

INTERNATIONAL STANDARD

ISO/IEC
9593-4

First edition
1991-12-15

AMENDMENT 1
1994-05-01

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

Part 4: C

AMENDMENT 1

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

Partie 4: C

AMENDEMENT 1



Reference number
ISO/IEC 9593-4:1991 / Amd. 1:1994 (E)
AVA Documents
Document Sharing Hub

Contents

1 Scope 1

2 Normative references 2

3 The C language binding of PHIGS 3

 3.1 Conformance 3

 3.2 Functions versus macros 3

 3.3 Character strings 3

 3.4 Function identifiers 3

 3.5 Registration 3

 3.6 Identifiers for graphical items 4

 3.7 Return values 4

 3.8 Header files 4

 3.9 Memory management 4

 3.9.1 Inquiry functions which return simple lists 4

 3.9.2 Inquiry functions which return complex data structures 4

 3.9.3 Meaning of the size of an element 4

 3.10 Inquiries returning structure elements 4

 3.11 Error handling 5

 3.11.1 Application defined error handlers 5

 3.11.2 Error codes 5

 3.11.3 C specific PHIGS errors 5

 3.12 Storage of two-dimensional data 5

 3.12.1 Storage of matrices 5

 3.12.2 Storage of colour arrays 6

 3.13 Data type descriptions 6

4 Tables 7

 4.1 Abbreviation policy for construction of identifiers 7

 4.2 Table of abbreviations 7

 4.3 Function names 8

 4.3.1 List ordered alphabetically by bound name 8

 4.3.2 List ordered alphabetically by PHIGS function name 10

5 Type definitions 14

6 Macro definitions 15

7 C PHIGS functions 16

8 C PHIGS PLUS type definitions 17

 8.1 Mapping of PHIGS PLUS data types 17

 8.2 Modifications to PHIGS data types 18

 8.3 Implementation dependent PHIGS PLUS type definitions 23

 8.4 Implementation independent PHIGS PLUS type definitions 33

© ISO/IEC 1994
All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from the publisher.

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

9	C PHIGS PLUS macro definitions	55
9.1	Function identifiers	55
9.2	Error codes	56
9.3	Miscellaneous	59
9.3.1	Colour mapping methods	59
9.3.2	Colour types	59
9.3.3	Curve approximation criteria types	60
9.3.4	Data mapping methods	60
9.3.5	Interior shading method	60
9.3.6	Light source types	60
9.3.7	Parametric surface characteristic types	60
9.3.8	Polyline shading method	61
9.3.9	Reflectance models	61
9.3.10	Reflectance properties	61
9.3.11	Rendering Colour Model	61
9.3.12	Surface approximation criteria types	61
9.3.13	Element enumeration	61
10	C PHIGS PLUS functions	62
10.1	Output primitive functions	62
10.2	Attribute specification functions	68
10.2.1	Bundled attribute selection	68
10.2.2	Individual attribute selection	69
10.2.3	Workstation attribute table definition	75
10.3	Inquiry functions	78
10.3.1	Inquiry functions for the workstation state list	78
10.3.2	Inquiry functions for the workstation description table	85
A	Data types in compilation order and external functions	95
A.1	Macro definitions	95
A.2	Types in compilation order	101
A.3	External functions	125
B	Example programs	144
B.1	star	144
B.2	iron	144
B.3	dyna_star	144
B.4	show_line	144
B.5	xform_pline	144
B.6	surface_trimmed	144
C	Macros for short function identifiers	149
C.1	Short function identifiers	149
D	Memory management	152
D.1	Introduction	152
D.2	Functions that return simple lists	152
D.3	Functions that return complex data structures	152
E	Function lists	153
E.1	List of functions ordered alphabetically by function name	153
E.2	List of functions ordered alphabetically by bound name	155

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 9593-4:1991 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*.

Introduction

Replace the current text with the following:

Part 1 of the Programmer's Hierarchical Interactive Graphics System (PHIGS) functional description is registered as ISO/IEC 9592-1:1989. Part 1 is extended by part 4, ISO/IEC 9592-4:1991 to incorporate the effects of lighting, shading and other properties important for the display of surfaces and multidimensional data. The purpose of this part of ISO/IEC 9593 is to define a standard binding for the C computer programming language to the functionality defined in ISO/IEC 9592-1:1989 and ISO/IEC 9592-4:1991.

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

Part 4: C

AMENDMENT 1

1 Scope

Replace the text with the following:

Part 1 of the Programmer's Hierarchical Interactive Graphics System (PHIGS), ISO/IEC 9592-1, and part 4 (PHIGS PLUS), ISO/IEC 9592-4, specify a language independent nucleus of a graphics system. For integration into a programming language, PHIGS and PHIGS PLUS are embedded in a language dependent layer obeying the particular conventions of that language. This part of ISO/IEC 9593 specifies such a language dependent layer for the C language.

2 Normative references

Add the following reference:

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 The C language binding of PHIGS

Change the clause heading to The C language binding of PHIGS and PHIGS PLUS.

3.1 Conformance

Replace the last two paragraphs with the following text:

In order to conform, an implementation of the C binding of PHIGS shall implement those functions specified in ISO/IEC 9592-1. The implementation shall make visible all of the declarations in the C binding specified in clauses 5 to 7 in this part of ISO/IEC 9593.

In order to conform, an implementation of the C 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 C binding specified in clauses 5 to 10 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.

3.2 Functions versus macros

No changes.

3.3 Character strings

No changes.

3.4 Function identifiers

No changes.

3.5 Registration

No changes.

3.6 Identifiers for graphical items

No changes.

3.7 Return values

No changes.

3.8 Header files

No changes.

3.9 Memory management

No changes.

3.9.1 Inquiry functions which return simple lists

No changes.

3.9.2 Inquiry functions which return complex data structures

No changes.

3.9.3 Meaning of the size of an element

No changes.

3.10 Inquiries returning structure elements

No changes.

3.11 Error handling

3.11.1 Application defined error handlers

No changes.

3.11.2 Error codes

No changes.

3.11.3 C specific PHIGS errors

Change the heading to C binding specific errors and add the following binding specific errors:

2207	Ignoring function, two or more vertices do not contain the same type of data Is issued when an output primitive is created in which one or more of the vertices are specified with a particular type of data, such as normals, and other vertices are specified without the same type of data.
------	---

3.12 Storage of two-dimensional data

No changes.

3.12.1 Storage of matrices

No changes.

3.12.2 Storage of colour arrays

Change the heading to be Storage of arrays and replace the text with the following:

The entries of the Ppat_rep and Px_array data types (where x is one of colrv, colrv_ctrl_point, ctrl_point3, data_ctrl_point, edge_flag, facet, float, vertex, and vertex3) shall be stored such that the $(i,j)^{\text{th}}$ element in the array is given by the following expression:

$$i + j * DX$$

where:

$$i = 0, \dots, DX-1,$$

$$j = 0, \dots, DY-1,$$

DX = the size_x value of the array's Pint_size field, and

DY = the size_y value of the array's Pint_size field.

3.13 Data type descriptions

Add the clause with the following text:

The Pedge_flag_array data type defines an edge visibility flag for each edge of the quadrilateral mesh. The edge visibility flags are arranged in a MXN array of Pedge_flag_pairs. For each (i,j) Pedge_flag_pair, the first element specifies the edge flag for the edge between the i,j and $i+1,j$ vertices and the second element specifies the edge flag for the edge between the i,j and $i,j+1$ vertices. The first element of the (M,j) Pedge_flag_pair, where $1 \leq j \leq N$, and second element of the (i,N) Pedge_flag_pair, where $1 \leq i \leq M$, are not used.

The Pedge_flag_triplet_list data type consists of a list of Pedge_flag_triplets; one for each triangle in a triangle set. The first element of a Pedge_flag_triplet specifies the edge flag between the first and the second vertices of the triangle, the second element specifies the edge flag between the second and third vertices of the triangle. The third element specifies the edge flag between the third and first vertices of the triangle.

The Px_set data types (where x is one of colrv, contour, edge_flag, float, vertex, and vertex3) contain a list of Px_lists.

The Px_set_list data types (where x is one of contour and edge_flag) contains a list of Px_sets.

4 Tables

4.1 Abbreviation policy for construction of identifiers

No changes.

4.2 Table of abbreviations

Add alphabetically, to table 1, the following abbreviations:

Table 1 - Abbreviations ordered alphabetically

Word or Phrase	Abbreviation
activation	act
ambient	amb
approximation	approx
attenuation	atten
between	NULL
characteristics	chars
coefficient	coeff
concentration	conc
constant	const
compute	comp
criteria	crit
culling	cull
deactivate	deact
deactivation	deact
deviation	dev
diffuse	diff
directional	dir
distinguishing	disting
exponent	exp
geometric	geom
homogeneous	homo
isoparametric	isoparam
parametric	param
positional	pos
product	prod
properties	props
quadrilateral	quad

Table 1 - Abbreviations ordered alphabetically

Word or Phrase	Abbreviation
reflectance	refl
rendering	rend
shading	shad
specular	spec
subdivision	subd
triangle	tri
triangular	tri
trimming	trim
with	NULL

4.3 Function names

4.3.1 List ordered alphabetically by bound name

In table 2 change the column heading PHIGS Name to PHIGS and PHIGS PLUS Name and merge alphabetically by bound name, the following function names:

Table 2 - Function names ordered by bound name

C Name	PHIGS and PHIGS PLUS Name
pcell_array3_plus	CELL ARRAY 3 PLUS
pfill_area_set_data	FILL AREA SET WITH DATA
pfill_area_set3_data	FILL AREA SET 3 WITH DATA
pinq_b_spline_surf_fac	INQUIRE B-SPLINE SURFACE FACILITIES
pinq_colr_map_fac	INQUIRE COLOUR MAPPING FACILITIES
pinq_colr_map_method_fac	INQUIRE COLOUR MAPPING METHOD FACILITIES
pinq_colr_map_rep	INQUIRE COLOUR MAPPING REPRESENTATION
pinq_colr_map_st	INQUIRE COLOUR MAPPING STATE
pinq_curve_fac	INQUIRE CURVE FACILITIES
pinq_data_map_fac	INQUIRE DATA MAPPING FACILITIES
pinq_data_map_rep	INQUIRE DATA MAPPING REPRESENTATION
pinq_depth_cue_fac	INQUIRE DEPTH CUE FACILITIES
pinq_depth_cue_rep	INQUIRE DEPTH CUE REPRESENTATION
pinq_direct_colr_model_fac	INQUIRE DIRECT COLOUR MODEL FACILITIES
pinq_dyns_ws_plus	INQUIRE DYNAMICS OF WORKSTATION PLUS
pinq_edge_rep_plus	INQUIRE EDGE REPRESENTATION PLUS
pinq_int_fac_plus	INQUIRE INTERIOR FACILITIES PLUS
pinq_int_rep_plus	INQUIRE INTERIOR REPRESENTATION PLUS
pinq_light_source_fac	INQUIRE LIGHT SOURCE FACILITIES
pinq_light_source_rep	INQUIRE LIGHT SOURCE REPRESENTATION
pinq_line_fac_plus	INQUIRE POLYLINE FACILITIES PLUS
pinq_line_rep_plus	INQUIRE POLYLINE REPRESENTATION PLUS
pinq_list_colr_map_inds	INQUIRE LIST OF COLOUR MAPPING INDICES

Table 2 - Function names ordered by bound name

C Name	PHIGS and PHIGS PLUS Name
pinq_list_data_map_inds	INQUIRE LIST OF DATA MAPPING INDICES
pinq_list_depth_cue_inds	INQUIRE LIST OF DEPTH CUE INDICES
pinq_list_light_source_inds	INQUIRE LIST OF LIGHT SOURCE INDICES
pinq_list_param_surf_inds	INQUIRE LIST OF PARAMETRIC SURFACE INDICES
pinq_list_refl_inds	INQUIRE LIST OF REFLECTANCE INDICES
pinq_marker_rep_plus	INQUIRE POLYMARKER REPRESENTATION PLUS
pinq_param_surf_fac	INQUIRE PARAMETRIC SURFACE FACILITIES
pinq_param_surf_rep	INQUIRE PARAMETRIC SURFACE REPRESENTATION
pinq_pat_rep_plus	INQUIRE PATTERN REPRESENTATION PLUS
pinq_pred_colr_map_rep	INQUIRE PREDEFINED COLOUR MAPPING REPRESENTATION
pinq_pred_data_map_rep	INQUIRE PREDEFINED DATA MAPPING REPRESENTATION
pinq_pred_depth_cue_rep	INQUIRE PREDEFINED DEPTH CUE REPRESENTATION
pinq_pred_edge_rep_plus	INQUIRE PREDEFINED EDGE REPRESENTATION PLUS
pinq_pred_int_rep_plus	INQUIRE PREDEFINED INTERIOR REPRESENTATION PLUS
pinq_pred_light_source_rep	INQUIRE PREDEFINED LIGHT SOURCE REPRESENTATION
pinq_pred_line_rep_plus	INQUIRE PREDEFINED POLYLINE REPRESENTATION PLUS
pinq_pred_marker_rep_plus	INQUIRE PREDEFINED POLYMARKER REPRESENTATION PLUS
pinq_pred_param_surf_rep	INQUIRE PREDEFINED PARAMETRIC SURFACE REPRESENTATION
pinq_pred_pat_rep_plus	INQUIRE PREDEFINED PATTERN REPRESENTATION PLUS
pinq_pred_refl_rep	INQUIRE PREDEFINED REFLECTANCE REPRESENTATION
pinq_pred_text_rep_plus	INQUIRE PREDEFINED TEXT REPRESENTATION PLUS
pinq_refl_fac	INQUIRE REFLECTANCE FACILITIES
pinq_refl_rep	INQUIRE REFLECTANCE REPRESENTATION
pinq_rend_colr_model_fac	INQUIRE RENDERING COLOUR MODEL FACILITIES
pinq_text_rep_plus	INQUIRE TEXT REPRESENTATION PLUS
pinq_trim_curve_fac	INQUIRE TRIMMING CURVE FACILITIES
pinq_ws_st_table_length_plus	INQUIRE WORKSTATION STATE TABLE LENGTHS PLUS
pnon_uniform_b_spline_curve3	NON-UNIFORM B-SPLINE CURVE 3
pnon_uniform_b_spline_curve3_colr	NON-UNIFORM B-SPLINE CURVE 3 WITH COLOUR
pnon_uniform_b_spline_surf3	NON-UNIFORM B-SPLINE SURFACE 3
pnon_uniform_b_spline_surf3_data	NON-UNIFORM B-SPLINE SURFACE 3 WITH DATA
ppolyline_set3_colr	POLYLINE SET 3 WITH COLOUR
pquad_mesh_data	QUADRILATERAL MESH WITH DATA
pquad_mesh3_data	QUADRILATERAL MESH 3 WITH DATA
pset_back_data_map_ind	SET BACK DATA MAPPING INDEX
pset_back_data_map_method	SET BACK DATA MAPPING METHOD
pset_back_int_colr	SET BACK INTERIOR COLOUR
pset_back_int_ind	SET BACK INTERIOR INDEX
pset_back_int_shad_method	SET BACK INTERIOR SHADING METHOD
pset_back_int_style	SET BACK INTERIOR STYLE
pset_back_int_style_ind	SET BACK INTERIOR STYLE INDEX
pset_back_refl_ind	SET BACK REFLECTANCE INDEX
pset_back_refl_model	SET BACK REFLECTANCE MODEL
pset_back_refl_props	SET BACK REFLECTANCE PROPERTIES
pset_colr_map_ind	SET COLOUR MAPPING INDEX
pset_colr_map_rep	SET COLOUR MAPPING REPRESENTATION
pset_curve_approx_crit	SET CURVE APPROXIMATION CRITERIA

Table 2 - Function names ordered by bound name

C Name	PHIGS and PHIGS PLUS Name
pset_data_map_ind	SET DATA MAPPING INDEX
pset_data_map_method	SET DATA MAPPING METHOD
pset_data_map_rep	SET DATA MAPPING REPRESENTATION
pset_depth_cue_ind	SET DEPTH CUE INDEX
pset_depth_cue_rep	SET DEPTH CUE REPRESENTATION
pset_edge_colr	SET EDGE COLOUR
pset_edge_rep_plus	SET EDGE REPRESENTATION PLUS
pset_facet_cull_mode	SET FACET CULLING MODE
pset_facet_disting_mode	SET FACET DISTINGUISHING MODE
pset_int_colr	SET INTERIOR COLOUR
pset_int_rep_plus	SET INTERIOR REPRESENTATION PLUS
pset_int_shad_method	SET INTERIOR SHADING METHOD
pset_light_source_rep	SET LIGHT SOURCE REPRESENTATION
pset_light_source_st	SET LIGHT SOURCE STATE
pset_line_colr	SET POLYLINE COLOUR
pset_line_rep_plus	SET POLYLINE REPRESENTATION PLUS
pset_line_shad_method	SET POLYLINE SHADING METHOD
pset_marker_colr	SET POLYMARKER COLOUR
pset_marker_rep_plus	SET POLYMARKER REPRESENTATION PLUS
pset_of_fill_area_sets_data	SET OF FILL AREA SETS WITH DATA
pset_of_fill_area_sets3_data	SET OF FILL AREA SETS 3 WITH DATA
pset_param_surf_chars	SET PARAMETRIC SURFACE CHARACTERISTICS
pset_param_surf_ind	SET PARAMETRIC SURFACE INDEX
pset_param_surf_rep	SET PARAMETRIC SURFACE REPRESENTATION
pset_pat_rep_plus	SET PATTERN REPRESENTATION PLUS
pset_refl_ind	SET REFLECTANCE INDEX
pset_refl_model	SET REFLECTANCE MODEL
pset_refl_props	SET REFLECTANCE PROPERTIES
pset_refl_rep	SET REFLECTANCE REPRESENTATION
pset_rend_colr_model	SET RENDERING COLOUR MODEL
pset_surf_approx_crit	SET SURFACE APPROXIMATION CRITERIA
pset_text_colr	SET TEXT COLOUR
pset_text_rep_plus	SET TEXT REPRESENTATION PLUS
ptri_set_data	TRIANGLE SET WITH DATA
ptri_set3_data	TRIANGLE SET 3 WITH DATA
ptri_strip_data	TRIANGLE STRIP WITH DATA
ptri_strip3_data	TRIANGLE STRIP 3 WITH DATA

4.3.2 List ordered alphabetically by PHIGS function name

In table 3 change the clause heading to List ordered alphabetically by PHIGS and PHIGS PLUS function name, change the table caption to Table 3 - Function names ordered by PHIGS and PHIGS PLUS names,

change the column heading PHIGS Name to PHIGS and PHIGS PLUS Name and merge alphabetically by function name the following function names:

Table 3 - Function names ordered by PHIGS and PHIGS PLUS function name

PHIGS and PHIGS PLUS Name	C Name
CELL ARRAY 3 PLUS	pcell_array3_plus
FILL AREA SET WITH DATA	pfill_area_set_data
FILL AREA SET 3 WITH DATA	pfill_area_set3_data
INQUIRE B-SPLINE SURFACE FACILITIES	pinq_b_spline_surf_fac
INQUIRE COLOUR MAPPING FACILITIES	pinq_colr_map_fac
INQUIRE COLOUR MAPPING METHOD FACILITIES	pinq_colr_map_method_fac
INQUIRE COLOUR MAPPING REPRESENTATION	pinq_colr_map_rep
INQUIRE COLOUR MAPPING STATE	pinq_colr_map_st
INQUIRE CURVE FACILITIES	pinq_curve_fac
INQUIRE DATA MAPPING FACILITIES	pinq_data_map_fac
INQUIRE DATA MAPPING REPRESENTATION	pinq_data_map_rep
INQUIRE DEPTH CUE FACILITIES	pinq_depth_cue_fac
INQUIRE DEPTH CUE REPRESENTATION	pinq_depth_cue_rep
INQUIRE DIRECT COLOUR MODEL FACILITIES	pinq_direct_colr_model_fac
INQUIRE DYNAMICS OF WORKSTATION PLUS	pinq_dyns_ws_plus
INQUIRE EDGE REPRESENTATION PLUS	pinq_edge_rep_plus
INQUIRE INTERIOR FACILITIES PLUS	pinq_int_fac_plus
INQUIRE INTERIOR REPRESENTATION PLUS	pinq_int_rep_plus
INQUIRE LIGHT SOURCE FACILITIES	pinq_light_source_fac
INQUIRE LIGHT SOURCE REPRESENTATION	pinq_light_source_rep
INQUIRE LIST OF COLOUR MAPPING INDICES	pinq_list_colr_map_inds
INQUIRE LIST OF DATA MAPPING INDICES	pinq_list_data_map_inds
INQUIRE LIST OF DEPTH CUE INDICES	pinq_list_depth_cue_inds
INQUIRE LIST OF LIGHT SOURCE INDICES	pinq_list_light_source_inds
INQUIRE LIST OF PARAMETRIC SURFACE INDICES	pinq_list_param_surf_inds
INQUIRE LIST OF REFLECTANCE INDICES	pinq_list_refl_inds
INQUIRE PARAMETRIC SURFACE FACILITIES	pinq_param_surf_fac
INQUIRE PARAMETRIC SURFACE REPRESENTATION	pinq_param_surf_rep
INQUIRE PATTERN REPRESENTATION PLUS	pinq_pat_rep_plus
INQUIRE POLYLINE FACILITIES PLUS	pinq_line_fac_plus
INQUIRE POLYLINE REPRESENTATION PLUS	pinq_line_rep_plus
INQUIRE POLYMARKER REPRESENTATION PLUS	pinq_marker_rep_plus
INQUIRE PREDEFINED COLOUR MAPPING REPRESENTATION	pinq_pred_colr_map_rep
INQUIRE PREDEFINED DATA MAPPING REPRESENTATION	pinq_pred_data_map_rep
INQUIRE PREDEFINED DEPTH CUE REPRESENTATION	pinq_pred_depth_cue_rep
INQUIRE PREDEFINED EDGE REPRESENTATION PLUS	pinq_pred_edge_rep_plus
INQUIRE PREDEFINED INTERIOR REPRESENTATION PLUS	pinq_pred_int_rep_plus
INQUIRE PREDEFINED LIGHT SOURCE REPRESENTATION	pinq_pred_light_source_rep
INQUIRE PREDEFINED PARAMETRIC SURFACE REPRESENTATION	pinq_pred_param_surf_rep
INQUIRE PREDEFINED PATTERN REPRESENTATION PLUS	pinq_pred_pat_rep_plus
INQUIRE PREDEFINED POLYLINE REPRESENTATION PLUS	pinq_pred_line_rep_plus
INQUIRE PREDEFINED POLYMARKER REPRESENTATION PLUS	pinq_pred_marker_rep_plus
INQUIRE PREDEFINED REFLECTANCE REPRESENTATION	pinq_pred_refl_rep
INQUIRE PREDEFINED TEXT REPRESENTATION PLUS	pinq_pred_text_rep_plus

Table 3 - Function names ordered by PHIGS and PHIGS PLUS function name

PHIGS and PHIGS PLUS Name	C Name
INQUIRE REFLECTANCE FACILITIES	pinq_refl_fac
INQUIRE REFLECTANCE REPRESENTATION	pinq_refl_rep
INQUIRE RENDERING COLOUR MODEL FACILITIES	pinq_rend_colr_model_fac
INQUIRE TEXT REPRESENTATION PLUS	pinq_text_rep_plus
INQUIRE TRIMMING CURVE FACILITIES	pinq_trim_curve_fac
INQUIRE WORKSTATION STATE TABLE LENGTHS PLUS	pinq_ws_st_table_length_plus
NON-UNIFORM B-SPLINE CURVE 3	pnon_uniform_b_spline_curve3
NON-UNIFORM B-SPLINE SURFACE 3	pnon_uniform_b_spline_surf3
NON-UNIFORM B-SPLINE SURFACE 3 WITH DATA	pnon_uniform_b_spline_surf3_data
NON-UNIFORM B-SPLINE CURVE 3 WITH COLOUR	pnon_uniform_b_spline_curve3_colr
POLYLINE SET 3 WITH COLOUR	ppolyline_set3_colr
QUADRILATERAL MESH WITH DATA	pquad_mesh_data
QUADRILATERAL MESH 3 WITH DATA	pquad_mesh3_data
SET BACK DATA MAPPING INDEX	pset_back_data_map_ind
SET BACK DATA MAPPING METHOD	pset_back_data_map_method
SET BACK INTERIOR COLOUR	pset_back_int_colr
SET BACK INTERIOR INDEX	pset_back_int_ind
SET BACK INTERIOR SHADING METHOD	pset_back_int_shad_method
SET BACK INTERIOR STYLE INDEX	pset_back_int_style_ind
SET BACK INTERIOR STYLE	pset_back_int_style
SET BACK REFLECTANCE INDEX	pset_back_refl_ind
SET BACK REFLECTANCE MODEL	pset_back_refl_model
SET BACK REFLECTANCE PROPERTIES	pset_back_refl_props
SET COLOUR MAPPING INDEX	pset_colr_map_ind
SET COLOUR MAPPING REPRESENTATION	pset_colr_map_rep
SET CURVE APPROXIMATION CRITERIA	pset_curve_approx_crit
SET DATA MAPPING INDEX	pset_data_map_ind
SET DATA MAPPING METHOD	pset_data_map_method
SET DATA MAPPING REPRESENTATION	pset_data_map_rep
SET DEPTH CUE INDEX	pset_depth_cue_ind
SET DEPTH CUE REPRESENTATION	pset_depth_cue_rep
SET EDGE COLOUR	pset_edge_colr
SET EDGE REPRESENTATION PLUS	pset_edge_rep_plus
SET FACET CULLING MODE	pset_facet_cull_mode
SET FACET DISTINGUISHING MODE	pset_facet_disting_mode
SET INTERIOR COLOUR	pset_int_colr
SET INTERIOR REPRESENTATION PLUS	pset_int_rep_plus
SET INTERIOR SHADING METHOD	pset_int_shad_method
SET LIGHT SOURCE REPRESENTATION	pset_light_source_rep
SET LIGHT SOURCE STATE	pset_light_source_st
SET OF FILL AREA SETS WITH DATA	pset_of_fill_area_sets_data
SET OF FILL AREA SETS 3 WITH DATA	pset_of_fill_area_sets3_data
SET PARAMETRIC SURFACE CHARACTERISTICS	pset_param_surf_chars
SET PARAMETRIC SURFACE INDEX	pset_param_surf_ind
SET PARAMETRIC SURFACE REPRESENTATION	pset_param_surf_rep
SET PATTERN REPRESENTATION PLUS	pset_pat_rep_plus
SET POLYLINE COLOUR	pset_line_colr

Table 3 - Function names ordered by PHIGS and PHIGS PLUS function name

PHIGS and PHIGS PLUS Name	C Name
SET POLYLINE REPRESENTATION PLUS	pset_line_rep_plus
SET POLYLINE SHADING METHOD	pset_line_shad_method
SET POLYMARKER COLOUR	pset_marker_colr
SET POLYMARKER REPRESENTATION PLUS	pset_marker_rep_plus
SET REFLECTANCE INDEX	pset_refl_ind
SET REFLECTANCE MODEL	pset_refl_model
SET REFLECTANCE PROPERTIES	pset_refl_props
SET REFLECTANCE REPRESENTATION	pset_refl_rep
SET RENDERING COLOUR MODEL	pset_rend_colr_model
SET SURFACE APPROXIMATION CRITERIA	pset_surf_approx_crit
SET TEXT COLOUR	pset_text_colr
SET TEXT REPRESENTATION PLUS	pset_text_rep_plus
TRIANGLE SET WITH DATA	ptri_set_data
TRIANGLE SET 3 WITH DATA	ptri_set3_data
TRIANGLE STRIP WITH DATA	ptri_strip_data
TRIANGLE STRIP 3 WITH DATA	ptri_strip3_data

5 Type definitions

Change the clause heading to C PHIGS type definitions.

6 Macro definitions

Change the clause heading to C PHIGS macro definitions.

7 C PHIGS functions

Change the following function definitions:

INQUIRE GENERALIZED DRAWING PRIMITIVE 3

```
void pinq_gdp3 (
    Pint          ws_type,          /* workstation type          */
    Pint          gdp,              /* GDP function identifier    */
    Pint          *err_ind,          /* OUT error indicator        */
    Pint          *num_attr,         /* OUT number of attributes used */
    Pattrrs       attr[7]           /* OUT list of attributes used */
);
```

INQUIRE GENERALIZED DRAWING PRIMITIVE

```
void pinq_gdp (
    Pint          ws_type,          /* workstation type          */
    Pint          gdp,              /* GDP function identifier    */
    Pint          *err_ind,          /* OUT error indicator        */
    Pint          *num_attr,         /* OUT number of attributes used */
    Pattrrs       attr[7]           /* OUT list of attributes used */
);
```

8 C PHIGS PLUS type definitions

8.1 Mapping of PHIGS PLUS data types

ISO/IEC 9592-4 specifies a set of abstract data types beyond the types defined in ISO/IEC 9592-1. Table 4 - gives the mapping from those additional data types defined in ISO/IEC 9592-4 to the data types defined later in this binding.

Table 4 - Mapping of PHIGS PLUS data types to C

PHIGS PLUS data type		C binding data type
COLRV	colour value	Pcolrv
COLRVH	homogeneous colour value	Phomo_colr_rep_ptr
GCOLR	general colour	Pgcolr
NORM	normal vector	Pvec3
P3H	3D homogeneous point	Phomo_point3
P4H	4D homogeneous point	Phomo_point4

8.2 Modifications to PHIGS data types

Paspect ASPECT

```
typedef enum {  
    /* start of PHIGS enumerations */  
    ...,  
    /* end of PHIGS enumerations */  
    /* start of PHIGS PLUS enumerations */  
    PASPECT_LINE_COLR,  
    PASPECT_MARKER_COLR,  
    PASPECT_TEXT_COLR,  
    PASPECT_INT_COLR,  
    PASPECT_EDGE_COLR,  
    PASPECT_LINE_SHAD_METHOD,  
    PASPECT_INT_SHAD_METHOD,  
    PASPECT_DATA_MAP_METHOD,  
    PASPECT_REFL_PROPS,  
    PASPECT_REFL_MODEL,  
    PASPECT_BACK_INT_STYLE,  
    PASPECT_BACK_INT_STYLE_IND,  
    PASPECT_BACK_INT_COLR,  
    PASPECT_BACK_INT_SHAD_METHOD,  
    PASPECT_BACK_DATA_MAP_METHOD,  
    PASPECT_BACK_REFL_PROPS,  
    PASPECT_BACK_REFL_MODEL,  
    PASPECT_CURVE_APPROX_CRIT,  
    PASPECT_SURF_APPROX_CRIT,  
    PASPECT_PARAM_SURF_CHARS  
    /* end of PHIGS PLUS enumerations */  
} Paspect;
```

Pattrr ATTRIBUTES USED

```
typedef enum {  
    /* start of PHIGS enumerations */  
    ...,  
    /* end of PHIGS enumerations */  
    /* start of PHIGS PLUS enumerations */  
    PATTR_REFL,  
    PATTR_PARAM_SURF  
    /* end of PHIGS PLUS enumerations */  
} Pattrr;
```

Pelem_data ELEMENT DATA

```

typedef union {

    /* start of PHIGS element data */
    ...,
    /* end of PHIGS element data */
    /* start of PHIGS PLUS element data */
    struct Pelem_polyline_set3_colr {
        Pint          colr_type;          /* colour type */
        Pvertex3_set   vertex_data;        /* list of vertex lists */
    } polyline_set3_colr;
    struct Pelem_fill_area_set3_data {
        Pint          colr_type;          /* colour type */
        Pfacet         *facet_data;        /* facet data */
        Pedge_flag_set *edge_flags;        /* list of edge flag lists */
        Pvertex3_set   vertex_data;        /* list of vertex lists */
    } fill_area_set3_data;
    struct Pelem_fill_area_set_data {
        Pint          colr_type;          /* colour type */
        Pfacet         *facet_data;        /* facet data */
        Pedge_flag_set *edge_flags;        /* list of edge flag lists */
        Pvertex_set    vertex_data;        /* list of vertex lists */
    } fill_area_set_data;
    struct Pelem_cell_array3_plus {
        Pparal         paral;              /* parallelogram P, Q, R */
        Ppat_rep_plus   pattern;           /* pattern colour array */
    } cell_array3_plus;
    struct Pelem_set_of_fill_area_sets3_data {
        Pint          colr_type;          /* colour type */
        Pset_of_fill_area_sets_data *set_data; /* fill area sets data */
        Pedge_flag_set_list *edge_flags;    /* list of edge flag sets */
        Pvertex3_list    vertex_data;        /* list of vertices */
        Pcontour_set_list contours;          /* list of contour sets */
    } set_of_fill_area_sets3_data;
    struct Pelem_set_of_fill_area_sets_data {
        Pint          colr_type;          /* colour type */
        Pset_of_fill_area_sets_data *set_data; /* fill area sets data */
        Pedge_flag_set_list *edge_flags;    /* list of edge flag sets */
        Pvertex_list    vertex_data;        /* list of vertices */
        Pcontour_set_list contours;          /* list of contour sets */
    } set_of_fill_area_sets_data;
    struct Pelem_tri_set3_data {
        Pint          colr_type;          /* colour type */
        Ptri_set_data  *set_data;          /* triangle set data */
        Pedge_flag_triplet_list *edge_flags; /* list of triangle edge flag triplets */
        Pvertex3_list  vertex_data;        /* list of vertices */
        Pvertex_ind_triplet_list triplets;  /* list of triangle vertex index triplets */
    }

```

```

} tri_set3_data;
struct Pelem_tri_set_data {
    Pint          colr_type;      /* colour type          */
    Ptri_set_data *set_data;      /* triangle set data    */
    Pedge_flag_triplet_list *edge_flags; /* list of triangle edge flag */
                                     /* triplets             */
    Pvertex_list   vertex_data;   /* list of vertices     */
    Pvertex_ind_triplet_list triplets; /* list of triangle vertex index */
                                     /* triplets             */
} tri_set_data;
struct Pelem_tri_strip3_data {
    Pint          colr_type;      /* colour type          */
    Pfacet_list   *facet_data;    /* list of facet data    */
    Pedge_flag_list *edge_flags;  /* list of edge flags    */
    Pvertex3_list vertex_data;    /* list of vertex data   */
} tri_strip3_data;
struct Pelem_tri_strip_data {
    Pint          colr_type;      /* colour type          */
    Pfacet_list   *facet_data;    /* list of facet data    */
    Pedge_flag_list *edge_flags;  /* list of edge flags    */
    Pvertex_list   vertex_data;   /* list of vertex data   */
} tri_strip_data;
struct Pelem_quad_mesh3_data {
    Pint          colr_type;      /* colour type          */
    Pfacet_array  *facet_data;    /* array of facet data   */
    Pedge_flag_array *edge_flags; /* array of edge flags   */
    Pvertex3_array vertex_data;   /* array of vertex data  */
} quad_mesh3_data;
struct Pelem_quad_mesh_data {
    Pint          colr_type;      /* colour type          */
    Pfacet_array  *facet_data;    /* array of facet data   */
    Pedge_flag_array *edge_flags; /* array of edge flags   */
    Pvertex_array vertex_data;   /* array of vertex data  */
} quad_mesh_data;
struct Pelem_non_uniform_b_spline_curve3 {
    Pcurve_geom_spline geom_spline; /* geometry spline      */
} non_uniform_b_spline_curve3;
struct Pelem_non_uniform_b_spline_curve3_colr {
    Pcurve_geom_spline geom_spline; /* geometry spline      */
    Pcurve_colr_spline *colr_spline; /* colour spline        */
} non_uniform_b_spline_curve3_colr;
struct Pelem_non_uniform_b_spline_surf3 {
    Psurf_geom_spline geom_spline; /* geometry spline      */
    Ptrim_curve_loop_list *trim_loops; /* list of trimming curve loops */
} non_uniform_b_spline_surf3;
struct Pelem_non_uniform_b_spline_surf3_data {
    Psurf_geom_spline geom_spline; /* geometry spline      */
    Ptrim_curve_loop_list *trim_loops; /* list of trimming curve loops */
    Psurf_colr_spline *colr_spline; /* colour spline        */
    Psurf_data_spline_list *data_spline; /* data spline         */
} non_uniform_b_spline_surf3_data;

```

```

Pgcclr          clr;          /* general colour          */
Pfacet_disting_mode facet_disting_mode; /*facet distinguishing mode */
Pfacet_cull_mode  facet_cull_mode; /* facet culling mode      */
Pdata_map_rep     data_map_method; /* data mapping method      */
Prefl_prop        refl_prop;    /* reflectance property data */
struct Pelem_light_source_st {
  Pint_list        act_list;     /* activation list          */
  Pint_list        deact_list;   /* deactivation list        */
} light_source_st;
Pcurve_approx_crit curve_approx_crit; /* curve approximation criteria */
/* data record          */
Psurf_approx_crit  surf_approx_crit; /* surface approximation criteria */
/* data record          */
Pparam_surf_chars  param_surf_chars; /* parametric surface        */
/* characteristics data record */
/* end of PHIGS PLUS element data */
} Pelem_data;

```

NOTE — A NULL value for the facet_data, edge_flags, set_data, clr_spline, trim_loops, or data_spline fields indicates the corresponding optional data is not defined.

Pelem_type ELEMENT TYPE

```

typedef enum {

    /* start of PHIGS enumerations */
    ...,
    /* end of PHIGS enumerations */
    /* start of PHIGS PLUS enumerations */
    PELEM_POLYLINE_SET3_COLR,
    PELEM_FILL_AREA_SET3_DATA,
    PELEM_FILL_AREA_SET_DATA,
    PELEM_CELL_ARRAY3_PLUS,
    PELEM_SET_OF_FILL_AREA_SETS3_DATA,
    PELEM_SET_OF_FILL_AREA_SETS_DATA,
    PELEM_TRI_SET3_DATA,
    PELEM_TRI_SET_DATA,
    PELEM_TRI_STRIP3_DATA,
    PELEM_TRI_STRIP_DATA,
    PELEM_QUAD_MESH3_DATA,
    PELEM_QUAD_MESH_DATA,
    PELEM_NON_UNIFORM_B_SPLINE_CURVE3,
    PELEM_NON_UNIFORM_B_SPLINE_CURVE3_COLR,
    PELEM_NON_UNIFORM_B_SPLINE_SURF3,
    PELEM_NON_UNIFORM_B_SPLINE_SURF3_DATA,
    PELEM_DATA_MAP_IND,
    PELEM_REFL_IND,
    PELEM_BACK_INT_IND,
    PELEM_BACK_DATA_MAP_IND,
    PELEM_BACK_REFL_IND,
    PELEM_PARAM_SURF_IND,
    PELEM_LINE_COLR,
    PELEM_LINE_SHAD_METHOD,
    PELEM_MARKER_COLR,
    PELEM_TEXT_COLR,
    PELEM_FACET_DISTING_MODE,
    PELEM_FACET_CULL_MODE,
    PELEM_INT_COLR,
    PELEM_INT_SHAD_METHOD,
    PELEM_DATA_MAP_METHOD,
    PELEM_REFL_PROPS,
    PELEM_REFL_MODEL,
    PELEM_BACK_INT_STYLE,
    PELEM_BACK_INT_STYLE_IND,
    PELEM_BACK_INT_COLR,
    PELEM_BACK_INT_SHAD_METHOD,
    PELEM_BACK_DATA_MAP_METHOD,
    PELEM_BACK_REFL_PROPS,
    PELEM_BACK_REFL_MODEL,
    PELEM_LIGHT_SOURCE_ST,
    PELEM_EDGE_COLR,

```

```

PELEM_CURVE_APPROX_CRIT,
PELEM_SURF_APPROX_CRIT,
PELEM_PARAM_SURF_CHARS,
PELEM_REND_COLR_MODEL,
PELEM_DEPTH_CUE_IND,
PELEM_COLR_MAP_IND
/* end of PHIGS PLUS enumerations */

} Pelem_type;

```

8.3 Implementation dependent PHIGS PLUS type definitions

The following type definitions may be modified by the implementation. If the implementation adds new fields to the unions, the convention specified in clause 5.3 shall be used for naming the new field with the following addition:

For the name of a new field in a PHIGS PLUS implementation dependent type, called “xxx” in clause 5.3, use “type.”

Pcolr_map_method COLOUR MAPPING METHOD

```

typedef struct {
    Pint          method;                /* colour mapping method          */
    union Pcolr_map_method_data {
        Pint      num_true_colrs;        /* number of true colours          */
        Pint      max_num_pseudo_colrs; /* maximum number of pseudo        */
                                          /* colours                          */
        ...      /* implementation defined          */
    } data;
} Pcolr_map_method;

```

Pcolr_map_rep COLOUR MAPPING REPRESENTATION

```

typedef struct {
    Pint          method;                /* colour mapping method      */
    union Pcolr_map_rep_data {
        struct Pcolr_map_rep_pseudo {
            Pint          model;          /* colour model               */
            Pfloat_list   weight_vector; /* list of weight vector      */
            Pcolrv_list   colrs;         /* list of colour values      */
        } pseudo;
        struct Pcolr_map_rep_pseudo_N {
            Pint          model;          /* colour model               */
            Pfloat_set    colrs;         /* list of lists of colour    */
                                   /* coordinates                 */
        } pseudo_N;
        ...                             /* implementation defined      */
    } data;
} Pcolr_map_rep;

```

Pcolr_map_st COLOUR MAPPING STATE

```

typedef struct {
    Pint          method;                /* colour mapping method      */
    union Pcolr_map_st_data {
        Pint          num_true_colrs;    /* number of available true    */
                                   /* colours                     */
        Pint          max_num_pseudo_colrs; /* maximum number of pseudo   */
                                   /* colour entries              */
        ...                             /* implementation defined      */
    } data;
} Pcolr_map_st;

```

Pcolr_rep_ptr COLOUR REPRESENTATION POINTER

```
typedef union {  
    Prgb          *rgb;          /* pointer to RGB colour values */  
    Pcieluv       *cieluv;       /* pointer to CIELUV colour values */  
    Phls          *hls;          /* pointer to HLS colour values */  
    Phsv          *hsv;          /* pointer to HSV colour values */  
    Pdata         *unsupp;       /* pointer to unsupported colour */  
                                /* values */  
    ...           /* implementation defined */  
} Pcolr_rep_ptr;
```

Pcurve_approx_crit CURVE APPROXIMATION CRITERIA

```

typedef struct {
    Pint                type;                /* curve approximation criteria */
                                           /* type */
    union Pcurve_approx_crit_data {
        struct Pcurve_approx_const_param_subd {
            Pint        count;                /* count */
        } const_param_subd;
        struct Pcurve_approx_chordal_size_wc {
            Pfloat       approx;               /* approximation value */
        } chordal_size_wc;
        struct Pcurve_approx_chordal_size_npc {
            Pfloat       approx;               /* approximation value */
        } chordal_size_npc;
        struct Pcurve_approx_chordal_size_dc {
            Pfloat       approx;               /* approximation value */
        } chordal_size_dc;
        struct Pcurve_approx_chordal_dev_wc {
            Pfloat       approx;               /* approximation value */
        } chordal_dev_wc;
        struct Pcurve_approx_chordal_dev_npc {
            Pfloat       approx;               /* approximation value */
        } chordal_dev_npc;
        struct Pcurve_approx_chordal_dev_dc {
            Pfloat       approx;               /* approximation value */
        } chordal_dev_dc;
        struct Pcurve_approx_rel_wc {
            Pfloat       approx;               /* approximation value */
        } rel_wc;
        struct Pcurve_approx_rel_npc {
            Pfloat       approx;               /* approximation value */
        } rel_npc;
        struct Pcurve_approx_rel_dc {
            Pfloat       approx;               /* approximation value */
        } rel_dc;
        Pdata           unsupp;               /* unsupported curve approximation*/
                                           /* criteria data */
        ...              /* implementation defined */
    } data;
} Pcurve_approx_crit;

```

Pdata_map_rep DATA MAPPING REPRESENTATION

```

typedef struct {
    Pint          method;          /* data mapping method          */
    Psource_select_list selectors;  /* list of source selectors     */
    union Pdata_map_rep_data {
        struct Pdata_map_rep_single_uniform {
            Pint          ind;      /* data value index            */
            Pfloat        lower_limit; /* lower range limit          */
            Pfloat        upper_limit; /* upper range limit          */
            Pint          colr_type; /* colour type                 */
            Pcolrv_list   colr_values; /* list of colour values      */
        } single_uniform;
        struct Pdata_map_rep_single_non_uniform {
            Pint          ind;      /* data value index            */
            Pfloat_list   range;    /* range boundaries            */
            Pint          colr_type; /* colour type                 */
            Pcolrv_list   colr_values; /* list of colour values      */
        } single_non_uniform;
        struct Pdata_map_rep_bi_uniform {
            Pint          inds[2];  /* data value indices          */
            Pfloat        lower_limit_a; /* lower limit of Ra range    */
            Pfloat        upper_limit_a; /* upper limit of Ra range    */
            Pfloat        lower_limit_b; /* lower limit of Rb range    */
            Pfloat        upper_limit_b; /* upper limit of Rb range    */
            Pint          colr_type; /* colour type                 */
            Pcolrv_set    colr_values; /* list of colour value lists */
        } bi_uniform;
        struct Pdata_map_rep_bi_non_uniform {
            Pint          inds[2];  /* data value indices          */
            Pfloat_list   range_a;  /* Ra range boundaries         */
            Pfloat_set    range_b;  /* Rb range boundaries         */
            Pint          colr_type; /* colour type                 */
            Pcolrv_set    colr_values; /* array of colour values     */
        } bi_non_uniform;
        ... /* implementation defined */
    } data;
} Pdata_map_rep;

```

NOTE — The indexing of data mapping values is 1 based.

Phomo_colr_rep_ptr HOMOGENEOUS COLOUR REPRESENTATION POINTER

```
typedef union {  
    Phomo_rgb          *rgb;          /* pointer to homogeneous rgb    */  
                                /* colour value                  */  
    Phomo_cieluv       *cieluv;       /* pointer to homogeneous cieluv */  
                                /* colour value                  */  
    Phomo_hls          *hls;          /* pointer to homogeneous hls    */  
                                /* colour value                  */  
    Phomo_hsv          *hsv;          /* pointer to homogeneous hsv    */  
                                /* colour value                  */  
    ...                /* implementation defined       */  
} Phomo_colr_rep_ptr;
```

Plight_source_rep LIGHT SOURCE REPRESENTATION

```

typedef struct {
    Pint                type;                /* light source type          */
    union Plight_source_rep_data {
        struct Plight_source_rep_amb {
            Pcolour      colour;            /* light source colour        */
        } amb;
        struct Plight_source_rep_dir {
            Pcolour      colour;            /* light source colour        */
            Pvec3         dir;              /* light source direction     */
        } dir;
        struct Plight_source_rep_pos {
            Pcolour      colour;            /* light source colour        */
            Ppoint3       pos;              /* light source position      */
            Pfloat        c1;               /* light source attenuation   */
            /* coefficient                  */
            Pfloat        c2;               /* light source attenuation   */
            /* coefficient                  */
        } pos;
        struct Plight_source_rep_spot {
            Pcolour      colour;            /* light source colour        */
            Ppoint3       pos;              /* light source position      */
            Pvec3         dir;              /* light source direction     */
            Pfloat        concent_exp;      /* concentration exponent     */
            Pfloat        c1;               /* light source attenuation   */
            /* coefficient                  */
            Pfloat        c2;               /* light source attenuation   */
            /* coefficient                  */
            Pfloat        spread_angle;     /* light source spread angle  */
        } spot;
        ...                               /* implementation defined     */
    } data;
} Plight_source_rep;

```

Pparam_surf_chars PARAMETRIC SURFACE CHARACTERISTICS

```

typedef struct {
    Pint          type;          /* parametric surface          */
                                /* characteristics type        */

    union Pparam_surf_chars_data {
        struct Pparam_surf_isoparam {
            Pplacement    placement;    /* curve placement: UNIFORM OVER */
                                        /* SURFACE, UNIFORM BETWEEN KNOTS */

            Pint          num_u_curves;  /* curve count in u direction    */
            Pint          num_v_curves;  /* curve count in v direction    */
        } isoparam;
        struct Pparam_surf_level_mc {
            Ppoint3        origin;        /* origin point                  */
            Pvec3          dir;           /* direction vector              */
            Pfloat_list    params;        /* parameter list                */
        } level_mc;
        struct Pparam_surf_level_wc {
            Ppoint3        origin;        /* origin point                  */
            Pvec3          dir;           /* direction vector              */
            Pfloat_list    params;        /* parameter list                */
        } level_wc;
        Pdata             unsupp;        /* unsupported parametric        */
                                /* surface characteristics data  */
        ...               /* implementation defined        */
    } data;
} Pparam_surf_chars;

```

Prefl_prop REFLECTANCE PROPERTY

```
typedef struct {
    Pint          type;                /* reflectance property type    */
    union Prefl_prop_data {
        struct Prefl_prop_simple_refl {
            Pfloat      amb_coeff;      /* ambient reflection          */
                                          /* coefficient                  */
            Pfloat      diff_coeff;     /* diffuse reflection          */
                                          /* coefficient                  */
            Pfloat      spec_coeff;     /* specular reflection         */
                                          /* coefficient                  */
            Pcolour     spec_colr;      /* specular colour             */
            Pfloat      spec_exp;       /* specular exponent           */
        } simple_refl;
        Pdata          unsupp;         /* unsupported reflectance     */
                                          /* property data               */
        ...            /* implementation defined      */
    } data;
} Prefl_prop;
```

Psurf_approx_crit SURFACE APPROXIMATION CRITERIA

```

typedef struct {
    Pint          type;          /* surface approximation criteria */
                                /* type                               */
    union Psurf_approx_crit_data {
        struct Psurf_approx_const_param_subd {
            Pint      u_count;    /* u count                        */
            Pint      v_count;    /* v count                        */
        } const_param_subd;
        struct Psurf_approx_chordal_size_wc {
            Pfloat     u_approx;   /* u approximation value         */
            Pfloat     v_approx;   /* v approximation value         */
        } chordal_size_wc;
        struct Psurf_approx_chordal_size_npc {
            Pfloat     u_approx;   /* u approximation value         */
            Pfloat     v_approx;   /* v approximation value         */
        } chordal_size_npc;
        struct Psurf_approx_chordal_aize_dc {
            Pfloat     u_approx;   /* u approximation value         */
            Pfloat     v_approx;   /* v approximation value         */
        } chordal_size_dc;
        struct Psurf_approx_planar_dev_wc {
            Pfloat     approx;     /* approximation value           */
        } planar_dev_wc;
        struct Psurf_approx_planar_dev_npc {
            Pfloat     approx;     /* approximation value           */
        } planar_dev_npc;
        struct Psurf_approx_planar_dev_dc {
            Pfloat     approx;     /* approximation value           */
        } planar_dev_dc;
        struct Psurf_approx_rel_wc {
            Pfloat     approx;     /* approximation value           */
        } rel_wc;
        struct Psurf_approx_rel_npc {
            Pfloat     approx;     /* approximation value           */
        } rel_npc;
        struct Psurf_approx_rel_dc {
            Pfloat     approx;     /* approximation value           */
        } rel_dc;
        Pdata         unsupp;      /* unsupported surface           */
                                /* approximation criteria data   */
        ...            /* implementation defined        */
    } data;
} Psurf_approx_crit;

```

8.4 Implementation independent PHIGS PLUS type definitions

Pcolr_ctrl_point_array COLOUR CONTROL POINT ARRAY

```
typedef struct {
    Prationality      rationality;    /* control point rationality    */
    Pint_size         dims;           /* dimensions of control point  */
                                /* array                        */
    Pcolr_ctrl_point_ptr colr_points; /* array of colour control points */
} Pcolr_ctrl_point_array;
```

Pcolr_ctrl_point_list COLOUR CONTROL POINT LIST

```
typedef struct {
    Prationality      rationality;    /* control point rationality    */
    Pint              num_points;     /* number of control points     */
    Pcolr_ctrl_point_ptr points;      /* list of colour control points */
} Pcolr_ctrl_point_list;
```

Pcolr_ctrl_point_ptr COLOUR CONTROL POINT POINTER

```
typedef union {
    Pcolr_rep_ptr      points;         /* pointer to colour points     */
    Phomo_colr_rep_ptr homo_points;    /* pointer to homogeneous colour */
                                /* points                       */
} Pcolr_ctrl_point_ptr;
```

Pcolrv COLOUR VALUE

```
typedef union {
    Pint              colr_ind;        /* colour index                 */
    Pcolr_rep         colr_rep;       /* colour representation        */
} Pcolrv;
```

Pcolrv_array COLOUR VALUE ARRAY

```
typedef struct {  
    Pint_size      dims;           /* colour value array dimensions */  
    Pcolrv_ptr     colrs;          /* array of colour values        */  
} Pcolrv_array;
```

Pcolrv_list COLOUR VALUE LIST

```
typedef struct {  
    Pint           num_colrs;       /* number of colours            */  
    Pcolrv_ptr     colrs;          /* list of colour values        */  
} Pcolrv_list;
```

Pcolrv_ptr COLOUR VALUE POINTER

```
typedef union {  
    Pint           *colr_inds;      /* pointer to colour indices    */  
    Pcolrv_ptr     colr_reps;       /* pointer to colour            */  
                                /* representations              */  
} Pcolrv_ptr;
```

Pcolrv_set COLOUR VALUE SET

```
typedef struct {  
    Pint           num_lists;       /* number of colour value lists */  
    Pcolrv_list    *colrs;         /* list of colour value lists   */  
} Pcolrv_set;
```

Pcontour_set CONTOUR SET

```
typedef struct {  
    Pint          num_lists;          /* number of contour lists      */  
    Pint_list     *contours;         /* list of contour lists       */  
} Pcontour_set;
```

Pcontour_set_list CONTOUR SET LIST

```
typedef struct {  
    Pint          num_contour_sets;   /* number of contour sets      */  
    Pcontour_set  *contours;         /* list of contour sets       */  
} Pcontour_set_list;
```

Pctrl_point3_array CONTROL POINT 3 ARRAY

```
typedef struct {  
    Prationality   rationality;       /* control point rationality   */  
    Pint_size      dims;              /* control point array u,v     */  
                                   /* dimensions                  */  
    Ppoint3_ptr    points;            /* array of control points     */  
} Pctrl_point3_array;
```

Pctrl_point_list CONTROL POINT LIST

```
typedef struct {  
    Prationality   rationality;       /* control point rationality   */  
    Pint          num_points;         /* number of control points    */  
    Ppoint_ptr     points;            /* list of control points      */  
} Pctrl_point_list;
```

Pctrl_point3_list CONTROL POINT 3 LIST

```
typedef struct {  
    Prationality    rationality;    /* control point rationality    */  
    Pint            num_points;     /* number of control points    */  
    Ppoint3_ptr     points;         /* list of control points      */  
} Pctrl_point3_list;
```

Pcurve_colr_spline CURVE COLOUR SPLINE

```
typedef struct {  
    Pint            order;          /* spline order                */  
    Pfloat_list     knots;         /* list of knot values         */  
    Pint            colr_type;     /* colour type                 */  
    Pcolr_ctrl_point_list colrs;   /* list of colour control points */  
} Pcurve_colr_spline;
```

Pcurve_geom_spline CURVE GEOMETRY SPLINE

```
typedef struct {  
    Pint            order;          /* spline order                */  
    Pfloat_list     knots;         /* knots                      */  
    Pfloat          low_limit;     /* lower parameter range limit */  
    Pfloat          high_limit;    /* high parameter range limit  */  
    Pctrl_point3_list ctrl_points; /* list of control points      */  
} Pcurve_geom_spline;
```

Pdata_ctrl_point_array DATA CONTROL POINT ARRAY

```

typedef struct {
    Prationality    rationality;          /* control point rationality    */
    Pint_size      dims;                  /* control point array u,v      */
                                          /* dimensions                    */
    Pint           points_dims;           /* control point dimensionality */
    Pfloat         *ctrl_points;          /* array of control points      */
} Pdata_ctrl_point_array;

```

Pdepth_cue_mode DEPTH CUE MODE

```

typedef enum {
    PDEPTH_CUE_SUPPR,
    PDEPTH_CUE_ALLOWED
} Pdepth_cue_mode;

```

Pdepth_cue_mode_list DEPTH CUE MODE LIST

```

typedef struct {
    Pint          num_modes;              /* number of depth cue modes    */
    Pdepth_cue_mode *modes;              /* list of depth cue modes      */
} Pdepth_cue_mode_list;

```

Pdepth_cue_rep DEPTH CUE REPRESENTATION

```

typedef struct {
    Pdepth_cue_mode mode;                /* depth cue mode               */
    Pfloat          ref_planes[2];        /* depth cue reference planes    */
                                          /* (min, max)                   */
    Pfloat          scale[2];             /* depth cue scale factors      */
                                          /* (min, max)                   */
    Pgcolr          gcolr;               /* depth cue colour             */
} Pdepth_cue_rep;

```

Pdyn_ws_attrs_plus DYNAMICS OF WORKSTATION ATTRIBUTES PLUS

```

typedef struct {
    Pdyn_mod      data_map;           /* data mapping representation */
    Pdyn_mod      refl;              /* reflectance representation */
    Pdyn_mod      param_surf;        /* parametric surface          */
                                   /* representation              */
    Pdyn_mod      light_source;      /* light source representation */
    Pdyn_mod      depth_cue;         /* depth cue representation    */
    Pdyn_mod      colr_map;          /* colour mapping representation */
} Pdyn_ws_attrs_plus;

```

Pedge_bundle_plus EDGE BUNDLE PLUS

```

typedef struct {
    Pedge_flag     flag;              /* edge flag                   */
    Pint           type;              /* edgetype                    */
    Pfloat         width;             /* edgewidth scale factor      */
    Pgcolr         colr;              /* edge colour                  */
} Pedge_bundle_plus;

```

Pedge_flag_array EDGE FLAG ARRAY

```

typedef struct {
    Pint_size      dims;              /* edge flag array dimensions */
    Pedge_flag_pair *edge_flags;      /* array of edge flag pairs    */
} Pedge_flag_array;

```

Pedge_flag_list EDGE FLAG LIST

```

typedef struct {
    Pint           num_edges;          /* number of edges in list     */
    Pedge_flag     *edge_flags;        /* list of edge flags          */
} Pedge_flag_list;

```

Pedge_flag_pair EDGE FLAG PAIR

```
typedef Pedge_flag Pedge_flag_pair[2];
```

Pedge_flag_set EDGE FLAG SET

```
typedef struct {  
    Pint          num_lists;          /* number of edge flag lists      */  
    Pedge_flag_list *edge_flags;      /* list of edge flag lists       */  
} Pedge_flag_set;
```

Pedge_flag_set_list EDGE FLAG SET LIST

```
typedef struct {  
    Pint          num_edge_flag_sets; /* number of edge flag sets      */  
    Pedge_flag_set *edge_flag_sets;  /* list of edge flag sets        */  
} Pedge_flag_set_list;
```

Pedge_flag_triplet EDGE FLAG TRIPLET

```
typedef Pedge_flag Pedge_flag_triplet[3];
```

Pedge_flag_triplet_list EDGE FLAG TRIPLET LIST

```
typedef struct {  
    Pint          num_edge_flag_triplets; /* number of edge flag triplets */  
    Pedge_flag_triplet *edge_flag_triplets; /* list of edge flag triplets   */  
} Pedge_flag_triplet_list;
```

Pfacet FACET

```
typedef struct {  
    Pint          num_data_per_facet; /* number of data values per-facet*/  
    Pcolrv        *facet_colrv;      /* facet colour */  
    Pvec3         *facet_norm;       /* facet normal */  
    Pfloat        *facet_data;       /* list of facet data */  
} Pfacet;
```

NOTE — A NULL value for the facet_colrv, facet_norm, or facet_data fields indicates the corresponding optional data is not supplied.

Pfacet_array FACET ARRAY

```
typedef struct {  
    Pint_size      dims;              /* facet array dimensions */  
    Pint          num_data_per_facet; /* number of data values per-facet*/  
    Pcolrv_ptr     facet_colrvs;      /* array of facet colour values */  
    Pvec3         *facet_norms;       /* array of facet normals */  
    Pfloat        *facet_data;       /* array of facet data lists */  
} Pfacet_array;
```

NOTE — A NULL value for the facet_colrvs, facet_norms, or facet_data fields indicates the corresponding optional data is not supplied.

Pfacet_cull_mode FACET CULLING MODE

```
typedef enum {  
    PFACET_CULL_NONE,  
    PFACET_CULL_BACKFACING,  
    PFACET_CULL_FRONTFACING  
} Pfacet_cull_mode;
```

Pfacet_disting_mode FACET DISTINGUISHING MODE

```
typedef enum {  
    PFACET_DISTING_OFF,  
    PFACET_DISTING_ON  
} Pfacet_disting_mode;
```

Pfacet_list FACET LIST

```
typedef struct {  
    Pint          num_facets;           /* number of facets          */  
    Pint          num_data_per_facet;   /* number of data values per-facet */  
    Pcolrv_ptr    facet_colrvs;        /* list of facet colour values */  
    Pvec3         *facet_norms;        /* list of facet normals      */  
    Pfloat        *facet_data;         /* list of facet data lists   */  
} Pfacet_list;
```

NOTE — A NULL value for the facet_colrvs, facet_norms, or facet_data fields indicates the corresponding optional data is not supplied.

Pfloat_list FLOAT LIST

```
typedef struct {  
    Pint          num_floats;           /* number of floats in list   */  
    Pfloat        *floats;             /* list of floats             */  
} Pfloat_list;
```

Pfloat_set FLOAT SET

```
typedef struct {  
    Pint          num_lists;           /* number of float lists      */  
    Pfloat_list   *floats;            /* list of float lists        */  
} Pfloat_set;
```

Pgcolr GENERAL COLOUR

```
typedef struct {  
    Pint          colr_type;          /* colour type          */  
    Pcolrv        colr_value;        /* colour value        */  
} Pgcolr;
```

Phomo_cieluv HOMOGENEOUS CIEL*U*V*

```
typedef struct {  
    Pfloat        cieluv_x;          /* x coefficient        */  
    Pfloat        cieluv_y;          /* y coefficient        */  
    Pfloat        cieluv_y_lum;      /* y luminance          */  
    Pfloat        w;                 /* homogeneous coordinate */  
} Phomo_cieluv;
```

Phomo_hls HOMOGENEOUS HUE LIGHTNESS SATURATION

```
typedef struct {  
    Pfloat        hue;               /* hue                  */  
    Pfloat        lightness;         /* lightness            */  
    Pfloat        satur;             /* saturation           */  
    Pfloat        w;                 /* homogeneous coordinate */  
} Phomo_hls;
```

Phomo_hsv HOMOGENEOUS HUE SATURATION VALUE

```
typedef struct {  
    Pfloat        hue;               /* hue                  */  
    Pfloat        satur;             /* saturation           */  
    Pfloat        value;             /* value                */  
    Pfloat        w;                 /* homogeneous coordinate */  
} Phomo_hsv;
```

Phomo_point3 HOMOGENEOUS POINT 3

```
typedef struct {
    Pfloat      wx;           /* wx coordinate          */
    Pfloat      wy;           /* wy coordinate          */
    Pfloat      w;           /* homogeneous coordinate */
} Phomo_point3;
```

Phomo_point4 HOMOGENEOUS POINT 4

```
typedef struct {
    Pfloat      wx;           /* wx coordinate          */
    Pfloat      wy;           /* wy coordinate          */
    Pfloat      wz;           /* wz coordinate          */
    Pfloat      w;           /* homogeneous coordinate */
} Phomo_point4;
```

Phomo_rgb HOMOGENEOUS RED GREEN BLUE

```
typedef struct {
    Pfloat      red;          /* red                    */
    Pfloat      green;        /* green                  */
    Pfloat      blue;         /* blue                   */
    Pfloat      w;           /* homogeneous coordinate */
} Phomo_rgb;
```

Pint_bundle_plus INTERIOR BUNDLE PLUS

```
typedef struct {
    Pint_style   style;       /* interior style          */
    Pint         style_ind;    /* interior style index    */
    Pgcolr       colr;        /* interior colour         */
    Pint         shad_method;  /* shading method          */
} Pint_bundle_plus;
```

Pint_facs_plus INTERIOR FACILITIES PLUS

```

typedef struct {
    Pint          num_int_styles;      /* number of interior styles */
    Pint_style    int_styles[5];      /* list of available interior */
                                          /* styles */
    Pint          num_hatch_styles;    /* number of available hatch */
                                          /* styles */
    Pint_list     hatch_styles;        /* list of available hatch styles */
    Pint_list     shad_methods;        /* list of available shading */
                                          /* methods */
    Pint          num_pred_int_inds;   /* number of predefined interior */
                                          /* indices */

} Pint_facs_plus;

```

Pline_bundle_plus POLYLINE BUNDLE PLUS

```

typedef struct {
    Pint          type;                /* line type */
    Pfloat        width;               /* linewidth scale factor */
    Pcolour       colour;              /* polyline colour */
    Pint          shad_method;         /* polyline shading method */
    Pcurve_approx_crit approx_crit;    /* curve approximation criteria */
                                          /* data record */

} Pline_bundle_plus;

```