInd
INQUIRE POLYLINE INDEX	L0a	GInqLineInd
INQUIRE LINETYPE	L0a	GInqLineType

Table 2 - Names ordered by Pascal name

GKS Function Name	Level	Pascal Name
INQUIRE LINEWIDTH SCALE FACTOR	L0a	GInqLineWidthScale
INQUIRE LIST OF COLOUR INDICES	L0a	GInqListColrInd
INQUIRE LIST OF FILL AREA INDICES	L1a	GInqListFillInd
INQUIRE LIST OF AVAILABLE GENERALIZED DRAWING PRIMITIVES	L0a	GInqListGDP
INQUIRE LIST OF NORMALIZATION TRANSFORMATION NUMBERS	L0a	GInqListNormTranNum
INQUIRE LIST OF PATTERN INDICES	L1a	GInqListPatternInd
INQUIRE LIST OF POLYLINE INDICES	L1a	GInqListPolylineInd
INQUIRE LIST OF POLYMARKER INDICES	L1a	GInqListPolymarkerInd
INQUIRE LIST OF FILL AREA INDICES	L1a	GInqListPrimInd(FillArea
INQUIRE LIST OF POLYLINE INDICES	L1a	GInqListPrimInd(Polyline
INQUIRE LIST OF POLYMARKER INDICES	L1a	GInqListPrimInd(Polymarker
INQUIRE LIST OF TEXT INDICES	L1a	GInqListPrimInd(Text
INQUIRE LIST OF TEXT INDICES	L1a	GInqListTextInd
INQUIRE LIST OF AVAILABLE WORKSTATION TYPES	L0a	GInqListWsTypes
INQUIRE LOCATOR DEVICE STATE	L0b	GInqLocatorDeviceSt
INQUIRE POLYMARKER COLOUR INDEX	L0a	GInqMarkerColrInd
INQUIRE POLYMARKER INDEX	L0a	GInqMarkerInd
INQUIRE POLYMARKER SIZE SCALE FACTOR	L0a	GInqMarkerSizeScale
INQUIRE POLYMARKER TYPE	L0a	GInqMarkerType
INQUIRE MAXIMUM NORMALIZATION TRANSFORMATION NUMBER	L0a	GInqMaxNormTranNum
INQUIRE MAXIMUM LENGTH OF WORKSTATION STATE TABLES	L0a	GInqMaxWsSt
INQUIRE MORE SIMULTANEOUS EVENTS	L0c	GInqMoreEvents
INQUIRE NORMALIZATION TRANSFORMATION	L0a	GInqNormTran
INQUIRE NUMBER OF AVAILABLE LOGICAL INPUT DEVICES	L0b	GInqNumInputDevices
INQUIRE NUMBER OF SEGMENT PRIORITIES SUPPORTED	L1a	GInqNumSegPriorities
INQUIRE OPERATING STATE VALUE	L0a	GInqOpSt
INQUIRE NAME OF OPEN SEGMENT	L1a	GInqOpenSeg
INQUIRE SET OF OPEN WORKSTATIONS	L0a	GInqOpenWs
INQUIRE PATTERN FACILITIES	L0a	GInqPatternFacil
INQUIRE PATTERN REFERENCE POINT	L0a	GInqPatternRefPoint
INQUIRE PATTERN REPRESENTATION	L1a	GInqPatternRep
INQUIRE PATTERN SIZE	L0a	GInqPatternSize
INQUIRE PICK DEVICE STATE	L1b	GInqPickDeviceSt
INQUIRE PIXEL	L0a	GInqPixel
INQUIRE PIXEL ARRAY	L0a	GInqPixelArray
INQUIRE PIXEL ARRAY DIMENSIONS	L0a	GInqPixelArrayDim
INQUIRE POLYLINE FACILITIES	L0a	GInqPolylineFacil
INQUIRE POLYLINE REPRESENTATION	L1a	GInqPolylineRep
INQUIRE POLYMARKER FACILITIES	L0a	GInqPolymarkerFacil
INQUIRE POLYMARKER REPRESENTATION	L1a	GInqPolymarkerRep
INQUIRE PREDEFINED COLOUR REPRESENTATION	L0a	GInqPredColrRep
INQUIRE PREDEFINED FILL AREA REPRESENTATION	L0a	GInqPredFillRep
INQUIRE PREDEFINED PATTERN REPRESENTATION	L0a	GInqPredPatternRep
INQUIRE PREDEFINED POLYLINE REPRESENTATION	L0a	GInqPredPolylineRep
INQUIRE PREDEFINED POLYMARKER REPRESENTATION	L0a	GInqPredPolymarkerRep
INQUIRE PREDEFINED FILL AREA REPRESENTATION	L0a	GInqPredPrimRep(FillArea
INQUIRE PREDEFINED POLYLINE REPRESENTATION	L0a	GInqPredPrimRep(Polyline
INQUIRE PREDEFINED POLYMARKER REPRESENTATION	L0a	GInqPredPrimRep(Polymarker
INQUIRE PREDEFINED TEXT REPRESENTATION	L0a	GInqPredPrimRep(Text
INQUIRE PREDEFINED TEXT REPRESENTATION	L0a	GInqPredTextRep
INQUIRE FILL AREA FACILITIES	L0a	GInqPrimFacil(FillArea