INTERNATIONAL STANDARD

ISO/IEC 9593-1

First edition
1990-06-01

AMENDMENT 1
1995-04-15

Information processing systems — Computer graphics — Programmer's Hierarchical Interactive Graphics System (PHIGS) language bindings —

Part 1: FORTRAN

AMENDMENT 1

Systèmes de traitement de l'information — Infographie — Interfaces langage entre un programme d'application et son support graphique

Partie 1: FORTRAN

AMENDEMENT 1

Reference number
ISO/IEC 9593-1:1990/Amd.1:1995(E)

Contents

ISO/IEC Copyright Office • Case postale 56 • CH-1211 Genève 20 • Switzerland

Printed in Switzerland

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work.

In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

Amendment 1 to International Standard ISO/IEC 9593-1:1990 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*.

Introduction

Replace the Introduction of ISO/IEC 9593-1 with the following text:

ISO/IEC 9592-1:1989, provides a set of functions for the display and modification of 2D or 3D graphical data. Part 1 is extended by Part 4 (PHIGS PLUS) to incorporate the effects of lighting, shading and other properties that are important for the display of surfaces and multidimensional data.

ISO/IEC 9592-1 and ISO/IEC 9592-4 are specified in a language independent manner and must be embedded in language dependent layers (language bindings) for use with particular programming languages.

The purpose of this part of ISO/IEC 9593 is to define the FORTRAN language binding for ISO/IEC 9592-1 and ISO/IEC 9592-4.

Information processing systems — Computer graphics — Programmer's Hierarchical Interactive Graphics System (PHIGS) language bindings —

Part 1: FORTRAN

AMENDMENT 1

1 Scope

Replace clause 1:

The "Programmer's Hierarchical Interactive Graphics System" (PHIGS), ISO/IEC 9592-1:1989, and ISO/IEC 9592-4:1992, specify a language independent nucleus of a graphics system. For integration into a programming language, PHIGS PLUS is embedded in a language dependent layer obeying the particular conventions of that language. This part of ISO/IEC 9593 specifies the FORTRAN language dependent layer.

2 Normative references

Add the following reference to clause 2:

ISO/IEC 9592-4:1992, *Information processing systems - Computer graphics - Programmer's Hierarchical Interactive Graphics System (PHIGS) Part 4 - Plus Lumière Und Surfaces (PHIGS PLUS).*

3 Principles

3.1 Specification

Replace subclause 3.1, Specification, of ISO/IEC 9593-1 with the following text:

This part of ISO/IEC 9593 defines the PHIGS and PHIGS PLUS language binding interface for FORTRAN 77, as described in ISO 1539: 1980. With some minor modifications, application programs can be transported between full FORTRAN77 and FORTRAN77 Subset PHIGS and PHIGS PLUS installations.

This binding incorporates the rules of conformance defined in the PHIGS (ISO/IEC 9592-1) and PHIGS PLUS (ISO/IEC 9592-4) Standard for PHIGS and PHIGS PLUS implementations, with those additional requirements specifically defined for FORTRAN language implementations defined in this part of ISO/IEC 9593. The following criteria are established for determining conformance of an implementation to this binding:

In order to conform, an implementation of the FORTRAN binding of PHIGS shall implement those functions specified in ISO/IEC 9592-1. The implementation shall make visible all of the declarations in the FORTRAN binding specified in clause 5 to 10 in this part of ISO/IEC 9593.

In order to conform, an implementation of the FORTRAN binding of PHIGS PLUS shall implement those functions specified in ISO/IEC 9592-1 and also those functions specified in ISO/IEC 9592-4. The implementation shall make visible all of the declarations in the FORTRAN binding specified in clause 11 to 14 and in clause 5 to 10, as modified by clause 11 to 14, in this part of ISO/IEC 9593.

Thus, for example, the syntax of the function names shall be precisely as specified in this part of ISO/IEC 9593 and the parameters shall be of the data types stated in this part of ISO/IEC 9593.

A PHIGS FORTRAN application should run without modification under a PHIGS PLUS FORTRAN binding implementation.

3.2 Mapping of PHIGS function names to FORTRAN subroutine names

Append the following paragraph to subclause 3.2,

However, two abbreviations are changed for PHIGS PLUS; MAPPING becomes M and WORKSTATION becomes W, due to the FORTRAN subroutine naming restrictions.

3.3 Parameters

No change to ISO/IEC 9593-1.

3.4 The FORTRAN subset

No change to ISO/IEC 9593-1.

3.5 Error handling

No change to ISO/IEC 9593-1.

4 Generating FORTRAN subroutine names

Add the following, alphabetically, to table 2 of clause 4:

Table 2 - Reduce compound terms for uniqueness

REPRESENTATION PLUS	-> P

Add the following, alphabetically, to table 3 of clause 4:

Table 3 - Deletions

GEOMETRIC	NON-UNIFORM	WITH

Add the following, alphabetically, to table 4 to clause 4:

Table 4 - Abbreviations

PHIGS PLUS word	abbreviation	remarks
APPROXIMATION	A	
B-SPLINE	BS	
BACK	B	
CHARACTERISTICS	C	
COORDINATE	CD	
CRITERIA	C	
CUE	C	
CULLING	C	
CURVE	C	
DATA	D	
DEPTH	D	SET DEPTH CUE INDEX:DP
DIRECT	D	
DISTINGUISHING	D	
DYNAMICS	DC	
FACET	F	
LIGHT	L	
MAPPING	M	
MESH	M	FOR PHIGS PLUS
METHOD	M	
PARAMETRIC	P	
PLACEMENT	P	
PLUS	P	
PROPERTIES	P	
QUADRILATERAL	Q	
REFLECTANCE	RF	
RENDERING	R	
SHADING	S	
SPLINE	S	
STRIP	ST	
SURFACE	S	
SOURCE	S	
TRIMMING	T	
TRIANGLE	T	
WORKSTATION	W	FOR PHIGS PLUS

5 Data types

Append the following data type definitions to clause 5:

COLRV colour value
INTEGER containing colour index when CTYPE is INDIRECT
INTEGER NCC when CTYPE is not INDIRECT
REAL$_1$,REAL$_2$,REAL$_3$,...REAL$_n$ containing $C_1,C_2,C_3,...C_n$ ($_n$=NCC) when CTYPE is not INDIRECT

COLRVH homogeneous colour value
REAL$_1$,REAL$_2$,REAL$_3$,...REAL$_n$ and REAL containing $WC_1,WC_2,WC_3,...WC_n$ and W ($_n$=NCC)

GCOLR general colour
a compound data type containing colour type and colour value of **COLRV**

NORM normal vector
REAL,REAL,REAL containing X-,Y- and Z- values

P2H two-dimensional homogeneous point
REAL,REAL,REAL containing WX-,WY- and W- values (or WU-,WV- and W- of a trimming curve control point) W value is ignored in case of non-rational type

P3H three-dimensional homogeneous point
REAL,REAL,REAL,REAL containing WX-,WY-,WZ- and W- values
W value is ignored in case of non-rational type

A(P3) array of coordinates of points
REAL(*),REAL(*),REAL(*) containing X-,Y- and Z- values(*=number of columns by number of rows)

L(L(P3{,COLRV})) list of vertex data lists
REAL PXA(*),PYA(*),PZA(*) *=last value of array of end indices for point lists(NP)
INTEGER VCOLI(*) *=NP, when CTYPE is INDIRECT
REAL VCOLR(*) *=number of components of colour value(NCC) × NP, when CTYPE is not INDIRECT

{COLRV}{,NORM}{,L(R)} facet data
INTEGER FCOLI when CTYPE is INDIRECT
REAL FCOLR(*) *=number of components of colour value(NCC), when CTYPE is not INDIRECT
REAL FNXA,FNYA,FNZA facet normal data
REAL FDLEN length of application-specific data
REAL FDATA(FDLEN) facet application-specific data

L(L(E)) list of edge flags lists
INTEGER EDATA(*) *=last value of array of end indices for point lists

L(L(P2{,COLRV}{,NORM}{,L(R)}))
REAL PXA(*),PYA(*) *=last value of array of end indices for point lists(NP)
INTEGER VCOLI(*) *=NP, when CTYPE is INDIRECT
REAL VCOLR(*) *=number of components of colour value(NCC) × NP, when CTYPE

is not INDIRECT
REAL VNXA(*),VNYA(*) vertex normal data(*=NP)
INTEGER VDLEN length of application-specific data
REAL VDATA(*) vertex application-specific data(*=VDLEN × NP)

L(L(P3{,COLRV}{,NORM}{,L(R)}))
REAL PXA(*),PYA(*),PZA(*) *=last value of array of end indices for point lists(NP)
INTEGER VCOLI(*) *=NP, when CTYPE is INDIRECT
REAL VCOLR(*) *=number of components of colour value(NCC) × NP, when CTYPE is not INDIRECT
REAL VNXA(*),VNYA(*),VNZA(*) vertex normal data(*=NP)
INTEGER VDLEN length of application-specific data
REAL VDATA(*) vertex application-specific data(*=VDLEN × NP)

L(L(L(I))) list of lists of vertex indices lists
REAL VIND(*) *=last value of array of end vertex indices for each fill area

L(L(L(E))) list of lists of edge flags lists
INTEGER EDATA(*) *=last value of array of end vertex indices for each fill area

L({COLRV}{,NORM}{,L(R)}) list of facet data
INTEGER FCOLI(*) *=number of fill area sets(N) in set of fill area sets 3 with data or set of fill area sets with data, number of triangles(N) at following 4 functions, triangle set 3 with data, triangle set with data, triangle strip 3 with data and triangle strip with data, when CTYPE is INDIRECT
REAL FCOLR(*) *=number of components of colour value(NCC) x NFS, when CTYPE is not INDIRECT
REAL FNXA(*),FNYA(*),FNZA(*) facet normal data(*=N)
INTEGER FDLEN length of application-specific data
REAL FDATA(*) facet application-specific data(*=FDLEN × N)

L(P3{,COLRV}{,NORM}{,L(R)}) list of vertex data
REAL PXA(*),PYA(*),PZA(*) *=number of points(NP)
INTEGER VCOLI(*) *=NP, when CTYPE is INDIRECT
REAL VCOLR(*) *=number of components of colour value(NCC) x NP, when CTYPE is not INDIRECT
REAL VNXA(*),VNYA(*),VNZA(*) vertex normal data(*=NP)
INTEGER VDLEN length of application-specific data
REAL VDATA(*) vertex application-specific data(*=VDLEN x NP)

L(P2{,COLRV}{,NORM}{,L(R)}) list of vertex data
REAL PXA(*),PYA(*) *=number of points(NP)
INTEGER VCOLI(*) *=NP, when CTYPE is INDIRECT
REAL VCOLR(*) *=number of components of colour value(NCC) x NP, when CTYPE is not INDIRECT
REAL VNXA(*),VNYA(*),VNZA(*) vertex normal data(*=NP)
INTEGER VDLEN length of application-specific data
REAL VDATA(*) vertex application-specific data(*=VDLEN x NP)

A(COLRV) colour array
INTEGER COLIA(*) *=number of columns(DIMX) × number of rows(DIMY), when CTYPE is INDIRECT
REAL COLRA(*) *=number of components of colour value(NCC) × DIMX × DIMY, when CTYPE is not INDIRECT

A({COLRV}{,NORM}{,L(R)}) array of facet data

INTEGER FCOLI(*) *=number of columns minus 1(NC-1) × number of rows minus 1(NR-1), when CTYPE is INDIRECT

REAL FCOLR(*) *=number of components of colour value(NCC) × NC-1 × NR-1, when CTYPE is not INDIRECT

REAL FNXA(*),FNYA(*),FNZA(*) facet normal data(*= NC-1 × NR-1)

INTEGER FDLEN length of application-specific data

REAL FDATA(*) facet application-specific data(*=FDLEN × NC-1 × NR-1)

A(P3{,COLRV}{,NORM}{,L(R)}) array of vertex data

REAL PXA(*),PYA(*),PZA(*) *=number of columns(NC) × number of rows(NR)

INTEGER VCOLI(*) *=NC × NR, when CTYPE is INDIRECT

REAL VCOLR(*) *=number of components of colour value(NCC) × NC × NR, when CTYPE is not INDIRECT

REAL VNXA(*),VNYA(*),VNZA(*) vertex normal data(*= NC × NR)

INTEGER VDLEN length of application-specific data

REAL VDATA(*) vertex application-specific data(*=VDLEN × NC × NR)

A(P2{,COLRV}{,NORM}{,L(R)}) array of vertex data

REAL PXA(*),PYA(*) *=number of columns(NC) × number of rows(NR)

INTEGER VCOLI(*) *=NC × NR, when CTYPE is INDIRECT

REAL VCOLR(*) *=number of components of colour value(NCC) × NC × NR, when CTYPE is not INDIRECT

REAL VNXA(*),VNYA(*),VNZA(*) vertex normal data(*= NC × NR)

INTEGER VDLEN length of application-specific data

REAL VDATA(*) vertex application-specific data(*=VDLEN × NC × NR)

A(2×E) array of edge data

INTEGER EDATA(*) *=2 × number of columns(NC) × number of rows(NR)

L(P3H)|L(P3) list of curve control points

REAL PXA(*),PYA(*),PZA(*),PWA(*) *=number of control points, when CRTYPE is PRAT

REAL PXA(*),PYA(*),PZA(*) *=number of control points, when CRTYPE is PNRAT

L(COLRVH)|L(COLRV) list of colour spline control points

REAL CSCR(*) *={number of components of colour value(NCC)+1} × number of colour spline control points(NCSCP), when CRTYPE is PRAT

REAL CSCR(*) *=number of components of colour value(NCC) × number of colour spline control points(NCSCP), when CRTYPE is PNRAT

A(P3H)|A(P3) array of surface control points

REAL PXA(*),PYA(*),PZA(*),PWA(*) *=u number of control points dimension(UNCP) × v number of control points dimension(VNCP), when CRTYPE is PRAT

REAL PXA(*),PYA(*),PZA(*) *=u number of control points dimension(UNCP) × v number of control points dimension(VNCP), when CRTYPE is PNRAT

L(L(TRIMCURVE)) list of trimming loop definitions lists

INTEGER TACRI(*) *=number of components of list of trimming loop definitions lists(NCLTL)

INTEGER TCVF(*) trimming curve visibility flag(*=NCLTL)

INTEGER TSORD(*) trimming curve spline order(*=NCLTL)

INTEGER TNKA(*) number of spline knots(*=NCLTL)

REAL TKNOTS(*) *=last value of array of number of spline knots in array(TNKA)

REAL TPARL(2,*) trimming curve parameter range limits(*=NCLTL)

INTEGER TRTYPE(*) trimming curve spline rationality(*=NCLTL)

INTEGER NTCCP(*) number of timming curve spline control points(*=NCLTL)
REAL TPWXA(*),TPWYA(*),TPWWA(*) *=last value of number of trimming curve spline control points, when CRTYPE is PRAT
REAL TPWXA(*),TPWYA(*) *=last value of number of trimming curve spline control points, when CRTYPE is PNRAT

L(P2H)|L(P2) list of trimming curve control points
REAL TPWUA(*),TPWVA(*),TPWWA(*) *=number of control points, when CRTYPE is PRAT
REAL TPWUA(*),TPWVA(*) *=number of control points, when CRTYPE is PNRAT

A(COLRVH)|A(COLRV) array of colour spline control points
REAL CSCP(*) *={number of components of colour value(NCC)+1} × u number of colour spline control points dimension(NUCSCP) × v number of colour spline control points dimension(NVCSCP), when CRTYPE is PRAT
REAL CSCP(*) *=number of components of colour value(NCC) × u number of colour spline control points dimension(NUCSCP) × v number of colour spline control points dimension(NVCSCP), when CRTYPE is PNRAT

L(DATASPLINE) list of data spline (for non-uniform B-spline surface with data)
INTEGER DUSORD(*) u data spline order(*=NDS)
INTEGER DVSORD(*) v data spline order(*=NDS)
INTEGER DNUKA(*) number of u spline knots(*=NDS)
INTEGER DNVKA(*) number of v spline knots(*=NDS)
REAL DUKNTS(*) *=last value of number of u spline knots(DNUKA) × NDS
REAL DVKNTS(*) *=last value of number of v spline knots(DNVKA) × NDS
INTEGER DRTYPE(*) data spline rationality(*=NDS)
INTEGER NUDSCP(*) u number of data spline control points dimension(*=NDS)
INTEGER NVDSCP(*) v number of data spline control points dimension(*=NDS)
INTEGER DDIMS(*) data dimension(*=NDS)
REAL DSCP(*) *=

$$\sum_{i=1}^{NDS} (NUDSCP(i) \times NVDSCP(i) \times DDIMS(i))$$

DDIMS(i) = n+1 (n = $d_1,d_2,d_3,...d_n$,w) when DRTYPE(i) is PRAT.
DDIMS(i) = n (n = $d_1,d_2,d_3,...d_n$) when DRTYPE(i) is PNRAT.

A(DATAH)|A(DATA) control points of data spline
REAL DSCP(*) *=

$$\sum_{i=1}^{NDS} (NUDSCP(i) \times NVDSCP(i) \times DDIMS(i))$$

DDIMS(i) = n+1 (n = $d_1,d_2,d_3,...d_n$,w) when DRTYPE(i) is PRAT.
DDIMS(i) = n (n = $d_1,d_2,d_3,...d_n$) when DRTYPE(i) is PNRAT.

NDS=number of data spline
NUDSCP=u number of data spline control points dimension
NVDSCP=v number of data spline control points dimension
DDIMS=data dimension for spline

6 Enumeration types

No change to ISO/IEC 9593-1.

7 List of the PHIGS function names

No change to ISO/IEC 9593-1.

8 PHIGS errors specific to the FORTRAN binding

No change to ISO/IEC 9593-1.

9. The PHIGS function interface

No change to ISO/IEC 9593-1.

10 Utility functions not defined in PHIGS

No change to ISO/IEC 9593-1.

Add the following new clauses 11, 12, 13 and 14:

11 PHIGS PLUS Enumeration types

All the enumeration types of PHIGS PLUS are mapped to FORTRAN INTEGERs. The correspondence between PHIGS PLUS scalars and FORTRAN INTEGERs is shown as follows in a list of symbolic FORTRAN constants that may be included in any application program. This clause contains a mapping of PHIGS PLUS enumeration types to FORTRAN variable names.

"Polyline shading method", "interior shading method", "data mapping method", "reflectance property type", "reflectance model", "curve placement", "light source type", "colour mapping method", "curve approximation type", "surface approximation type" and "parametric surface charcteristic type" are defined as INTEGER rather than enumeration types in PHIGS PLUS. Constant definitions for the explicitly defined and required values of these conceptually unbounded ranges are provided as a convenience.

Also, a numbering of all PHIGS PLUS functions is given for use in the error handling procedures.

Mnemonic FORTRAN names and their values for PHIGS PLUS ENUMERATION type values:

aspect identifier

	polyline colour				
INTEGER	PPLCOL,	PPMCOL,	PTXCOL,	PINCOL,	PEDCOL,
1	PPSHMD,	PISHMD,	PDMPMD,	PREFPR,	PREFM,
2	PBINSY,	PBINSI,	PBICOL,	PBISHM,	PBDMPM,
3	PBREFP,	PBREFM,	PCAPCR,	PSAPCR,	PPASUC
PARAMETER(	PPLCOL=18,	PPMCOL=19,	PTXCOL=20,	PINCOL=21,	PEDCOL=22,
1	PPSHMD=23,	PISHMD=24,	PDMPMD=25,	PREFPR=26,	PREFM=27,
2	PBINSY=28,	PBINSI=29,	PBICOL=30,	PBISHM=31,	PBDMPM=32,
3	PBREFP=33,	PBREFM=34,	PCAPCR=35,	PSAPCR=36,	PPASUC=37)

colour mapping method

	true	*pseudo*	*pseudo-n*
INTEGER	PTRUE,	PSUD,	PSUDN
PARAMETER(	PTRUE=1,	PSUD=2,	PSUDN=3)

INTEGER rather than enumeration type. Explicitly defined and required portion of conceptually unbounded range defined here.

colour type

	indirect
INTEGER	PINDIR
PARAMETER(	PINDIR=0)

The following colour models are also colour types.
RGB,CIELUV,HSV,HLS

culling mode

	none	*backfacing*	*frontfacing*
INTEGER	PNOFC,	PBKFC,	PFTFC
PARAMETER(	PNOFC=0,	PBKFC=1,	PFTFC=2)