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

# 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 ) |

**curve approximation criteria type**

| | *workstation dependent chordal deviation in WC relative DC* | *constant between knots chordal deviation in NPC* | *chordal size in WC chordal deviation in DC* | *chordal size in NPC relative WC* | *chordal size in DC relative NPC* |
|---|---|---|---|---|---|
| INTEGER | PWDCA, | PCBKCA, | PCSWCA, | PCSNCA, | PCSDCA, |
| 1 | PCDWCA, | PCDNCA, | PCDDCA, | PRWCA, | PRNCA, |
| 2 | PRDCA | | | | |
| PARAMETER( | PWDCA=1, | PCBKCA=2, | PCSWCA=3, | PCSNCA=4, | PCSDCA=5, |
| 1 | PCDWCA=6, | PCDNCA=7, | PCDDCA=8, | PRWCA=9, | PRNCA=10, |
| 2 | PRDCA=11) | | | | |

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

**curve placement (of ISOPARAMETRIC CURVE)**

| | *over surface* | *between knots* |
|---|---|---|
| INTEGER | PUOSCP, | PUBKCP |
| PARAMETER( | PUOSCP=0, | PUBKCP=1) |

**curve visibility flag**

Uses the POFF/PON enumeration defined for edge flag and error handling mode.

**data mapping method**

| | *colour* | *single uniform* | *single non-uniform* | *BI uniform* | *BI non-uniform* |
|---|---|---|---|---|---|
| INTEGER | PCDM, | PSUDM, | PSNUDM, | PBUDM, | PBNUDM |
| PARAMETER( | PCDM=1, | PSUDM=2, | PSNUDM=3, | PBUDM=4, | PBNUDM=5) |

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

**depth cue mode**

| | *suppressed* | *allowed* |
|---|---|---|
| INTEGER | PSUPPR | PALLOW |
| PARAMETER( | PSUPPR=0, | PALLOW=1 ) |

**distinguishing mode**

Uses the POFF/PON enumeration defined for edge flag and error handling mode.

**edge data flag**

| | *none* | *edge visibility flag* |
|---|---|---|
| INTEGER | PENO | PEVF |
| PARAMETER( | PENO=0, | PEVF=1 ) |

Not defined in PHIGS PLUS but used to indicate presence of optional data for FORTRAN binding.

**element type**

| | *polyline set 3 with data .....* | | | | |
|---|---|---|---|---|---|
| INTEGER | PEPLS3, | PEFS3D, | PEFSD, | PECA3P, | PESFS3, |
| 1 | PESFSD, | PETS3D, | PETSD, | PETST3, | PETSTD, |
| 2 | PEQM3D, | PEQMD, | PEBC3, | PEBC3C, | PEBS3, |
| 3 | PEBS3D, | PEDMI, | PERFI, | PEBII, | PEBDMI, |
| 4 | PEBRFI, | PEPRSI, | PEPLC, | PEPLSM, | PEPMC, |
| 5 | PETXC, | PEFDM, | PEFCM, | PEIC, | PEISM, |
| 6 | PEDMM, | PERFP, | PERFM, | PEBIS, | PEBISI, |
| 7 | PEBIC, | PEBISM, | PEBDMM, | PEBRFP, | PEBRFM, |
| 8 | PELSS, | PEEDC, | PECAC, | PESAC, | PEPSC, |
| 9 | PERCM, | PEDPCI, | PECMI) | | |
| PARAMETER( | PEPLS3=71, | PEFS3D=72, | PEFSD=73, | PECA3P=74, | PESFS3=75, |
| 1 | PESFSD=76, | PETS3D=77, | PETSD=78, | PETST3=79, | PETSTD=80, |
| 2 | PEQM3D=81, | PEQMD=82, | PEBC3=83, | PEBC3C=84, | PEBS3=85, |
| 3 | PEBS3D=86, | PEDMI=87, | PERFI=88, | PEBII=89, | PEBDMI=90, |
| 4 | PEBRFI=91, | PEPRSI=92, | PEPLC=93, | PEPLSM=94, | PEPMC=95, |
| 5 | PETXC=96, | PEFDM=97, | PEFCM=98, | PEIC=99, | PEISM=100, |
| 6 | PEDMM=101, | PERFP=102, | PERFM=103, | PEBIS=104, | PEBISI=105, |
| 7 | PEBIC=106, | PEBISM=107, | PEBDMM=108, | PEBRFP=109, | PEBRFM=110, |
| 8 | PELSS=111, | PEEDC=112, | PECAC=113, | PESAC=114, | PEPSC=115, |
| 9 | PERCM=116, | PEDPCI=117, | PECMI=118) | | |

**facet data flag**

| | *no facet data colour* | *normal* | *data* | *colour/normal* | |
|---|---|---|---|---|---|
| | *colour/data* | *normal/data* | *colour/normal/data* | | |
| INTEGER | PFNO, | PFC, | PFN, | PFD, | PFCN, |
| 1 | PFCD, | PFND, | PFCND | | |
| PARAMETER( | PFNO=0, | PFC=1, | PFN=2, | PFD=3, | PFCN=4, |
| 1 | PFCD=5, | PFND=6, | PFCND=7 ) | | |

Not defined in PHIGS PLUS but used to indicate presence of optional data for FORTRAN binding.

**GDP attributes**

| | *reflectance* | *parametric surface* |
|---|---|---|
| INTEGER | PRFATT, | PPSATT |
| PARAMETER( | PRFATT=5, | PPSATT=6 ) |

**interior shading method**

| | *none* | *colour* | *data* | *data/dot* | *data/normal* |
|---|---|---|---|---|---|
| INTEGER | PNOIS, | PCIS, | PDIS, | PDDIS, | PDNIS |
| PARAMETER( | PNOIS=1, | PCIS=2, | PDIS=3, | PDDIS=4, | PDNIS=5) |

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

**light source type**

| | *ambient* | *directional* | *positional* | *spot* |
|---|---|---|---|---|
| INTEGER | PAMB, | PDIRE, | PPOSI, | PSPOT |
| PARAMETER( | PAMB=1, | PDIRE=2, | PPOSI=3, | PSPOT=4) |

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

**parametric surface characteristic type**

| | *none* | *workstation dependent* | *isoparametric curve* | *level curve in MC* | *level curve in WC* |
|---|---|---|---|---|---|
| INTEGER | PNOPC, | PWDPC, | PICPC, | PLCMPC, | PLCWPC |
| PARAMETER( | PNOPC=1, | PWDPC=2, | PICPC=3, | PLCMPC=4, | PLCWPC=5) |

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

**polyline shading method**

| | *none* | *colour* |
|---|---|---|
| INTEGER | PNOPS, | PCPS |
| PARAMETER( | PNOPS=1, | PCPS=2 ) |

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

**rationality**

| | *rational* | *non-rational* |
|---|---|---|
| INTEGER | PRAT, | PNRAT |
| PARAMETER( | PRAT=0, | PNRAT=1 ) |

**reflectance model**

| | *no reflectance* | *ambient* | *ambient/diffuse* | *ambient/diffuse/specular* |
|---|---|---|---|---|
| INTEGER | PNORM, | PARM, | PADRM, | PADSRM |
| PARAMETER( | PNORM=1, | PARM=2, | PADRM=3, | PADSRM=4) |

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

**reflectance property type**

| | *simple reflectance* |
|---|---|
| INTEGER | PSRPT |
| PARAMETER( | PSRPT=1) |

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

**surface approximation criteria**

| | *workstation dependent planar deviation in WC relative in DC* | *constant between knots planar deviation in NPC* | *chordal size in WC planar deviation in DC* | *chordal size in NPC relative in WC* | *chordal size in DC relative in NPC* |
|---|---|---|---|---|---|
| INTEGER | PWDSA, | PCBKSA, | PCSWSA, | PCSNSA, | PCSDSA, |
| 1 | PPDWSA, | PPDNSA, | PPDDSA, | PRWSA, | PRNSA, |
| 2 | PRDSA | | | | |
| PARAMETER( | PWDSA=1, | PCBKSA=2, | PCSWSA=3, | PCSNSA=4, | PCSDSA=5, |
| 1 | PPDWSA=6, | PPDNSA=7, | PPDDSA=8, | PRWSA=9, | PRNSA=10, |
| 2 | PRDSA=11) | | | | |

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

**source selector**

| | *colour aspect* | *vertex colour* | *vertex data* | *facet colour* | *facet data* |
|---|---|---|---|---|---|
| INTEGER | PSCASF, | PSVC, | PSVD, | PSFC, | PSFD |
| PARAMETER( | PSCASF=0, | PSVC=1, | PSVD=2, | PSFC=3, | PSFD=4 ) |

**trimming curve visibility flag**

Uses the POFF/PON enumeration defined for edge flag and error handling mode.

**vertex data flag**

| | *coordinate only colour/data* | *colour normal/data* | *normal colour/normal/data* | *data* | *colour/normal* |
|---|---|---|---|---|---|
| INTEGER | PCD, | PCDC, | PCDN, | PCDD, | PCDCN, |
| 1 | PCDCD, | PCDND, | PCDCND | | |
| PARAMETER( | PCD=0, | PCDC=1, | PCDN=2, | PCDD=3, | PCDCN=4, |
| 1 | PCDCD=5, | PCDND=6, | PCDCND=7 ) | | |

Not defined in PHIGS PLUS but used to indicate presence of optional data for FORTRAN binding.

**PHIGS PLUS** functions. These names are used for error handling. The names are the same as the PHIGS PLUS function names except that the sentinel character 'P' is replaced by 'E'. The same function identification used for full FORTRAN 77.

```
INTEGER     EPLS3C,     EFAS3D,     EFASD,     ECA3P,     ESFAS3
PARAMETER(  EPLS3C=183, EFAS3D=184, EFASD=185, ECA3P=186, ESFAS3=187)
INTEGER     ESFASD,     ETS3D,      ETSD,      ETST3D,    ETSTD
PARAMETER(  ESFASD=188, ETS3D=189,  ETSD=190,  ETST3D=191,ETSTD=192)
INTEGER     EQM3D,      EQMD,       EBSC3,     EBSC3C,    EBSS3
PARAMETER(  EQM3D=193,  EQMD=194,   EBSC3=195, EBSC3C=196,EBSS3=197)
INTEGER     EBSS3D,     ESDMI,      ESRFI,     ESBII,     ESBDMI
PARAMETER(  EBSS3D=198, ESDMI=199,  ESRFI=200, ESBII=201, ESBDMI=202)
INTEGER     ESBRFI,     ESPSI,      ESPLC,     ESPLSM,    ESPMC
PARAMETER(  ESBRFI=203, ESPSI=204,  ESPLC=205, ESPLSM=206,ESPMC=207)
INTEGER     ESTXC,      ESFDM,      ESFCM,     ESIC,      ESISM
PARAMETER(  ESTXC=208,  ESFDM=209,  ESFCM=210, ESIC=211,  ESISM=212)
INTEGER     ESDMM,      ESRFP,      ESRFM,     ESBIS,     ESBISI
PARAMETER(  ESDMM=213,  ESRFP=214,  ESRFM=215, ESBIS=216, ESBISI=217)
INTEGER     ESBIC,      ESBISM,     ESBDMM,    ESBRFP,    ESBRFM
PARAMETER(  ESBIC=218,  ESBISM=219, ESBDMM=220,ESBRFP=221,ESBRFM=222)
INTEGER     ESLSS,      ESEDC,      ESCAC,     ESSAC,     ESPSC
PARAMETER(  ESLSS=223,  ESEDC=224,  ESCAC=225, ESSAC=226, ESPSC=227)
INTEGER     ESRCM,      ESDPCI,     ESCMI,     ESPLP,     ESPMP
PARAMETER(  ESRCM=228,  ESDPCI=229, ESCMI=230, ESPLP=231, ESPMP=232)
INTEGER     ESTXRP,     ESIP,       ESEDP,     ESDMR,     ESRFR
PARAMETER(  ESTXRP=233, ESIP=234,   ESEDP=235, ESDMR=236, ESRFR=237)
INTEGER     ESPSR,      ESPAP,      ESLSR,     ESDCR,     ESCMR
PARAMETER(  ESPSR=238,  ESPAP=239,  ESLSR=240, ESDCR=241, ESCMR=242)
```

## 12. List of the PHIGS PLUS function names

The complete list of PHIGS PLUS function names follows.

### 12.1. List of functions ordered alphabetically by bound name

| | | |
|---|---|---|
| PBSC3 | BS-C-3 | NON-UNIFORM B-SPLINE CURVE 3 |
| PBSC3C | BS-C-3-C | NON-UNIFORM B-SPLINE CURVE 3 WITH COLOUR |
| PBSS3 | BS-S-3 | NON-UNIFORM B-SPLINE SURFACE 3 |
| PBSS3D | BS-S-3-D | NON-UNIFORM B-SPLINE SURFACE 3 WITH DATA |
| PCA3P | C-A-3-P | CELL ARRAY 3 PLUS |
| PFAS3D | F-A-S-3-D | FILL AREA SET 3 WITH DATA |
| PFASD | F-A-S-D | FILL AREA SET WITH DATA |
| PPCSC | P-C-S-C | PACK COLOUR SPLINE CURVE |
| PPCSS | P-C-S-S | PACK COLOUR SPLINE SURFACE |
| PPDSS | P-D-S-S | PACK DATA SPLINE SURFACE |
| PPLS3C | PL-S-3-C | POLYLINE SET 3 WITH COLOUR |
| PPTC | P-T-C | PACK TRIMMING CURVE |
| PQBSSF | Q-BS-S-F | INQUIRE B-SPLINE SURFACE FACILITIES |
| PQCMF | Q-C-M-F | INQUIRE COLOUR MAPPING FACILITIES |
| PQCMMF | Q-C-M-M-F | INQUIRE COLOUR MAPPING METHOD FACILITIES |
| PQCMR | Q-C-M-R | INQUIRE COLOUR MAPPING REPRESENTATION |
| PQCMS | Q-C-M-S | INQUIRE COLOUR MAPPING STATE |
| PQCVF | Q-CV-F | INQUIRE CURVE FACILITIES |
| PQDCF | Q-D-C-F | INQUIRE DEPTH CUE FACILITIES |
| PQDCMF | Q-D-C-M-F | INQUIRE DIRECT COLOUR MODEL FACILITIES |
| PQDCR | Q-D-C-R | INQUIRE DEPTH CUE REPRESENTATION |
| PQDCWP | Q-DC-W-P | INQUIRE DYNAMICS OF WORKSTATION PLUS |
| PQDMF | Q-D-M-F | INQUIRE DATA MAPPING FACILITIES |
| PQDMR | Q-D-M-R | INQUIRE DATA MAPPING REPRESENTATION |
| PQECMI | Q-E-C-M-I | INQUIRE LIST element OF COLOUR MAPPING INDICES |
| PQEDCI | Q-E-D-C-I | INQUIRE LIST element OF DEPTH CUE INDICES |
| PQEDMI | Q-E-D-M-I | INQUIRE LIST element OF DATA MAPPING INDICES |
| PQEDP | Q-ED-P | INQUIRE EDGE REPRESENTATION PLUS |
| PQELSI | Q-E-L-S-I | INQUIRE LIST element OF LIGHT SOURCE INDICES |
| PQEPSI | Q-E-P-S-I | INQUIRE LIST element OF PARAMETRIC SURFACE INDICES |
| PQERFI | Q-E-R-I | INQUIRE LIST element OF REFLECTANCE INDICES |
| PQIFP | Q-I-F-P | INQUIRE INTERIOR FACILITIES PLUS |
| PQIP | Q-I-P | INQUIRE INTERIOR REPRESENTATION PLUS |
| PQLSF | Q-L-S-F | INQUIRE LIGHT SOURCE FACILITIES |
| PQLSR | Q-L-S-R | INQUIRE LIGHT SOURCE REPRESENTATION |
| PQM3D | Q-M-3-D | QUADRILATERAL MESH 3 WITH DATA |
| PQMD | Q-M-D | QUADRILATERAL MESH WITH DATA |
| PQPAP | Q-PA-P | INQUIRE PATTERN REPRESENTATION PLUS |
| PQPCMR | Q-P-C-M-R | INQUIRE PREDEFINED COLOUR MAPPING REPRESENTATION |
| PQPDCR | Q-P-D-C-R | INQUIRE PREDEFINED DEPTH CUE REPRESENTATION |
| PQPDMR | Q-P-D-M-R | INQUIRE PREDEFINED DATA MAPPING REPRESENTATION |
| PQPEDP | Q-P-ED-P | INQUIRE PREDEFINED EDGE REPRESENTATION PLUS |
| PQPIP | Q-P-I-P | INQUIRE PREDEFINED INTERIOR REPRESENTATION PLUS |
| PQPLFP | Q-PL-F-P | INQUIRE POLYLINE FACILITIES PLUS |
| PQPLP | Q-PL-P | INQUIRE POLYLINE REPRESENTATION PLUS |
| PQPLSR | Q-P-L-S-R | INQUIRE PREDEFINED LIGHT SOURCE REPRESENTATION |
| PQPMP | Q-PM-P | INQUIRE POLYMARKER REPRESENTATION PLUS |
| PQPPAP | Q-P-PA-P | INQUIRE PREDEFINED PATTERN REPRESENTATION PLUS |
| PQPPLP | Q-P-PL-P | INQUIRE PREDEFINED POLYLINE REPRESENTATION PLUS |
| PQPPMP | Q-P-PM-P | INQUIRE PREDEFINED POLYMARKER REPRESENTATION PLUS |
| PQPPSR | Q-P-P-S-R | INQUIRE PREDEFINED PARAMETRIC SURFACE REPRESENTATION |
| PQPRFR | Q-P-RF-R | INQUIRE PREDEFINED REFLECTANCE REPRESENTATION |
| PQPSF | Q-P-S-F | INQUIRE PARAMETRIC SURFACE FACILITIES |
| PQPSR | Q-P-S-R | INQUIRE PARAMETRIC SURFACE REPRESENTATION |
| PQPTXP | Q-P-TX-P | INQUIRE PREDEFINED TEXT REPRESENTATION PLUS |
| PQRCMF | Q-R-C-M-F | INQUIRE RENDERING COLOUR MODEL FACILITIES |
| PQRFF | Q-RF-F | INQUIRE REFLECTANCE FACILITIES |
| PQRFR | Q-RF-R | INQUIRE REFLECTANCE REPRESENTATION |

| | | |
|---|---|---|
| PQTCF | Q-T-C-F | INQUIRE TRIMMING CURVE FACILITIES |
| PQTXP | Q-TX-P | INQUIRE TEXT REPRESENTATION PLUS |
| PQWSLP | Q-W-S-L-P | INQUIRE WORKSTATION STATE TABLE LENGTHS PLUS |
| PSBDMI | S-B-D-M-I | SET BACK DATA MAPPING INDEX |
| PSBDMM | S-B-D-M-M | SET BACK DATA MAPPING METHOD |
| PSBIC | S-B-I-C | SET BACK INTERIOR COLOUR |
| PSBII | S-B-I-I | SET BACK INTERIOR INDEX |
| PSBIS | S-B-I-S | SET BACK INTERIOR STYLE |
| PSBISI | S-B-I-S-I | SET BACK INTERIOR STYLE INDEX |
| PSBISM | S-B-I-S-M | SET BACK INTERIOR SHADING METHOD |
| PSBRFI | S-B-RF-I | SET BACK REFLECTANCE INDEX |
| PSBRFM | S-B-RF-M | SET BACK REFLECTANCE MODEL |
| PSBRFP | S-B-RF-P | SET BACK REFLECTANCE PROPERTIES |
| PSCAC | S-C-A-C | SET CURVE APPROXIMATION CRITERIA |
| PSCMI | S-C-M-I | SET COLOUR MAPPING INDEX |
| PSCMR | S-C-M-R | SET COLOUR MAPPING REPRESENTATION |
| PSDCI | S-D-C-I | SET DEPTH CUE INDEX |
| PSDCR | S-D-C-R | SET DEPTH CUE REPRESENTATION |
| PSDMI | S-D-M-I | SET DATA MAPPING INDEX |
| PSDMM | S-D-M-M | SET DATA MAPPING METHOD |
| PSDMR | S-D-M-R | SET DATA MAPPING REPRESENTATION |
| PSEDC | S-ED-C | SET EDGE COLOUR |
| PSEDP | S-ED-P | SET EDGE REPRESENTATION PLUS |
| PSFAS3 | S-F-A-S-3 | SET OF FILL AREA SETS 3 WITH DATA |
| PSFASD | S-F-A-S-D | SET OF FILL AREA SETS WITH DATA |
| PSFCM | S-F-C-M | SET FACET CULLING MODE |
| PSFDM | S-F-D-M | SET FACET DISTINGUISHING MODE |
| PSIC | S-I-C | SET INTERIOR COLOUR |
| PSIP | S-I-P | SET INTERIOR REPRESENTATION PLUS |
| PSISM | S-I-S-M | SET INTERIOR SHADING METHOD |
| PSLSR | S-L-S-R | SET LIGHT SOURCE REPRESENTATION |
| PSLSS | S-L-S-S | SET LIGHT SOURCE STATE |
| PSPAP | S-PA-P | SET PATTERN REPRESENTATION PLUS |
| PSPLC | S-PL-C | SET POLYLINE COLOUR |
| PSPLP | S-PL-P | SET POLYLINE REPRESENTATION PLUS |
| PSPLSM | S-PL-S-M | SET POLYLINE SHADING METHOD |
| PSPMC | S-PM-C | SET POLYMARKER COLOUR |
| PSPMP | S-PM-P | SET POLYMARKER REPRESENTATION PLUS |
| PSPSC | S-P-S-C | SET PARAMETRIC SURFACE CHARACTERISTICS |
| PSPSI | S-P-S-I | SET PARAMETRIC SURFACE INDEX |
| PSPSR | S-P-S-R | SET PARAMETRIC SURFACE REPRESENTATION |
| PSRCM | S-R-C-M | SET RENDERING COLOUR MODEL |
| PSRFI | S-RF-I | SET REFLECTANCE INDEX |
| PSRFM | S-RF-M | SET REFLECTANCE MODEL |
| PSRFP | S-RF-P | SET REFLECTANCE PROPERTIES |
| PSRFR | S-RF-R | SET REFLECTANCE REPRESENTATION |
| PSSAC | S-S-A-C | SET SURFACE APPROXIMATION CRITERIA |
| PSTXC | S-TX-C | SET TEXT COLOUR |
| PSTXRP | S-TX-R-P | SET TEXT REPRESENTATION PLUS |
| PTS3D | T-S-3-D | TRIANGLE SET 3 WITH DATA |
| PTSD | T-S-D | TRIANGLE SET WITH DATA |
| PTST3D | T-ST-3-D | TRIANGLE STRIP 3 WITH DATA |
| PTSTD | T-ST-D | TRIANGLE STRIP WITH DATA |
| PUCSC | U-C-S-C | UNPACK COLOUR SPLINE CURVE |
| PUCSS | U-C-S-S | UNPACK COLOUR SPLINE SURFACE |
| PUDSS | U-D-S-S | UNPACK DATA SPLINE SURFACE |
| PUTC | U-T-C | UNPACK TRIMMING CURVE |

## 12.2 List of functions ordered alphabetically by PHIGS PLUS function name

| | | |
|---|---|---|
| PCA3P | C-A-3-P | CELL ARRAY 3 PLUS |
| PFAS3D | F-A-S-3-D | FILL AREA SET 3 WITH DATA |
| PFASD | F-A-S-D | FILL AREA SET WITH DATA |
| PQBSSF | Q-BS-S-F | INQUIRE B-SPLINE SURFACE FACILITIES |
| PQCMF | Q-C-M-F | INQUIRE COLOUR MAPPING FACILITIES |
| PQCMMF | Q-C-M-M-F | INQUIRE COLOUR MAPPING METHOD FACILITIES |
| PQCMR | Q-C-M-R | INQUIRE COLOUR MAPPING REPRESENTATION |
| PQCMS | Q-C-M-S | INQUIRE COLOUR MAPPING STATE |
| PQCVF | Q-CV-F | INQUIRE CURVE FACILITIES |
| PQDMF | Q-D-M-F | INQUIRE DATA MAPPING FACILITIES |
| PQDMR | Q-D-M-R | INQUIRE DATA MAPPING REPRESENTATION |
| PQDCF | Q-D-C-F | INQUIRE DEPTH CUE FACILITIES |
| PQDCR | Q-D-C-R | INQUIRE DEPTH CUE REPRESENTATION |
| PQDCMF | Q-D-C-M-F | INQUIRE DIRECT COLOUR MODEL FACILITIES |
| PQDCWP | Q-DC-W-P | INQUIRE DYNAMICS OF WORKSTATION PLUS |
| PQEDP | Q-ED-P | INQUIRE EDGE REPRESENTATION PLUS |
| PQIFP | Q-I-F-P | INQUIRE INTERIOR FACILITIES PLUS |
| PQIP | Q-I-P | INQUIRE INTERIOR REPRESENTATION PLUS |
| PQLSF | Q-L-S-F | INQUIRE LIGHT SOURCE FACILITIES |
| PQLSR | Q-L-S-R | INQUIRE LIGHT SOURCE REPRESENTATION |
| PQECMI | Q-E-C-M-I | INQUIRE LIST element OF COLOUR MAPPING INDICES |
| PQEDMI | Q-E-D-M-I | INQUIRE LIST element OF DATA MAPPING INDICES |
| PQEDCI | Q-E-D-C-I | INQUIRE LIST element OF DEPTH CUE INDICES |
| PQELSI | Q-E-L-S-I | INQUIRE LIST element OF LIGHT SOURCE INDICES |
| PQEPSI | Q-E-P-S-I | INQUIRE LIST element OF PARAMETRIC SURFACE INDICES |
| PQERFI | Q-E-R-I | INQUIRE LIST element OF REFLECTANCE INDICES |
| PQPSF | Q-P-S-F | INQUIRE PARAMETRIC SURFACE FACILITIES |
| PQPSR | Q-P-S-R | INQUIRE PARAMETRIC SURFACE REPRESENTATION |
| PQPAP | Q-PA-P | INQUIRE PATTERN REPRESENTATION PLUS |
| PQPLFP | Q-PL-F-P | INQUIRE POLYLINE FACILITIES PLUS |
| PQPLP | Q-PL-P | INQUIRE POLYLINE REPRESENTATION PLUS |
| PQPMP | Q-PM-P | INQUIRE POLYMARKER REPRESENTATION PLUS |
| PQPCMR | Q-P-C-M-R | INQUIRE PREDEFINED COLOUR MAPPING REPRESENTATION |
| PQPDMR | Q-P-D-M-R | INQUIRE PREDEFINED DATA MAPPING REPRESENTATION |
| PQPDCR | Q-P-D-C-R | INQUIRE PREDEFINED DEPTH CUE REPRESENTATION |
| PQPEDP | Q-P-ED-P | INQUIRE PREDEFINED EDGE REPRESENTATION PLUS |
| PQPIR | Q-P-I-P | INQUIRE PREDEFINED INTERIOR REPRESENTATION PLUS |
| PQPLSR | Q-P-L-S-R | INQUIRE PREDEFINED LIGHT SOURCE REPRESENTATION |
| PQPPSR | Q-P-P-S-R | INQUIRE PREDEFINED PARAMETRIC SURFACE REPRESENTATION |
| PQPPAP | Q-P-PA-P | INQUIRE PREDEFINED PATTERN REPRESENTATION PLUS |
| PQPPLP | Q-P-PL-P | INQUIRE PREDEFINED POLYLINE REPRESENTATION PLUS |
| PQPPMP | Q-P-PM-P | INQUIRE PREDEFINED POLYMARKER REPRESENTATION PLUS |
| PQPRFR | Q-P-RF-R | INQUIRE PREDEFINED REFLECTANCE REPRESENTATION |
| PQPTXP | Q-P-TX-P | INQUIRE PREDEFINED TEXT REPRESENTATION PLUS |
| PQRFF | Q-RF-F | INQUIRE REFLECTANCE FACILITIES |
| PQRFR | Q-RF-R | INQUIRE REFLECTANCE REPRESENTATION |
| PQRCMF | Q-R-C-M-F | INQUIRE RENDERING COLOUR MODEL FACILITIES |
| PQTXP | Q-TX-P | INQUIRE TEXT REPRESENTATION PLUS |
| PQTCF | Q-T-C-F | INQUIRE TRIMMING CURVE FACILITIES |
| PQWSLP | Q-W-S-L-P | INQUIRE WORKSTATION STATE TABLE LENGTHS PLUS |
| PBSC3 | BS-C-3 | NON-UNIFORM B-SPLINE CURVE 3 |
| PBSCC3 | BS-C-3-C | NON-UNIFORM B-SPLINE CURVE 3 WITH COLOUR |
| PBSS3 | BS-S-3 | NON-UNIFORM B-SPLINE SURFACE 3 |
| PBSSD3 | BS-S-3-D | NON-UNIFORM B-SPLINE SURFACE 3 WITH DATA |
| PPCSC | P-C-S-C | PACK COLOUR SPLINE CURVE |
| PPCSS | P-C-S-S | PACK COLOUR SPLINE SURFACE |
| PPDSS | P-D-S-S | PACK DATA SPLINE SURFACE |
| PPTC | P-T-C | PACK TRIMMING CURVE |
| PPLS3C | PL-S-3-C | POLYLINE SET 3 WITH COLOUR |
| PQM3D | Q-M-3-D | QUADRILATERAL MESH 3 WITH DATA |
| PQMD | Q-M-D | QUADRILATERAL MESH WITH DATA |
| PSBDMI | S-B-D-M-I | SET BACK DATA MAPPING INDEX |

| | | |
|---|---|---|
| PSBDMM | S-B-D-M-M | SET BACK DATA MAPPING METHOD |
| PSBIC | S-B-I-C | SET BACK INTERIOR COLOUR |
| PSBII | S-B-I-I | SET BACK INTERIOR INDEX |
| PSBISM | S-B-I-S-M | SET BACK INTERIOR SHADING METHOD |
| PSBIS | S-B-I-S | SET BACK INTERIOR STYLE |
| PSBISI | S-B-I-S-I | SET BACK INTERIOR STYLE INDEX |
| PSBRFI | S-B-RF-I | SET BACK REFLECTANCE INDEX |
| PSBRFM | S-B-RF-M | SET BACK REFLECTANCE MODEL |
| PSBRFP | S-B-RF-P | SET BACK REFLECTANCE PROPERTIES |
| PSCMI | S-C-M-I | SET COLOUR MAPPING INDEX |
| PSCMR | S-C-M-R | SET COLOUR MAPPING REPRESENTATION |
| PSCAC | S-C-A-C | SET CURVE APPROXIMATION CRITERIA |
| PSDMI | S-D-M-I | SET DATA MAPPING INDEX |
| PSDMM | S-D-M-M | SET DATA MAPPING METHOD |
| PSDMR | S-D-M-R | SET DATA MAPPING REPRESENTATION |
| PSDCI | S-D-C-I | SET DEPTH CUE INDEX |
| PSDCR | S-D-C-R | SET DEPTH CUE REPRESENTATION |
| PSEDC | S-ED-C | SET EDGE COLOUR |
| PSEDP | S-ED-P | SET EDGE REPRESENTATION PLUS |
| PSFCM | S-F-C-M | SET FACET CULLING MODE |
| PSFDM | S-F-D-M | SET FACET DISTINGUISHING MODE |
| PSIC | S-I-C | SET INTERIOR COLOUR |
| PSIP | S-I-P | SET INTERIOR REPRESENTATION PLUS |
| PSISM | S-I-S-M | SET INTERIOR SHADING METHOD |
| PSLSR | S-L-S-R | SET LIGHT SOURCE REPRESENTATION |
| PSLSS | S-L-S-S | SET LIGHT SOURCE STATE |
| PSFAS3 | S-F-A-S-3 | SET OF FILL AREA SETS 3 WITH DATA |
| PSFASD | S-F-A-S-D | SET OF FILL AREA SETS WITH DATA |
| PSPSC | S-P-S-C | SET PARAMETRIC SURFACE CHARACTERISTICS |
| PSPSI | S-P-S-I | SET PARAMETRIC SURFACE INDEX |
| PSPSR | S-P-S-R | SET PARAMETRIC SURFACE REPRESENTATION |
| PSPAP | S-PA-P | SET PATTERN REPRESENTATION PLUS |
| PSPLC | S-PL-C | SET POLYLINE COLOUR |
| PSPLP | S-PL-P | SET POLYLINE REPRESENTATION PLUS |
| PSPLSM | S-PL-S-M | SET POLYLINE SHADING METHOD |
| PSPMC | S-PM-C | SET POLYMARKER COLOUR |
| PSPMP | S-PM-P | SET POLYMARKER REPRESENTATION PLUS |
| PSRFI | S-RF-I | SET REFLECTANCE INDEX |
| PSRFM | S-RF-M | SET REFLECTANCE MODEL |
| PSRFP | S-RF-P | SET REFLECTANCE PROPERTIES |
| PSRFR | S-RF-R | SET REFLECTANCE REPRESENTATION |
| PSRCM | S-R-C-M | SET RENDERING COLOUR MODEL |
| PSSAC | S-S-A-C | SET SURFACE APPROXIMATION CRITERIA |
| PSTXC | S-TX-C | SET TEXT COLOUR |
| PSTXRP | S-TX-R-P | SET TEXT REPRESENTATION PLUS |
| PTS3D | T-S-3-D | TRIANGLE SET 3 WITH DATA |
| PTSD | T-S-D | TRIANGLE SET WITH DATA |
| PTST3D | T-ST-3-D | TRIANGLE STRIP 3 WITH DATA |
| PTSTD | T-ST-D | TRIANGLE STRIP WITH DATA |
| PUCSC | U-C-S-C | UNPACK COLOUR SPLINE CURVE |
| PUCSS | U-C-S-S | UNPACK COLOUR SPLINE SURFACE |
| PUDSS | U-D-S-S | UNPACK DATA SPLINE SURFACE |
| PUTC | U-T-C | UNPACK TRIMMING CURVE |

# 13 The PHIGS PLUS function interface

## 13.1 General principles

For each PHIGS PLUS function the corresponding FORTRAN SUBROUTINE declaration is given. The name of the PHIGS PLUS function is listed, followed by its FORTRAN name and the corresponding parameters. After that, the list of parameters is described by type and a brief identifying phrase.

For the mapping of ENUMERATION types see clause 6 and clause 11.

## 13.2 Output primitive functions

### POLYLINE SET 3 WITH COLOUR

SUBROUTINE PPLS3C(VFLAG,CTYPE,NCC,NPL,IXA,PXA,PYA,PZA,VCOLI,VCOLR)

*Input Parameters:*

| | |
|---|---|
| INTEGER VFLAG | data per vertex data flag (PCD,PCDC) |
| INTEGER CTYPE | colour type |
| INTEGER NCC | number of components of colour value |
| INTEGER NPL | number of point lists |
| INTEGER IXA(NPL) | array of end indices for point lists |
| REAL PXA(IXA(NPL)),PYA(IXA(NPL)),PZA(IXA(NPL)) | coordinates of points(MC) |
| INTEGER VCOLI(IXA(NPL)) | vertex colour indices |
| REAL VCOLR(NCC*IXA(NPL)) | vertex colour values |

**FILL AREA SET 3 WITH DATA**

```
SUBROUTINE PFAS3D(FFLAG,EFLAG,VFLAG,CTYPE,NCC,FCOLI,FCOLR,FNXA,
*FNYA,FNZA,FDLEN,FDATA,NPL,IXA,EDATA,PXA,PYA,PZA,VCOLI,VCOLR,
*VNXA,VNYA,VNZA,VDLEN,VDATA)
```

*Input Parameters:*

| | |
|---|---|
| INTEGER FFLAG | data per facet data flag (PFNO,PFC,PFN,PFD,PFCN,PFCD,PFND,PFCND) |
| INTEGER EFLAG | data per edge data flag (PENO,PEVF) |
| INTEGER VFLAG | data per vertex data flag(PCD,PCDC,PCDN, PCDD,PCDCN,PCDCD,PCDND,PCDCND) |
| INTEGER CTYPE | colour type |
| INTEGER NCC | number of components of colour value |
| INTEGER FCOLI | facet colour index |
| REAL FCOLR(NCC) | facet colour values |
| REAL FNXA,FNYA,FNZA | facet normal data(MC) |
| INTEGER FDLEN | length of application-specific data list per facet |
| REAL FDATA(FDLEN) | facet application-specific data |
| INTEGER NPL | number of point lists |
| INTEGER IXA(NPL) | array of end indices for point lists |
| INTEGER EDATA(IXA(NPL)) | edge data(POFF,PON) |
| REAL PXA(IXA(NPL)),PYA(IXA(NPL)),PZA(IXA(NPL)) | coordinates of points(MC) |
| INTEGER VCOLI(IXA(NPL)) | vertex colour indices |
| REAL VCOLR(NCC*IXA(NPL)) | vertex colour values |
| REAL VNXA(IXA(NPL)),VNYA(IXA(NPL)),VNZA(IXA(NPL)) | vertex normal data(MC) |
| INTEGER VDLEN | length of application-specific data list per vertex |
| REAL VDATA(VDLEN*IXA(NPL)) | vertex application-specific data |

**FILL AREA SET WITH DATA**

```
SUBROUTINE PFASD(FFLAG,EFLAG,VFLAG,CTYPE,NCC,FCOLI,FCOLR,FNXA,
*FNYA,FNZA,FDLEN,FDATA,NPL,IXA,EDATA,PXA,PYA,VCOLI,VCOLR,VNXA,
*VNYA,VNZA,VDLEN,VDATA)
```

*Input Parameters:*

| | |
|---|---|
| INTEGER FFLAG | data per facet data flag (PFNO,PFC,PFN,PFD,PFCN,PFCD,PFND,PFCND) |
| INTEGER EFLAG | data per edge data flag (PENO,PEVF) |
| INTEGER VFLAG | data per vertex data flag(PCD,PCDC,PCDN, PCDD,PCDCN,PCDCD,PCDND,PCDCND) |
| INTEGER CTYPE | colour type |
| INTEGER NCC | number of components of colour value |
| INTEGER FCOLI | facet colour index |
| REAL FCOLR(NCC) | facet colour values |
| REAL FNXA,FNYA,FNZA | facet normal data(MC) |
| INTEGER FDLEN | length of application-specific data list per facet |
| REAL FDATA(FDLEN) | facet application-specific data |
| INTEGER NPL | number of points lists |
| INTEGER IXA(NPL) | array of end indices for point lists |
| INTEGER EDATA(IXA(NPL)) | edge data(POFF,PON) |
| REAL PXA(IXA(NPL)),PYA(IXA(NPL)) | coordinates of points(MC) |
| INTEGER VCOLI(IXA(NPL)) | vertex colour indices |
| REAL VCOLR(NCC*IXA(NPL)) | vertex colour values |
| REAL VNXA(IXA(NPL)),VNYA(IXA(NPL)),VNZA(IXA(NPL)) | vertex normal data(MC) |
| INTEGER VDLEN | length of application-specific data list per vertex |
| REAL VDATA(VDLEN*IXA(NPL)) | vertex application-specific data |

**CELL ARRAY 3 PLUS**

```
SUBROUTINE PCA3P(CPXA,CPYA,CPZA,DIMX,DIMY,ISC,ISR,DX,DY,CTYPE,
*NCC,COLIA,COLRA)
```

*Input Parameters:*

| | |
|---|---|
| REAL CPXA(3),CPYA(3),CPZA(3) | cell parallelogram(P,Q,R)(MC) |
| INTEGER DIMX,DIMY | the number of columns and number of rows in COLIA or COLRA |
| INTEGER ISC,ISR | indices of start column, start row |
| INTEGER DX,DY | number of columns,number of rows |
| INTEGER CTYPE | colour type |
| INTEGER NCC | number of components of colour value |
| INTEGER COLIA(DIMX,DIMY) | colour index array |
| REAL COLRA(NCC*DIMX*DIMY) | colour value array |

## SET OF FILL AREA SETS 3 WITH DATA

```
 SUBROUTINE PSFAS3(FFLAG,EFLAG,VFLAG,CTYPE,NCC,NFS,FCOLI,FCOLR,
*FNXA,FNYA,FNZA,FDLEN,FDATA,NF,NP,IXA,EDATA,PXA,PYA,PZA,VCOLI,
*VCOLR,VNXA,VNYA,VNZA,VDLEN,VDATA,VIND)
```

*Input Parameters:*

| | |
|---|---|
| INTEGER FFLAG | data per facet data flag (PFNO,PFC,PFN,PFD,PFCN,PFCD,PFND,PFCND) |
| INTEGER EFLAG | data per edge data flag (PENO,PEVF) |
| INTEGER VFLAG | data per vertex data flag(PCD,PCDC,PCDN, PCDD,PCDCN,PCDCD,PCDND,PCDCND) |
| INTEGER CTYPE | colour type |
| INTEGER NCC | number of components of colour value |
| INTEGER NFS | number of fill area sets |
| INTEGER FCOLI(NFS) | facet colour indices |
| REAL FCOLR(NCC*NFS) | facet colour values |
| REAL FNXA(NFS),FNYA(NFS),FNZA(NFS) | facet normal data(MC) |
| INTEGER FDLEN | length of application-specific data list per facet |
| REAL FDATA(FDLEN*NFS) | facet application-specific data |
| INTEGER NF(NFS) | array of end vertex indices in IXA for each fill area set |
| INTEGER NP | number of points |
| INTEGER IXA(NF(NFS)) | array of end vertex indices in VIND for each fill area |
| INTEGER EDATA(IXA(NF(NFS))) | edge data(POFF,PON) |
| REAL PXA(NP),PYA(NP),PZA(NP) | coordinates of points(MC) |
| INTEGER VCOLI(NP) | vertex colour indices |
| REAL VCOLR(NCC*NP) | vertex colour values |
| REAL VNXA(NP),VNYA(NP),VNZA(NP) | vertex normal data(MC) |
| INTEGER VDLEN | length of application-specific data list per vertex |
| REAL VDATA(VDLEN*NP) | vertex application-specific data |
| INTEGER VIND(IXA(NF(NFS))) | vertex indices for each fill area |

## SET OF FILL AREA SETS WITH DATA

```
 SUBROUTINE PSFASD(FFLAG,EFLAG,VFLAG,CTYPE,NCC,NFS,FCOLI,FCOLR,
*FNXA,FNYA,FNZA,FDLEN,FDATA,NF,NP,IXA,EDATA,PXA,PYA,VCOLI,VCOLR,
*VNXA,VNYA,VNZA,VDLEN,VDATA,VIND)
```

*Input Parameters:*

| | |
|---|---|
| INTEGER FFLAG | data per facet data flag (PFNO,PFC,PFN,PFD,PFCN,PFCD,PFND,PFCND) |
| INTEGER EFLAG | data per edge data flag (PENO,PEVF) |
| INTEGER VFLAG | data per vertex data flag(PCD,PCDC,PCDN, PCDD,PCDCN,PCDCD,PCDND,PCDCND) |
| INTEGER CTYPE | colour type |
| INTEGER NCC | number of components of colour value |
| INTEGER NFS | number of fill area sets |
| INTEGER FCOLI(NFS) | facet colour indices |
| REAL FCOLR(NCC*NFS) | facet colour values |
| REAL FNXA(NFS),FNYA(NFS),FNZA(NFS) | facet normal data(MC) |
| INTEGER FDLEN | length of application-specific data list per facet |
| REAL FDATA(FDLEN*NFS) | facet application-specific data |
| INTEGER NF(NFS) | array of end indices in IXA for each fill area set |
| INTEGER NP | number of points |
| INTEGER IXA(NF(NFS)) | array of end vertex indices in VIND for each fill area |
| INTEGER EDATA(IXA(NF(NFS)) | edge data(POFF,PON) |
| REAL PXA(NP),PYA(NP) | coordinates of points(MC) |
| INTEGER VCOLI(NP) | vertex colour indices |
| REAL VCOLR(NCC*NP) | vertex colour values |
| REAL VNXA(NP),VNYA(NP),VNZA(NP) | vertex normal data(MC) |
| INTEGER VDLEN | length of application-specific data list per vertex |
| REAL VDATA(VDLEN*NP) | vertex application-specific data |
| INTEGER VIND(IXA(NF(NFS))) | vertex indices for each fill area |

**TRIANGLE SET 3 WITH DATA**

```
 SUBROUTINE PTS3D(FFLAG,EFLAG,VFLAG,CTYPE,NCC,NTRI,FCOLI,FCOLR,
*FNXA,FNYA,FNZA,FDLEN,FDATA,NP,EDATA,PXA,PYA,PZA,VCOLI,VCOLR,
*VNXA,VNYA,VNZA,VDLEN,VDATA,VIND)
```

*Input Parameters:*

| | |
|---|---|
| INTEGER FFLAG | data per facet data flag (PFNO,PFC,PFN,PFD,PFCN,PFCD,PFND,PFCND) |
| INTEGER EFLAG | data per edge data flag (PENO,PEVF) |
| INTEGER VFLAG | data per vertex data flag(PCD,PCDC,PCDN, PCDD,PCDCN,PCDCD,PCDND,PCDCND) |
| INTEGER CTYPE | colour type |
| INTEGER NCC | number of components of colour value |
| INTEGER NTRI | number of triangles |
| INTEGER FCOLI(NTRI) | facet colour indices |
| REAL FCOLR(NCC*NTRI) | facet colour values |
| REAL FNXA(NTRI),FNYA(NTRI),FNZA(NTRI) | facet normal data(MC) |
| INTEGER FDLEN | length of application-specific data list per facet |
| REAL FDATA(FDLEN*NTRI) | facet application-specific data |
| INTEGER NP | number of points |
| INTEGER EDATA(3*NTRI) | edge data(POFF,PON) |
| REAL PXA(NP),PYA(NP),PZA(NP) | coordinates of points(MC) |
| INTEGER VCOLI(NP) | vertex colour indices |
| REAL VCOLR(NCC*NP) | vertex colour values |
| REAL VNXA(NP),VNYA(NP),VNZA(NP) | vertex normal data(MC) |
| INTEGER VDLEN | length of application-specific data list per vertex |
| REAL VDATA(VDLEN*NP) | vertex application-specific data |
| INTEGER VIND(NTRI*3) | vertex indices |

**TRIANGLE SET WITH DATA**

```
 SUBROUTINE PTSD(FFLAG,EFLAG,VFLAG,CTYPE,NCC,NTRI,FCOLI,FCOLR,
*FNXA,FNYA,FNZA,FDLEN,FDATA,NP,EDATA,PXA,PYA,VCOLI,VCOLR,VNXA,
*VNYA,VNZA,VDLEN,VDATA,VIND)
```

*Input Parameters:*

| | |
|---|---|
| INTEGER FFLAG | data per facet data flag (PFNO,PFC,PFN,PFD,PFCN,PFCD,PFND,PFCND) |
| INTEGER EFLAG | data per edge data flag (PENO,PEVF) |
| INTEGER VFLAG | data per vertex data flag(PCD,PCDC,PCDN, PCDD,PCDCN,PCDCD,PCDND,PCDCND) |
| INTEGER CTYPE | colour type |
| INTEGER NCC | number of components of colour value |
| INTEGER NTRI | number of triangles |
| INTEGER FCOLI(NTRI) | facet colour indices |
| REAL FCOLR(NCC*NTRI) | facet colour values |
| REAL FNXA(NTRI),FNYA(NTRI),FNZA(NTRI) | facet normal data(MC) |
| INTEGER FDLEN | length of application-specific data list per facet |
| REAL FDATA(FDLEN*NTRI) | facet application-specific data |
| INTEGER NP | number of points |
| INTEGER EDATA(3*NTRI) | edge data(POFF,PON) |
| REAL PXA(NP),PYA(NP) | coordinates of points(MC) |
| INTEGER VCOLI(NP) | vertex colour indices |
| REAL VCOLR(NCC*NP) | vertex colour values |
| REAL VNXA(NP),VNYA(NP),VNZA(NP) | vertex normal data(MC) |
| INTEGER VDLEN | length of application-specific data list per vertex |
| REAL VDATA(VDLEN*NP) | vertex application-specific data |
| INTEGER VIND(NTRI*3) | vertex indices |

**TRIANGLE STRIP 3 WITH DATA**

```
 SUBROUTINE PTST3D(FFLAG,EFLAG,VFLAG,CTYPE,NCC,NP,FCOLI,FCOLR,
*FNXA,FNYA,FNZA,FDLEN,FDATA,EDATA,PXA,PYA,PZA,VCOLI,VCOLR,VNXA,
*VNYA,VNZA,VDLEN,VDATA)
```

*Input Parameters:*

| | |
|---|---|
| INTEGER FFLAG | data per facet data flag (PFNO,PFC,PFN,PFD,PFCN,PFCD,PFND,PFCND) |
| INTEGER EFLAG | data per edge data flag (PENO,PEVF) |
| INTEGER VFLAG | data per vertex data flag(PCD,PCDC,PCDN, PCDD,PCDCN,PCDCD,PCDND,PCDCND) |
| INTEGER CTYPE | colour type |
| INTEGER NCC | number of components of colour value |
| INTEGER NP | number of points |
| INTEGER FCOLI(NP-2) | facet colour indices |
| REAL FCOLR(NCC*(NP-2)) | facet colour values |
| REAL FNXA(NP-2),FNYA(NP-2),FNZA(NP-2) | facet normal data(MC) |
| INTEGER FDLEN | length of application-specific data list per facet |
| REAL FDATA(FDLEN*(NP-2)) | facet application-specific data |
| INTEGER EDATA(2*NP-3) | edge data(POFF,PON) |
| REAL PXA(NP),PYA(NP),PZA(NP) | coordinates of points(MC) |
| INTEGER VCOLI(NP) | vertex colour indices |
| REAL VCOLR(NCC*NP) | vertex colour values |
| REAL VNXA(NP),VNYA(NP),VNZA(NP) | vertex normal data(MC) |
| INTEGER VDLEN | length of application-specific data list per vertex |
| REAL VDATA(VDLEN*NP) | vertex application-specific data |

**TRIANGLE STRIP WITH DATA**

```
 SUBROUTINE PTSTD(FFLAG,EFLAG,VFLAG,CTYPE,NCC,NP,FCOLI,FCOLR,
*FNXA,FNYA,FNZA,FDLEN,FDATA,EDATA,PXA,PYA,VCOLI,VCOLR,VNXA,VNYA,
*VNZA,VDLEN,VDATA)
```

*Input Parameters:*

| | |
|---|---|
| INTEGER FFLAG | data per facet data flag (PFNO,PFC,PFN,PFD,PFCN,PFCD,PFND,PFCND) |
| INTEGER EFLAG | data per edge data flag (PENO,PEVF) |
| INTEGER VFLAG | data per vertex data flag(PCD,PCDC,PCDN, PCDD,PCDCN,PCDCD,PCDND,PCDCND) |
| INTEGER CTYPE | colour type |
| INTEGER NCC | number of components of colour value |
| INTEGER NP | number of points |
| INTEGER FCOLI(NP-2) | facet colour indices |
| REAL FCOLR(NCC*(NP-2)) | facet colour values |
| REAL FNXA(NP-2),FNYA(NP-2),FNZA(NP-2) | facet normal data(MC) |
| INTEGER FDLEN | length of application-specific data list per facet |
| REAL FDATA(FDLEN*(NP-2)) | facet application-specific data |
| INTEGER EDATA(2*NP-3) | edge data(POFF,PON) |
| REAL PXA(NP),PYA(NP) | coordinates of points(MC) |
| INTEGER VCOLI(NP) | vertex colour indices |
| REAL VCOLR(NCC*NP) | vertex colour values |
| REAL VNXA(NP),VNYA(NP),VNZA(NP) | vertex normal data(MC) |
| INTEGER VDLEN | length of application-specific data list per vertex |
| REAL VDATA(VDLEN*NP) | vertex application-specific data |

**QUADRILATERAL MESH 3 WITH DATA**

```
 SUBROUTINE PQM3D(FFLAG,EFLAG,VFLAG,CTYPE,NCC,NC,NR,FCOLI,FCOLR,
*FNXA,FNYA,FNZA,FDLEN,FDATA,EDATA,PXA,PYA,PZA,VCOLI,VCOLR,VNXA,
*VNYA,VNZA,VDLEN,VDATA)
```

*Input Parameters:*

| | |
|---|---|
| INTEGER FFLAG | data per facet data flag (PFNO,PFC,PFN,PFD,PFCN,PFCD,PFND,PFCND) |
| INTEGER EFLAG | data per edge data flag (PENO,PEVF) |
| INTEGER VFLAG | data per vertex data flag(PCD,PCDC,PCDN, PCDD,PCDCN,PCDCD,PCDND,PCDCND) |
| INTEGER CTYPE | colour type |
| INTEGER NCC | number of components of colour value |
| INTEGER NC,NR | number of columns,number of rows |
| INTEGER FCOLI((NC-1)*(NR-1)) | facet colour indices |
| REAL FCOLR(NCC*(NC-1)*(NR-1)) | facet colour values |
| REAL FNXA((NC-1)*(NR-1)),FNYA((NC-1)*(NR-1)),FNZA((NC-1)*(NR-1)) | facet normal data(MC) |
| INTEGER FDLEN | length of application-specific data list per facet |
| REAL FDATA(FDLEN*(NC-1)*(NR-1)) | facet application-specific data |
| INTEGER EDATA(2*NC*NR) | edge data(POFF,PON) |
| REAL PXA(NC*NR),PYA(NC*NR),PZA(NC*NR) | coordinates of points(MC) |
| INTEGER VCOLI(NC*NR) | vertex colour indices |
| REAL VCOLR(NCC*NC*NR) | vertex colour values |
| REAL VNXA(NC*NR),VNYA(NC*NR),VNZA(NC*NR) | vertex normal data(MC) |
| INTEGER VDLEN | length of application-specific data list per vertex |
| REAL VDATA(VDLEN*NC*NR) | vertex application-specific data |

**QUADRILATERAL MESH WITH DATA**

```
 SUBROUTINE PQMD(FFLAG,EFLAG,VFLAG,CTYPE,NCC,NC,NR,FCOLI,FCOLR,
*FNXA,FNYA,FNZA,FDLEN,FDATA,EDATA,PXA,PYA,VCOLI,VCOLR,VNXA,
*VNYA,VNZA,VDLEN,VDATA)
```

*Input Parameters:*

| | |
|---|---|
| INTEGER FFLAG | data per facet data flag (PFNO,PFC,PFN,PFD,PFCN,PFCD,PFND,PFCND) |
| INTEGER EFLAG | data per edge data flag (PENO,PEVF) |
| INTEGER VFLAG | data per vertex data flag(PCD,PCDC,PCDN, PCDD,PCDCN,PCDCD,PCDND,PCDCND) |
| INTEGER CTYPE | colour type |
| INTEGER NCC | number of components of colour value |
| INTEGER NC,NR | number of columns,number of rows |
| INTEGER FCOLI((NC-1)*(NR-1)) | facet colour indices |
| REAL FCOLR(NCC*(NC-1)*(NR-1)) | facet colour values |
| REAL FNXA((NC-1)*(NR-1)),FNYA((NC-1)*(NR-1)),FNZA((NC-1)*(NR-1)) | facet normal data(MC) |
| INTEGER FDLEN | length of application-specific data list per facet |
| REAL FDATA(FDLEN*(NC-1)*(NR-1)) | facet application-specific data |
| INTEGER EDATA(2*NC*NR) | edge data(POFF,PON) |
| REAL PXA(NC*NR),PYA(NC*NR) | coordinates of points(MC) |
| INTEGER VCOLI(NC*NR) | vertex colour indices |
| REAL VCOLR(NCC*NC*NR) | vertex colour values |
| REAL VNXA(NC*NR),VNYA(NC*NR),VNZA(NC*NR) | vertex normal data(MC) |
| INTEGER VDLEN | length of application-specific data list per vertex |
| REAL VDATA(VDLEN*NC*NR) | vertex application-specific data |

**NON-UNIFORM B-SPLINE CURVE 3**

SUBROUTINE PBSC3(SORD,NKA,KNOTS,PARL,RTYPE,NCP,PXA,PYA,PZA,PWA)

*Input Parameters:*

| | |
|---|---|
| INTEGER SORD | spline order |
| INTEGER NKA | number of knots |
| REAL KNOTS(NKA) | knots |
| REAL PARL(2) | parameter range limits |
| INTEGER RTYPE | rationality (PRAT,PNRAT) |
| INTEGER NCP | number of control points |
| REAL PXA(NCP),PYA(NCP),PZA(NCP),PWA(NCP) | control points(MC) |

**NON-UNIFORM B-SPLINE CURVE 3 WITH COLOUR**

SUBROUTINE PBSC3C(SORD,NKA,KNOTS,PARL,RTYPE,NCP,PXA,PYA,PZA,PWA
*CSCLDR,CSCREC)

*Input Parameters:*

| | |
|---|---|
| INTEGER SORD | spline order |
| INTEGER NKA | number of knots |
| REAL KNOTS(NKA) | knots |
| REAL PARL(2) | parameter range limits |
| INTEGER RTYPE | rationality (PRAT,PNRAT) |
| INTEGER NCP | number of control points |
| REAL PXA(NCP),PYA(NCP),PZA(NCP),PWA(NCP) | control points(MC) |
| INTEGER CSCLDR | number of array elements used in CSCREC |
| CHARACTER*80 CSCREC(CSCLDR) | colour spline curve data record[1] |

**NON-UNIFORM B-SPLINE SURFACE 3**

SUBROUTINE PBSS3(USORD,VSORD,UNKA,VNKA,UKNOTS,VKNOTS,RTYPE,
*UNCP,VNCP,PXA,PYA,PZA,PWA,TCLDR,TCREC)

*Input Parameters:*

| | |
|---|---|
| INTEGER USORD | u spline order |
| INTEGER VSORD | v spline order |
| INTEGER UNKA | number of knots for u |
| INTEGER VNKA | number of knots for v |
| REAL UKNOTS(UNKA) | u knots |
| REAL VKNOTS(VNKA) | v knots |
| INTEGER RTYPE | rationality (PRAT,PNRAT) |
| INTEGER UNCP | u number of control points dimension |
| INTEGER VNCP | v number of control points dimension |
| REAL PXA(UNCP*VNCP),PYA(UNCP*VNCP),PZA(UNCP*VNCP),PWA(UNCP*VNCP) | control points(MC) |
| INTEGER TCLDR | number of array elements used in TCREC |

[1] See PACK COLOUR SPLINE CURVE and UNPACK COLOUR SPLINE CURVE for a description of the colour spline curve data record.

CHARACTER*80 TCREC(TCLDR) trimming curve data record[2]

## NON-UNIFORM B-SPLINE SURFACE 3 WITH DATA

SUBROUTINE PBSS3D(USORD,VSORD,UNKA,VNKA,UKNOTS,VKNOTS,RTYPE,
*UNCP,VNCP,PXA,PYA,PZA,PWA,TCLDR,TCREC,CSSLDR,CSSREC,DSSLDR,
*DSSREC)

*Input Parameters:*

| | |
|---|---|
| INTEGER USORD | u spline order |
| INTEGER VSORD | v spline order |
| INTEGER UNKA | number of knots for u |
| INTEGER VNKA | number of knots for v |
| REAL UKNOTS(UNKA) | u knots |
| REAL VKNOTS(VNKA) | v knots |
| INTEGER RTYPE | rationality (PRAT,PNRAT) |
| INTEGER UNCP | u number of control points dimension |
| INTEGER VNCP | v number of control points dimension |
| REAL PXA(UNCP*VNCP),PYA(UNCP*VNCP),PZA(UNCP*VNCP),PWA(UNCP*VNCP) | control points(MC) |
| INTEGER TCLDR | number of array elements used in TCREC |
| CHARACTER*80 TCREC(TCLDR) | trimming curve data record[3] |
| INTEGER CSSLDR | number of array elements used in CSSREC |
| CHARACTER*80 CSSREC(CSSLDR) | colour spline surface data record[4] |
| INTEGER DSSLDR | number of array elements used in DSSREC |
| CHARACTER*80 DSSREC(DSSLDR) | data spline surface data record[5] |



[2] See PACK TRIMMING CURVE and UNPACK TRIMMING CURVE for a description of the trimming curve data record.

[3] See PACK TRIMMING CURVE and UNPACK TRIMMING CURVE for a description of the trimming curve data record.

[4] See PACK COLOUR SPLINE SURFACE and UNPACK COLOUR SPLINE SURFACE for a description of the colour spline surface data record.

[5] See PACK DATA SPLINE SURFACE and UNPACK DATA SPLINE SURFACE for a description of the data spline surface data record.

## 13.3 Attribute specification functions

### 13.3.1. Bundled attribute selection

**SET DATA MAPPING INDEX**

SUBROUTINE PSDMI(DMI)

*Input Parameters:*
INTEGER DMI — data mapping index

**SET REFLECTANCE INDEX**

SUBROUTINE PSRFI(RFI)

*Input Parameters:*
INTEGER RFI — reflectance index

**SET BACK INTERIOR INDEX**

SUBROUTINE PSBII(BII)

*Input Parameters:*
INTEGER BII — back interior index

**SET BACK DATA MAPPING INDEX**

SUBROUTINE PSBDMI(BDMI)

*Input Parameters:*
INTEGER BDMI — back data mapping index

**SET BACK REFLECTANCE INDEX**

SUBROUTINE PSBRFI(BRFI)

*Input Parameters:*
INTEGER BRFI — back reflectance index

**SET PARAMETRIC SURFACE INDEX**

SUBROUTINE PSPSI(PSI)

*Input Parameters:*
INTEGER PSI — parametric surface index

### 13.3.2 Individual attribute selection

**SET POLYLINE COLOUR**

SUBROUTINE PSPLC(CTYPE,COLI,NCC,COLR)

*Input Parameters:*

| | |
|---|---|
| INTEGER CTYPE | polyline colour type |
| INTEGER COLI | colour index |
| INTEGER NCC | number of components of colour value |
| REAL COLR(NCC) | colour values |

**SET POLYLINE SHADING METHOD**

SUBROUTINE PSPLSM(PLSM)

*Input Parameters:*

| | |
|---|---|
| INTEGER PLSM | polyline shading method |

**SET POLYMARKER COLOUR**

SUBROUTINE PSPMC(CTYPE,COLI,NCC,COLR)

*Input Parameters:*

| | |
|---|---|
| INTEGER CTYPE | polymarker colour type |
| INTEGER COLI | colour index |
| INTEGER NCC | number of components of colour value |
| REAL COLR(NCC) | colour values |

**SET TEXT COLOUR**

SUBROUTINE PSTXC(CTYPE,COLI,NCC,COLR)

*Input Parameters:*

| | |
|---|---|
| INTEGER CTYPE | text colour type |
| INTEGER COLI | colour index |
| INTEGER NCC | number of components of colour value |
| REAL COLR(NCC) | colour values |

**SET FACET DISTINGUISHING MODE**

SUBROUTINE PSFDM(DMODE)

*Input Parameters:*

| | |
|---|---|
| INTEGER DMODE | distinguishing mode (POFF,PON) |

**SET FACET CULLING MODE**

SUBROUTINE PSFCM(CMODE)

*Input Parameters:*

| | |
|---|---|
| INTEGER CMODE | culling mode (PNOFC,PBKFC,PFTFC) |

**SET INTERIOR COLOUR**

SUBROUTINE PSIC(CTYPE,COLI,NCC,COLR)

*Input Parameters:*

| | |
|---|---|
| INTEGER CTYPE | interior colour type |
| INTEGER COLI | colour index |
| INTEGER NCC | number of components of colour value |
| REAL COLR(NCC) | colour values |

**SET INTERIOR SHADING METHOD**

SUBROUTINE PSISM(ISM)

*Input Parameters:*

| | |
|---|---|
| INTEGER ISM | interior shading method |

**SET DATA MAPPING METHOD**

SUBROUTINE PSDMM(DMM,LDR,DATREC)

*Input Parameters:*

| | |
|---|---|
| INTEGER DMM | data mapping method |
| INTEGER LDR | length of data record array |
| CHARACTER*80 DATREC(LDR) | data record |

DATREC parameters to build DATA MAPPING METHOD DATA RECORD for DMM = PCDM:
The IA array contains:
- The number of source selectors
- The actual list of source selectors

The other arrays are empty.

Common element:
IL= 1+*
IA(1)= number of source selectors = NSS
IA= element 2 through NSS+1 contain source selectors
RL= 0
RA=( )
SL= 0
LSTR= ( )
STR= ( )

DATREC parameters to build DATA MAPPING METHOD DATA RECORD for DMM = PSUDM:
The IA array contains:
- The number source selectors
- The actual list of source selectors
- The index of the data value
- The colour type(CT)
- The number of colour values(NCV)
- The list of colour indices.There are NCV colour indices, if CT is indirect.
- The number of components of colour value(NCC), if CT is not indirect.

The RA array contains:

- The lower range limit
- The upper range limit
- The list of colour values. There are NCV colour values, each having as many components as required by CT.

Common element:
IL= 5+*
IA(1)= number ofsource selectors = NSS
IA= element 2 through NSS+1 contain source selectors
IA(NSS+2)= data value index
IA(NSS+3)= colour type = CT
IA(NSS+4)= number of specified colour = NCV
RL= 2+*
RA(1)= lower range limit
RA(2)= upper range limit
SL= 0
LSTR= ( )
STR= ( )
Individual element:
CASE colour type=PINDDIR:
IA=element NSS+5 through NSS+4+NCV contain the colour indices
CASE colour type=PRGB or PCIE or PHSV or PHLS:
IA(NSS+5)= number of components of colour value = NCC
RA=element 3 through NCC*NCV+2 contain the colour values
example r1,g1,b1,r2,g2,b2,...

DATREC parameters to build DATA MAPPING METHOD DATA RECORD for
DMM = PSNUDM:
The IA array contains:
- The number of source selectors
- The actual list of source selectors
- The index of the data value
- The colour type(CT)
- The number of range boundaries(NRB)
- The list of colour indices.There are NRB+1 colour indices, if CT is indirect.
- The number of components of colour value(NCC), if CT is not indirect.

The RA array contains:
- The actual data of range boundaries
- The list of colour values. There are NRB+1 colour values, each having as many components as required by CT.

Common element:
IL= 5+NSS+*
IA(1)= number of source selectors = NSS
IA= element 2 through NSS+1 contain source selectors
IA(NSS+2)= data value index
IA(NSS+3)= colour type = CT
IA(NSS+4)= number of range boundaries = number of specified colour -1 = NRB
RL= NRB+*
RA=element 1 through NRB contain the range boundary
SL= 0
LSTR= ( )
STR= ( )
Individual element:

CASE colour type=PINDIR:
IA=element NSS+5 through NSS+4+NRB+1 contain the colour indices
CASE colour type=PRGB or PCIE or PHSV or PHLS:
IA(NSS+5)= number of components of colour value = NCC
RA=element NRB+1 through NCC*(NRB+1)+NRB contain the colour values
example r1,g1,b1,r2,g2,b2,...

DATREC parameters to build DATA MAPPING METHOD DATA RECORD for DMM = PBUDM:
The IA array contains:
- The number of source selectors
- The actual list of source selectors
- The two indices of the data value
- The colour type(CT)
- The number of colour lists(NCL)
- The array of end indices of colour lists
- The list of colour indices.There are a lot of colour indices, if CT is indirect.
- The number of components of colour value(NCC), if CT is not indirect.

The RA array contains:
- The actual data of lower and upper limit of Ra range boundary
- The actual data of lower and upper limit of Rb range boundary
- The list of colour values. There are a lot of colour values, each having as many components as required by CT.

Common element:
IL= 6+NSS+NCL
IA(1)= number of source selectors = NSS
IA= element 2 through NSS+1 contain source selectors
IA(NSS+2)= data value index 1
IA(NSS+3)= data value index 2
IA(NSS+4)= colour type = CT
IA(NSS+5)= number of colour lists = NCL
IA= element NSS+6 through NSS+5+NCL contain array of end indices of colour lists
RL= 4+*
RA(1)= lower limit of Ra range
RA(2)= upper limit of Ra range
RA(3)= lower limit of Rb range
RA(4)= upper limit of Rb range
SL= 0
LSTR= ( )
STR= ( )
Individual element:
CASE colour type=PINDIR:
IA=element NSS+5+NCL+1 through NSS+5+NCL+IA(NSS+5+NCL) contain the colour indices
CASE colour type=PRGB or PCIE or PHSV or PHLS:
IA(NSS+5+NCL+1)= number of components of colour value = NCC
RA=element 5 through NCC*IA(NSS+5+NCL)+4 contain the colour values
example r1,g1,b1,r2,g2,b2,...

DATREC parameters to build DATA MAPPING METHOD DATA RECORD for DMM = PBNUDM:
The IA array contains:
- The number of source selectors
- The actual list of source selectors

- The two indices of the data value
- The colour type(CT)
- The number of Ra range boundaries(NRAB)
- The array of end indices of Rb range boundaries data
- The list of colour indices.There is a lot of colour indices
  (number of Rb range boundaries data + number of Ra range boundaries),
  if CT is indirect.
- The number of components of colour value(NCC), if CT is not indirect.

The RA array contains:
- The actual data of Ra range boundaries
- The actual data of Rb range boundaries
- The list of colour values. There is a lot of colour values
  (number of Rb range boundaries data + number of Ra range boundaries),
  each having as many components as required by CT.

Common element:
IL= 6+*
IA(1)= number of source selectors = NSS
IA= element 2 through NSS+1 contain source selectors
IA(NSS+2)= data value index 1
IA(NSS+3)= data value index 2
IA(NSS+4)= colour type = CT
IA(NSS+5)= number of Ra range boundaries = NRAB
IA=element NSS+6 through NSS+5+NRAB array of end indices of Rb range boundaries
RL= NRAB+IA(NSS+5+NRAB)+*
RA=element 1 through NRAB contain Ra range boundary
RA=element NRAB+1 through NRAB+IA(NSS+5+NRAB) contain Rb range boundary
SL= 0
LSTR= ( )
STR= ( )
Individual element:
  CASE colour type=PINDIR:
    IA=element NSS+5+NRAB+1 through NSS+4+NRAB+1+IA(NSS+5+NRAB)
          contain colour indices
  CASE colour type=PRGB or PCIE or PHSV or PHLS:
    IA(NSS+5+NRAB+1)= number of components of colour value = NCC
    RA=element NRAB+IA(NSS+5+NRAB)+1 through
       NRAB+IA(NSS+5+NRAB)+NCC*(IA(NSS+5+NRAB))
       contain the colour values
    example r1,g1,b1,r2,g2,b2,...

**SET REFLECTANCE PROPERTIES**

SUBROUTINE PSRFP(RPTYPE,LDR,DATREC)

*Input Parameters:*

| | |
|---|---|
| INTEGER RPTYPE | reflectance properties type |
| INTEGER LDR | length of data record array |
| CHARACTER*80 DATREC(LDR) | data record |

DATREC parameters to build REFLECTANCE PROPERTIES DATA RECORD for
RPTYPE = PSRPT:
The IA array contains:
- The colour type(CT)
- The number of components of colour value(NCC)

- The colour index of the specular colour if CT is indirect.

The RA array contains:
- The actual data of ambient reflection coefficient
- The actual data of diffuse reflection coefficient
- The actual data of specular reflection coefficient
- The actual data of specular exponent
- The actual data of colour values as required by CT.

Common element:
IL= 2+*
IA(1)= colour type = CT
IA(2)= number of components of colour value = NCC
RL= 4+*
RA(1)= ambient reflection coefficient
RA(2)= diffuse reflection coefficient
RA(3)= specular reflection coefficient
RA(4)= specular exponent
SL= 0
LSTR= ( )
STR= ( )

Individual element:
CASE colour type=PINDIR:
IA(3)= specular colour index
CASE colour type=PRGB or PCIE or PHSV or PHLS:
RA=element 5 through 4+NCC contain the colour values

**SET REFLECTANCE MODEL**

SUBROUTINE PSRFM(RFM)

*Input Parameters:*

| | |
|---|---|
| INTEGER RFM | reflectance model |

**SET BACK INTERIOR STYLE**

SUBROUTINE PSBIS(INTS)

*Input Parameters:*

| | |
|---|---|
| INTEGER INTS | interior style (PHOLLO,PSOLID,PPATTR,PPATCH,PISEMP) |

**SET BACK INTERIOR STYLE INDEX**

SUBROUTINE PSBISI(ISTYLI)

*Input Parameters:*

| | |
|---|---|
| INTEGER ISTYLI | interior style index |

**SET BACK INTERIOR COLOUR**

SUBROUTINE PSBIC(CTYPE,COLI,NCC,COLR)

*Input Parameters:*

| | |
|---|---|
| INTEGER CTYPE | back interior colour type |
| INTEGER COLI | colour index |
| INTEGER NCC | number of components of colour value |
| REAL COLR(NCC) | colour values |

**SET BACK INTERIOR SHADING METHOD**

SUBROUTINE PSBISM(ISM)

*Input Parameters:*

| | |
|---|---|
| INTEGER ISM | interior shading method |

**SET BACK DATA MAPPING METHOD**

SUBROUTINE PSBDMM(DMM,LDR,DATREC)

*Input Parameters:*

| | |
|---|---|
| INTEGER DMM | data mapping method |
| INTEGER LDR | length of data record array |
| CHARACTER*80 DATREC(LDR) | data record[6] |

**SET BACK REFLECTANCE PROPERTIES**

SUBROUTINE PSBRFP(RPTYPE,LDR,DATREC)

*Input Parameters:*

| | |
|---|---|
| INTEGER RPTYPE | reflectance properties type |
| INTEGER LDR | length of data record array |
| CHARACTER*80 DATREC(LDR) | data record[7] |

**SET BACK REFLECTANCE MODEL**

SUBROUTINE PSBRFM(RFM)

*Input Parameters:*

| | |
|---|---|
| INTEGER RFM | reflectance model |



[6] See SET DATA MAPPING METHOD for a description of the data mapping method data records.

[7] See SET REFLECTANCE PROPERTIES for a description of the reflectance properties data records.

**SET LIGHT SOURCE STATE**

SUBROUTINE PSLSS(NACTI,ACTLST,NDEAI,DEALST)

*Input Parameters:*

| | |
|---|---|
| INTEGER NACTI | number of activation indices |
| INTEGER ACTLST(NACTI) | activation list |
| INTEGER NDEAI | number of deactivation indices |
| INTEGER DEALST(NDEAI) | deactivation list |

**SET EDGE COLOUR**

SUBROUTINE PSEDC(CTYPE,COLI,NCC,COLR)

*Input Parameters:*

| | |
|---|---|
| INTEGER CTYPE | edge colour type |
| INTEGER COLI | colour index |
| INTEGER NCC | number of components of colour value |
| REAL COLR(NCC) | colour values |

**SET CURVE APPROXIMATION CRITERIA**

SUBROUTINE PSCAC(ACRI,LDR,DATREC)

*Input Parameters:*

| | |
|---|---|
| INTEGER ACRI | curve approximation criteria type |
| INTEGER LDR | length of data record array |
| CHARACTER*80 DATREC(LDR) | data record |

DATREC parameters to build CURVE APPROXIMATION CRITERIA DATA RECORD for ACRI=PWDCA: No data record

DATREC parameters to build CURVE APPROXIMATION CRITERIA DATA RECORD for ACRI =PCBKCA:
The IA array contains:
- The actual data of count
The other arrays are empty.

IL= 1
IA(1)= count
RL= 0
RA= ( )
SL= 0
LSTR= ( )
STR= ( )

DATREC parameters to build CURVE APPROXIMATION CRITERIA DATA RECORD for ACRI =PCSWCA or PCSNCA or PCSDCA or PCDWCA or PCDNCA or PCDDCA or PRWCA or PRNCA or PRDCA:
The RA array contains:
- The actual data of approximation value
The other arrays are empty.

IL= 0

IA= ( )
RL= 1
RA(1)= approximation value
SL= 0
LSTR= ( )
STR= ( )

## SET SURFACE APPROXIMATION CRITERIA

SUBROUTINE PSSAC(ACRI,LDR,DATREC)

*Input Parameters:*

| | |
|---|---|
| INTEGER ACRI | surface approximation criteria type |
| INTEGER LDR | length of data record array |
| CHARACTER*80 DATREC(LDR) | data record |

DATREC parameters to build SURFACE APPROXIMATION CRITERIA DATA RECORD for ACRI=PWDCA: No data record

DATREC parameters to build SURFACE APPROXIMATION CRITERIA DATA RECORD for ACRI=PCBKSA:
The IA array contains:
- The actual data of u count
- The actual data of v count

The other arrays are empty.
IL= 2
IA(1)= u count
IA(2)= v count
RL= 0
RA= ( )
SL= 0
LSTR= ( )
STR= ( )

DATREC parameters to build SURFACE APPROXIMATION CRITERIA DATA RECORD for ACRI = PCSWSA or PCSNSA or PCSDSA:
The RA array contains:
- The actual data of u approximation value
- The actual data of v approximation value

The other arrays are empty.
IL= 0
IA= ( )
RL= 2
RA(1)= u approximation value
RA(2)= v approximation value
SL= 0
LSTR= ( )
STR= ( )

DATREC parameters to build SURFACE APPROXIMATION CRITERIA DATA RECORD for ACRI = PPDWSA or PPDNSA or PPDDSA or PRWSA or PRNSA or PRDSA:
The RA array contains:
- The actual data of approximation value

The other arrays are empty.
IL= 0
IA= ( )
RL= 1
RA(1)= approximation value
SL= 0
LSTR= ( )
STR= ( )

**SET PARAMETRIC SURFACE CHARACTERISTICS**

SUBROUTINE PSPSC(PSCTY,LDR,DATREC)

*Input Parameters:*

| | |
|---|---|
| INTEGER PSCTY | parametric surface characteristics type |
| INTEGER LDR | length of data record array |
| CHARACTER*80 DATREC(LDR) | data record |

DATREC parameters to build PARAMETRIC SURFACE CHARACTERISTICS DATA RECORD for PSCTY = PNOPC or PWDPC: No data record

DATREC parameters to build PARAMETRIC SURFACE CHARACTERISTICS DATA RECORD for PSCTY = PICPC:
The IA array contains:
- The actual data of curve placement
- The actual data of curve count in u direction
- The actual data of curve count in v direction

The other arrays are empty.

IL= 3
IA(1)= curve placement(PUOSCP,PUBKCP)
IA(2)= curve count in u direction
IA(3)= curve count in v direction
RL= 0
RA= ( )
SL= 0
LSTR= ( )
STR= ( )

DATREC parameters to build PARAMETRIC SURFACE CHARACTERISTICS DATA RECORD for PSCTY = PLCMPC:
The IA array contains:
- The number of parameters(NP)

The RA array contains:
- The actual data of origin point
- The actual data of direction vector
- The actual data of parameters. There are NP parameters.

IL= 1
IA(1)= number of parameters = NP
RL= 6+NP
RA(1)= x-coordinate of origin point(MC)
RA(2)= y-coordinate of origin point(MC)
RA(3)= z-coordinate of origin point(MC)
RA(4)= x-value of direction vector(MC)
RA(5)= y-value of direction vector(MC)
RA(6)= z-value of direction vector(MC)
RA= element 7 through NP+6 contain parameters
SL= 0
LSTR= ( )
STR= ( )

DATREC parameters to build PARAMETRIC SURFACE CHARACTERISTICS DATA

RECORD for PSCTY = PLCWPC:
The IA array contains:
- The number of parameters(NP)
The RA array contains:
- The actual data of origin point
- The actual data of direction vector
- The actual data of parameters. There are NP parameters.

IL= 1
IA(1)= number of parameters = NP
RL= 6+NP
RA(1)= x-coordinate of origin point(WC)
RA(2)= y-coordinate of origin point(WC)
RA(3)= z-coordinate of origin point(WC)
RA(4)= x-value of direction vector(WC)
RA(5)= y-value of direction vector(WC)
RA(6)= z-value of direction vector(WC)
RA= element 7 through NP+6 contain parameters
SL= 0
LSTR= ( )
STR= ( )

**SET RENDERING COLOUR MODEL**

SUBROUTINE PSRCM(RCOLM)

*Input Parameters:*

| | |
|---|---|
| INTEGER RCOLM | rendering colour model |

**SET DEPTH CUE INDEX**

SUBROUTINE PSDCI(DCI)

*Input Parameters:*

| | |
|---|---|
| INTEGER DCI | depth cue index |

**SET COLOUR MAPPING INDEX**

SUBROUTINE PSCMI(CMI)

*Input Parameters:*

| | |
|---|---|
| INTEGER CMI | colour mapping index |

### 13.3.3 Aspect source flag setting

**SET INDIVIDUAL ASF**

SUBROUTINE PSIASF(ASPCID,ASFVAL)

*Input Parameters:*

| | |
|---|---|
| INTEGER ASPCID | aspect identifier<br>*(PLN,PLWSC,PPLCI,PMK,PMKSC,PPMCI,PTXFN,*<br>*PTXPR,PCHXP,PCHSP,PTXCI,PIS,PISI,PICI,*<br>*PEDFG,PEDTY,PEWSC,PEDCI,*[8]<br>PPLCOL,PPMCOL,PTXCOL,PINCOL,PEDCOL,<br>PPSHMD,PISHMD,PDMPMD,PREFPR,PREFM,<br>PBINSY,PBINSI,PBICOL,PBISHM,PBDMPM,<br>PBREFP,PBREFM,PCAPCR,PSAPCR,PPASUC) |
| INTEGER ASFVAL | aspect source flag value<br>(PBUNDL,PINDIV) |

This function is defined in PHIGS part 1. Additional aspect identifiers are defined in PHIGS part 4. This binding shows those additional aspect identifiers.



[8] already defined in PHIGS

### 13.3.4 Workstation attribute table definition

**SET POLYLINE REPRESENTATION PLUS**

SUBROUTINE PSPLP(WKID,PLI,LTYPE,LWIDTH,CTYPE,COLI,NCC,COLR,PLSM, *ACRI,LDR,DATREC)

*Input Parameters:*

| | |
|---|---|
| INTEGER WKID | workstation identifier |
| INTEGER PLI | polyline index |
| INTEGER LTYPE | linetype |
| REAL LWIDTH | linewidth scale factor |
| INTEGER CTYPE | polyline colour type |
| INTEGER COLI | colour index |
| INTEGER NCC | number of components of colour value |
| REAL COLR(NCC) | colour values |
| INTEGER PLSM | polyline shading method |
| INTEGER ACRI | curve approximation criteria type |
| INTEGER LDR | length of data record array |
| CHARACTER*80 DATREC(LDR) | curve approximation criteria data record[9] |

**SET POLYMARKER REPRESENTATION PLUS**

SUBROUTINE PSPMP(WKID,PMI,MTYPE,MSZSF,CTYPE,COLI,NCC,COLR)

*Input Parameters:*

| | |
|---|---|
| INTEGER WKID | workstation identifier |
| INTEGER PMI | polymarker index |
| INTEGER MTYPE | marker type |
| REAL MSZSF | marker size scale factor |
| INTEGER CTYPE | polymarker colour type |
| INTEGER COLI | colour index |
| INTEGER NCC | number of components of colour value |
| REAL COLR(NCC) | colour values |



[9] See SET CURVE APPROXIMATION CRITERIA for a description of the curve approximation criteria data records.

**SET TEXT REPRESENTATION PLUS**

SUBROUTINE PSTXRP(WKID,TXI,FONT,PREC,CHXP,CHSP,CTYPE,COLI,NCC,
*COLR)

*Input Parameters:*

| | |
|---|---|
| INTEGER WKID | workstation identifier |
| INTEGER TXI | text index |
| INTEGER FONT | text font |
| INTEGER PREC | text precision (PSTRP,PCHARP,PSTRKP) |
| REAL CHXP | character expansion factor |
| REAL CHSP | character spacing |
| INTEGER CTYPE | text colour type |
| INTEGER COLI | colour index |
| INTEGER NCC | number of components of colour value |
| REAL COLR(NCC) | colour values |

**SET INTERIOR REPRESENTATION PLUS**

SUBROUTINE PSIP(WKID,II,INTS,STYLI,CTYPE,COLI,NCC,COLR,ISM)

*Input Parameters:*

| | |
|---|---|
| INTEGER WKID | workstation identifier |
| INTEGER II | interior index |
| INTEGER INTS | interior style (PHOLLO,PSOLID,PPATTR,PPATCH,PISEMP) |
| INTEGER STYLI | interior style index |
| INTEGER CTYPE | interior colour type |
| INTEGER COLI | colour index |
| INTEGER NCC | number of components of colour value |
| REAL COLR(NCC) | colour values |
| INTEGER ISM | interior shading method |

**SET EDGE REPRESENTATION PLUS**

SUBROUTINE PSEDP(WKID,EDI,EDFLAG,EDTYPE,EWIDTH,CTYPE,COLI,NCC,
*COLR)

*Input Parameters:*

| | |
|---|---|
| INTEGER WKID | workstation identifier |
| INTEGER EDI | edge index |
| INTEGER EDFLAG | edge flag (POFF,PON) |
| INTEGER EDTYPE | edge type |
| REAL EWIDTH | edgewidth scale factor |
| INTEGER CTYPE | edge colour type |
| INTEGER COLI | colour index |
| INTEGER NCC | number of components of colour value |
| REAL COLR(NCC) | colour values |

**SET DATA MAPPING REPRESENTATION**

SUBROUTINE PSDMR(WKID,DMI,DMM,LDR,DATREC)

*Input Parameters:*

| | |
|---|---|
| INTEGER WKID | workstation identifier |
| INTEGER DMI | data mapping index |
| INTEGER DMM | data mapping method |
| INTEGER LDR | length of data record array |
| CHARACTER*80 DATREC(LDR) | data record[10] |

**SET REFLECTANCE REPRESENTATION**

SUBROUTINE PSRFR(WKID,RFI,RFM,RPTYPE,LDR,DATREC)

*Input Parameters:*

| | |
|---|---|
| INTEGER WKID | workstation identifier |
| INTEGER RFI | reflectance index |
| INTEGER RFM | reflectance model |
| INTEGER RPTYPE | reflectance properties type |
| INTEGER LDR | length of data record array |
| CHARACTER*80 DATREC(LDR) | data record[11] |

**SET PARAMETRIC SURFACE REPRESENTATION**

SUBROUTINE PSPSR(WKID,PSI,ACRI,LDRAC,DARECA,PSCTY,LDRPS,DARECP)

*Input Parameters:*

| | |
|---|---|
| INTEGER WKID | workstation identifier |
| INTEGER PSI | parametric surface index |
| INTEGER ACRI | surface approximation criteria type |
| INTEGER LDRAC | length of surface approximation criteria data record array |
| CHARACTER*80 DARECA(LDRAC) | surface approximation criteria data record[12] |
| INTEGER PSCTY | parametric surface characteristics type |
| INTEGER LDRPS | length of parametric surface characteristics data record array |
| CHARACTER*80 DARECP(LDRPS) | parametric surface characteristics data record[13] |



[10] See SET DATA MAPPING METHOD for a description of the data mapping method data records.

[11] See SET REFLECTANCE PROPERTIES for a description of the reflectance properties data records.

[12] See SET SURFACE APPROXIMATION CRITERIA for a description of the surface approximation criteria data records.

[13] See SET PARAMETRIC SURFACE CHARACTERISTICS for a description of the parametric surface characteristics data records.

## SET PATTERN REPRESENTATION PLUS

SUBROUTINE PSPAP(WKID,PAI,DIMX,DIMY,ISC,ISR,DX,DY,CTYPE,COLIA,
*NCC,COLRA)

*Input Parameters:*

| | |
|---|---|
| INTEGER WKID | workstation identifier |
| INTEGER PAI | pattern index |
| INTEGER DIMX,DIMY | number of rows, number of columns in COLIA orr COLRA |
| INTEGER ISC,ISR | indices of start column, start row for pattern |
| INTEGER DX,DY | number of columns, number of rows in pattern |
| INTEGER CTYPE | interior colour type |
| INTEGER COLIA(DIMX,DIMY) | colour index array |
| INTEGER NCC | number of components of colour value |
| REAL COLRA(NCC*DIMX*DIMY) | colour value array |

## SET LIGHT SOURCE REPRESENTATION

SUBROUTINE PSLSR(WKID,LSI,LSTYP,LDR,DATREC)

*Input Parameters:*

| | |
|---|---|
| INTEGER WKID | workstation identifier |
| INTEGER LSI | light source index |
| INTEGER LSTYP | light source type |
| INTEGER LDR | length of data record array |
| CHARACTER*80 DATREC(LDR) | data record |

DATREC parameters to build LIGHT SOURCE DATA RECORD for LSTYP = PAMB:
The IA array contains:
- The colour type(CT)
- The actual data of colour index, if CT is indirect
- The number of components of colour value(NCC)

The RA array contains:
- The actual data of colour values as required by CT

Common element:
IL= 2
IA(1)= colour type = CT
RL= *
RA= ( )
SL= 0
LSTR= ( )
STR= ( )
Individual element:
CASE colour type=PINDIR:
IA(2)= colour index
CASE colour type=PRGB or PCIE or PHSV or PHLS:
IA(2)= number of components of colour value = NCC
RA= element 1 through NCC contain the colour values

DATREC parameters to build LIGHT SOURCE DATA RECORD for LSTYP = PDIRE:
The IA array contains:
- The colour type(CT)
- The actual data of colour index, if CT is indirect.

- The number of components of colour value(NCC)
The RA array contains:
- The actual data of direction vector
- The actual data of colour values as required by CT.
Common element:
IL= 2
IA(1)= colour type = CT
RL= 3+*
RA(1)=x-value of direction vector(WC)
RA(2)=y-value of direction vector(WC)
RA(3)=z-value of direction vector(WC)
SL= 0
LSTR= ( )
STR= ( )
Individual element:
  CASE colour type=PINDIR:
    IA(2)=colour index
  CASE colour type=PRGB or PCIE or PHSV or PHLS:
    IA(2)= number of components of colour value = NCC
    RA= element 4 through 3+NCC contain the colour values

DATREC parameters to build LIGHT SOURCE DATA RECORD for LSTYP = PPOSI:
The IA array contains:
- The colour type(CT)
- The actual data of colour index, if CT is indirect.
- The number of components of colour value(NCC)
The RA array contains:
- The actual data of direction vector
- The actual data of attenuation coefficients
- The actual data of colour values as required by CT.
Common element:
IL= 2
IA(1)= colour type = CT
RL= 5+*
RA(1)=x-value of direction vector(WC)
RA(2)=y-value of direction vector(WC)
RA(3)=z-value of direction vector(WC)
RA(4)=attenuation coefficient 1
RA(5)=attenuation coefficient 2
SL= 0
LSTR= ( )
STR= ( )
Individual element:
  CASE colour type=PINDIR:
    IA(2)=colour index
  CASE colour type=PRGB or PCIE or PHSV or PHLS:
    IA(2)= number of components of colour value = NCC
    RA= element 6 through 5+NCC contain the colour values

DATREC parameters to build LIGHT SOURCE DATA RECORD for LSTYP = PSPOT:
The IA array contains:
- The colour type(CT)
- The actual data of colour index, if CT is indirect.
- The number of components of colour value(NCC)

The RA array contains:
- The actual data of light source position
- The actual data of direction vector of light source
- The actual data of concentration exponent
- The actual data of attenuatioon coefficients
- The actual data of spread angle
- The actual data of colour values as required by CT.

Common element:
IL= 2
IA(1)= colour type = CT
RL= 10+*
RA(1)=x-coordinate of light source position(WC)
RA(2)=y-coordinate of light source position(WC)
RA(3)=z-coordinate of light source position(WC)
RA(4)=x-value of light source direction(WC)
RA(5)=y-value of light source direction(WC)
RA(6)=z-value of light source direction(WC)
RA(7)= concentration exponent
RA(8)=attenuation coefficient 1
RA(9)=attenuation coefficient 2
RA(10)= spread angle
SL= 0
LSTR= ( )
STR= ( )
Individual element:
CASE colour type=PINDIR:
IA(2)=colour index
CASE colour type=PRGB or PCIE or PHSV or PHLS:
IA(2)= number of components of colour value = NCC
RA= element 11 through 10+NCC contain the colour values

**SET DEPTH CUE REPRESENTATION**

SUBROUTINE PSDCR(WKID,DCI,DCMODE,DQMIN,DQMAX,DCSFMI,DCSFMX,
*CTYPE,COLI,NCC,COLR)

*Input Parameters:*

| | |
|---|---|
| INTEGER WKID | workstation identifier |
| INTEGER DCI | depth cue index |
| INTEGER DCMODE | depth cue mode (PSUPPR,PALLOW) |
| REAL DQMIN,DQMAX | depth cue reference planes(NPC) |
| REAL DCSFMI,DCSFMX | depth cue scale factors |
| INTEGER CTYPE | depth cue colour type |
| INTEGER COLI | colour index |
| INTEGER NCC | number of components of colour value |
| REAL COLR(NCC) | colour values |

## SET COLOUR MAPPING REPRESENTATION

SUBROUTINE PSCMR(WKID,CMI,CMM,LDR,DATREC)

*Input Parameters:*

| | |
|---|---|
| INTEGER WKID | workstation identifier |
| INTEGER CMI | colour mapping index |
| INTEGER CMM | colour mapping method |
| INTEGER LDR | length of data record array |
| CHARACTER*80 DATREC(LDR) | data record |

DATREC parameters to build COLOUR MAPPING METHOD DATA RECORD
for CMM = PTRUE: no data record

DATREC parameters to build COLOUR MAPPING METHOD DATA RECORD
for CMM = PSUD:
The IA array contains:
- The colour model(CM)
- The number of components of colour value(NCC), same as the number of weight vector values
- The number of list of colours in the specified colour model(NLC)

The RA array contains:
- The actual data of weight vector
- The list of colour values. There are NLC colour values,
  each having as many components as required by CM.

IL= 3
IA(1)= colour model = CM
IA(2)= number of components of colour value, same as number of weight vector value = NCC
IA(3)= number of list of colours in the specified colour model = NLC
RL= NCC+NCC*NLC
RA= element 1 through NCC contain the weight vector
RA= element NCC+1 through NCC*NLC+NCC contain the colour value
  example r1,g1,b1,r2,g2,b2,...
SL= 0
LSTR= ( )
STR= ( )

DATREC parameters to build COLOUR MAPPING METHOD DATA RECORD
for CMM = PSUDN:
The IA array contains:
- The colour model(CM)
- The number of components in the colour model(NCC)
- The array of end indices for list of colour coordinates(IXA)

The RA array contains:
- The list of colour coordinates.

IL= 2
IA(1)= colour model = CM
IA(2)= number of components in the colour model = NCC
IA= element 3 through NCC+2 contain the array of end indices for lists of colour coordinates
RL= IA(NCC+2)
RA=element 1 through IA(NCC+2) contain the lists of colour coordinates
  example for colour model=PRGB:
    RA element 1 through IA(3) contain red values;
    RA element IA(3)+1 through IA(4) contain green values
    RA element IA(4)+1 through IA(5) contain blue values

SL= 0
LSTR= ( )
STR= ( )

## 13.4 Inquiry functions

### 13.4.1 Inquiry functions for workstation state list

#### INQUIRE POLYLINE REPRESENTATION PLUS

SUBROUTINE PQPLP(WKID,PLI,CCSBSZ,TYPE,MLDR,ERRIND,LTYPE,
*LWIDTH,CTYPE,COLI,NCC,COLR,PLSM,ACRI,LDR,DATREC)

*Input Parameters:*

| | |
|---|---|
| INTEGER WKID | workstation identifier |
| INTEGER PLI | polyline index |
| INTEGER CCSBSZ | colour component specification buffer size |
| INTEGER TYPE | type of return value(PSET,PREALI) |
| INTEGER MLDR | dimension of curve approximation criteria data record array |

*Output Parameters:*

| | |
|---|---|
| INTEGER ERRIND | error indicator |
| INTEGER LTYPE | linetype |
| REAL LWIDTH | linewidth scale factor |
| INTEGER CTYPE | colour type |
| INTEGER COLI | colour index |
| INTEGER NCC | number of components of colour value |
| REAL COLR(CCSBSZ) | colour values |
| INTEGER PLSM | polyline shading method |
| INTEGER ACRI | curve approximation criteria type |
| INTEGER LDRCA | length of curve approximation criteria data record array |
| CHARACTER*80 DATREC(MLDR) | curve approximation criteria data record[14] |

#### INQUIRE POLYMARKER REPRESENTATION PLUS

SUBROUTINE PQPMP(WKID,PMI,CCSBSZ,TYPE,ERRIND,MTYPE,MSZSF,CTYPE,
*COLI,NCC,COLR)

*Input Parameters:*

| | |
|---|---|
| INTEGER WKID | workstation identifier |
| INTEGER PMI | polymarker index |
| INTEGER CCSBSZ | colour component specification buffer size |
| INTEGER TYPE | type of return value(PSET,PREALI) |

*Output Parameters:*

| | |
|---|---|
| INTEGER ERRIND | error indicator |
| INTEGER MTYPE | marker type |
| REAL MSZSF | marker scale factor |
| INTEGER CTYPE | colour type |
| INTEGER COLI | colour index |
| INTEGER NCC | number of components of colour value |
| REAL COLR(CCSBSZ) | colour values |

[14] See SET CURVE APPROXIMATION CRITERIA for a description of the curve approximation criteria data records.

## INQUIRE TEXT REPRESENTATION PLUS

SUBROUTINE PQTXP(WKID,TXI,CCSBSZ,TYPE,ERRIND,FONT,PREC,CHXP, *CHSP,CTYPE,COLI,NCC,COLR)

*Input Parameters:*

| | |
|---|---|
| INTEGER WKID | workstation identifier |
| INTEGER TXI | text index |
| INTEGER CCSBSZ | colour component specification buffer size |
| INTEGER TYPE | type of return value(PSET,PREALI) |

*Output Parameters:*

| | |
|---|---|
| INTEGER ERRIND | error indicator |
| INTEGER FONT | text font |
| INTEGER PREC | text precision(PSTRP,PCHARP,PSTRKP) |
| REAL CHXP | character expansion factor |
| REAL CHSP | character spacing |
| INTEGER CTYPE | colour type |
| INTEGER COLI | colour index |
| INTEGER NCC | number of components of colour value |
| REAL COLR(CCSBSZ) | colour values |

## INQUIRE INTERIOR REPRESENTATION PLUS

SUBROUTINE PQIP(WKID,II,CCSBSZ,TYPE,ERRIND,INTS,STYLI,CTYPE,COLI, *NCC,COLR,ISM)

*Input Parameters:*

| | |
|---|---|
| INTEGER WKID | workstation identifier |
| INTEGER II | interior index |
| INTEGER CCSBSZ | colour component specification buffer size |
| INTEGER TYPE | type of return value(PSET,PREALI) |

*Output Parameters:*

| | |
|---|---|
| INTEGER ERRIND | error indicator |
| INTEGER INTS | interior style |
| INTEGER STYLI | interior style index |
| INTEGER CTYPE | colour type |
| INTEGER COLI | colour index |
| INTEGER NCC | number of components of colour value |
| REAL COLR(CCSBSZ) | colour values |
| INTEGER ISM | interior shading method |

**INQUIRE EDGE REPRESENTATION PLUS**

SUBROUTINE PQEDP(WKID,EDI,CCSBSZ,TYPE,ERRIND,EDFLAG,EDTYPE,
*EWIDTH,CTYPE,COLI,NCC,COLR)

*Input Parameters:*

| | |
|---|---|
| INTEGER WKID | workstation identifier |
| INTEGER EDI | edge index |
| INTEGER CCSBSZ | colour component specification buffer size |
| INTEGER TYPE | type of return value(PSET,PREALI) |

*Output Parameters:*

| | |
|---|---|
| INTEGER ERRIND | error indicator |
| INTEGER EDFLAG | edge flag (POFF,PON) |
| INTEGER EDTYPE | edge type |
| REAL EWIDTH | edge scale factor |
| INTEGER CTYPE | colour type |
| INTEGER COLI | colour index |
| INTEGER NCC | number of components of colour value |
| REAL COLR(CCSBSZ) | colour values |

**INQUIRE LIST element OF DATA MAPPING INDICES**

SUBROUTINE PQEDMI(WKID,N,ERRIND,OL,DMI)

*Input Parameters:*

| | |
|---|---|
| INTEGER WKID | workstation identifier |
| INTEGER N | list element requested |

*Output Parameters:*

| | |
|---|---|
| INTEGER ERRIND | error indicator |
| INTEGER OL | number of data mapping bundle table entries |
| INTEGER DMI | Nth element of list of defined data mapping indices |

**INQUIRE DATA MAPPING REPRESENTATION**

SUBROUTINE PQDMR(WKID,DMI,TYPE,MLDR,ERRIND,DMM,LDR,DATREC)

*Input Parameters:*

| | |
|---|---|
| INTEGER WKID | workstation identifier |
| INTEGER DMI | data mapping index |
| INTEGER TYPE | type of return value(PSET,PREALI) |
| INTEGER MLDR | dimension of data record array |

*Output Parameters:*

| | |
|---|---|
| INTEGER ERRIND | error indicator |
| INTEGER DMM | data mapping method |
| INTEGER LDR | length of data record array |
| CHARACTER*80 DATREC(MLDR) | data mapping data record[15] |

[15] See SET DATA MAPPING METHOD for a description of the data mapping method data records.

**INQUIRE LIST element OF REFLECTANCE INDICES**

SUBROUTINE PQERFI(WKID,N,ERRIND,OL,RFI)

*Input Parameters:*

| | |
|---|---|
| INTEGER WKID | workstation identifier |
| INTEGER N | list element requested |

*Output Parameters:*

| | |
|---|---|
| INTEGER ERRIND | error indicator |
| INTEGER OL | number of reflectance bundle table entries |
| INTEGER RFI | Nth element of list of defined reflectance indices |

**INQUIRE REFLECTANCE REPRESENTATION**

SUBROUTINE PQRFR(WKID,RFI,TYPE,MLDR,ERRIND,RFM,RPTYPE,LDR, *DATREC)

*Input Parameters:*

| | |
|---|---|
| INTEGER WKID | workstation identifier |
| INTEGER RFI | reflectance index |
| INTEGER TYPE | type of return value(PSET,PREALI) |
| INTEGER MLDR | dimension of data record array |

*Output Parameters:*

| | |
|---|---|
| INTEGER ERRIND | error indicator |
| INTEGER RFM | reflectance model |
| INTEGER RPTYPE | reflectance propety type |
| INTEGER LDR | length of data record array |
| CHARACTER*80 DATREC(MLDR) | reflectance property colour value data record[16] |

**INQUIRE LIST element OF PARAMETRIC SURFACE INDICES**

SUBROUTINE PQEPSI(WKID,N,ERRIND,OL,PSI)

*Input Parameters:*

| | |
|---|---|
| INTEGER WKID | workstation identifier |
| INTEGER N | list element requested |

*Output Parameters:*

| | |
|---|---|
| INTEGER ERRIND | error indicator |
| INTEGER OL | number of parametric surface table entries |
| INTEGER PSI | Nth element of list of defined parametric surface indices |

[16] See SET REFLECTANCE PROPERTIES for a description of the reflectance properties data records.

## INQUIRE PARAMETRIC SURFACE REPRESENTATION

SUBROUTINE PQPSR(WKID,PSI,TYPE,MLDRSA,MLDRPS,ERRIND,ACRI,LDRSA,
*DARECS,PSCH,LDRPS,DARECP)

*Input Parameters:*

| | |
|---|---|
| INTEGER WKID | workstation identifier |
| INTEGER PSI | parametric surface bundle index |
| INTEGER TYPE | type of return value(PSET,PREALI) |
| INTEGER MLDRSA | dimension of surface approximation criteria data record array |
| INTEGER MLDRPS | dimension of parametric surface characteristics data record array |

*Output Parameters:*

| | |
|---|---|
| INTEGER ERRIND | error indicator |
| INTEGER ACRI | surface approximation criteria type |
| INTEGER LDRSA | length of surface approximation criteria data record |
| CHARACTER*80 DARECS(MLDRSA) | surface approximation criteria data record[17] |
| INTEGER PSCH | parametric surface characteristics type |
| INTEGER LDRPS | length of parametric surface characteristics data record |
| CHARACTER*80 DARECP(MLDRPS) | parametric surface characteristics data record[18] |

## INQUIRE PATTERN REPRESENTATION PLUS

SUBROUTINE PQPAP(WKID,PAI,CCSBSZ,TYPE,DIMX,DIMY,ERRIND,DX,DY,
*CTYPE,COLIA,NCC,COLRA)

*Input Parameters:*

| | |
|---|---|
| INTEGER WKID | workstation identifier |
| INTEGER PAI | pattern index |
| INTEGER CCSBSZ | colour component specification buffer size |
| INTEGER TYPE | type of return value(PSET,PREALI) |
| INTEGER DIMX,DIMY | maximum pattern array dimension |

*Output Parameters:*

| | |
|---|---|
| INTEGER ERRIND | error indicator |
| INTEGER DX,DY | pattern array dimensions |
| INTEGER CTYPE | colour type |
| INTEGER COLIA(DIMX,DIMY) | colour index |
| INTEGER NCC | number of components of colour value |
| REAL COLRA(CCSBSZ) | colour values |



[17] See SET SURFACE APPROXIMATION CRITERIA for a description of the surface approximation criteria data records.

[18] See SET PARAMETRIC SURFACE CHARACTERISTICS for a description of the parametric surface characteristics data records.

**INQUIRE LIST element OF LIGHT SOURCE INDICES**

SUBROUTINE PQELSI(WKID,N,ERRIND,OL,LSI)

*Input Parameters:*

| | |
|---|---|
| INTEGER WKID | workstation identifier |
| INTEGER N | list element requested |

*Output Parameters:*

| | |
|---|---|
| INTEGER ERRIND | error indicator |
| INTEGER OL | number of light source table entries |
| INTEGER LSI | Nth element of list of defined light source indices |

**INQUIRE LIGHT SOURCE REPRESENTATION**

SUBROUTINE PQLSR(WKID,LSI,TYPE,MLDR,ERRIND,LSTYP,LDR,DATREC)

*Input Parameters:*

| | |
|---|---|
| INTEGER WKID | workstation identifier |
| INTEGER LSI | light source index |
| INTEGER TYPE | type of return value(PSET,PREALI) |
| INTEGER MLDR | dimension of data record array |

*Output Parameters:*

| | |
|---|---|
| INTEGER ERRIND | error indicator |
| INTEGER LSTYP | light source type |
| INTEGER LDR | length of data record array |
| CHARACTER*80 DATREC(MLDR) | data record[19] |

**INQUIRE LIST element OF DEPTH CUE INDICES**

SUBROUTINE PQEDCI(WKID,N,ERRIND,NCC,DCI)

*Input Parameters:*

| | |
|---|---|
| INTEGER WKID | workstation identifier |
| INTEGER N | list element requested |

*Output Parameters:*

| | |
|---|---|
| INTEGER ERRIND | error indicator |
| INTEGER OL | number of depth cue table entries |
| INTEGER DCI | Nth element of list of defined depth cue indices |



[19] See SET LIGHT SOURCE REPRESENTATION for a description of the light source data record.

**INQUIRE DEPTH CUE REPRESENTATION**

SUBROUTINE PQDCR(WKID,DCI,CCSBSZ,TYPE,ERRIND,DCMODE,DQMIN,
*DQMAX,DCSFMI,DCSFMX,CTYPE,COLI,NCC,COLR)

*Input Parameters:*

| | |
|---|---|
| INTEGER WKID | workstation identifier |
| INTEGER DCI | depth cue index |
| INTEGER CCSBSZ | colour component specification buffer size |
| INTEGER TYPE | type of return value(PSET,PREALI) |

*Output Parameters:*

| | |
|---|---|
| INTEGER ERRIND | error indicator |
| INTEGER DCMODE | depth cue mode(PSUPPR,PALLOW) |
| REAL DQMIN,DQMAX | depth cue reference plane(NPC) |
| REAL DCSFMI,DCSFMX | depth cue scale factors |
| INTEGER CTYPE | colour type |
| INTEGER COLI | colour index |
| INTEGER NCC | number of components of colour value |
| REAL COLR(CCSBSZ) | colour values |

**INQUIRE COLOUR MAPPING STATE**

SUBROUTINE PQCMS(WKID,CMM,MLDR,ERRIND,LDR,DATREC)

*Input Parameters:*

| | |
|---|---|
| INTEGER WKID | workstation identifier |
| INTEGER CMM | colour mapping method |
| INTEGER MLDR | dimension of data record array |

*Output Parameters:*

| | |
|---|---|
| INTEGER ERRIND | error indicator |
| INTEGER LDR | length of data record array |
| CHARACTER*80 DATREC(MLDR) | data record |

DATREC parameters to build COLOUR MAPPING DATA RECORD for CMM = PTRUE:
The IA array contains:
- The number of true colours available
The other array are empty.
IL= 1
IA(1)= number of true colours available
RL= 0
RA= ( )
SL= 0
LSTR= ( )
STR= ( )

DATREC parameters to build COLOUR MAPPING DATA RECORD for CMM = PSUD:
The IA array contains:
- The number of pseudo colour entries available
The other array are empty.
IL= 1
IA(1)= number of pseudo colour entries available
RL= 0

RA= ( )
SL= 0
LSTR= ( )
STR= ( )

DATREC parameters to build COLOUR MAPPING DATA RECORD for CMM = PSUDN:
No data record

**INQUIRE LIST element OF COLOUR MAPPING INDICES**

SUBROUTINE PQECMI(WKID,N,ERRIND,OL,CMI)

*Input Parameters:*

| | |
|---|---|
| INTEGER WKID | workstation identifier |
| INTEGER N | list element requested |

*Output Parameters:*

| | |
|---|---|
| INTEGER ERRIND | error indicator |
| INTEGER OL | number of colour mapping table entries |
| INTEGER CMI | Nth element of list of defined colour mapping indices |

**INQUIRE COLOUR MAPPING REPRESENTATION**

SUBROUTINE PQCMR(WKID,CMI,TYPE,MLDR,ERRIND,CMM,LDR,DATREC)

*Input Parameters:*

| | |
|---|---|
| INTEGER WKID | workstation identifier |
| INTEGER CMI | colour mapping index |
| INTEGER TYPE | type of return value(PSET,PREALI) |
| INTEGER MLDR | dimension of data record array |

*Output Parameters:*

| | |
|---|---|
| INTEGER ERRIND | error indicator |
| INTEGER CMM | colour mapping method |
| INTEGER LDR | length of data record array |
| CHARACTER*80 DATREC(MLDR) | data record[20] |

[20] See SET COLOUR MAPPING REPRESENTATION for a description of the colour mapping data record.

## 13.4.2 Inquiry functions for workstation description table

### INQUIRE DIRECT COLOUR MODEL FACILITIES

SUBROUTINE PQDCMF(WTYPE,N,ERRIND,NDCM,DCM)

*Input Parameters:*

| | |
|---|---|
| INTEGER WTYPE | workstation type |
| INTEGER N | list element requested of direct colour model |

*Output Parameters:*

| | |
|---|---|
| INTEGER ERRIND | error indicator |
| INTEGER NDCM | number of available direct colour model |
| INTEGER DCM | Nth element of list of defined direct colour model |

### INQUIRE RENDERING COLOUR MODEL FACILITIES

SUBROUTINE PQRCMF(WTYPE,N,ERRIND,NRCM,RCM)

*Input Parameters:*

| | |
|---|---|
| INTEGER WTYPE | workstation type |
| INTEGER N | list element requested of rendering colour model |

*Output Parameters:*

| | |
|---|---|
| INTEGER ERRIND | error indicator |
| INTEGER NRCM | number of available rendering colour model |
| INTEGER RCM | Nth element of list of defined rendering colour model |

### INQUIRE DYNAMICS OF WORKSTATION PLUS

SUBROUTINE PQDCWP(WTYPE,ERRIND,DMR,RFR,PSR,LSR,DCR,CMR)

*Input Parameters:*

| | |
|---|---|
| INTEGER WTYPE | workstation type |

*Output Parameters:*

| | |
|---|---|
| INTEGER ERRIND | error indicator |
| INTEGER DMR | data mapping representation(PIRG,PIMM,PCBS) |
| INTEGER RFR | reflectance representation(PIRG,PIMM,PCBS) |
| INTEGER PSR | parametric surface representation (PIRG,PIMM,PCBS) |
| INTEGER LSR | light source representation(PIRG,PIMM,PCBS) |
| INTEGER DCR | depth cue representation(PIRG,PIMM,PCBS) |
| INTEGER CMR | colour mapping representation(PIRG,PIMM,PCBS) |

## INQUIRE POLYLINE FACILITIES PLUS

SUBROUTINE PQPLFP(WTYPE,NRLT,NRSM,ERRIND,NLT,LT,NLW,NOMLW,
*RLWMIN,RLWMAX,NSM,SM,NPPLI)

*Input Parameters:*

| | |
|---|---|
| INTEGER WTYPE | workstation type |
| INTEGER NRLT | list of element requested of linetypes |
| INTEGER NRSM | list of element requested of shading method |

*Output Parameters:*

| | |
|---|---|
| INTEGER ERRIND | error indicator |
| INTEGER NLT | number of available linetypes |
| INTEGER LT | element NRLT of list of available linetypes |
| INTEGER NLW | number of available linewidths |
| REAL NOMLW | nominal linewidth |
| REAL RLWMIN,RLWMAX | range of linewidths |
| INTEGER NSM | number of available shading method |
| INTEGER SM | element NRSM of list of available shading method |
| INTEGER NPPLI | number of predefined polyline indices |

## INQUIRE PREDEFINED POLYLINE REPRESENTATION PLUS

SUBROUTINE PQPPLP(WTYPE,PLI,CCSBSZ,MLDR,ERRIND,LTYPE,LWIDTH,
*CTYPE,COLI,NCC,COLR,PLSM,ACRI,LDR,DATREC)

*Input Parameters:*

| | |
|---|---|
| INTEGER WTYPE | workstation type |
| INTEGER PLI | predefined polyline index |
| INTEGER CCSBSZ | colour component specification buffer size |
| INTEGER MLDR | dimension of curve approximation criteria data record array |

*Output Parameters:*

| | |
|---|---|
| INTEGER ERRIND | error indicator |
| INTEGER LTYPE | linetype |
| REAL LWIDTH | linewidth scale factor |
| INTEGER CTYPE | colour type |
| INTEGER COLI | colour index |
| INTEGER NCC | number of components of colour value |
| REAL COLR(CCSBSZ) | colour values |
| INTEGER PLSM | polyline shading method |
| INTEGER ACRI | curve approximation criteria type |
| INTEGER LDR | length of data record array |
| CHARACTER*80 DATREC(MLDR) | curve approximation criteria data record[21] |

[21] See SET CURVE APPROXIMATION CRITERIA for a description of the curve approximation criteria data records.

**INQUIRE PREDEFINED POLYMARKER REPRESENTATION PLUS**

SUBROUTINE PQPPMP(WTYPE,PMI,CCSBSZ,ERRIND,MTYPE,MSZSF,
*CTYPE,COLI,NCC,COLR)

*Input Parameters:*

| | |
|---|---|
| INTEGER WTYPE | workstation type |
| INTEGER PMI | predefined polymarker index |
| INTEGER CCSBSZ | colour component specification buffer size |

*Output Parameters:*

| | |
|---|---|
| INTEGER ERRIND | error indicator |
| INTEGER MTYPE | marker type |
| REAL MSZSF | marker scale factor |
| INTEGER CTYPE | colour type |
| INTEGER COLI | colour index |
| INTEGER NCC | number of components of colour value |
| REAL COLR(CCSBSZ) | colour values |

**INQUIRE PREDEFINED TEXT REPRESENTATION PLUS**

SUBROUTINE PQPTXP(WTYPE,PTXI,CCSBSZ,ERRIND,FONT,PREC,CHXP,CHSP,
*CTYPE,COLI,NCC,COLR)

*Input Parameters:*

| | |
|---|---|
| INTEGER WTYPE | workstation type |
| INTEGER PTXI | predefined text index |
| INTEGER CCSBSZ | colour component specification buffer size |

*Output Parameters:*

| | |
|---|---|
| INTEGER ERRIND | error indicator |
| INTEGER FONT | text font |
| INTEGER PREC | text precision(PSTRP,PCHARP,PSTRKP) |
| REAL CHXP | character expansion factor |
| REAL CHSP | character spacing |
| INTEGER CTYPE | colour type |
| INTEGER COLI | colour index |
| INTEGER NCC | number of components of colour value |
| REAL COLR(CCSBSZ) | colour values |

**INQUIRE INTERIOR FACILITIES PLUS**

```
SUBROUTINE PQIFP(WTYPE,NI,NH,NRSM,ERRIND,NIS,IS,NHS,HS,NSM,
*SM,NPFAI)
```

*Input Parameters:*

| | |
|---|---|
| INTEGER WTYPE | workstation type |
| INTEGER NI | list element of interior style requested |
| INTEGER NH | list element of hatch style requested |
| INTEGER NRSM | list element of shading method requested |

*Output Parameters:*

| | |
|---|---|
| INTEGER ERRIND | error indicator |
| INTEGER NIS | number of available interior styles |
| INTEGER IS | element NI of list of available interior styles |
| INTEGER NHS | number of available hatch style |
| INTEGER HS | element NH of list of available hatch styles |
| INTEGER NSM | number of available shading method |
| INTEGER SM | element NRSM of list of available shading method |
| INTEGER NPFAI | number of predefined interior indices |

**INQUIRE PREDEFINED INTERIOR REPRESENTATION PLUS**

```
SUBROUTINE PQPIP(WTYPE,PII,CCSBSZ,ERRIND,INTS,STYLI,
*CTYPE,COLI,NCC,COLR,ISM)
```

*Input Parameters:*

| | |
|---|---|
| INTEGER WTYPE | workstation type |
| INTEGER PII | predefined interior index |
| INTEGER CCSBSZ | colour component specification buffer size |

*Output Parameters:*

| | |
|---|---|
| INTEGER ERRIND | error indicator |
| INTEGER INTS | interior style |
| INTEGER STYLI | interior style index |
| INTEGER CTYPE | colour type |
| INTEGER COLI | colour index |
| INTEGER NCC | number of components of colour value |
| REAL COLR(CCSBSZ) | colour values |
| INTEGER ISM | interior shading method |

**INQUIRE PREDEFINED EDGE REPRESENTATION PLUS**

SUBROUTINE PQPEDP(WTYPE,PEDI,CCSBSZ,ERRIND,EDFLAG,EDTYPE, *EWIDTH,CTYPE,COLI,NCC,COLR)

*Input Parameters:*

| | |
|---|---|
| INTEGER WTYPE | workstation type |
| INTEGER PEDI | predefined edge index |
| INTEGER CCSBSZ | colour component specification buffer size |

*Output Parameters:*

| | |
|---|---|
| INTEGER ERRIND | error indicator |
| INTEGER EDFLAG | edge flag (POFF,PON) |
| INTEGER EDTYPE | edge type |
| REAL EWIDTH | edge scale factor |
| INTEGER CTYPE | colour type |
| INTEGER COLI | colour index |
| INTEGER NCC | number of components of colour value |
| REAL COLR(CCSBSZ) | colour values |

**INQUIRE DATA MAPPING FACILITIES**

SUBROUTINE PQDMF(WTYPE,N,ERRIND,NDM,DM,NPDAI)

*Input Parameters:*

| | |
|---|---|
| INTEGER WTYPE | workstation type |
| INTEGER N | list element requested of data mapping methods |

*Output Parameters:*

| | |
|---|---|
| INTEGER ERRIND | error indicator |
| INTEGER NDM | number of available data mapping methods |
| INTEGER DM | Nth element of list of available data mapping methods |
| INTEGER NPDAI | number of predefined data mapping indices |

## INQUIRE PREDEFINED DATA MAPPING REPRESENTATION

SUBROUTINE PQPDMR(WTYPE,PDMI,MLDR,ERRIND,DMM,LDR,DATREC)

*Input Parameters:*

| | |
|---|---|
| INTEGER WTYPE | workstation type |
| INTEGER PDMI | predefined data mapping index |
| INTEGER MLDR | dimension of data record array |

*Output Parameters:*

| | |
|---|---|
| INTEGER ERRIND | error indicator |
| INTEGER DMM | data mapping method |
| INTEGER LDR | length of data record array |
| CHARACTER*80 DATREC(MLDR) | data mapping data record[22] |

## INQUIRE REFLECTANCE FACILITIES

SUBROUTINE PQRFF(WTYPE,N,NRRFP,ERRIND,NRFM,RFM,NPRFI,NRFP,RFP)

*Input Parameters:*

| | |
|---|---|
| INTEGER WTYPE | workstation type |
| INTEGER N | list element requested of reflectance models |
| INTEGER NRRFP | list of element requested of reflectance properties |

*Output Parameters:*

| | |
|---|---|
| INTEGER ERRIND | error indicator |
| INTEGER NRFM | number of available reflectance model |
| INTEGER RFM | Nth element of list of available reflectance models |
| INTEGER NPRFI | number of predefined reflectance indices |
| INTEGER NRFP | number of available reflectance properties |
| INTEGER RFP | element NRRFP of list of available reflectance properties |

## INQUIRE PREDEFINED REFLECTANCE REPRESENTATION

SUBROUTINE PQPRFR(WTYPE,PRFI,MLDR,ERRIND,RFM,RPTYPE,LDR,DATREC)

*Input Parameters:*

| | |
|---|---|
| INTEGER WTYPE | workstation type |
| INTEGER PRFI | predefined reflectance index |
| INTEGER MLDR | dimension of data record array |

*Output Parameters:*

| | |
|---|---|
| INTEGER ERRIND | error indicator |
| INTEGER RFM | reflectance model |
| INTEGER RPTYPE | reflectance properties type |
| INTEGER LDR | length of data record array |
| CHARACTER*80 DATREC(MLDR) | data record[23] |

[22] See SET DATA MAPPING METHOD for a description of the data mapping data records.

[23] See SET REFLECTANCE PROPERTIES for a description of the reflectance properties data records.

## INQUIRE CURVE AND SURFACE FACILITIES

It has been broken into four related functions: INQUIRE CURVE FACILITIES, INQUIRE PARAMETRIC SURFACE FACILITIES, INQUIRE B-SPLINE SURFACE FACILITIES, and INQUIRE TRIMMING CURVE FACILITIES.

## INQUIRE CURVE FACILITIES

SUBROUTINE PQCVF(WTYPE,N,ERRIND,MNUBSC,NCACT,CACT)

*Input Parameters:*

| | |
|---|---|
| INTEGER WTYPE | workstation type |
| INTEGER N | list element requested of curve approximation criteria types |

*Output Parameters:*

| | |
|---|---|
| INTEGER ERRIND | error indicator |
| INTEGER MNUBSC | maximum non-uniform b-spline curve order supported |
| INTEGER NCACT | number of available curve approximation criteria types |
| INTEGER CACT | Nth element of list of available curve approximation criteria types |

## INQUIRE PARAMETRIC SURFACE FACILITIES

SUBROUTINE PQPSF(WTYPE,N,ERRIND,NPSCT,PSCT,NPPSI)

*Input Parameters:*

| | |
|---|---|
| INTEGER WTYPE | workstation type |
| INTEGER N | list element requested of parametric surface characteristics types |

*Output Parameters:*

| | |
|---|---|
| INTEGER ERRIND | error indicator |
| INTEGER NPSCT | number of available parametric surface characteristics |
| INTEGER PSCT | Nth element of list of available parametric surface characteristics types |
| INTEGER NPPSI | number of predefined parametric surface indices |

**INQUIRE B-SPLINE SURFACE FACILITIES**

SUBROUTINE PQBSSF(WTYPE,N,ERRIND,MNUBSS,NCAST,CAST)

*Input Parameters:*

| | |
|---|---|
| INTEGER WTYPE | workstation type |
| INTEGER N | list element requested of surface approximation criteria types |

*Output Parameters:*

| | |
|---|---|
| INTEGER ERRIND | error indicator |
| INTEGER MNUBSS | maximum non-uniform b-spline surface order supported |
| INTEGER NCAST | number of available surface approximation criteria types |
| INTEGER CAST | Nth element of list of available surface approximation criteria types |

**INQUIRE TRIMMING CURVE FACILITIES**

SUBROUTINE PQTCF(WTYPE,N,ERRIND,MTRIMC,NTCACT,TCACT)

*Input Parameters:*

| | |
|---|---|
| INTEGER WTYPE | workstation type |
| INTEGER N | list element requested of trimming curve approximation criteria types |

*Output Parameters:*

| | |
|---|---|
| INTEGER ERRIND | error indicator |
| INTEGER MTRIMC | maximum trimming curve order supported |
| INTEGER NTCACT | number of available trimming curve approximation criteria types |
| INTEGER TCACT | Nth element of list of available trimming curve approximation criteria types |

## INQUIRE PREDEFINED PARAMETRIC SURFACE REPRESENTATION

SUBROUTINE PQPPSR(WTYPE,PPSI,MLDRSA,MLDRPS,ERRIND,ACRI,LDRSA,
*DARECS,PSCH,LDRPS,DARECP)

*Input Parameters:*

| | |
|---|---|
| INTEGER WTYPE | workstation type |
| INTEGER PPSI | predefined parametric surface bundle index |
| INTEGER MLDRSA | dimension of surface approximation criteria data record array |
| INTEGER MLDRPS | dimension of parametric surface characteristics data record array |

*Output Parameters:*

| | |
|---|---|
| INTEGER ERRIND | error indicator |
| INTEGER ACRI | surface approximation criteria type |
| INTEGER LDRSA | length of surface approximation criteria data record |
| CHARACTER*80 DARECS(MLDRSA) | surface approximation criteria data record[24] |
| INTEGER PSCH | parametric surface characteristics type |
| INTEGER LDRPS | length of parametric surface characteristics data record |
| CHARACTER*80 DARECP(MLDRPS) | parametric surface characteristics data record[25] |

## INQUIRE PREDEFINED PATTERN REPRESENTATION PLUS

SUBROUTINE PQPPAP(WTYPE,PAI,CCSBSZ,DIMX,DIMY,ERRIND,
*DX,DY,CTYPE,COLIA,NCC,COLRA)

*Input Parameters:*

| | |
|---|---|
| INTEGER WTYPE | workstation type |
| INTEGER PAI | predefined pattern index |
| INTEGER CCSBSZ | colour component specification buffer size |
| INTEGER DIMX,DIMY | maximum pattern array dimension |

*Output Parameters:*

| | |
|---|---|
| INTEGER ERRIND | error indicator |
| INTEGER DX,DY | pattern array dimensions |
| INTEGER CTYPE | colour type |
| INTEGER COLIA(DIMX,DIMY) | colour index |
| INTEGER NCC | number of components of colour value specification |
| REAL COLRA(CCSBSZ) | colour values |



[24] See SET SURFACE APPROXIMATION CRITERIA for a description of the surface approximation criteria data records.

[25] See SET PARAMETRIC SURFACE CHARACTERISTICS for a description of the parametric surface characteristics data records.

## INQUIRE LIGHT SOURCE FACILITIES

SUBROUTINE PQLSF(WTYPE,N,ERRIND,NLSTYP,LSTYP,MSNALS,NPLSI)

*Input Parameters:*

| | |
|---|---|
| INTEGER WTYPE | workstation type |
| INTEGER N | list element requested of light source indices |

*Output Parameters:*

| | |
|---|---|
| INTEGER ERRIND | error indicator |
| INTEGER NLSTYP | number of available light source types |
| INTEGER LSTYP | Nth element of list of available light source types |
| INTEGER MSNALS | maximum number of simultaneously active non-ambient light source |
| INTEGER NPLSI | number of predefined light source indices |

## INQUIRE PREDEFINED LIGHT SOURCE REPRESENTATION

SUBROUTINE PQPLSR(WTYPE,PLSI,MLDR,ERRIND,LSTYP,LDR,DATREC)

*Input Parameters:*

| | |
|---|---|
| INTEGER WTYPE | workstation type |
| INTEGER PLSI | predefined light source index |
| INTEGER MLDR | dimension of data record array |

*Output Parameters:*

| | |
|---|---|
| INTEGER ERRIND | error indicator |
| INTEGER LSTYP | light source type |
| INTEGER LDR | length of data record array |
| CHARACTER*80 DATREC(MLDR) | light source data record[26] |

## INQUIRE DEPTH CUE FACILITIES

SUBROUTINE PQDCF(WTYPE,N,ERRIND,NPDCI,NDCM,DCM)

*Input Parameters:*

| | |
|---|---|
| INTEGER WTYPE | workstation type |
| INTEGER N | list element requested of depth cue modes |

*Output Parameters:*

| | |
|---|---|
| INTEGER ERRIND | error indicator |
| INTEGER NPDCI | number of predefined depth cue indices |
| INTEGER NDCM | number of depth cue modes |
| INTEGER DCM | Nth element of list of depth cue mode |

[26] See SET LIGHT SOURCE REPRESENTATION for a description of the light source data record.

**INQUIRE PREDEFINED DEPTH CUE REPRESENTATION**

SUBROUTINE PQPDCR(WTYPE,PDCI,CCSBSZ,ERRIND,DCMODE,DQMIN,DQMAX, *DCSFMI,DCSFMX,CTYPE,COLI,NCC,COLR)

*Input Parameters:*

| | |
|---|---|
| INTEGER WTYPE | workstation type |
| INTEGER PDCI | predefined depth cue index |
| INTEGER CCSBSZ | colour component specification buffer size |

*Output Parameters:*

| | |
|---|---|
| INTEGER ERRIND | error indicator |
| INTEGER DCMODE | depth cue mode(PSUPPR,PALLOW) |
| REAL DQMIN,DQMAX | depth cue reference planes(NPC) |
| REAL DCSFMI,DCSFMX | depth cue scale factors |
| INTEGER CTYPE | colour type |
| INTEGER COLI | colour index |
| INTEGER NCC | number of components of colour value |
| REAL COLR(CCSBSZ) | colour cvalues |

**INQUIRE COLOUR MAPPING FACILITIES**

SUBROUTINE PQCMF(WTYPE,N,ERRIND,NCMTYP,CMTYP,NPCMI)

*Input Parameters:*

| | |
|---|---|
| INTEGER WTYPE | workstation type |
| INTEGER N | list element requested of colour mapping indices |

*Output Parameters:*

| | |
|---|---|
| INTEGER ERRIND | error indicator |
| INTEGER NCMTYP | number of available colour mapping methods |
| INTEGER CMTYP | Nth element of list of available colour mapping methods |
| INTEGER NPCMI | number of predefined colour mapping indices |

## INQUIRE COLOUR MAPPING METHOD FACILITIES

SUBROUTINE PQCMMF(WTYPE,CMM,MLDR,ERRIND,LDR,DATREC)

*Input Parameters:*

| | |
|---|---|
| INTEGER WTYPE | workstation type |
| INTEGER CMM | colour mapping method |
| INTEGER MLDR | dimension of data record array |

*Output Parameters:*

| | |
|---|---|
| INTEGER ERRIND | error indicator |
| INTEGER LDR | length of data record array |
| CHARACTER*80 DATREC(MLDR) | data record |

DATREC parameters to build COLOUR MAPPING METHOD DATA RECORD for CMM=PTRUE:
The IA array contains:
- The number of true colours available
The other array are empty.
IL= 1
IA(1)= number of true colours available
RL= 0
RA= ( )
SL= 0
LSTR= ( )
STR= ( )

DATREC parameters to build COLOUR MAPPING METHOD DATA RECORD for CMM=PSUD:
The IA array contains:
- The maximum number of pseudo colour entries
The other array are empty.
IL= 1
IA(1)= maximum number of pseudo colour entries
RL= 0
RA= ( )
SL= 0
LSTR= ( )
STR= ( )

DATREC parameters to build COLOUR MAPPING METHOD DATA RECORD for CMM = PSUDN: No data record

**INQUIRE PREDEFINED COLOUR MAPPING REPRESENTATION**

SUBROUTINE PQPCMR(WTYPE,PCMI,MLDR,ERRIND,CMM,LDR,DATREC)

*Input Parameters:*

| | |
|---|---|
| INTEGER WTYPE | workstation type |
| INTEGER PCMI | predefined colour mapping index |
| INTEGER MLDR | dimension of data record array |

*Output Parameters:*

| | |
|---|---|
| INTEGER ERRIND | error indicator |
| INTEGER CMM | colour mapping method |
| INTEGER LDR | length of data record array |
| CHARACTER*80 DATREC(MLDR) | data record[27] |

**INQUIRE WORKSTATION STATE TABLE LENGTHS PLUS**

SUBROUTINE PQWSLP(WTYPE,ERRIND,MDMBTE,MRFBTE,MPSBTE,MLSTE,
*MDCTE,MCMTE)

*Input Parameters:*

| | |
|---|---|
| INTEGER WTYPE | workstation type |

*Output Parameters:*

| | |
|---|---|
| INTEGER ERRIND | error indicator |
| INTEGER MDMBTE | maximum number of data mapping bundle table entries |
| INTEGER MRFBTE | maximum number of reflectance bundle table entries |
| INTEGER MPSBTE | maximum number of parametric surface bundle table entries |
| INTEGER MLSTE | maximum number of light source table entries |
| INTEGER MDCTE | maximum number of depth cue table entries |
| INTEGER MCMTE | maximum number of colour mapping table entries |

**INQUIRE GENERALIZED DRAWING PRIMITIVE 3**

SUBROUTINE PQGDP3(WTYPE,GDP,ERRIND,NBND,BNDL)

*Input Parameters:*

| | |
|---|---|
| INTEGER WTYPE | workstation type |
| INTEGER GDP | 3D GDP identifier |

*Output Parameters:*

| | |
|---|---|
| INTEGER ERRIND | error indicator |
| INTEGER NBND | number of sets of attributes used |
| INTEGER BNDL(7) | list of sets of attributes used (*PPLATT,PPMATT,PTXATT,PINATT,PEDATT,*[28] PRFATT,PPSATT) |

This function is defined in PHIGS part 1. Additional attributes are defined in PHIGS part 4. This binding shows those additional attributes.

---

[27] See SET COLOUR MAPPING REPRESENTATION for a description of the colour mapping data record.

[28] already defined in PHIGS

## INQUIRE GENERALIZED DRAWING PRIMITIVE

SUBROUTINE PQGDP(WTYPE,GDP,ERRIND,NBND,BNDL)

*Input Parameters:*

| | |
|---|---|
| INTEGER WTYPE | workstation type |
| INTEGER GDP | GDP identifier |

*Output Parameters:*

| | |
|---|---|
| INTEGER ERRIND | error indicator |
| INTEGER NBND | number of sets of attributes used |
| INTEGER BNDL(7) | list of sets of attributes used<br>*(PPLATT,PPMATT,PTXATT,PINATT,PEDATT,*[29]<br>PRFATT,PPSATT) |

This function is defined in PHIGS part 1. Additional attributes are defined in PHIGS part 4. This binding shows those additional attributes.



[29] already defined in PHIGS

**INQUIRE CURRENT ELEMENT TYPE AND SIZE**

SUBROUTINE PQCETS(ERRIND,ELTYPE,IL,RL,SL)

*Output Parameters:*

| | |
|---|---|
| INTEGER ERRIND | error indicator |
| INTEGER ELTYPE | element type<br>*(PENIL,PEPL3,PEPL,PEPM3,PEPM,PETX3,PETX,*<br>*PEATR3,PEATR,PEFA3,PEFA,PEFAS3,PEFAS,*<br>*PECA3,PECA,PEGDP3,PEGDP,PEPLI,PEPMI,*<br>*PETXI,PEII,PEEDI,PELN,PELWSC,PEPLCI,*<br>*PEMK,PEMKSC,PEPMCI,PETXFN,PETXPR,*<br>*PECHXP,PECHSP,PETXCI,PECHHI,PECHUP,*<br>*PETXP,PETXAL,PEATCH,PEATCV,PEATP,*<br>*PEATAL,PEANST,PEIS,PEISI,PEICI,PEEDFG,*<br>*PEEDT,PEEWSC,PEEDCI,PEPA,PEPRPV,*<br>*PEPARF,PEADS,PERES,PEIASF,*<br>*PEHRID,PELMT3,PELMT,PEGMT3,PEGMT,*<br>*PEMCV3,PEMCV,PEMCLI,PERMCV,PEVWI,*<br>*PEEXST,PELB,PEAP,PEGSE,PEPKID,*[30]<br>PEPLS3,PEPS3D,PEFSD,PECA3P,PESFS3,<br>PESFSD,PETS3D,PETSD,PETST3,PETSTD,<br>PEQM3D,PEQMD,PEBC3,PEBC3C,PEBS3,<br>PEBS3D,PEDMI,PERFI,PEBII,PEBDMI,<br>PEBRFI,PEPRSI,PEPLC,PEPLSM,PEPMC,<br>PETXC,PEFDM,PEFCM,PEIC,PEISM,<br>PEDMM,PERFP,PERFM,PEBIS,PEBISI,<br>PEBIC,PEBISM,PEBDMM,PEBRFP,PEBRFM,<br>PELSS,PEEDC,PECAC,PESAC,PEPSC,<br>PERCM,PEDPCI,PECMI) |
| INTEGER IL | dimension of integer array |
| INTEGER RL | dimension of real array |
| INTEGER SL | dimension of character array |

This function is defined in PHIGS part 1. Additional element types are defined in PHIGS part 4. This binding shows those additional element types.



[30] already defined in PHIGS

## INQUIRE ELEMENT TYPE AND SIZE

SUBROUTINE PQETS(STRID,ELENUM,ERRIND,ELTYPE,IL,RL,SL)

| | |
|---|---|
| *Input Parameters:* | |
| INTEGER STRID | structure identifier |
| INTEGER ELENUM | element number |
| *Output Parameters:* | |
| INTEGER ERRIND | error indicator |
| INTEGER ELTYPE | element type<br>*(PENIL,PEPL3,PEPL,PEPM3,PEPM,PETX3,PETX,*<br>*PEATR3,PEATR,PEFA3,PEFA,PEFAS3,PEFAS,*<br>*PECA3,PECA,PEGDP3,PEGDP,PEPLI,PEPMI,*<br>*PETXI,PEII,PEEDI,PELN,PELWSC,PEPLCI,*<br>*PEMK,PEMKSC,PEPMCI,PETXFN,PETXPR,*<br>*PECHXP,PECHSP,PETXCI,PECHH,PECHUP,*<br>*PETXP,PETXAL,PEATCH,PEATCU,PEATP,*<br>*PEATAL,PEANST,PEIS,PEISI,PEICI,PEEDFG,*<br>*PEEDT,PEEWSC,PEEDCI,PEPA,PEPRPV,*<br>*PEPARF,PEADS,PERES,PEIASF,*<br>*PEHRID,PELMT3,PELMT,PEGMT3,PEGMT,*<br>*PEMCV3,PEMCV,PEMCLI,PERMCV,PEVWI,*<br>*PEEXST,PELB,PEAP,PEGSE,PEPKID,*[31]<br>PEPLS3,PEFS3D,PEFSD,PECA3P,PESFS3,<br>PESFSD,PETS3D,PETSD,PETST3,PETSTD,<br>PEQM3D,PEQMD,PEBC3,PEBC3C,PEBS3,<br>PEBS3D,PEDMI,PERFI,PEBII,PEBDMI,<br>PEBRFI,PEPRSI,PEPLC,PEPLSM,PEPMC,<br>PETXC,PEFDM,PEFCM,PEIC,PEISM,<br>PEDMM,PERFR,PERFM,PEBIS,PEBISI,<br>PEBIC,PEBISM,PEBDMM,PEBRFP,PEBRFM,<br>PELSS,PEEDC,PECAC,PESAC,PEPSC,<br>PERCM,PEDPCI,PECMI) |
| INTEGER IL | dimension of integer array |
| INTEGER RL | dimension of real array |
| INTEGER SL | dimension of character array |

This function is defined in PHIGS part 1. Additional element types are defined in PHIGS part 4. This binding shows those additional element types.

[31] already defined in PHIGS

## INQUIRE ELEMENT CONTENT

SUBROUTINE PQECO(STRID,ELENUM,IIL,IRL,ISL,ERRIND,IL,IA,RL,RA,SL,
*LSTR,STR)

*Input Parameters:*

| | |
|---|---|
| INTEGER STRID | structure identifier |
| INTEGER ELENUM | element position |
| INTEGER IIL | dimension of integer array |
| INTEGER IRL | dimension of real array |
| INTEGER ISL | dimension of character array |

*Output Parameters:*

| | |
|---|---|
| INTEGER ERRIND | error indicator |
| INTEGER IL | number of integer entries |
| INTEGER IA(IIL) | array of containing integer entries |
| INTEGER RL | number of real entries |
| REAL RA(IRL) | array of containing real entries |
| INTEGER SL | number of character string entries |
| INTEGER LSTR(ISL) | length of each character string entry |
| CHARACTER*80 STR(ISL) | character string entries[32] |

This function is defined in PHIGS part 1. Additional attributes are defined in PHIGS part 4. This binding shows those additional element types.



[32]See INQUIRE CURRENT ELEMENT CONTENT for details on interpreting the information returned in the integer, real and character arrays.

## INQUIRE CURRENT ELEMENT CONTENT

SUBROUTINE PQCECO(IIL,IRL,ISL,ERRIND,IL,IA,RL,RA,SL,LSTR,STR)

*Input Parameters:*

| | |
|---|---|
| INTEGER IIL | dimension of integer array |
| INTEGER IRL | dimension of real array |
| INTEGER ISL | dimension of character array |

*Output Parameters:*

| | |
|---|---|
| INTEGER ERRIND | error indicator |
| INTEGER IL | number of integer entries |
| INTEGER IA(IIL) | array of containing integer entries |
| INTEGER RL | number of real entries |
| REAL RA(IRL) | array of containing real entries |
| INTEGER SL | number of character string entries |
| INTEGER LSTR(ISL) | length of each character string entry |
| CHARACTER*80 STR(ISL) | character string entries |

Output parameters for STRUCTURE CONTENT DATA RECORD for
ELTYPE = PEPLS3:
The IA array contain:
- The start element number and the end element number of following data
  vertex colour indices
  vertex colour components
  If the optional data is not present, the value of the "start element number" is set to zero and the value of the "end element number" is set to -1.
- data per vertex flag
- The colour type
- The number of components of colour value
- The number of point list
- The array of end indices for point lists
- The vertex colour indices

The RA array contains:
- The coordinates of points
- The vertex colour components

IL= 8+IA(8)+(IA(2)-IA(1)+1)
IA(1)= start element number in IA of vertex colour indices
IA(2)= end element number in IA of vertex colour indices
IA(3)= start element number in RA of vertex colour components
IA(4)= end element number in RA of vertex colour components
IA(5)= data per vertex flag(PCD,PCDC)
IA(6)= colour type
IA(7)= number of colour components of colour value
IA(8)= number of point lists in polyline set 3 with colour
IA= element 9 through IA(8)+8 contain the array of end indices for point lists in polyline set 3 with data
IA=element IA(1) through IA(2) contain the vertex colour indices
RL= IA(IA(8)+8)*3
RA=element 1 through IA(IA(8)+8)*3 contain the coordinates of points

This function is defined in PHIGS part 1. Additional attributes are defined in PHIGS part 4. This binding shows those additional element types.

example x1,x2,...y1,y2,...z1,z2...
RA=element IA(3) through IA(4) contain the vertex colour components
example(colour type =PRGB) r1,g1,b1,r2,g2,b2,.....
SL= 0
LSTR= ( )
STR= ( )

Output parameters for STRUCTURE CONTENT DATA RECORD for
ELTYPE = PEFS3D:
The IA array contains:
- The start element number and the end element number of following data
  edge data
  vertex colour indices
  vertex colour components
  vertex normal data
  vertex application-specific data
  If the optional data is not present, the value of the "start element number" is set to zero and the value of the "end element number" is set to -1.
- The data per facet flag
- The data per edge flag
- The data per vertex flag
- The colour type
- The number of components of colour value(NCC)
- The number of point lists
- The length of application-specific data list per facet
- The length of application-specific data list per vertex
- The array of end indices for point lists
- The facet colour indices
- The edge data
- The vertex colour indices
The RA array contains:
- The coordinates of points
- The facet colour components
- The facet normal data
- The facet application-specific data
- The vertex colour components
- The vertex normal data
- The vertex application-specific data

IL=19+IA(16)+(IA(2)-IA(1)+1)+(IA(4)-IA(3)+1)
IA(1)= start element number in IA of edge data
IA(2)= end element number in IA of edge data
IA(3)= start element number in IA of vertex colour indices
IA(4)= end element number in IA of vertex colour indices
IA(5)= start element number in RA of vertex colour components
IA(6)= end element number in RA of vertex colour components
IA(7)= start element number in RA of vertex normal data
IA(8)= end element number in RA of vertex normal data
IA(9)= start element number in RA of vertex application-specific data
IA(10)= end element number in RA of vertex application-specific data
IA(11)= data per facet flag(PFNO,PFC,PFN,PFD,PFCN,PFCD,PFND,PFCND)
IA(12)= data per edge flag(PENO,PEVF)
IA(13)= data per vertex flag
(PCD,PCDC,PCDN,PCDD,PCDCN,PCDCD,PCDND,PCDCND)

IA(14)= colour type
IA(15)= number of components of colour value = NCC
IA(16)= number of point lists in fill area set 3 with data
IA(17)= length of application-specific data list per facet
IA(18)= length of application-specific data list per vertex
IA(19)= facet colour index
IA=element 20 through IA(16)+19 contain the array of end indices for point lists in fill area set 3 with data
IA=element IA(1) through IA(2) contain the edge data
IA=element IA(3) through IA(4) contain the vertex colour indices
RL=IA(IA(16)+19)*3+IA(15)+3+IA(17)+(IA(6)-IA(5)+1)+(IA(8)-IA(7)+1)+
(IA(10)-IA(9)-1)
RA=element 1 through IA(IA(16)+19)*3=NUM1 contain the coordinates of points
example x1,x2,...y1,y2,...z1,z2...
RA=element NUM1+1 through NUM1+NCC=NUM2 contain the facet colour components
example(colour type =PRGB) r1,g1,b1,r2,g2,b2,.....
RA=element NUM2+1 through NUM2+3=NUM3 contain the facet normal data
RA=element NUM3+1 through NUM3+IA(17) contain the facet application-specific data
RA=element IA(5) through IA(6) contain the vertex colour components
RA=element IA(7) through IA(8) contain the vertex normal data
example x1,x2,...y1,y2,...z1,z2...
RA=element IA(9) through IA(10) contain the vertex application-specific data
SL= 0
LSTR= ( )
STR= ( )

Output parameters for STRUCTURE CONTENT DATA RECORD for
ELTYPE = PEFSD:
The IA array contains:
- The start element number and the end element number of following data
  - edge data
  - vertex colour indices
  - vertex colour components
  - vertex normal data
  - vertex application-specific data

  If the optional data is not present, the value of the "start element number" is set to zero and the value of the "end element number" is set to -1.
- The data per facet flag
- The data per edge flag
- The data per vertex flag
- The colour type
- The number of components of colour value(NCC)
- The number of point lists
- The length of application-specific data list per facet
- The length of application-specific data list per vertex
- The array of end indices for point lists
- The facet colour indices
- The edge data
- The vertex colour indices

The RA array contains:
- The coordinates of points
- The facet colour components
- The facet normal data
- The facet application-specific data

- The vertex colour components
- The vertex normal data
- The vertex application-specific data

IL=19+IA(16)+(IA(2)-IA(1)+1)+(IA(4)-IA(3)+1)
IA(1)= start element number in IA of edge data
IA(2)= end element number in IA of edge data
IA(3)= start element number in IA of vertex colour indices
IA(4)= end element number in IA of vertex colour indices
IA(5)= start element number in RA of vertex colour components
IA(6)= end element number in RA of vertex colour components
IA(7)= start element number in RA of vertex normal data
IA(8)= end element number in RA of vertex normal data
IA(9)= start element number in RA of vertex application-specific data
IA(10)= end element number in RA of vertex application-specific data
IA(11)= data per facet flag(PFNO,PFC,PFN,PFD,PFCN,PFCD,PFND,PFCND)
IA(12)= data per edge flag(PENO,PEVF)
IA(13)= data per vertex flag
(PCD,PCDC,PCDN,PCDD,PCDCN,PCDCD,PCDND,PCDCND)
IA(14)= colour type
IA(15)= number of components of colour value = NCC
IA(16)= number of point lists in fill area set with data
IA(17)= length of application-specific data list per facet
IA(18)= length of application-specific data list per vertex
IA(19)= facet colour indices
IA=element 20 through IA(16)+19 contain the array of end indices for point lists in fill area set with data
IA=element IA(1) through IA(2) contain the edge data
IA=element IA(3) through IA(4) contain the vertex colour indices
RL=IA(IA(16)+19)*2+IA(15)+IA(17)+(IA(6)-IA(5)+1)+(IA(8)-IA(7)+1)+
(IA(10)-IA(9)-1)
RA=element 1 through IA(IA(16)+19)*2=NUM1 contain the coordinates of points
example x1,x2,...y1,y2...
RA=element NUM1+1 through NUM1+NCC=NUM2 contain the facet colour components
example(colour type = PRGB) r1,g1,b1,r2,g2,b2,.....
RA=element NUM2+1 through NUM2+3=NUM3 contain the facet normal data
RA=element NUM3+1 through NUM3+IA(17) contain the facet application-specific data
RA=element IA(5) through IA(6) contain the vertex colour components
RA=element IA(7) through IA(8) contain the vertex normal data
example x1,x2,...y1,y2,...z1,z2...
RA=element IA(9) through IA(10) contain the vertex application-specific data
SL= 0
LSTR= ( )
STR= ( )

Output parameters for STRUCTURE CONTENT DATA RECORD for
ELTYPE = PECA3P:
The IA array contains:
- The number of columns(NC)
- The number of rows(NR)
- The colour type(CT)
- The number of components of colour value
- The list of colour indices. There are NC*NR colour indices, if CT is indirect.

The RA array contains:

- The cell parallelogram(P)
- The cell parallelogram(Q)
- The cell parallelogram(R)
- The list of colour values. There are NCC*NC*NR colour values, that having as many components as required by CT.

Common element:
IL= 3+*
IA(1)= number of columns = NC
IA(2)= number of rows = NR
IA(3)= colour type = CT
IA(4)= number of components of colour value = NCC
RL= 9+*
RA=element 1 through 3 contain the cell parallelogram(P)
RA=element 4 through 6 contain the cell parallelogram(Q)
RA=element 7 through 9 contain the cell parallelogram(R)
SL= 0
LSTR= ( )
STR= ( )
Individual element:
CASE colour type=PINDIR:
IA=element 4 through NC*NR+3 contain colour indices
CASE colour type=PRGB or PCIE or PHSV or PHLS:
RA=element 10 through NCC*NC*NR+9 contain the colour components

Output parameters for STRUCTURE CONTENT DATA RECORD for
ELTYPE = PESFS3:
The IA array contains:
- The start element number and the end element number of following data
  - facet colour indices
  - edge data
  - vertex colour indices
  - facet colour components
  - facet normal data
  - facet application-specific data
  - vertex colour components
  - vertex normal data
  - vertex application-specific data

  If the optional data is not present, the value of the "start element number" is set to zero and the value of the "end element number" is set to -1.
- The data per facet flag
- The data per edge flag
- The data per vertex flag
- The colour type
- The number of components of colour value
- The number of fill area sets
- The number of points
- The length of application-specific data list per facet
- The length of application-specific data list per vertex
- The array of end indices of fill area indices for each fill area set
- The array of end indices of vertex indices for each fill area
- The vertex indices
- The facet colour indices
- The edge data

- The vertex colour indices

The RA array contains:

- The coordinates of points
- The facet colour components
- The facet normal data
- The facet application-specific data
- The vertex colour components
- The vertex normal data
- The vertex application-specific data

IL= 27+(IA(2)-IA(1)+1)+(IA(4)-IA(3)+1)+(IA(6)-IA(5)+1)+IA(24)+IA(IA(24)+27)
+IA(IA(24)+27)+IA(27+IA(24)+IA(27+IA(24)))

IA(1)= start element number in IA of facet colour indices
IA(2)= end element number in IA of facet colour indices
IA(3)= start element number in IA of edge data
IA(4)= end element number in IA of edge data
IA(5)= start element number in IA of vertex colour indices
IA(6)= end element number in IA of vertex colour indices
IA(7)= start element number in RA of facet colour components
IA(8)= end element number in RA of facet colour components
IA(9)= start element number in RA of facet normal data
IA(10)= end element number in RA of facet normal data
IA(11)= start element number in RA of facet application-specific data
IA(12)= end element number in RA of facet application-specific data
IA(13)= start element number in RA of vertex colour components
IA(14)= end element number in RA of vertex colour components
IA(15)= start element number in RA of vertex normal data
IA(16)= end element number in RA of vertex normal data
IA(17)= start element number in RA of vertex application-specific data
IA(18)= end element number in RA of vertex application-specific data
IA(19)= data per facet flag(PFNO,PFC,PFN,PFD,PFCN,PFCD,PFND,PFCND)
IA(20)= data per edge flag(PENO,PEVF)
IA(21)= data per vertex flag
(PCD,PCDC,PCDN,PCDD,PCDCN,PCDCD,PCDND,PCDCND)
IA(22)= colour type
IA(23)= number of components of colour value
IA(24)= number of fill area sets in set of fill area sets 3 with data
IA(25)= number of points in set of fill area sets 3 with data
IA(26)= length of application-specific data list per facet
IA(27)= length of application-specific data list per vertex
IA=element 28 through IA(24)+27 contain the array of end indices of vertex indices
for each fill area set in set of fill area sets 3 with data
IA=element IA(24)+28 through IA(24)+27+IA(IA(24)+27)=NUM1 contain the array of
end indices for vertex indices in for each fill area in set of fill area sets 3 with data
IA=element NUM1+1 through IA(NUM1)+NUM1 contain the vertex indices
IA=element IA(1) through IA(2) contain the facet colour indices
IA=element IA(3) through IA(4) contain the edge data
IA=element IA(5) through IA(6) contain the vertex colour indices
RL= IA(25)*3+(IA(8)-IA(7)+1)+(IA(10)-IA(9)+1)+(IA(12)-IA(11)+1)+
(IA(14)-IA(13)+1)+(IA(16)-IA(15)+1)+(IA(18)-IA(17)+1)
RA=element 1 through IA(25)*3 contain the coordinates of points
example x1,x2,...y1,y2,...z1,z2...
RA=element IA(7) through IA(8) contain the facet colour components
example(colour type =PRGB) r1,g1,b1,r2,g2,b2,.....

RA=element IA(9) through IA(10) contain the facet normal data
example x1,x2,...y1,y2,...z1,z2...
RA=element IA(11) through IA(12) contain the facet application-specific data
RA=element IA(13) through IA(14) contain the vertex colour components
RA=element IA(15) through IA(16) contain the vertex normal data
RA=element IA(17) through IA(18) contain the vertex application-specific data
SL= 0
LSTR= ( )
STR= ( )

Output parameters for STRUCTURE CONTENT DATA RECORD for
ELTYPE = PESFSD:
The IA array contains:
- The start element number and the end element number of following data
  facet colour indices
  edge data
  vertex colour indices
  facet colour components
  facet normal data
  facet application-specific data
  vertex colour components
  vertex normal data
  vertex application-specific data
  If the optional data is not present, the value of the "start element number" is set to zero and the value of the "end element number" is set to -1.
- The data per facet flag
- The data per edge flag
- The data per vertex flag
- The colour type
- The number of components of colour value
- The number of fill area sets
- The number of points
- The length of application-specific data list per facet
- The length of application-specific data list per vertex
- The array of end indices of vertex indices for each fill area set
- The array of end indices of vertex indices for each fill area
- The vertex indices
- The facet colour indices
- The edge data
- The vertex colour indices

The RA array contains:
- The coordinates of points
- The facet colour components
- The facet normal data
- The facet application-specific data
- The vertex colour components
- The vertex normal data
- The vertex application-specific data

IL= 27+(IA(2)-IA(1)+1)+(IA(4)-IA(3)+11)+(IA(6)-IA(5)+1)+IA(24)+IA(IA(24)+27)
+IA(27+IA(24)+IA(27+IA(24))
IA(1)= start element number in IA of facet colour indices
IA(2)= end element number in IA of facet colour indices
IA(3)= start element number in IA of edge data

IA(4)= end element number in IA of edge data
IA(5)= start element number in IA of vertex colour indices
IA(6)= end element number in IA of vertex colour indices
IA(7)= start element number in RA of facet colour components
IA(8)= end element number in RA of facet colour components
IA(9)= start element number in RA of facet normal data
IA(10)= end element number in RA of facet normal data
IA(11)= start element number in RA of facet application-specific data
IA(12)= end element number in RA of facet application-specific data
IA(13)= start element number in RA of vertex colour components
IA(14)= end element number in RA of vertex colour components
IA(15)= start element number in RA of vertex normal data
IA(16)= end element number in RA of vertex normal data
IA(17)= start element number in RA of vertex application-specific data
IA(18)= end element number in RA of vertex application-specific data
IA(19)= data per facet flag(PFNO,PFC,PFN,PFD,PFCN,PFCD,PFND,PFCND)
IA(20)= data per edge flag(PENO,PEVF)
IA(21)= data per vertex flag
(PCD,PCDC,PCDN,PCDD,PCDCN,PCDCD,PCDND,PCDCND)
IA(22)= colour type
IA(23)= number of components of colour value
IA(24)= number of fill area sets in set of fill area sets with data
IA(25)= number of points in set of fill area sets with data
IA(26)= length of application-specific data list per facet
IA(27)= length of application-specific data list per vertex
IA=element 28 through IA(24)+27 contain the array of end indices of vertex indices for each fill area set in set of fill area sets with data
IA=element IA(24)+28 through IA(24)+27+IA(IA(24)+27)=NUM1 contain the array of end indices for vertex indices in set of fill area sets with data
IA=element NUM1+1 through IA(NUM1)+NUM1 contain the vertex indices
IA=element IA(1) through IA(2) contain the facet colour indices
IA=element IA(3) through IA(4) contain the edge data
IA=element IA(5) through IA(6) contain the vertex colour indices
RL= IA(25)*2+(IA(8)-IA(7)+1)+(IA(10)-IA(9)+1)+(IA(12)-IA(11)+1)+(IA(14)-IA(13)+1)+(IA(16)-IA(15)+1)+(IA(18)-IA(17)+1)
RA=element 1 through IA(25)*2 contain the coordinates of points
example x1,x2,...y1,y2,...
RA=element IA(7) through IA(8) contain the facet colour components
example(colour type =PRGB) r1,g1,b1,r2,g2,b2,.....
RA=element IA(9) through IA(10) contain the facet normal data
example x1,x2,...y1,y2,...z1,z2...
RA=element IA(11) through IA(12) contain the facet application-specific data
RA=element IA(13) through IA(14) contain the vertex colour components
RA=element IA(15) through IA(16) contain the vertex normal data
RA=element IA(17) through IA(18) contain the vertex application-specific data
SL= 0
LSTR= ( )
STR= ( )

Output parameters for STRUCTURE CONTENT DATA RECORD for
ELTYPE = PETS3D:
The IA array contains:
- The start element number and the end element number of following data
facet colour indices

edge data
vertex colour indices
facet colour components
facet normal data
facet application-specific data
vertex colour components
vertex normal data
vertex application-specific data

If the optional data is not present, the value of the "start element number" is set to zero and the value of the "end element number" is set to -1.

- The data per facet flag
- The data per edge flag
- The data per vertex flag
- The colour type
- The number of components of colour value
- The number of triangles
- The number of points
- The length of application-specific data list per facet
- The length of application-specific data list per vertex
- The vertex indices
- The facet colour indices
- The edge data
- The vertex colour indices

The RA array contains:

- The coordinates of points
- The facet colour components
- The facet normal data
- The facet application-specific data
- The vertex colour components
- The vertex normal data
- The vertex application-specific data

IL=27+3*IA(24)+(IA(2)-IA(1)+1)+(IA(4)-IA(3)-1)+(IA(6)-IA(5)+1)
IA(1)= start element number in IA of facet colour indices
IA(2)= end element number in IA of facet colour indices
IA(3)= start element number in IA of edge data
IA(4)= end element number in IA of edge data
IA(5)= start element number in IA of vertex colour indices
IA(6)= end element number in IA of vertex colour indices
IA(7)= start element number in RA of facet colour components
IA(8)= end element number in RA of facet colour components
IA(9)= start element number in RA of facet normal data
IA(10)= end element number in RA of facet normal data
IA(11)= start element number in RA of facet application-specific data
IA(12)= end element number in RA of facet application-specific data
IA(13)= start element number in RA of vertex colour components
IA(14)= end element number in RA of vertex colour components
IA(15)= start element number in RA of vertex normal data
IA(16)= end element number in RA of vertex normal data
IA(17)= start element number in RA of vertex application-specific data
IA(18)= end element number in RA of vertex application-specific data
IA(19)= data per facet flag(PFNO,PFC,PFN,PFD,PFCN,PFCD,PFND,PFCND)
IA(20)= data per edge flag(PENO,PEVF)
IA(21)= data per vertex flag

(PCD,PCDC,PCDN,PCDD,PCDCN,PCDCD,PCDND,PCDCND)
IA(22)= colour type
IA(23)= number of components of colour value
IA(24)= number of triangles
IA(25)= number of points in triangle set 3 with data
IA(26)= length of application-specific data list per facet
IA(27)= length of application-specific data list per vertex
IA=element 28 through 3*IA(24)+27 contain vertex indices
IA=element IA(1) through IA(2) contain the facet colour indices
IA=element IA(3) through IA(4) contain the edge data
IA=element IA(5) through IA(6) contain the vertex colour indices
RL= IA(25)*3+(IA(8)-IA(7)+1)+(IA(10)-IA(9)+1)+(IA(12)-IA(11)+1)+
(IA(14)-IA(13)+1)+(IA(16)-IA(15)+1)+(IA(18)-IA(17)+1)
RA=element 1 through IA(25)*3 contain the coordinates of points
example x1,x2,...y1,y2,...z1,z2...
RA=element IA(7) through IA(8) contain the facet colour components
example(colour type =PRGB) r1,g1,b1,r2,g2,b2,.....
RA=element IA(9) through IA(10) contain the facet normal data
example x1,x2,...y1,y2,...z1,z2...
RA=element IA(11) through IA(12) contain the facet application-specific data
RA=element IA(13) through IA(14) contain the vertex colour components
RA=element IA(15) through IA(16) contain the vertex normal data
RA=element IA(17) through IA(18) contain the vertex application-specific data
SL= 0
LSTR= ( )
STR= ( )

Output parameters for STRUCTURE CONTENT DATA RECORD for
ELTYPE = PETSD:
The IA array contains:
- The start element number and the end element number of following data
facet colour indices
edge data
vertex colour indices
facet colour components
facet normal data
facet application-specific data
vertex colour components
vertex normal data
vertex application-specific data
If the optional data is not present, the value of the "start element number"
is set to zero and the value of the "end element number" is set to -1.
- The data per facet flag
- The data per edge flag
- The data per vertex flag
- The colour type
- The number of components of colour value
- The number of triangles
- The number of points
- The length of application-specific data list per facet
- The length of application-specific data list per vertex
- The vertex indices
- The facet colour indices
- The edge data

- The vertex colour indices
The RA array contains:
- The coordinates of points
- The facet colour components
- The facet normal data
- The facet application-specific data
- The vertex colour components
- The vertex normal data
- The vertex application-specific data

IL=27+3*IA(24)+(IA(2)-IA(1)+1)+(IA(4)-IA(3)+1)+(IA(6)-IA(5)+1)
IA(1)= start element number in IA of facet colour indices
IA(2)= end element number in IA of facet colour indices
IA(3)= start element number in IA of edge data
IA(4)= end element number in IA of edge data
IA(5)= start element number in IA of vertex colour indices
IA(6)= end element number in IA of vertex colour indices
IA(7)= start element number in RA of facet colour components
IA(8)= end element number in RA of facet colour components
IA(9)= start element number in RA of facet normal data
IA(10)= end element number in RA of facet normal data
IA(11)= start element number in RA of facet application-specific data
IA(12)= end element number in RA of facet application-specific data
IA(13)= start element number in RA of vertex colour components
IA(14)= end element number in RA of vertex colour components
IA(15)= start element number in RA of vertex normal data
IA(16)= end element number in RA of vertex normal data
IA(17)= start element number in RA of vertex application-specific data
IA(18)= end element number in RA of vertex application-specific data
IA(19)= data per facet flag(PFNO,PFC,PFN,PFD,PFCN,PFCD,PFND,PFCND)
IA(20)= data per edge flag(PENO,PEVF)
IA(21)= data per vertex flag
(PCD,PCDC,PCDN,PCDD,PCDCN,PCDCD,PCDND,PCDCND)
IA(22)= colour type
IA(23)= number of components of colour value
IA(24)= number of triangles
IA(25)= number of points in triangle set 3 with data
IA(26)= length of application-specific data list per facet
IA(27)= length of application-specific data list per vertex
IA=element 28 through 3*IA(24)+27 contain vertex indices
IA=element IA(1) through IA(2) contain the facet colour indices
IA=element IA(3) through IA(4) contain the edge data
IA=element IA(5) through IA(6) contain the vertex colour indices
RL= IA(25)*2+(IA(8)-IA(7)+1)+(IA(10)-IA(9)+1)+(IA(12)-IA(11)+1)+
(IA(14)-IA(13)+1)+(IA(16)-IA(15)+1)+(IA(18)-IA(17)+1)
RA=element 1 through IA(25)*2 contain the coordinates of points
example x1,x2,...y1,y2,...
RA=element IA(7) through IA(8) contain the facet colour components
example(colour type =PRGB) r1,g1,b1,r2,g2,b2,.....
RA=element IA(9) through IA(10) contain the facet normal data
example x1,x2,...y1,y2,...z1,z2...
RA=element IA(11) through IA(12) contain the facet application-specific data
RA=element IA(13) through IA(14) contain the vertex colour components
RA=element IA(15) through IA(16) contain the vertex normal data

RA=element IA(17) through IA(18) contain the vertex application-specific data
SL= 0
LSTR= ( )
STR= ( )

Output parameters for STRUCTURE CONTENT DATA RECORD for
ELTYPE = PETST3:
The IA array contains:
- The start element number and the end element number of following data
    facet colour indices
    edge data
    vertex colour indices
    facet colour components
    facet normal data
    facet application-specific data
    vertex colour components
    vertex normal data
    vertex application-specific data

  If the optional data is not present, the value of the "start element number" is set to zero and the value of the "end element number" is set to -1.
- The data per facet flag
- The data per edge flag
- The data per vertex flag
- The colour type
- The number of components of colour value
- The number of points
- The length of application-specific data list per facet
- The length of application-specific data list per vertex
- The facet colour indices
- The edge data
- The vertex colour indices

The RA array contains:
- The coordinates of points
- The facet colour components
- The facet normal data
- The facet application-specific data
- The vertex colour components
- The vertex normal data
- The vertex application-specific data

IL=26+(IA(2)-IA(1)+1)+(IA(4)-IA(3)+1)+(IA(6)-IA(5)+1)
IA(1)= start element number in IA of facet colour indices
IA(2)= end element number in IA of facet colour indices
IA(3)= start element number in IA of edge data
IA(4)= end element number in IA of edge data
IA(5)= start element number in IA of vertex colour indices
IA(6)= end element number in IA of vertex colour indices
IA(7)= start element number in RA of facet colour components
IA(8)= end element number in RA of facet colour components
IA(9)= start element number in RA of facet normal data
IA(10)= end element number in RA of facet normal data
IA(11)= start element number in RA of facet application-specific data
IA(12)= end element number in RA of facet application-specific data
IA(13)= start element number in RA of vertex colour components

IA(14)= end element number in RA of vertex colour components
IA(15)= start element number in RA of vertex normal data
IA(16)= end element number in RA of vertex normal data
IA(17)= start element number in RA of vertex application-specific data
IA(18)= end element number in RA of vertex application-specific data
IA(19)= data per facet flag(PFNO,PFC,PFN,PFD,PFCN,PFCD,PFND,PFCND)
IA(20)= data per edge flag(PENO,PEVF)
IA(21)= data per vertex flag
(PCD,PCDC,PCDN,PCDD,PCDCN,PCDCD,PCDND,PCDCND)
IA(22)= colour type
IA(23)= number of components of colour value
IA(24)= number of points in triangle strip 3 with data
IA(25)= length of application-specific data list per facet
IA(26)= length of application-specific data list per vertex
IA=element IA(1) through IA(2) contain the facet colour indices
IA=element IA(3) through IA(4) contain the edge data
IA=element IA(5) through IA(6) contain the vertex colour indices
RL= IA(24)*3+(IA(8)-IA(7)+1)+(IA(10)-IA(9)+1)+(IA(12)-IA(11)+1)+
(IA(14)-IA(13)+1)+(IA(16)-IA(15)+1)+(IA(18)-IA(17)+1)
RA=element 1 through IA(24)*3 contain the coordinates of points
example x1,x2,...y1,y2,...z1,z2...
RA=element IA(7) through IA(8) contain the facet colour components
example(colour type =PRGB) r1,g1,b1,r2,g2,b2,.....
RA=element IA(9) through IA(10) contain the facet normal data
example x1,x2,...y1,y2,...z1,z2...
RA=element IA(11) through IA(12) contain the facet application-specific data
RA=element IA(13) through IA(14) contain the vertex colour components
RA=element IA(15) through IA(16) contain the vertex normal data
RA=element IA(17) through IA(18) contain the vertex application-specific data
SL= 0
LSTR= ( )
STR= ( )

Output parameters for STRUCTURE CONTENT DATA RECORD for
ELTYPE = PETSTD:
The IA array contains:
- The start element number and the end element number of following data
facet colour indices
edge data
vertex colour indices
facet colour components
facet normal data
facet application-specific data
vertex colour components
vertex normal data
vertex application-specific data
If the optional data is not present, the value of the "start element number"
is set to zero and the value of the "end element number" is set to -1.
- The data per facet flag
- The data per edge flag
- The data per vertex flag
- The colour type
- The number of components of colour value
- The number of points

- The length of application-specific data list per facet
- The length of application-specific data list per vertex
- The facet colour indices
- The edge data
- The vertex colour indices

The RA array contains:

- The coordinates of points
- The facet colour components
- The facet normal data
- The facet application-specific data
- The vertex colour components
- The vertex normal data
- The vertex application-specific data

IL=26+(IA(2)-IA(1)+1)+(IA(4)-IA(3)+1)+(IA(6)-IA(5)+1)
IA(1)= start element number in IA of facet colour indices
IA(2)= end element number in IA of facet colour indices
IA(3)= start element number in IA of edge data
IA(4)= end element number in IA of edge data
IA(5)= start element number in IA of vertex colour indices
IA(6)= end element number in IA of vertex colour indices
IA(7)= start element number in RA of facet colour components
IA(8)= end element number in RA of facet colour components
IA(9)= start element number in RA of facet normal data
IA(10)= end element number in RA of facet normal data
IA(11)= start element number in RA of facet application-specific data
IA(12)= end element number in RA of facet application-specific data
IA(13)= start element number in RA of vertex colour components
IA(14)= end element number in RA of vertex colour components
IA(15)= start element number in RA of vertex normal data
IA(16)= end element number in RA of vertex normal data
IA(17)= start element number in RA of vertex application-specific data
IA(18)= end element number in RA of vertex application-specific data
IA(19)= data per facet flag(PFNO,PFC,PFN,PFD,PFCN,PFCD,PFND,PFCND)
IA(20)= data per edge flag(PENO,PEVF)
IA(21)= data per vertex flag
(PCD,PCDC,PCDN,PCDD,PCDCN,PCDCD,PCDND,PCDCND)
IA(22)= colour type
IA(23)= number of components of colour value
IA(24)= number of points in triangle strip 3 with data
IA(25)= length of application-specific data list per facet
IA(26)= length of application-specific data list per vertex
IA=element IA(1) through IA(2) contain the facet colour indices
IA=element IA(3) through IA(4) contain the edge data
IA=element IA(5) through IA(6) contain the vertex colour indices
RL= IA(24)*2+(IA(8)-IA(7)+1)+(IA(10)-IA(9)+1)+(IA(12)-IA(11)+1)+
(IA(14)-IA(13)+1)+(IA(16)-IA(15)+1)+(IA(18)-IA(17)+1)
RA=element 1 through IA(24)*2 contain the coordinates of points
example x1,x2,...y1,y2,...
RA=element IA(7) through IA(8) contain the facet colour components
example(colour type =PRGB) r1,g1,b1,r2,g2,b2,.....
RA=element IA(9) through IA(10) contain the facet normal data
example x1,x2,...y1,y2,...z1,z2...
RA=element IA(11) through IA(12) contain the facet application-specific data

RA=element IA(13) through IA(14) contain the vertex colour components
RA=element IA(15) through IA(16) contain the vertex normal data
RA=element IA(17) through IA(18) contain the vertex application-specific data
SL= 0
LSTR= ( )
STR= ( )

Output parameters for STRUCTURE CONTENT DATA RECORD for
ELTYPE = PEQM3D:
The IA array contains:
- The start element number and the end element number of following data
  - facet colour indices
  - edge data
  - vertex colour indices
  - facet colour components
  - facet normal data
  - facet application-specific data
  - vertex colour components
  - vertex normal data
  - vertex application-specific data

  If the optional data is not present, the value of the "start element number" is set to zero and the value of the "end element number" is set to -1.
- The data per facet flag
- The data per edge flag
- The data per vertex flag
- The colour type
- The number of components of colour value
- The columns number of point dimension
- The rows number of point dimension
- The length of application-specific data list per facet
- The length of application-specific data list per vertex
- The facet colour indices
- The edge data
- The vertex colour indices

The RA array contains:
- The coordinates of points
- The facet colour components
- The facet normal data
- The facet application-specific data
- The vertex colour components
- The vertex normal data
- The vertex application-specific data

IL= 27+(IA(2)-IA(1)+1)+(IA(4)-IA(3)+1)+(IA(6)-IA(5)+1)
IA(1)= start element number in IA of facet colour indices
IA(2)= end element number in IA of facet colour indices
IA(3)= start element number in IA of edge data
IA(4)= end element number in IA of edge data
IA(5)= start element number in IA of vertex colour indices
IA(6)= end element number in IA of vertex colour indices
IA(7)= start element number in RA of facet colour components
IA(8)= end element number in RA of facet colour components
IA(9)= start element number in Ra of facet normal data
IA(10)= end element number in RA of facet normal data

IA(11)= start element number in RA of facet application-specific data
IA(12)= end element number in RA of facet application-specific data
IA(13)= start element number in RA of vertex colour components
IA(14)= end element number in RA of vertex colour components
IA(15)= start element number in RA of vertex normal data
IA(16)= end element number in RA of vertex normal data
IA(17)= start element number in RA of vertex application-specific data
IA(18)= end element number in RA of vertex application-specific data
IA(19)= data per facet flag(PFNO,PFC,PFN,PFD,PFCN,PFCD,PFND,PFCND)
IA(20)= data per edge flag(PENO,PEVF)
IA(21)= data per vertex flag
(PCD,PCDC,PCDN,PCDD,PCDCN,PCDCD,PCDND,PCDCND)
IA(22)= colour type
IA(23)= number of components of colour value
IA(24)= columns number of points dimension
IA(25)= rows number of points dimension
IA(26)= length of application-specific data list per facet
IA(27)= length of application-specific data list per vertex
IA=element IA(1) through IA(2) contain the facet colour indices
IA=element IA(3) through IA(4) contain the edge data
IA=element IA(5) through IA(6) contain the vertex colour indices
RL=IA(24)*IA(25)*3+(IA(8)-IA(7)+1)+(IA(10)-IA(9)+1)+(IA(12)-IA(11)+1)+
(IA(14)-IA(13)+1)+(IA(16)-IA(15)+1)+(IA(18)-IA(17)+1)
RA=element 1 through IA(24)*IA(25)*3 contain the coordinates of points
example x1,x2,...y1,y2,...z1,z2...
RA=element IA(7) through IA(8) contain the facet colour components
example(colour type =PRGB) r1,g1,b1,r2,g2,b2,.....
RA=element IA(9) through IA(10) contain the facet normal data
example x1,x2,...y1,y2,...z1,z2...
RA=element IA(11) through IA(12) contain the facet application-specific data
RA=element IA(13) through IA(14) contain the vertex colour components
RA=element IA(15) through IA(16) contain the vertex normal data
RA=element IA(17) through IA(18) contain the vertex application-specific data
SL= 0
LSTR= ( )
STR= ( )

Output parameters for STRUCTURE CONTENT DATA RECORD for
ELTYPE = PEQMD:
The IA array contains:
- The start element number and the end element number of following data
facet colour indices
edge data
vertex colour indices
facet colour components
facet normal data
facet application-specific data
vertex colour components
vertex normal data
vertex application-specific data
If the optional data is not present, the value of the "start element number"
is set to zero and the value of the "end element number" is set to -1.
- The data per facet flag
- The data per edge flag

- The data per vertex flag
- The colour type
- The number of components of colour value
- The columns number of point dimension
- The rows number of point dimension
- The length of application-specific data list per facet
- The length of application-specific data list per vertex
- The facet colour indices
- The edge data
- The vertex colour indices

The RA array contains:

- The coordinates of points
- The facet colour components
- The facet normal data
- The facet application-specific data
- The vertex colour components
- The vertex normal data
- The vertex application-specific data

IL= 27+(IA(2)-IA(1)+1)+(IA(4)-IA(3)+1)+(IA(6)-IA(5)+1)
IA(1)= start element number in IA of facet colour indices
IA(2)= end element number in IA of facet colour indices
IA(3)= start element number in IA of edge data
IA(4)= end element number in IA of edge data
IA(5)= start element number in IA of vertex colour indices
IA(6)= end element number in IA of vertex colour indices
IA(7)= start element number in RA of facet colour components
IA(8)= end element number in RA of facet colour components
IA(9)= start element number in RA of facet normal data
IA(10)= end element number in RA of facet normal data
IA(11)= start element number in RA of facet application-specific data
IA(12)= end element number in RA of facet application-specific data
IA(13)= start element number in RA of vertex colour components
IA(14)= end element number in RA of vertex colour components
IA(15)= start element number in RA of vertex normal data
IA(16)= end element number in RA of vertex normal data
IA(17)= start element number in RA of vertex application-specific data
IA(18)= end element number in RA of vertex application-specific data
IA(19)= data per facet flag(PFNO,PFC,PFN,PFD,PFCN,PFCD,PFND,PFCND)
IA(20)= data per edge flag(PENO,PEVF)
IA(21)= data per vertex flag
(PCD,PCDC,PCDN,PCDD,PCDCN,PCDCD,PCDND,PCDCND)
IA(22)= colour type
IA(23)= number of components of colour value
IA(24)= columns number of points dimension
IA(25)= rows number of points dimension
IA(26)= length of application-specific data list per facet
IA(27)= length of application-specific data list per vertex
IA=element IA(1) through IA(2) contain the facet colour indices
IA=element IA(3) through IA(4) contain the edge data
IA=element IA(5) through IA(6) contain the vertex colour indices
RL=IA(24)*IA(25)*2+(IA(8)-IA(7)+1)+(IA(10)-IA(9)+1)+(IA(12)-IA(11)+1)+
(IA(14)-IA(13)+1)+(IA(16)-IA(15)+1)+(IA(18)-IA(17)+1)
RA=element 1 through IA(23)*IA(24)*2 contain the coordinates of points

example x1,x2,...y1,y2,...
RA=element IA(7) through IA(8) contain the facet colour components
example(colour type =PRGB) r1,g1,b1,r2,g2,b2,.....
RA=element IA(9) through IA(10) contain the facet normal data
example x1,x2,...y1,y2,...z1,z2...
RA=element IA(11) through IA(12) contain the facet application-specific data
RA=element IA(13) through IA(14) contain the vertex colour components
RA=element IA(15) through IA(16) contain the vertex normal data
RA=element IA(17) through IA(18) contain the vertex application-specific data
SL= 0
LSTR= ( )
STR= ( )

Output parameters for STRUCTURE CONTENT DATA RECORD for
ELTYPE = PEBC3:
The IA array contains:
- The spline order
- The number of knots
- The rationality
- The number of control points
The RA array contains:
- The knots
- The parameter range limits
- The coordinates of points
If "rationality" is PNRAT, coordinates of points are PXA, PYA and PZA.
If "rationality" is PRAT, coordinates of points are PXA, PYA, PZA and PWA.

IL= 4
IA(1)= spline order
IA(2)= number of knots
IA(3)= rationality(PRAT,PNRAT)
IA(4)= number of control points
RL= IA(2)+2+*
RA=element 1 through IA(2) contain the knots
RA=element IA(2)+1 through IA(2)+2 contain the parameter range limits
first minimum parameter, second maximum parameter.
if rationality=PRAT
RA=element IA(2)+4 through IA(2)+2+IA(4)*4 contain the coordinates of points
example x1,x2,...y1,y2,...z1,z2,...w1,w2,...
if rationality=PNRAT
RA=element IA(2)+3 through IA(2)+2+IA(4)*3 contain the coordinates of points
example x1,x2,...y1,y2,...z1,z2,...
SL= 0
LSTR= ( )
STR= ( )

Output parameters for STRUCTURE CONTENT DATA RECORD for
ELTYPE = PEBC3C:
The IA array contains:
- The spline order
- The number of knots
- The rationality
- The number of control points
The RA array contains:

- The knots
- The parameter range limits
- The coordinates of points
  If "rationality" is PNRAT, coordinates of points are PXA, PYA and PZA.
  If "rationality" is PRAT, coordinates of points are PXA, PYA, PZA and PWA.

The STR array contains:
- The colour spline curve data record.

IL= 4
IA(1)= spline order
IA(2)= number of knots
IA(3)= rationality(PRAT,PNRAT)
IA(4)= number of control points
RL= IA(2)+2+*
RA=element 1 through IA(2) contain the knots
RA=element IA(2)+1 through IA(2)+2 contain the parameter range limits
if rationality=PRAT
  RA=element IA(2)+4 through IA(2)+2+IA(4)*4 contain the coordinates of points
    example x1,x2,...y1,y2,...z1,z2,...w1,w2,...
if rationality=PNRAT
  RA=element IA(2)+3 through IA(2)+2+IA(4)*3 contain the coordinates of points
    example x1,x2,...y1,y2,...z1,z2,...
SL= number of array elements used in data record
For i= 1 to SL
    LSTR(i)= 80
STR= colour spline curve data record

Output parameters for STRUCTURE CONTENT DATA RECORD for
ELTYPE = PEBS3:
The IA array contains:
- The u spline order
- The v spline order
- The u number of knots
- The v number of knots
- The rationality
- The u number of control points
- The v number of control points

The RA array contains:
- The u knots
- The v knots
- The coordinates of points
  If "rationality" is PNRAT, coordinates of points are PXA, PYA and PZA.
  If "rationality" is PRAT, coordinates of points are PXA, PYA, PZA and PWA.

The STR array contains:
- The trimming curve data record

IL=7
IA(1)= u spline order
IA(2)= v spline order
IA(3)= number of knots in u
IA(4)= number of knots in v
IA(5)= rationality(PRAT,PNRAT)
IA(6)= u number of control points dimension
IA(7)= v number of control points dimension

RL= IA(3)+IA(4)+*
RA=element 1 through IA(3) contain the u knots
RA=element IA(3)+1 through IA(4)+IA(3) contain the v knots
if rationality=PRAT
  RA=element IA(4)+IA(3)+1 through 4*IA(6)*IA(7)+IA(4)+IA(3) contain
        the coordinates of points
    example x1,x2,...y1,y2,...z1,z2,...w1,w2,...
if rationality=PNRAT
  RA=element IA(4)+IA(3)+1 through 3*IA(6)*IA(7)+IA(4)+IA(3) contain
        the coordinates of points
    example x1,x2,...y1,y2,...z1,z2,...
SL= number of array elements used in trimming curve data record
For i=1 to SL
  LSTR(i)= 80
STR= trimming curve data record

Output parameters for STRUCTURE CONTENT DATA RECORD for
ELTYPE = PEBS3D:
The IA array contains:
- The u spline order
- The v spline order
- The u number of knots
- The v number of knots
- The rationality
- The u number of control points
- The v number of control points
The RA array contains:
- The u knots
- The v knots
- The coordinates of points
  If "rationality" is PNRAT, coordinates of points are PXA, PYA and PZA.
  If "rationality" is PRAT, coordinates of points are PXA, PYA, PZA and PWA.
The STR array contains:
- The trimming curve data record
- The colour spline surface data record
- The data spline surface data record

IL=7
IA(1)= spline order u
IA(2)= spline order v
IA(3)= number of knots in u
IA(4)= number of knots in v
IA(5)= rationality(PRAT,PNRAT)
IA(6)= u number of control points dimension
IA(7)= v number of control points dimension
RL= IA(3)+IA(4)+*
RA=element 1 through IA(3) contain the u knots
RA=element IA(3)+1 through IA(4)+IA(3) contain the v knots
if rationality=PRAT
  RA=element IA(4)+IA(3)+1 through 4*IA(6)*IA(7)+IA(4)+IA(3) contain
        the coordinates of points
    example x1,x2,...y1,y2,...z1,z2,...w1,w2,...
if rationality=PNRAT
  RA=element IA(4)+IA(3)+1 through 3*IA(6)*IA(7)+IA(4)+IA(3) contain

the coordinates of points
example x1,x2,...y1,y2,...z1,z2,...
SL= number of array elements used in trimming curve data record +
number of array elements used in colour spline surface data record
number of array elements used in data spline surface data record
For i=1 to SL
LSTR(i)= 80
STR= trimming curve data record, colour spline surface data record and
data spline surface data record

Output parameters for STRUCTURE CONTENT DATA RECORD for
ELTYPE = PEDMI:
The IA array contains:
- The data mapping index
The other arrays are empty.

IL= 1
IA(1)= data mapping index
RL= 0
RA= ( )
SL= 0
LSTR= ( )
STR= ( )

Output parameters for STRUCTURE CONTENT DATA RECORD for
ELTYPE = PERFI:
The IA array contains:
- The reflectance index
The other arrays are empty.

IL= 1
IA(1)= reflectance index
RL= 0
RA= ( )
SL= 0
LSTR= ( )
STR= ( )

Output parameters for STRUCTURE CONTENT DATA RECORD for
ELTYPE = PEBII:
The IA array contains:
- The back interior index
The other arrays are empty.

IL= 1
IA(1)= back interior index
RL= 0
RA= ( )
SL= 0
LSTR= ( )
STR= ( )

Output parameters for STRUCTURE CONTENT DATA RECORD for
ELTYPE = PEBDMI:

The IA array contains:
- The back data mapping index
The other arrays are empty.

IL= 1
IA(1)= back data mapping index
RL= 0
RA= ( )
SL= 0
LSTR= ( )
STR= ( )

Output parameters for STRUCTURE CONTENT DATA RECORD for ELTYPE = PEBRFI:
The IA array contains:
- The back reflectance index
The other arrays are empty.

IL= 1
IA(1)= back reflectance index
RL= 0
RA= ( )
SL= 0
LSTR= ( )
STR= ( )

Output parameters for STRUCTURE CONTENT DATA RECORD for ELTYPE = PEPRSI:
The IA array contains:
- The parametric surface index
The other arrays are empty.

IL= 1
IA(1)= parametric surface index
RL= 0
RA= ( )
SL= 0
LSTR= ( )
STR= ( )

Output parameters for STRUCTURE CONTENT DATA RECORD for ELTYPE = PEPLC:
The IA array contains:
- The colour type(CT)
- The number of components of colour value(NCC)
- The actual data of colour index, if CT is indirect.
The RA array contains:
- The actual data of colour component values as required by CT.

Common element:
IL= 2+*
IA(1)= colour type = CT
IA(2)= number of components of colour value = NCC
RL= *

SL= 0
LSTR= ( )
STR= ( )
Individual element:
  CASE colour type=PINDIR:
    IA(3)= colour index
  CASE colour type=PRGB or PCIE or PHSV or PHLS:
    RA= element 1 through NCC contain the colour components

Output parameters for STRUCTURE CONTENT DATA RECORD for ELTYPE = PEPLSM:
The IA array contains:
- The polyline shading method
The other arrays are empty.

IL= 1
IA(1)= polyline shading method
RL= 0
RA= ( )
SL= 0
LSTR= ( )
STR= ( )

Output parameters for STRUCTURE CONTENT DATA RECORD for ELTYPE = PEPMC:
The IA array contains:
- The colour type(CT)
- The number of components of colour value(NCC)
- The actual data of colour index, if CT is indirect.
The RA array contains:
- The actual data of colour component values as required by CT.

Common element:
IL= 2+*
IA(1)= colour type = CT
IA(2)= number of components of colour value
RL= *
SL= 0
LSTR= ( )
STR= ( )
Individual element:
  CASE colour type=PINDIR:
    IA(3)= colour index
  CASE colour type=PRGB or PCIE or PHSV or PHLS:
    RA= element 1 through NCC contain the colour components

Output parameters for STRUCTURE CONTENT DATA RECORD for ELTYPE = PETXC:
The IA array contains:
- The colour type(CT)
- The number of components of colour value(NCC)
- The actual data of colour index, if CT is indirect.
The RA array contains:
- The actual data of colour component values as required by CT.

Common element:
IL= 2+*
IA(1)= colour type = CT
IA(2)= number of components of colour value
RL= *
SL= 0
LSTR= ( )
STR= ( )
Individual element:
CASE colour type=PINDIR:
IA(3)= colour index
CASE colour type=PRGB or PCIE or PHSV or PHLS:
RA= element 1 through NCC contain the colour components

Output parameters for STRUCTURE CONTENT DATA RECORD for ELTYPE = PEFDM:
The IA array contains:
- The distinguishing mode
The other arrays are empty.

IL= 1
IA(1)= distinguishing mode(POFF,PON)
RL= 0
RA= ( )
SL= 0
LSTR= ( )
STR= ( )

Output parameters for STRUCTURE CONTENT DATA RECORD for ELTYPE = PEFCM:
The IA array contains:
- The culling mode
The other arrays are empty.

IL= 1
IA(1)= culling mode(PNOFC,PBKFC,PFTFC)
RL= 0
RA= ( )
SL= 0
LSTR= ( )
STR= ( )

Output parameters for STRUCTURE CONTENT DATA RECORD for ELTYPE = PEIC:
The IA array contains:
- The colour type(CT)
- The number of components of colour value(NCC)
- The actual data of colour index, if CT is indirect.
The RA array contains:
- The actual data of colour component values as required by CT.

Common element:
IL= 2+*
IA(1)= colour type = CT

IA(2)= number of components of colour value
RL= *
SL= 0
LSTR= ( )
STR= ( )
Individual element:
  CASE colour type=PINDIR:
    IA(3)= colour index
  CASE colour type=PRGB or PCIE or PHSV or PHLS:
    RA= element 1 through NCC contain the colour components

Output parameters for STRUCTURE CONTENT DATA RECORD for ELTYPE = PEISM:
The IA array contains:
- The interior shading mode
The other arrays are empty.

IL= 1
IA(1)= interior shading method
RL= 0
RA= ( )
SL= 0
LSTR= ( )
STR= ( )

Output parameters for STRUCTURE CONTENT DATA RECORD for ELTYPE = PEDMM:
The IA array contains:
- The data mapping method
The STR array contains:
- The data mapping data record

IL= 1
IA(1)= data mapping method
RL= 0
RA= ( )
SL= length of data record array
FOR i=1 TO length of data record array
LSTR(i)= 80
STR= data mapping data record[33]

Output parameters for STRUCTURE CONTENT DATA RECORD for ELTYPE = PERFP:
The IA array contains:
- The properties type
The STR array contains:
- The reflectance propeties data record

Common element:
IL= 1
IA(1)= properties type

[33] See SET DATA MAPPING METHOD for a description of the data mapping method data data record.

RL= 0
RA( )
SL= length of reflectance properties data record array
For i=1 to SL
LSTR(i)= 80
STR= reflectace propeties data record

Output parameters for STRUCTURE CONTENT DATA RECORD for ELTYPE = PERFM:
The IA array contains:
- The reflectance model
The other arrays are empty.

IL= 1
IA(1)= reflectance model
RL= 0
RA= ( )
SL= 0
LSTR= ( )
STR= ( )

Output parameters for STRUCTURE CONTENT DATA RECORD for ELTYPE = PEBIS:
The IA array contains:
- The interior style
The other arrays are empty.

IL= 1
IA(1)= interior style(PHOLLO,PSOLID,PPATTR,PPATCH,PISEMP)
RL= 0
RA= ( )
SL= 0
LSTR= ( )
STR= ( )

Output parameters for STRUCTURE CONTENT DATA RECORD for ELTYPE = PEBISI:
The IA array contains:
- The interior style index
The other arrays are empty.

IL= 1
IA(1)= interior style index
RL= 0
RA= ( )
SL= 0
LSTR= ( )
STR= ( )

Output parameters for STRUCTURE CONTENT DATA RECORD for ELTYPE = PEBIC:
The IA array contains:
- The colour type(CT)
- The number of components of colour value(NCC)

- The actual data of colour index, if CT is indirect.
The RA array contains:
- The actual data of colour component values as required by CT.

Common element:
IL= 2+*
IA(1)= colour type = CT
IA(2)= number of components of colour value
RL= *
SL= 0
LSTR= ( )
STR= ( )
Individual element:
  CASE colour type=PINDIR:
    IA(3)= colour index
  CASE colour type=PRGB or PCIE or PHSV or PHLS:
    RA= element 1 through NCC contain the colour components

Output parameters for STRUCTURE CONTENT DATA RECORD for
ELTYPE = PEBISM:
The IA array contains:
- The interior shading method
The other arrays are empty.

IL= 1
IA(1)= interior shading method
RL= 0
RA= ( )
SL= 0
LSTR= ( )
STR= ( )

Output parameters for STRUCTURE CONTENT DATA RECORD for
ELTYPE = PEBDMM:
The IA array contains:
- The data mapping method
The STR array contains:
- The data mapping data record

IL= 1
IA(1)= data mapping method
RL= 0
RA= ( )
SL= length of data record array
FOR i=1 TO length of data record array
LSTR(i)= 80
STR= data mapping data record[34]

Output parameters for STRUCTURE CONTENT DATA RECORD for
ELTYPE = PEBRFP:
The IA array contains:

[34] See SET DATA MAPPING METHOD for a description of the data mapping method data data record.

- The properties type
The STR array contains:
- The reflectance propetries data record

IL= 1
IA(1)= properties type
RL= 0
RA( )
SL= length of reflectance properties data record array
For i=1 to SL
LSTR(i)= 80
STR= reflectace propeties data record

Output parameters for STRUCTURE CONTENT DATA RECORD for ELTYPE = PEBRFM:
The IA array contains:
- The reflectance model
The other arrays are empty.

IL= 1
IA(1)= reflectance model
RL= 0
RA= ( )
SL= 0
LSTR= ( )
STR= ( )

Output parameters for STRUCTURE CONTENT DATA RECORD for ELTYPE = PELSS:
The IA array contains:
- The number of activation indices(NAI)
- The list of activation indices
- The number of deactivation indices(NDI)
- The list of deactivation indices
The other arrays are empty.

IL= 1+*
IA(1)= number of activation indices = NAI
IA(2)= number of deactivation indices = NDI
IA= element 3 through NAI+2 contain the activation list
IA= element NAI+3 through NDI+NAI+2 contain the deactivation list
RL= 0
RA= ( )
SL= 0
LSTR= ( )
STR= ( )

Output parameters for STRUCTURE CONTENT DATA RECORD for ELTYPE = PEEDC:
The IA array contains:
- The colour type(CT)
- The number of components of colour value
- The actual data of colour index, if CT is indirect.
The RA array contains:

- The actual data of colour component values as required by CT.

Common element:
IL= 2+*
IA(1)= colour type = CT
IA(2)= number of components of colour value
RL= *
SL= 0
LSTR= ( )
STR= ( )
Individual element:
  CASE colour type=PINDIR:
    IA(3)= colour index
  CASE colour type= PRGB or PCIE or PHSV or PHLS:
    RA= element 1 through NCC contain the colour components

Output parameters for STRUCTURE CONTENT DATA RECORD for
ELTYPE = PECAC:
The IA array contains:
- The curve approximation criteria type
The STR array contains:
- The curve approximation data record

IL= 1
IA(1)= curve approximation criteria type
RL= 0
RA=( )
SL= length of curve approximation data record array
For i= 1 to SL
  LSTR(i)= 80
STR= curve approximation data record

Output parameters for STRUCTURE CONTENT DATA RECORD for
ELTYPE = PESAC:
The IA array contains:
- The surface approximation criteria type
The STR array contains:
- The surface approximation data record

IL= 1
IA(1)= surface approximation criteria type
RL= 0
RA=( )
SL= length of surface approximation data record array
For i= 1 to SL
  LSTR(i)= 80
STR= surface approximation data record

Output parameters for STRUCTURE CONTENT DATA RECORD for
ELTYPE = PEPSC:
The IA array contains:
- The parametric surface characteristics type
The STR array contains:
- The parametric surface characteristics data record

IL= 1
IA(1)= parametric surface characteristics type
RL= 0
RA= ( )
SL= length of data record array
FOR i=1 TO length of data record array
LSTR(i)= 80
STR= parametric surface characteristics data record[35]

Output parameters for STRUCTURE CONTENT DATA RECORD for ELTYPE = PERCM:
The IA array contains:
- The rendering colour model
The other arrays are empty.

IL= 1
IA(1)= rendering colour model
RL= 0
RA= ( )
SL= 0
LSTR= ( )
STR= ( )

Output parameters for STRUCTURE CONTENT DATA RECORD for ELTYPE = PEDPCI:
The IA array contains:
- The depth cue index
The other arrays are empty.

IL= 1
IA(1)= depth cue index
RL= 0
RA= ( )
SL= 0
LSTR= ( )
STR= ( )

Output parameters for STRUCTURE CONTENT DATA RECORD for ELTYPE = PECMI:
The IA array contains:
- The colour mapping index
The other arrays are empty.

IL= 1
IA(1)= colour mapping index
RL= 0
RA= ( )
SL= 0
LSTR= ( )
STR= ( )

[35] See SET PARAMETRIC SURFACE CHARACTERISTICS for a description of the parametric surface characteristics data record.

## ELEMENT SEARCH

SUBROUTINE PELS(STRID,STRTEP,SRCDIR,EISN,EIS,EESN,EES,ERRIND,
*STATUS,FNDEP)

*Input Parameters:*

| | |
|---|---|
| INTEGER STRID | structure identifier |
| INTEGER STRTEP | start element position |
| INTEGER SRCDIR | search direction(PBWD,PFWD) |
| INTEGER EISN | number of elements in element inclusion set |
| INTEGER EIS(EISN) | element inclusion set<br>*(PENIL,PEPL3,PEPL,PEPM3,PEPM,PETX3,PETX,*<br>*PEATR3,PEATR,PEFA3,PEFA,PEFAS3,PEFAS,*<br>*PECA3,PECA,PEGDP3,PEGDP,PEPLI,PEPMI,*<br>*PETXI,PEII,PEEDI,PELN,PELWSC,PEPLCI,*<br>*PEMK,PEMKSC,PEPMCI,PETXFN,PETXPR,*<br>*PECHXP,PECHSP,PETXCI,PECHH,PECHUP,*<br>*PETXP,PETXAL,PEATCH,PEATCD,PEATP,*<br>*PEATAL,PEANST,PEIS,PEISI,PEICI,PEEDFG,*<br>*PEEDT,PEEWSC,PEEDCI,PEPA,PEPRPV,*<br>*PEPARF,PEADS,PERES,PEIASF,*<br>*PEHRID,PELMT3,PELMT,PEGMT3,PEGMT,*<br>*PEMCV3,PEMCV,PEMCLI,PERMCV,PEVWI,*<br>*PEEXST,PELB,PEAP,PEGSE,PEPKID,*[36]<br>PEPLS3,PEFS3D,PEFSD,PECA3P,PESFS3,<br>PESFSD,PETS3D,PETSD,PETST3,PETSTD,<br>PEQM3D,PEQMD,PEBC3,PEBC3C,PEBS3,<br>PEBS3D,PEDMI,PERFI,PEBII,PEBDMI,<br>PEBRFI,PEPRSI,PEPLC,PEPLSM,PEPMC,<br>PETXC,PEFDM,PEFCM,PEIC,PEISM,<br>PEDMM,PERFP,PERFM,PEBIS,PEBISI,<br>PEBIC,PEBISM,PEBDMM,PEBRFP,PEBRFM,<br>PELSS,PEEDC,PECAC,PESAC,PEPSC,<br>PERCM,PEDPCI,PECMI) |
| INTEGER EESN | number of elements in element exclusion set |
| INTEGER EES(EESN) | element exclusion set<br>*(enumerated type the same as that of element inclusion set)* |

*Output Parameters:*

| | |
|---|---|
| INTEGER ERRIND | error indicator |
| INTEGER STATUS | status indicator(PFAIL,PSUCC) |
| INTEGER FNDEP | found element position |

This function is defined in PHIGS part 1. Additional enumeration types are defined in PHIGS part 4. This binding shows those additional enumeration types.

[36] already defined in PHIGS

# 14 Utility functions not defined in PHIGS and PHIGS PLUS

## PACK COLOUR SPLINE CURVE

SUBROUTINE PPCSC (CSORD,CNKA,CKNOTS,CRTYPE,CTYPES,NCC,NCSCP,
*CSCP,MLDR,ERRIND,CSCLDR,CSCREC)

*Input Parameters:*

| | |
|---|---|
| INTEGER CSORD | colour spline order |
| INTEGER CNKA | number of colour spline knots |
| REAL CKNOTS(CNKA) | colour spline knots |
| INTEGER CRTYPE | colour spline rationality (PRAT,PNRAT) |
| INTEGER CTYPES | colour type for colour spline |
| INTEGER NCC | number of colour components (count does not include homogeneous component in rational colour splines) |
| INTEGER NCSCP | number of colour spline control points |
| REAL CSCP(*) | colour spline control points |
| INTEGER MLDR | dimension of colour spline curve data record array |

*Output Parameters:*

| | |
|---|---|
| INTEGER ERRIND | error indicator (zero if no error) |
| INTEGER CSCLDR | number of array elements used in CSCREC |
| CHARACTER*80 CSCREC(MLDR) | colour spline curve data record |

## UNPACK COLOUR SPLINE CURVE

SUBROUTINE PUCSC (CSCLDR,CSCREC,ICNKA,INCSCP,ERRIND,CSORD,CNKA,
*CKNOTS,CRTYPE,CTYPES,NC,NCSCP,CSCP)

*Input Parameters:*

| | |
|---|---|
| INTEGER CSCLDR | number of array elements used in CSCREC |
| CHARACTER*80 CSCREC(CSCLDR) | colour spline curve data record |
| INTEGER ICNKA | dimension of CKNOTS array |
| INTEGER INCSCP | dimension of CSCP array |

*Output Parameters:*

| | |
|---|---|
| INTEGER ERRIND | error indicator (zero if no error) |
| INTEGER CSORD | colour spline order |
| INTEGER CNKA | number of colour spline knots in CKNOTS array |
| REAL CKNOTS(ICNKA) | colour spline knots |
| INTEGER CRTYPE | colour spline rationality (PRAT,PNRAT) |
| INTEGER CTYPES | colour type for colour spline |
| INTEGER NC | number of colour components (count does not include homogeneous component in rational colour splines) |
| INTEGER NCSCP | number of colour spline control points |
| REAL CSCP(INCSCP) | colour spline control points |

**PACK TRIMMING CURVE**

```
SUBROUTINE PPTC (INITFL,LOOPFL,TACRI,LDR,DATREC,TCVF,TSORD,
*TNKA,TKNOTS,TPARL,TRTYPE,NTCCP,TPWUA,TPWVA,TPWWA,MLDR,
*ERRIND,TCLDR,TCREC)
```

*Input Parameters:*

| | |
|---|---|
| LOGICAL INITFL | data record initialization flag |
| LOGICAL LOOPFL | data record new loop flag |
| INTEGER TACRI | trimming curve approximation criteria type |
| INTEGER LDR | length of data record array for trimming curve approximation criteria |
| CHARACTER*80 DATREC(LDR) | data record for trimming curve approximation criteria[37] |
| INTEGER TCVF | trimming curve visibility flag (POFF,PON) |
| INTEGER TSORD | trimming curve spline order |
| INTEGER TNKA | number of spline knots in array |
| REAL TKNOTS(TNKA) | trimming curve spline knots |
| REAL TPARL(2) | trimming curve parameter range limits |
| INTEGER TRTYPE | trimming curve spline rationality (PRAT,PNRAT) |
| INTEGER NTCCP | number of trimming curve spline control points |
| REAL TPWUA(NTCCP) | trimming curve U control point components |
| REAL TPWVA(NTCCP) | trimming curve V control point components |
| REAL TPWWA(NTCCP) | trimming curve W control point components (unused if trimming curve is non-rational) |
| INTEGER MLDR | dimension of trimming curve data record |

*Output Parameters:*

| | |
|---|---|
| INTEGER ERRIND | error indicator (zero if no error) |
| INTEGER TCLDR | total number of array elements used in TCREC |
| CHARACTER*80 TCREC(MLDR) | trimming curve data record |



[37] See SET CURVE APPROXIMATION CRITERIA for description of the curve approximation criteria data recode.

## UNPACK TRIMMING CURVE

```
 SUBROUTINE PUTC(TCLDR,TCREC,ILOOP,ITCRV,MLDR,ITNKA,INTCCP,
*ERRIND,NLOOP,NCURVE,TACRI,LDR,DATREC,TCVF,TSORD,TNKA,TKNOTS,
*TPARL,TRTYPE,NTCCP,TPWUA,TPWVA,TPWWA)
```

*Input Parameters:*

| | |
|---|---|
| INTEGER TCLDR | number of array elements used in TCREC |
| CHARACTER*80 TCREC(TCLDR) | trimming curve data record |
| INTEGER ILOOP | which loop |
| INTEGER ITCRV | which curve in loop |
| INTEGER MLDR | dimension of data record for trimming curve approximation criteria |
| INTEGER ITNKA | dimension of TKNOTS array |
| INTEGER INTCCP | dimension of TPWUA, TPWVA, and TPWWA arrays |

*Output Parameters:*

| | |
|---|---|
| INTEGER ERRIND | error indicator (zero if no error) |
| INTEGER NLOOP | Number of trimming loops |
| INTEGER NCURVE | Number of curves in current loop |
| INTEGER TACRI | trimming curve approximation criteria type |
| INTEGER LDR | number of array elements used in DATREC |
| CHARACTER*80 DATREC(MLDR) | data record for trimming curve approximation criteria[38] |
| INTEGER TCVF | trimming curve visibility flag (POFF,PON) |
| INTEGER TSORD | trimming curve spline order |
| INTEGER TNKA | number of trimming curve spline knots in TKNOTS array |
| REAL TKNOTS(ITNKA) | trimming curve spline knots |
| REAL TPARL(2) | trimming curve parameter range limits |
| INTEGER TRTYPE | trimming curve spline rationality (PRAT,PNRAT) |
| INTEGER NTCCP | number of trimming curve spline control points |
| REAL TPWUA(INTCCP) | trimming curve U control point components |
| REAL TPWVA(INTCCP) | trimming curve V control point components |
| REAL TPWWA(INTCCP) | trimming curve W control point components (unused if trimming curve is non-rational) |



[38] See SET CURVE APPROXIMATION CRITERIA for a description of the curve approximation criteria data recode.

## PACK COLOUR SPLINE SURFACE

```
 SUBROUTINE PPCSS (CUSORD,CVSORD,CUNKA,CVNKA,CUKNTS,CVKNTS,
*CRTYPE,CTYPES,NCC,UNCSCP,VNCSCP,CSCP,MLDR,ERRIND,
*CSSLDR,CSSREC)
```

*Input Parameters:*

| | |
|---|---|
| INTEGER CUSORD | u colour spline order |
| INTEGER CVSORD | v colour spline order |
| INTEGER CUNKA | number of u spline knots in array |
| INTEGER CVNKA | number of v spline knots in array |
| REAL CUKNTS(CUNKA) | u colour spline knots |
| REAL CVKNTS(CVNKA) | v colour spline knots |
| INTEGER CRTYPE | colour spline rationality (PRAT,PNRAT) |
| INTEGER CTYPES | colour type for spline |
| INTEGER NCC | number of colour components<br>(count does not include homogeneous component<br>in rational colour splines) |
| INTEGER UNCSCP | u number of colour spline control points dimension |
| INTEGER VNCSCP | v number of colour spline control points dimension |
| REAL CSCP(*) | colour spline control points |
| INTEGER MLDR | dimension of colour spline surface data record array |

*Output Parameters:*

| | |
|---|---|
| INTEGER ERRIND | error indicator (zero if no error) |
| INTEGER CSSLDR | number of array elements used in CSSREC |
| CHARACTER*80 CSSREC(MLDR) | colour spline surface data record |

**UNPACK COLOUR SPLINE SURFACE**

```
 SUBROUTINE PUCSS (CSSLDR,CSSREC,ICUNKA,ICVNKA,ICNSCP,ERRIND,
*CUSORD,CVSORD,CUNKA,CVNKA,CUKNTS,CVKNTS,CRTYPE,CTYPES,NCC,
*UNCSCP,VNCSCP,CSCP)
```

*Input Parameters:*

| | |
|---|---|
| INTEGER CSSLDR | number of array elements used in CSSREC |
| CHARACTER*80 CSSREC(CSSLDR) | colour spline surface data record |
| INTEGER ICUNKA | dimension of CUKNTS array |
| INTEGER ICVNKA | dimension of CVKNTS array |
| INTEGER INCSCP | dimension of CSCP array |

*Output Parameters:*

| | |
|---|---|
| INTEGER ERRIND | error indicator (zero if no error) |
| INTEGER CUSORD | u colour spline order |
| INTEGER CVSORD | v colour spline order |
| INTEGER CUNKA | number of u spline knots in CUKNTS array |
| INTEGER CVNKA | number of v spline knots in CVKNTS array |
| REAL CUKNTS(ICUNKA) | u colour spline knots |
| REAL CVKNTS(ICVNKA) | v colour spline knots |
| INTEGER CRTYPE | colour spline rationality (PRAT,PNRAT) |
| INTEGER CTYPES | colour type for spline |
| INTEGER NCC | number of colour components (count does not include homogeneous component in rational colour splines) |
| INTEGER UNCSCP | u number of colour spline control points dimension |
| INTEGER VNCSCP | v number of colour spline control points dimension |
| REAL CSCP(INCSCP) | colour spline control points |

**PACK DATA SPLINE SURFACE**

```
 SUBROUTINE PPDSS (INITFL,DATAFL,DUSORD,DVSORD,DUNKA,DVNKA,
*DUKNTS,DVKNTS,DRTYPE,UNDSCP,UNDSCP,DDIMS,DSCP,MLDR,ERRIND,
*DSSLDR,DSSREC)
```

*Input Parameters:*

| | |
|---|---|
| LOGICAL INITFL | data record initialization flag |
| LOGICAL DATAFL | data record new data spline flag |
| INTEGER DUSORD | u data spline order |
| INTEGER DVSORD | v data spline order |
| INTEGER DUNKA | number of u spline knots in DUKNTS array |
| INTEGER DVNKA | number of u spline knots in DVKNTS array |
| REAL DUKNTS(DUNKA) | u data spline knots |
| REAL DVKNTS(DVNKA) | v data spline knots |
| INTEGER DRTYPE | data spline rationality (PRAT,PNRAT) |
| INTEGER UNDSCP | u number of data spline control points dimension |
| INTEGER UNDSCP | v number of data spline control points dimension |
| INTEGER DDIMS | data dimension for spline |
| REAL DSCP(*) | data spline control points |
| INTEGER MLDR | dimension of data spline surface data record array |

*Output Parameters:*

| | |
|---|---|
| INTEGER ERRIND | error indicator (zero if no error) |
| INTEGER DSSLDR | number of array elements used in DSSREC |
| CHARACTER*80 DSSREC(MLDR) | data spline surface data record |

**UNPACK DATA SPLINE SURFACE**

```
 SUBROUTINE PUDSS (DSSLDR,DSSREC,IDS,IDUNKA,IDVNKA,IDNSCP,ERRIND,
*DUSORD,DVSORD,DUNKA,DVNKA,DUKNTS,DVKNTS,DRTYPE,UNDSCP,
*UNDSCP,DDIMS,DSCP)
```

*Input Parameters:*

| | |
|---|---|
| INTEGER DSSLDR | number of array elements used in DSSREC |
| CHARACTER*80 DSSREC(DSSLDR) | data record |
| INTEGER IDS | which data spline |
| INTEGER IDUNKA | dimension of DUKNTS array |
| INTEGER IDVNKA | dimension of DVKNTS array |
| INTEGER IDNSCP | dimension of DSCP array |

*Output Parameters:*

| | |
|---|---|
| INTEGER ERRIND | error indicator (zero if no error) |
| INTEGER DUSORD | u data spline order |
| INTEGER DVSORD | v data spline order |
| INTEGER DUNKA | number of u spline knots in array |
| INTEGER DVNKA | number of v spline knots in array |
| REAL DUKNTS(IDUNKA) | u data spline knots |
| REAL DVKNTS(IDVNKA) | v data spline knots |
| INTEGER DRTYPE | data spline rationality (PRAT,PNRAT) |
| INTEGER UNDSCP | u number of data spline control points dimension |
| INTEGER UNDSCP | v number of data spline control points dimension |
| INTEGER DDIMS | data dimension for spline |
| REAL DSCP(IDNSCP) | data spline control points |

# Annex A
# (informative)
# FORTRAN Examples

Add the following to AnnexA.
The following sample programs, using the PHIGS and PHIGS PLUS FORTRAN binding,illustrate the use of PHIGS and PHIGS PLUS functions.
Example 1:

```
C     PROGRAM SURF
C
C     DESCRIPTION:
C      This program draws a surface hollowed out by trimming curve.
C      Copyright 1992 O'Reilly and Associates, Inc. Permission to use, copy,
C      and modify this program is hereby granted,as long as this
C      copyright notice appears in each copy of the program source code.
C      Program adapted from C to Fortran for use as language binding example.
C
C     Define PHIGS constants.
C
      INTEGER PNRAT,PHOLLO,PUBKCP
      PARAMETER (PNRAT=0,PHOLLO=0,PUBKCP=1)
      INTEGER PICPC,PCBKSA,PON
      PARAMETER (PICPC=3,PCBKSA=2,PON=1)
      INTEGER PPERFO
      PARAMETER (PPERFO=1)
C
C     Define the surface data.
C
      INTEGER USORD, VSORD
      INTEGER UNKA, VNKA
      INTEGER UNCP, VNCP
      INTEGER RTYPE
      REAL UKNOTS(6)/0.0,0.0,0.0,1.0,1.0,1.0/
      REAL VKNOTS(8)/0.0,0.0,0.0,0.0,1.0,1.0,1.0,1.0/
      REAL PXA(12)/0.0,0.3,0.6,0.9,
     -             0.0,0.3,0.6,0.9,
     -             0.0,0.3,0.6,0.9/
      REAL PYA(12)/0.5,0.5,0.4,0.5,
     -             0.6,1.0,0.5,0.3,
     -             0.5,0.5,0.4,0.2/
      REAL PZA(12)/0.0,0.0,0.0,0.0,
     -             0.5,0.5,0.5,0.5,
     -             1.0,1.0,1.0,1.0/
      REAL PWA(12)/12*1.0/
C
C     Define the trimming curve data.
C
      LOGICAL INITFL,LOOPFL
      INTEGER TACRI,TCVF
      INTEGER TSORD,TNKA,TRTYPE,NTCCP
      REAL TKNOTS(12),TPWUA(9),TPWVA(9)
      DATA TKNOTS(1),TKNOTS(2),TKNOTS(3),TKNOTS(4)/0.0,0.0,0.0,1.0/
```

```
      DATA TKNOTS(5),TKNOTS(6),TKNOTS(7),TKNOTS(8)/1.0,2.0,2.0,3.0/
      DATA TKNOTS(9),TKNOTS(10),TKNOTS(11),TKNOTS(12)/3.0,4.0,4.0,4.0/
      DATA TPWUA(1),TPWUA(2),TPWUA(3)/0.5,0.68,0.68/
      DATA TPWUA(4),TPWUA(5),TPWUA(6)/0.68,0.5,0.32/
      DATA TPWUA(7),TPWUA(8),TPWUA(9)/0.32,0.32,0.5/
      DATA TPWVA(1),TPWVA(2),TPWVA(3)/0.32,0.32,0.5/
      DATA TPWVA(4),TPWVA(5),TPWVA(6)/0.68,0.68,0.68/
      DATA TPWVA(7),TPWVA(8),TPWVA(9)/0.5,0.32,0.32/
      REAL TPWWA/0.0/
      REAL TPARL(2)
      INTEGER TCLDR
      CHARACTER*80 TCREC(9)
C
      INTEGER PSCTY
      INTEGER ACRI
      INTEGER ERRIND
      INTEGER IA(3),LDR,MLDR
      CHARACTER*80 DATREC
C
C     Open PHIGS and a workstation.
C
      CALL POPPH(99,-1)
      CALL POPWK(1,0,0)
C
C     Open the structure.
C
      CALL POPST(1)
C
C     Select the view index for the surface.
C
      CALL PSVWI(1)
C
C     Hollow the surface.
C
      CALL PSIS(PHOLLO)
C
C     Set the parametric surface characteristics.
C
      IL    = 3
      IA(1) = PUBKCP
      IA(2) = 20
      IA(3) = 20
      MLDR  = 1
      CALL PPREC(IL,IA,0,0,0,0,0,MLDR,ERRIND,LDR,DATREC)
      PSCTY = PICPC
      CALL PSPSC(PSCTY,LDR,DATREC)
C
C     Set the surface approximation criteria.
C
      IL    = 2
      IA(1) = 10
      IA(2) = 10
      MLDR  = 1
      CALL PPREC(IL,IA,0,0,0,0,0,MLDR,ERRIND,LDR,DATREC)
```

```
      ACRI  = PCBKSA
      CALL PSSAC(ACRI,LDR,DATREC)
C
C     Display the edge of surface.
C
      CALL PSEDFG(PON)
C
C     Set the first trimming  curve
C
      INITFL = .TRUE.
      LOOPFL = .TRUE.
      TACRI  = PCBKCA
      IL     = 1
      IA(1)  = 10
      MLDR   = 1
      CALL PPREC(IL,IA,0,0,0,0,0,MLDR,ERRIND,LDR,DATREC)
      TCVF   = PON
      TSORD  = 3
      TNKA   = 12
      TPARL(1) = 0.0
      TPARL(2) = 4.0
      TRTYPE = PNRAT
      NTCCP  = 9
      MLDR   = 3
      CALL PPTCV(INITFL,LOOPFL,TACRI,LDR,DATREC,TCVF,TSORD,TNKA,
     -           TKNOTS,TPARL,TRTYPE,NTCCP,TPWUA,TPWVA,
     -           TPWWA,MLDR,ERRIND,TCLDR,TCREC)
C
C     Set the second trimming  curve
C
      INITFL = .FALSE.
      LOOPFL = .FALSE.
      TACRI  = PCBKCA
      IL     = 1
      IA(1)  = 10
      MLDR   = 1
      CALL PPREC(IL,IA,0,0,0,0,0,MLDR,ERRIND,LDR,DATREC)
      TCVF   = PON
      TSORD  = 3
      TNKA   = 12
      TKNOTS(1)  = 0.0
      TKNOTS(2)  = 0.0
      TKNOTS(3)  = 0.0
      TKNOTS(4)  = 1.0
      TKNOTS(5)  = 1.0
      TKNOTS(6)  = 2.0
      TKNOTS(7)  = 2.0
      TKNOTS(8)  = 3.0
      TKNOTS(9)  = 3.0
      TKNOTS(10) = 4.0
      TKNOTS(11) = 4.0
      TKNOTS(12) = 4.0
      TPARL(1) = 0.0
      TPARL(2) = 4.0
```