9593 PT*1 90 ■ 4851903 0572942 768 ■

INTERNATIONAL STANDARD ISO/IEC 9593-3:1990
TECHNICAL CORRIGENDUM 2

Published 1994-08-01

INTERNATIONAL ORGANIZATION FOR STANDARDIZATION • МЕЖДУНАРОДНАЯ ОРГАНИЗАЦИЯ ПО СТАНДАРТИЗАЦИИ • ORGANISATION INTERNATIONALE DE NORMALISATION
INTERNATIONAL ELECTROTECHNICAL COMMISSION • МЕЖДУНАРОДНАЯ ЭЛЕКТРОТЕХНИЧЕСКАЯ КОМИССИЯ • COMMISSION ELECTROTECHNIQUE INTERNATIONALE

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

Part 3:
Ada

TECHNICAL CORRIGENDUM 2

Technologies de l'information — Infographie — Interfaces langage avec PHIGS —

Partie 3: Ada

RECTIFICATIF TECHNIQUE 2

Technical corrigendum 2 to International Standard ISO/IEC 9593-3:1990 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 24, *Computer graphics and image processing*.

Page 46

Subclause 4.2.2

Change the comment associated with PATH_ORDER to read:

```
-- Provides for specifying the order of pick paths.
```

UDC 681.3.01:681.3.072.2(084):800.92 **Ref. No. ISO/IEC 9593-3:1990/Cor.2:1994(E)**

Descriptors : data processing, information interchange, graphic data processing, programming (computers), programming languages, Ada.

© **ISO/IEC 1994**

Printed in Switzerland

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS

Not for Resale

Page 62

Subclause 4.2.2

Add the following text after the TRANSFORMATION_MATRIX_3 entry:

TRUNCATION_METHOD

type TRUNCATION_METHOD is (HEAD, TAIL);

-- Provides for specifying if the HEAD or TAIL portion of a reference path should be returned.

Page 97

Subclause 5.8

RETRIEVE PATHS TO ANCESTORS, change the third parameter from

ORDER : in PATH_ORDER;

to

TRUNCATION : in TRUNCATION_METHOD;

RETRIEVE PATHS TO DESCENDANTS, change the third parameter from

ORDER : in PATH_ORDER;

to

TRUNCATION : in TRUNCATION_METHOD;

Page 130

Subclause 5.11

INQUIRE PATHS TO ANCESTORS, change the second parameter from

ORDER : in PATH_ORDER;

to

TRUNCATION : in TRUNCATION_METHOD;

Page 131

Subclause 5.11

INQUIRE PATHS TO DESCENDANTS, change the second parameter from

ORDER : in PATH_ORDER;

to

TRUNCATION : in TRUNCATION_METHOD;

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Page 185

Annex A

Change the comment associated with PATH_ORDER to read:

-- Provides for specifying the order of pick paths.

Page 192

Annex A

Add the following text after the TRANSFORMATION_MATRIX_3 entry:

TRUNCATION_METHOD

type TRUNCATION_METHOD is (HEAD, TAIL);

-- Provides for specifying if the HEAD or TAIL portion of a reference path should be returned.

Page 213

Annex A

RETRIEVE PATHS TO ANCESTORS, change the third parameter from

ORDER : in PATH_ORDER;

to

TRUNCATION : in TRUNCATION_METHOD;

RETRIEVE PATHS TO DESCENDANTS, change the third parameter from

ORDER : in PATH_ORDER;

to

TRUNCATION : in TRUNCATION_METHOD;

Page 232

Annex A

INQUIRE PATHS TO ANCESTORS, change the second parameter from

ORDER : in PATH_ORDER;

to

TRUNCATION : in TRUNCATION_METHOD;

Page 233

Annex A

INQUIRE PATHS TO DESCENDANTS, change the second parameter from

ORDER : in PATH_ORDER;

to

TRUNCATION : in TRUNCATION_METHOD;

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567187 227 ■

INTERNATIONAL STANDARD

ISO/IEC 9593-3

First edition
1990-04-15

AMENDMENT 1
1994-06-15

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

Part 3:
Ada

AMENDMENT 1: Incorporation of PHIGS PLUS

Technologies de l'information — Infographie — Interfaces langage avec système graphique hiérarchisé interactif de programmation —

Partie 3: Ada

AMENDEMENT 1: Incorporation du PHIGS PLUS

Reference number
ISO/IEC 9593-3:1990/Amd.1:1994(E)

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS

Not for Resale

Contents

Page

Foreword v

Introduction vi

1 Scope 1

2 Normative references 2

3 Principles 3
- 3.1 Conformance 3
- 3.2 Implications of the Language 3
  - 3.2.1 Functional Mapping 3
  - 3.2.2 Implementations and Host Dependencies 3
  - 3.2.3 Error Handling 4
  - 3.2.4 Data Mapping 4
  - 3.2.5 Multi-tasking 4
  - 3.2.6 Packaging 4
  - 3.2.7 Application Program Environment 4
  - 3.2.8 Registration 4

4 Tables 5
- 4.1 Abbreviations used in procedure names 5
  - 4.1.1 List of procedures using the abbreviations 5
  - 4.1.2 Alphabetical by bound name 6
  - 4.1.3 Alphabetical PHIGS functions 8
- 4.2 Data type definitions 8
  - 4.2.1 Abbreviations used in the data type definitions 8
  - 4.2.2 Alphabetical list of type definitions 8
  - 4.2.3 Alphabetical list of private type definitions 8
  - 4.2.4 List of constant declarations 8
  - 4.2.5 PHIGS configuration values 8
- 4.3 Error Codes 9

© 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

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

9593 PT*3 90 ■ 4851903 0567189 0TT ■

4.3.1 Precluded Error Codes 9
4.3.2 Binding Specific Error Codes 9

5 Functions in the Ada Binding of PHIGS 10
5.1 Control functions 10
5.2 Output primitive functions 10
5.3 Attribute specification functions 10
5.4 Transformation and clipping functions 10
5.5 Structure content functions 10
5.6 Structure manipulation functions 11
5.7 Structure display functions 11
5.8 Structure archive functions 11
5.9 Input functions 11
5.10 Metafile functions 11
5.11 Inquiry functions 11
5.12 Error control functions 11
5.13 Special interface functions 12
5.14 Additional functions 12
5.14.1 Subprograms for manipulating input data records 12
5.14.2 PHIGS generic coordinate system package 12
5.14.3 PHIGS generic list utility package 12
5.14.4 PHIGS name set facility package 12
5.14.5 Deallocation of structure element records 12
5.14.6 Metafile function utilities 13
5.15 Conformal variants 13

6 Tables for PHIGS PLUS 14
6.1 Data type definitions 14
6.1.1 Abbreviations used in the data type definitions 14
6.1.2 Replacement definition for type ASPECT 14
6.1.3 Replacement definition for type ATTRIBUTES_USED_TYPE 16
6.1.4 Replacement definition for type ELEMENT_TYPE 16
6.1.5 Replacement definition for type STRUCTURE_ELEMENT_RECORD 20
6.1.6 Additions to alphabetical list of PHIGS type definitions 30
6.1.7 Additions to list of constant declarations 67
6.1.8 PHIGS PLUS configuration values 69

7 Functions in the Ada Binding of PHIGS PLUS 70
7.1 Output primitive functions 70
7.2 Attribute specification functions 75
7.3 Inquiry functions 82
7.4 Additional functions 92
7.4.1 Changes to PHIGS generic coordinate system package 93
7.4.1 Additions to PHIGS generic coordinate system package 93
7.4.2 PHIGS PLUS generic colour package 105
7.4.3 Deallocation of PHIGS PLUS structure element records 108

Compilable PHIGS Specification 110

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567190 811 ■

Cross Reference Listing of Implementation Defined Items 298

Example Programs 299
C.1 Example Program 1: STAR 299
C.2 Example Program 2: IRON 299
C.3 Example Program 3: DYNASTAR 299
C.4 Example Program 4: TRANSFORM_POLYLINE 299
C.5 Example Program 5: SHOW_LINETYPES 299
C.6 Example Program 6: DODECAHEDRON 300
C.7 Example Program 7: TRIMMED_SURFACE 308

PHIGS Multi-Tasking 313

Index 314

--`,,`-`-`,,`,,`,`,,`---

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567191 758 ■

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 take part in this 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 9593-3:1990 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology.*

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567192 694 ■

Introduction

page vi: The following text should replace the text in the Introduction.

Part 1 of PHIGS, 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 document is to define a standard binding of ISO/IEC 9592-4 to the Ada computer programming language.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567193 520 ■

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

Part 3:
Ada

AMENDMENT 1: Incorporation of PHIGS PLUS

page 1: The following phrase should be inserted on a line following the word "Ada" in the title.

to include PHIGS Part 4 (PHIGS PLUS)

1 Scope

page 1: The following text should replace the text in clause 1 of ISO/IEC 9593-3:

ISO/IEC 9592-1 and 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 Ada computer programming language.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

9593 PT*3 90 ■ 4851903 0567194 467 ■

2 Normative references

page 2: The following reference should be added:

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).*

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567195 3T3 ■

3 Principles

page 3: No changes.

3.1 Conformance

page 3: The following should be added to the list:

- To conform with PHIGS, the implementation shall correctly implement the binding defined in clauses 4 and 5; to conform with PHIGS PLUS, the implementation shall correctly implement the binding defined in clauses 4, 5, 6, and 7.

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

3.2 Implications of the Language

page 3: No changes.

3.2.1 Functional Mapping

pages 3 and 4: No changes.

3.2.2 Implementations and Host Dependencies

page 4: No changes.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

3.2.3 Error Handling

page 4: No changes.

3.2.4 Data Mapping

pages 4 to 6: No changes.

3.2.5 Multi-tasking

page 6: No changes.

3.2.6 Packaging

page 6 and 7: No changes.

3.2.7 Application Program Environment

page 7: No changes.

3.2.8 Registration

page 7: No changes.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567197 176 ■

4 Tables

page 8: No changes.

4.1 Abbreviations used in procedure names

page 8: No changes.

4.1.1 List of procedures using the abbreviations

pages 8 to 10: The following should be added to the list of procedures using the abbreviation INQ:

INQ	INQ_COLOUR_MAPPING_FACILITIES
	INQ_COLOUR_MAPPING_METHOD_FACILITIES
	INQ_COLOUR_MAPPING_REPRESENTATION
	INQ_COLOUR_MAPPING_STATE
	INQ_CURVE_AND_SURFACE_FACILITIES
	INQ_DATA_MAPPING_FACILITIES
	INQ_DATA_MAPPING_REPRESENTATION
	INQ_DEPTH_CUE_FACILITIES
	INQ_DEPTH_CUE_REPRESENTATION
	INQ_DIRECT_COLOUR_MODEL_FACILITIES
	INQ_DYNAMICS_OF_WS_ATTRIBUTES (PLUS)
	INQ_EDGE_REPRESENTATION (PLUS)
	INQ_INTERIOR_FACILITIES (PLUS)
	INQ_INTERIOR_REPRESENTATION (PLUS)
	INQ_LIGHT_SOURCE_FACILITIES
	INQ_LIGHT_SOURCE_REPRESENTATION
	INQ_LIST_OF_COLOUR_MAPPING_INDICES
	INQ_LIST_OF_DATA_MAPPING_INDICES
	INQ_LIST_OF_DEPTH_CUE_INDICES
	INQ_LIST_OF_LIGHT_SOURCE_INDICES
	INQ_LIST_OF_PARAMETRIC_SURFACE_INDICES
	INQ_LIST_OF_REFLECTANCE_INDICES
	INQ_PARAMETRIC_SURFACE_REPRESENTATION
	INQ_PATTERN_REPRESENTATION (PLUS)
	INQ_POLYLINE_FACILITIES (PLUS)
	INQ_POLYLINE_REPRESENTATION (PLUS)
	INQ_POLYMARKER_REPRESENTATION (PLUS)
	INQ_PREDEFINED_COLOUR_MAPPING_REPRESENTATION
	INQ_PREDEFINED_DATA_MAPPING_REPRESENTATION
	INQ_PREDEFINED_DEPTH_CUE_REPRESENTATION
	INQ_PREDEFINED_EDGE_REPRESENTATION (PLUS)
	INQ_PREDEFINED_INTERIOR_REPRESENTATION (PLUS)
	INQ_PREDEFINED_LIGHT_SOURCE_REPRESENTATION
	INQ_PREDEFINED_PARAMETRIC_SURFACE_REPRESENTATION

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

INQ_PREDEFINED_PATTERN_REPRESENTATION (PLUS)
INQ_PREDEFINED_POLYLINE_REPRESENTATION (PLUS)
INQ_PREDEFINED_POLYMARKER_REPRESENTATION (PLUS)
INQ_PREDEFINED_REFLECTANCE_REPRESENTATION
INQ_PREDEFINED_TEXT_REPRESENTATION (PLUS)
INQ_REFLECTANCE_FACILITIES
INQ_REFLECTANCE_REPRESENTATION
INQ_RENDERING_COLOUR_MODEL_FACILITIES
INQ_TEXT_REPRESENTATION (PLUS)
INQ_WS_STATE_TABLE_LENGTHS (PLUS)

page 10: The following should be added to the list of procedures using the abbreviation WS:

WS	INQ_DYNAMICS_OF_WS_ATTRIBUTES (PLUS)
	INQ_WS_STATE_TABLE_LENGTHS (PLUS)

4.1.2 Alphabetical by bound name

pages 11 to 15: The following list of functions should be added alphabetically to the alphabetical list of bound names:

CELL_ARRAY	cell array 3 plus
FILL_AREA_SET	fill area set 3 with data
FILL_AREA_SET	fill area set with data
INQ_COLOUR_MAPPING_FACILITIES	inquire colour mapping facilities
INQ_COLOUR_MAPPING_METHOD_FACILITIES	inquire colour mapping method facilities
INQ_COLOUR_MAPPING_REPRESENTATION	inquire colour mapping representation
INQ_COLOUR_MAPPING_STATE	inquire colour mapping state
INQ_CURVE_AND_SURFACE_FACILITIES	inquire curve and surface facilities
INQ_DATA_MAPPING_FACILITIES	inquire data mapping facilities
INQ_DATA_MAPPING_REPRESENTATION	inquire data mapping representation
INQ_DEPTH_CUE_FACILITIES	inquire depth cue facilities
INQ_DEPTH_CUE_REPRESENTATION	inquire depth cue representation
INQ_DIRECT_COLOUR_MODEL_FACILITIES	inquire direct colour model facilities
INQ_DYNAMICS_OF_WS_ATTRIBUTES	inquire dynamics of workstation attributes plus
INQ_EDGE_REPRESENTATION	inquire edge representation plus
INQ_INTERIOR_FACILITIES	inquire interior facilities plus
INQ_INTERIOR_REPRESENTATION	inquire interior representation plus
INQ_LIGHT_SOURCE_FACILITIES	inquire light source facilities
INQ_LIGHT_SOURCE_REPRESENTATION	inquire light source representation
INQ_LIST_OF_COLOUR_MAPPING_INDICES	inquire list of colour mapping indices
INQ_LIST_OF_DATA_MAPPING_INDICES	inquire list of data mapping indices
INQ_LIST_OF_DEPTH_CUE_INDICES	inquire list of depth cue indices
INQ_LIST_OF_LIGHT_SOURCE_INDICES	inquire list of light source indices
INQ_LIST_OF_PARAMETRIC_SURFACE_INDICES	inquire list of parametric surface indices
INQ_LIST_OF_REFLECTANCE_INDICES	inquire list of reflectance indices
INQ_PARAMETRIC_SURFACE_REPRESENTATION	inquire parametric surface representation
INQ_PATTERN_REPRESENTATION	inquire pattern representation plus
INQ_POLYLINE_FACILITIES	inquire polyline facilities plus
INQ_POLYLINE_REPRESENTATION	inquire polyline representation plus
INQ_POLYMARKER_REPRESENTATION	inquire polymarker representation plus
INQ_PREDEFINED_COLOUR_MAPPING_REPRESENTATION	inquire predefined colour mapping representation
INQ_PREDEFINED_DATA_MAPPING_REPRESENTATION	inquire predefined data mapping representation
INQ_PREDEFINED_DEPTH_CUE_REPRESENTATION	inquire predefined depth cue representation
INQ_PREDEFINED_EDGE_REPRESENTATION	inquire predefined edge representation plus
INQ_PREDEFINED_INTERIOR_REPRESENTATION	inquire predefined interior representation plus
INQ_PREDEFINED_LIGHT_SOURCE_REPRESENTATION	inquire predefined light source representation
INQ_PREDEFINED_PARAMETRIC_SURFACE_REPRESENTATION	inquire predefined parametric surface representation
INQ_PREDEFINED_PATTERN_REPRESENTATION	inquire predefined pattern representation plus
INQ_PREDEFINED_POLYLINE_REPRESENTATION	inquire predefined polyline representation plus
INQ_PREDEFINED_POLYMARKER_REPRESENTATION	inquire predefined polymarker representation plus
INQ_PREDEFINED_REFLECTANCE_REPRESENTATION	inquire predefined reflectance representation

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

INQ_PREDEFINED_TEXT_REPRESENTATION	inquire predefined text representation plus
INQ_REFLECTANCE_FACILITIES	inquire reflectance facilities
INQ_REFLECTANCE_REPRESENTATION	inquire reflectance representation
INQ_RENDERING_COLOUR_MODEL_FACILITIES	inquire rendering colour model facilities
INQ_TEXT_REPRESENTATION	inquire text representation plus
INQ_WS_STATE_TABLE_LENGTHS	inquire workstation state table lengths plus
NON_UNIFORM_B_SPLINE_CURVE	non-uniform B-spline curve
NON_UNIFORM_B_SPLINE_CURVE	non-uniform B-spline curve with colour
NON_UNIFORM_B_SPLINE_SURFACE	non-uniform B-spline surface
NON_UNIFORM_B_SPLINE_SURFACE	non_uniform B-spline surface with data
POLYLINE_SET	polyline set 3 with colour
QUADRILATERAL_MESH	quadrilateral mesh 3 with data
QUADRILATERAL_MESH	quadrilateral mesh with data
SET_BACK_DATA_MAPPING_INDEX	set back data mapping index
SET_BACK_DATA_MAPPING_METHOD	set back data mapping method
SET_BACK_INTERIOR_COLOUR	set back interior colour
SET_BACK_INTERIOR_INDEX	set back interior index
SET_BACK_INTERIOR_SHADING_METHOD	set back interior shading method
SET_BACK_INTERIOR_STYLE	set back interior style
SET_BACK_INTERIOR_STYLE_INDEX	set back interior style index
SET_BACK_REFLECTANCE_INDEX	set back reflectance index
SET_BACK_REFLECTANCE_MODEL	set back reflectance model
SET_BACK_REFLECTANCE_PROPERTIES	set back reflectance properties
SET_COLOUR_MAPPING_INDEX	set colour mapping index
SET_COLOUR_MAPPING_REPRESENTATION	set colour mapping representation
SET_CURVE_APPROXIMATION_CRITERIA	set curve approximation criteria
SET_DATA_MAPPING_INDEX	set data mapping index
SET_DATA_MAPPING_METHOD	set data mapping method
SET_DATA_MAPPING_REPRESENTATION	set data mapping representation
SET_DEPTH_CUE_INDEX	set depth cue index
SET_DEPTH_CUE_REPRESENTATION	set depth cue representation
SET_EDGE_COLOUR	set edge colour
SET_EDGE_REPRESENTATION	set edge representation plus
SET_FACET_CULLING_MODE	set facet culling mode
SET_FACET_DISTINGUISHING_MODE	set facet distinguishing mode
SET_INTERIOR_COLOUR	set interior colour
SET_INTERIOR_REPRESENTATION	set interior representation plus
SET_INTERIOR_SHADING_METHOD	set interior shading method
SET_LIGHT_SOURCE_REPRESENTATION	set light source representation
SET_LIGHT_SOURCE_STATE	set light source state
SET_OF_FILL_AREA_SETS	set of fill area sets 3 with data
SET_OF_FILL_AREA_SETS	set of fill area sets with data
SET_PARAMETRIC_SURFACE_CHARACTERISTICS	set parametric surface characteristics
SET_PARAMETRIC_SURFACE_INDEX	set parametric surface index
SET_PARAMETRIC_SURFACE_REPRESENTATION	set parametric surface representation
SET_PATTERN_REPRESENTATION	set pattern representation plus
SET_POLYLINE_COLOUR	set polyline colour
SET_POLYLINE_REPRESENTATION	set polyline representation plus
SET_POLYLINE_SHADING_METHOD	set polyline shading method
SET_POLYMARKER_COLOUR	set polymarker colour
SET_POLYMARKER_REPRESENTATION	set polymarker representation plus
SET_REFLECTANCE_INDEX	set reflectance index
SET_REFLECTANCE_MODEL	set reflectance model
SET_REFLECTANCE_PROPERTIES	set reflectance properties
SET_REFLECTANCE_REPRESENTATION	set reflectance representation
SET_RENDERING_COLOUR_MODEL	set rendering colour model
SET_SURFACE_APPROXIMATION_CRITERIA	set surface approximation criteria
SET_TEXT_COLOUR	set text colour
SET_TEXT_REPRESENTATION	set text representation plus
TRIANGLE_SET	triangle set 3 with data
TRIANGLE_SET	triangle set with data
TRIANGLE_STRIP	triangle strip 3 with data
TRIANGLE_STRIP	triangle strip with data

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

© ISO/IEC

4.1.3 Alphabetical PHIGS functions

page 15: No changes.

4.2 Data type definitions

page 15: No changes.

4.2.1 Abbreviations used in the data type definitions

page 16: No changes.

4.2.2 Alphabetical list of type definitions

pages 16 to 66: No changes.

4.2.3 Alphabetical list of private type definitions

pages 66 to 68: No changes.

4.2.4 List of constant declarations

pages 68 to 69: No changes.

4.2.5 PHIGS configuration values

pages 69 to 71: No changes.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

4.3 Error Codes

page 72: No changes.

4.3.1 Precluded Error Codes

page 72: No changes.

page 72: The following text should be added after clause 4.3.1 as clause 4.3.2 of ISO/IEC 9593-3.

4.3.2 Binding Specific Error Codes

The following binding specific error has been defined for use with this binding:

2502 Ignoring function, the parameters have inconsistent dimensions.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567202 363 ■

5 Functions in the Ada Binding of PHIGS

page 73: No changes.

5.1 Control functions

pages 73 and 74: No changes.

5.2 Output primitive functions

pages 74 to 77: No changes.

5.3 Attribute specification functions

pages 78 to 84: No changes.

5.4 Transformation and clipping functions

pages 85 to 91: No changes.

5.5 Structure content functions

pages 91 to 94: No changes.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567203 2TT ■

5.6 Structure manipulation functions

pages 94 and 95: No changes.

5.7 Structure display functions

page 95: No changes.

5.8 Structure archive functions

pages 95 to 97: No changes.

5.9 Input functions

pages 98 to 106: No changes.

5.10 Metafile functions

pages 106 and 107: No changes.

5.11 Inquiry functions

pages 107 to 132: No changes.

5.12 Error control functions

page 132: No changes.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

5.13 Special interface functions

pages 133 and 134: No changes.

5.14 Additional functions

page 134: No changes.

5.14.1 Subprograms for manipulating input data records

pages 134 to 138: No changes.

5.14.2 PHIGS generic coordinate system package

pages 138 and 141: No changes.

5.14.3 PHIGS generic list utility package

pages 141 to 143: No changes.

5.14.4 PHIGS name set facility package

pages 144 to 147: No changes.

5.14.5 Deallocation of structure element records

pages 147 to 149: No changes.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567205 072 ■

5.14.6 Metafile function utilities

page 149: No changes.

5.15 Conformal variants

page 150: No changes.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

page 150: The following text should be added after clause 5 as clauses 6 and 7 of ISO/IEC 9593-3:

6 Tables for PHIGS PLUS

6.1 Data type definitions

This clause contains modifications to type definitions used for PHIGS as well as new type definitions used to support PHIGS PLUS.

6.1.1 Abbreviations used in the data type definitions

page 16: The following should be added to the list of abbreviations used in the data type definitions in clause 4.2.1:

APPROX approximation

6.1.2 Replacement definition for type ASPECT

The following definition should replace the type ASPECT in 4.2.2:

ASPECT

type ASPECT is (TYPE_OF_LINE,
LINEWIDTH_SF,
LINE_COLOUR,

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567207 945 ■

TYPE_OF_MARKER,
SIZE,
MARKER_COLOUR,

FONT,
PRECISION,
EXPANSION,
SPACING,
TEXT_COLOUR,

STYLE_OF_INTERIOR,
STYLE_IND,
INTERIOR_COLOUR,

FLAG,
TYPE_OF_EDGE,
EDGEWIDTH_SF,
EDGE_COLOUR,

GENERAL_POLYLINE_COLOUR,
GENERAL_POLYMARKER_COLOUR,
GENERAL_TEXT_COLOUR,
GENERAL_INTERIOR_COLOUR,
GENERAL_EDGE_COLOUR,

METHOD_OF_POLYLINE_SHADING,
METHOD_OF_INTERIOR_SHADING,
METHOD_OF_DATA_MAPPING,
REFLECTANCE_PROPERTIES,
MODEL_OF_REFLECTANCE,

BACK_STYLE_OF_INTERIOR,
BACK_STYLE_IND,
BACK_INTERIOR_COLOUR_DIR,
BACK_METHOD_OF_INTERIOR_SHADING,
BACK_METHOD_OF_DATA_MAPPING,
BACK_REFLECTANCE_PROPERTIES,
BACK_REFLECTANCE_MODEL,

CRITERIA_FOR_CURVE_APPROX,
CRITERIA_FOR_SURFACE_APPROX,
CHARACTERISTICS_OF_PARAMETRIC_SURFACE);

-- This type lists the aspects for which an aspect source flag exists in PHIGS PLUS.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

6.1.3 Replacement definition for type ATTRIBUTES_USED_TYPE

The following definition should replace the type ATTRIBUTES_USED_TYPE in 4.2.2:

ATTRIBUTES_USED_TYPE

```
type ATTRIBUTES_USED_TYPE is ( POLYLINE_ATTRIBUTES,
                               POLYMARKER_ATTRIBUTES,
                               TEXT_ATTRIBUTES,
                               INTERIOR_ATTRIBUTES,
                               EDGE_ATTRIBUTES,
                               REFLECTANCE_ATTRIBUTES,
                               PARAMETRIC_SURFACE_ATTRIBUTES);

    -- The types of attributes which may be used in generating output for a GDP and in generating
    -- prompt and echo information for certain prompt and echo types of certain classes of input
    -- devices.
```

6.1.4 Replacement definition for type ELEMENT_TYPE

The following definition should replace the type ELEMENT_TYPE in 4.2.2:

ELEMENT_TYPE

```
type ELEMENT_TYPE is
        (ALL_ELEMENT_TYPES,
         NIL,

         POLYLINE_3,
         POLYLINE,

         POLYMARKER_3,
         POLYMARKER,

         TEXT_3,
         TEXT,

         ANNOTATION_TEXT_RELATIVE_3,
         ANNOTATION_TEXT_RELATIVE,
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567209 718 ■

FILL_AREA_3,
FILL_AREA,
FILL_AREA_SET_3,
FILL_AREA_SET,

CELL_ARRAY_3,
CELL_ARRAY,

GDP_3,
GDP,

SET_POLYLINE_INDEX,
SET_POLYMARKER_INDEX,
SET_TEXT_INDEX,
SET_INTERIOR_INDEX,
SET_EDGE_INDEX,

SET_LINETYPE,
SET_LINEWIDTH_SCALE_FACTOR,
SET_POLYLINE_COLOUR_INDEX,

SET_MARKER_TYPE,
SET_MARKER_SIZE_SCALE_FACTOR,
SET_POLYMARKER_COLOUR_INDEX,

SET_TEXT_FONT,
SET_TEXT_PRECISION,
SET_CHAR_EXPANSION_FACTOR,
SET_CHAR_SPACING,
SET_TEXT_COLOUR_INDEX,
SET_CHAR_HEIGHT,
SET_CHAR_UP_VECTOR,
SET_TEXT_PATH,
SET_TEXT_ALIGNMENT,

SET_ANNOTATION_TEXT_CHAR_HEIGHT,
SET_ANNOTATION_TEXT_CHAR_UP_VECTOR,
SET_ANNOTATION_TEXT_PATH,
SET_ANNOTATION_TEXT_ALIGNMENT,
SET_ANNOTATION_STYLE,

SET_INTERIOR_STYLE,
SET_INTERIOR_STYLE_INDEX,
SET_INTERIOR_COLOUR_INDEX,

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567210 43T ■

SET_EDGE_FLAG,
SET_EDGETYPE,
SET_EDGEWIDTH_SCALE_FACTOR,
SET_EDGE_COLOUR_INDEX,

SET_PATTERN_SIZE,
SET_PATTERN_REFERENCE_POINT_AND_VECTORS,
SET_PATTERN_REFERENCE_POINT,

ADD_NAMES_TO_SET,
REMOVE_NAMES_FROM_SET,

SET_INDIVIDUAL_ASF,
SET_HLHSR_IDENTIFIER,

SET_LOCAL_TRANSFORMATION_3,
SET_LOCAL_TRANSFORMATION,
SET_GLOBAL_TRANSFORMATION_3,
SET_GLOBAL_TRANSFORMATION,
SET_MODELLING_CLIPPING_VOLUME_3,
SET_MODELLING_CLIPPING_VOLUME,
SET_MODELLING_CLIPPING_INDICATOR,
RESTORE_MODELLING_CLIPPING_VOLUME,
SET_VIEW_INDEX,

EXECUTE_STRUCTURE,

LABEL,
APPLICATION_DATA,
GSE,
SET_PICK_IDENTIFIER,

POLYLINE_SET_3_WITH_COLOUR,
FILL_AREA_SET_3_WITH_DATA,
FILL_AREA_SET_WITH_DATA,
CELL_ARRAY_3_PLUS,
SET_OF_FILL_AREA_SETS_3_WITH_DATA,
SET_OF_FILL_AREA_SETS_WITH_DATA,
TRIANGLE_SET_3_WITH_DATA,
TRIANGLE_SET_WITH_DATA,
TRIANGLE_STRIP_3_WITH_DATA,
TRIANGLE_STRIP_WITH_DATA,
QUADRILATERAL_MESH_3_WITH_DATA,
QUADRILATERAL_MESH_WITH_DATA,
NON_UNIFORM_B_SPLINE_CURVE,
NON_UNIFORM_B_SPLINE_CURVE_WITH_COLOUR,
NON_UNIFORM_B_SPLINE_SURFACE,
NON_UNIFORM_B_SPLINE_SURFACE_WITH_DATA,

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567211 376 ■

SET_DATA_MAPPING_INDEX,
SET_REFLECTANCE_INDEX,

SET_BACK_INTERIOR_INDEX,
SET_BACK_DATA_MAPPING_INDEX,
SET_BACK_REFLECTANCE_INDEX,

SET_PARAMETRIC_SURFACE_INDEX,

SET_POLYLINE_COLOUR,
SET_POLYLINE_SHADING_METHOD,

SET_POLYMARKER_COLOUR,

SET_TEXT_COLOUR,

SET_FACET_DISTINGUISHING_MODE,
SET_FACET_CULLING_MODE,

SET_INTERIOR_COLOUR,
SET_INTERIOR_SHADING_METHOD,

SET_DATA_MAPPING_METHOD,
SET_REFLECTANCE_PROPERTIES,
SET_REFLECTANCE_MODEL,

SET_BACK_INTERIOR_STYLE,
SET_BACK_INTERIOR_STYLE_INDEX,
SET_BACK_INTERIOR_COLOUR,
SET_BACK_INTERIOR_SHADING_METHOD,
SET_BACK_DATA_MAPPING_METHOD,
SET_BACK_REFLECTANCE_PROPERTIES,
SET_BACK_REFLECTANCE_MODEL,

SET_LIGHT_SOURCE_STATE,

SET_EDGE_COLOUR,

SET_CURVE_APPROX_CRITERIA,
SET_SURFACE_APPROX_CRITERIA,
SET_PARAMETRIC_SURFACE_CHARACTERISTICS,

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
        SET_RENDERING_COLOUR_MODEL,
        SET_DEPTH_CUE_INDEX,
        SET_COLOUR_MAPPING_INDEX);

    -- This type lists the element types which exist in PHIGS PLUS.
```

6.1.5 Replacement definition for type STRUCTURE_ELEMENT_RECORD

The following definition should replace the type STRUCTURE_ELEMENT_RECORD in 4.2.2:

```
STRUCTURE_ELEMENT_RECORD

    type STRUCTURE_ELEMENT_RECORD
      (ELEMENT_TYPE : STRUCTURE_ELEMENT_TYPE := NIL) is
        record
          case ELEMENT_TYPE is

            -- The empty element

            when NIL =>
              null;

            -- PHIGS Primitive Elements

            when POLYLINE_3 =>
              POLYLINE_3_POINTS : MC.ACCESS_POINT_LIST_3;

            when POLYLINE =>
              POLYLINE_POINTS : MC.ACCESS_POINT_LIST_2;

            when POLYMARKER_3 =>
              POLYMARKER_3_POINTS : MC.ACCESS_POINT_LIST_3;

            when POLYMARKER =>
              POLYMARKER_POINTS : MC.ACCESS_POINT_LIST_2;

            when TEXT_3 =>
              TEXT_3_POINT : MC.POINT_3;
              TEXT_DIRECTION_VECTORS : MC.VECTOR_PAIR_3;
              TEXT_3_CHAR_STRING : ACCESS_STRING;

            when TEXT =>
              TEXT_POINT : MC.POINT_2;
              TEXT_CHAR_STRING : ACCESS_STRING;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567213 149 ■


```
when ANNOTATION_TEXT_RELATIVE_3 =>
  ANNOTATION_TEXT_RELATIVE_3_REF_POINT : MC.POINT_3;
  ANNOTATION_TEXT_RELATIVE_3_OFFSET : NPC.POINT_3;
  ANNOTATION_TEXT_3_CHAR_STRING : ACCESS_STRING;

when ANNOTATION_TEXT_RELATIVE =>
  ANNOTATION_TEXT_RELATIVE_REF_POINT : MC.POINT_2;
  ANNOTATION_TEXT_RELATIVE_OFFSET : NPC.POINT_2;
  ANNOTATION_TEXT_CHAR_STRING : ACCESS_STRING;

when FILL_AREA_3 =>
  FILL_AREA_3_POINTS : MC.ACCESS_POINT_LIST_3;

when FILL_AREA =>
  FILL_AREA_POINTS : MC.ACCESS_POINT_LIST_2;

when FILL_AREA_SET_3 =>
  FILL_AREA_SET_3_POINTS : MC.ACCESS_LIST_OF_POINT_LIST_3;

when FILL_AREA_SET =>
  FILL_AREA_SET_POINTS : MC.ACCESS_LIST_OF_POINT_LIST_2;

when CELL_ARRAY_3 =>
  CORNER_P_3 : MC.POINT_3;
  CORNER_Q_3 : MC.POINT_3;
  CORNER_R_3 : MC.POINT_3;
  CELL_ARRAY_3_CELLS : ACCESS_COLOUR_MATRIX;

when CELL_ARRAY =>
  CORNER_P : MC.POINT_2;
  CORNER_Q : MC.POINT_2;
  CELL_ARRAY_CELLS : ACCESS_COLOUR_MATRIX;

when GDP_3 =>
  GDP_3_POINTS : MC.ACCESS_POINT_LIST_3;
  GDP_3_DATA : GDP_3_RECORD;

when GDP =>
  GDP_POINTS : MC.ACCESS_POINT_LIST_2;
  GDP_DATA : GDP_RECORD;

-- PHIGS Bundle Index Elements

when SET_POLYLINE_INDEX =>
  POLYLINE_IND : POLYLINE_INDEX;

when SET_POLYMARKER_INDEX =>
  POLYMARKER_IND : POLYMARKER_INDEX;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567214 085 ■


```
when SET_TEXT_INDEX =>
  TEXT_IND : TEXT_INDEX;

when SET_INTERIOR_INDEX =>
  INTERIOR_IND : INTERIOR_INDEX;

when SET_EDGE_INDEX =>
  EDGE_IND : EDGE_INDEX;

-- PHIGS Individual Aspect Elements

when SET_LINETYPE =>
  TYPE_OF_LINE : LINETYPE;

when SET_LINEWIDTH_SCALE_FACTOR =>
  LINEWIDTH_SF : LINEWIDTH;

when SET_POLYLINE_COLOUR_INDEX =>
  LINE_COLOUR : COLOUR_INDEX;

when SET_MARKER_TYPE =>
  TYPE_OF_MARKER : MARKER_TYPE;

when SET_MARKER_SIZE_SCALE_FACTOR =>
  SIZE : MARKER_SIZE;

when SET_POLYMARKER_COLOUR_INDEX =>
  MARKER_COLOUR : COLOUR_INDEX;

when SET_TEXT_FONT =>
  FONT : TEXT_FONT;

when SET_TEXT_PRECISION =>
  PRECISION : TEXT_PRECISION;

when SET_CHAR_EXPANSION_FACTOR =>
  EXPANSION : CHAR_EXPANSION;

when SET_CHAR_SPACING =>
  SPACING : CHAR_SPACING;

when SET_TEXT_COLOUR_INDEX =>
  TEXT_COLOUR : COLOUR_INDEX;

when SET_CHAR_HEIGHT =>
  HEIGHT : MC.MAGNITUDE;

when SET_CHAR_UP_VECTOR =>
  CHAR_UP_VECTOR : MC.VECTOR_2;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
when SET_TEXT_PATH =>
  PATH : TEXT_PATH;

when SET_TEXT_ALIGNMENT =>
  ALIGNMENT : TEXT_ALIGNMENT;

when SET_ANNOTATION_TEXT_CHAR_HEIGHT =>
  ANNOTATION_HEIGHT : NPC.MAGNITUDE;

when SET_ANNOTATION_TEXT_CHAR_UP_VECTOR =>
  ANNOTATION_CHAR_UP_VECTOR : NPC.VECTOR_2;

when SET_ANNOTATION_TEXT_PATH =>
  ANNOTATION_PATH : TEXT_PATH;

when SET_ANNOTATION_TEXT_ALIGNMENT =>
  ANNOTATION_ALIGNMENT : TEXT_ALIGNMENT;

when SET_ANNOTATION_STYLE =>
  STYLE_OF_ANNOTATION : ANNOTATION_STYLE;

when SET_INTERIOR_STYLE =>
  STYLE_OF_INTERIOR : INTERIOR_STYLE;

when SET_INTERIOR_STYLE_INDEX =>
  STYLE_IND : STYLE_INDEX;

when SET_INTERIOR_COLOUR_INDEX =>
  INTERIOR_COLOUR : COLOUR_INDEX;

when SET_EDGE_FLAG =>
  FLAG : EDGE_FLAG;

when SET_EDGETYPE =>
  TYPE_OF_EDGE : EDGETYPE;

when SET_EDGEWIDTH_SCALE_FACTOR =>
  EDGEWIDTH_SF : EDGEWIDTH;

when SET_EDGE_COLOUR_INDEX =>
  EDGE_COLOUR : COLOUR_INDEX;

-- PHIGS Pattern Attribute Elements

when SET_PATTERN_SIZE =>
  PATTERN_SIZE : MC.SIZE_2;

when SET_PATTERN_REFERENCE_POINT_AND_VECTORS =>
  PATTERN_REFERENCE_POINT_3 : MC.POINT_3;
  PATTERN_REFERENCE_VECTORS : MC.VECTOR_PAIR_3;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567216 958 ■


```
when SET_PATTERN_REFERENCE_POINT =>
  PATTERN_REFERENCE_POINT : MC.POINT_2;

-- PHIGS Name Set Elements

when ADD_NAMES_TO_SET =>
  NAMES_TO_ADD : NAME_SET;

when REMOVE_NAMES_FROM_SET =>
  NAMES_TO_REMOVE : NAME_SET;

-- PHIGS ASF Elements

when SET_INDIVIDUAL_ASF =>
  ATTRIBUTE_ID : ASPECT;
  SOURCE_FLAG : ASF;

-- PHIGS HLHSR Elements

when SET_HLHSR_IDENTIFIER =>
  HLHSR_IDENTIFIER : HLHSR_ID;

-- PHIGS Transformation Elements

when SET_LOCAL_TRANSFORMATION_3 =>
  LOCAL_MATRIX_3 : TRANSFORMATION_MATRIX_3;
  HOW_APPLIED_3 : COMPOSITION_TYPE;

when SET_LOCAL_TRANSFORMATION =>
  LOCAL_MATRIX : TRANSFORMATION_MATRIX_2;
  HOW_APPLIED : COMPOSITION_TYPE;

when SET_GLOBAL_TRANSFORMATION_3 =>
  GLOBAL_MATRIX_3 : TRANSFORMATION_MATRIX_3;

when SET_GLOBAL_TRANSFORMATION =>
  GLOBAL_MATRIX : TRANSFORMATION_MATRIX_2;

when SET_MODELLING_CLIPPING_VOLUME_3 =>
  MODELLING_CLIPPING_OPERATOR_3
                    : MODELLING_CLIP_OPERATION_TYPE;
  MODELLING_CLIPPING_LIMITS_3
                    : MC.ACCESS_HALF_SPACE_LIST_3;

when SET_MODELLING_CLIPPING_VOLUME =>
  MODELLING_CLIPPING_OPERATOR
                    : MODELLING_CLIP_OPERATION_TYPE;
  MODELLING_CLIPPING_LIMITS : MC.ACCESS_HALF_SPACE_LIST_2;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
when SET_MODELLING_CLIPPING_INDICATOR =>
  MODELLING_CLIPPING_INDICATOR : CLIPPING_INDICATOR;

when RESTORE_MODELLING_CLIPPING_VOLUME =>
  null;

when SET_VIEW_INDEX =>
  VIEW_IND : VIEW_INDEX;

-- PHIGS Invocation Elements

when EXECUTE_STRUCTURE =>
  STRUCTURE_IDENTIFIER : STRUCTURE_ID;

-- PHIGS Structure Content Identification Elements

when LABEL =>
  LABEL_IDENTIFIER : LABEL_ID;

when APPLICATION_DATA =>
  DATA : APPLICATION_DATA_RECORD;

when GSE =>
  GSE_DATA : GSE_RECORD;

when SET_PICK_IDENTIFIER =>
  PICK_IDENTIFIER : PICK_ID;

-- PHIGS PLUS Primitive Elements

when POLYLINE_SET_3_WITH_COLOUR =>
  POLYLINE_SET_3_VERTICES
                        : MC.ACCESS_LIST_OF_VERTEX_COLOUR_LIST_SET_3;

when FILL_AREA_SET_3_WITH_DATA =>
  FILL_AREA_SET_3_FACET : MC.ACCESS_FACET_DATA_SET;
  FILL_AREA_SET_3_EDGES : ACCESS_LIST_OF_EDGE_FLAG_LIST;
  FILL_AREA_SET_3_VERTICES
                        : MC.ACCESS_LIST_OF_VERTEX_DATA_LIST_SET_3;

when FILL_AREA_SET_WITH_DATA =>
  FILL_AREA_SET_FACET : MC.ACCESS_FACET_DATA_SET;
  FILL_AREA_SET_EDGES : ACCESS_LIST_OF_EDGE_FLAG_LIST;
  FILL_AREA_SET_VERTICES
                        : MC.ACCESS_LIST_OF_VERTEX_DATA_LIST_SET_2;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
when CELL_ARRAY_3_PLUS =>
  CORNER_P_3_PLUS : MC.POINT_3;
  CORNER_Q_3_PLUS : MC.POINT_3;
  CORNER_R_3_PLUS : MC.POINT_3;
  CELL_ARRAY_3_PLUS_CELLS : ACCESS_COLOUR_VALUE_ARRAY;

when SET_OF_FILL_AREA_SETS_3_WITH_DATA =>
  SET_OF_FILL_AREA_SETS_3_FACETS
                        : MC.ACCESS_FACET_DATA_LIST_SET;
  SET_OF_FILL_AREA_SETS_3_EDGES
                        : ACCESS_LIST_OF_LIST_OF_EDGE_FLAG_LIST;
  SET_OF_FILL_AREA_SETS_3_VERTICES
                        : MC.ACCESS_VERTEX_DATA_LIST_SET_3;
  SET_OF_FILL_AREA_SETS_3_INDICES
                        : ACCESS_LIST_OF_LIST_OF_VERTEX_INDEX_LIST;

when SET_OF_FILL_AREA_SETS_WITH_DATA =>
  SET_OF_FILL_AREA_SETS_FACETS : MC.ACCESS_FACET_DATA_LIST_SET;
  SET_OF_FILL_AREA_SETS_EDGES
                        : ACCESS_LIST_OF_LIST_OF_EDGE_FLAG_LIST;
  SET_OF_FILL_AREA_SETS_VERTICES
                        : MC.ACCESS_VERTEX_DATA_LIST_SET_2;
  SET_OF_FILL_AREA_SETS_INDICES
                        : ACCESS_LIST_OF_LIST_OF_VERTEX_INDEX_LIST;

when TRIANGLE_SET_3_WITH_DATA =>
  TRIANGLE_SET_3_FACETS : MC.ACCESS_FACET_DATA_LIST_SET;
  TRIANGLE_SET_3_EDGES : ACCESS_LIST_OF_EDGE_FLAG_TRIPLET;
  TRIANGLE_SET_3_VERTICES : MC.ACCESS_VERTEX_DATA_LIST_SET_3;
  TRIANGLE_SET_3_INDICES
                        : ACCESS_LIST_OF_VERTEX_INDEX_TRIPLET;

when TRIANGLE_SET_WITH_DATA =>
  TRIANGLE_SET_FACETS : MC.ACCESS_FACET_DATA_LIST_SET;
  TRIANGLE_SET_EDGES : ACCESS_LIST_OF_EDGE_FLAG_TRIPLET;
  TRIANGLE_SET_VERTICES : MC.ACCESS_VERTEX_DATA_LIST_SET_2;
  TRIANGLE_SET_INDICES
                        : ACCESS_LIST_OF_VERTEX_INDEX_TRIPLET;

when TRIANGLE_STRIP_3_WITH_DATA =>
  TRIANGLE_STRIP_3_FACETS : MC.ACCESS_FACET_DATA_LIST_SET;
  TRIANGLE_STRIP_3_EDGES : ACCESS_EDGE_FLAG_LIST;
  TRIANGLE_STRIP_3_VERTICES
                        : MC.ACCESS_VERTEX_DATA_LIST_SET_3;

when TRIANGLE_STRIP_WITH_DATA =>
  TRIANGLE_STRIP_FACETS : MC.ACCESS_FACET_DATA_LIST_SET;
  TRIANGLE_STRIP_EDGES : ACCESS_EDGE_FLAG_LIST;
  TRIANGLE_STRIP_VERTICES : MC.ACCESS_VERTEX_DATA_LIST_SET_2;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567219 667 ■


```
when QUADRILATERAL_MESH_3_WITH_DATA =>
  QUADRILATERAL_MESH_3_FACETS
                        : MC.ACCESS_FACET_DATA_ARRAY_SET;
  QUADRILATERAL_MESH_3_EDGES
                        : ACCESS_ARRAY_OF_EDGE_FLAG_PAIR;
  QUADRILATERAL_MESH_3_VERTICES
                        : MC.ACCESS_VERTEX_DATA_ARRAY_SET_3;

when QUADRILATERAL_MESH_WITH_DATA =>
  QUADRILATERAL_MESH_FACETS
                        : MC.ACCESS_FACET_DATA_ARRAY_SET;
  QUADRILATERAL_MESH_EDGES
                        : ACCESS_ARRAY_OF_EDGE_FLAG_PAIR;
  QUADRILATERAL_MESH_VERTICES
                        : MC.ACCESS_VERTEX_DATA_ARRAY_SET_2;

when NON_UNIFORM_B_SPLINE_CURVE =>
  NURB_CURVE_SPLINE : MC.ACCESS_CURVE_GEOMETRYSPLINE_3;
  NURB_CURVE_LIMITS : SPC.RANGE_OF_MAGNITUDES;

when NON_UNIFORM_B_SPLINE_CURVE_WITH_COLOUR =>
  NURB_CURVE_COLOUR_GEOMETRYSPLINE
                        : MC.ACCESS_CURVE_GEOMETRYSPLINE_3;
  NURB_CURVE_COLOUR_LIMITS : SPC.RANGE_OF_MAGNITUDES;
  NURB_CURVE_COLOUR_COLOURSPLINE
                        : ACCESS_CURVE_COLOURSPLINE;

when NON_UNIFORM_B_SPLINE_SURFACE =>
  NURB_SURFACE_GEOMETRYSPLINE
                        : MC.ACCESS_SURFACE_GEOMETRYSPLINE;
  NURB_SURFACE_TRIM_CURVES
                        : ACCESS_LIST_OF_TRIMCURVE_LIST;

when NON_UNIFORM_B_SPLINE_SURFACE_WITH_DATA =>
  NURB_SURFACE_DATA_GEOMETRYSPLINE
                        : MC.ACCESS_SURFACE_GEOMETRYSPLINE;
  NURB_SURFACE_DATA_TRIM_CURVES
                        : ACCESS_LIST_OF_TRIMCURVE_LIST;
  NURB_SURFACE_DATA_COLOURSPLINE
                        : ACCESS_SURFACE_COLOURSPLINE;
  NURB_SURFACE_DATA_DATASPLINES
                              : ACCESS_DATASPLINE_LIST;

-- PHIGS PLUS Bundle Index Elements

when SET_DATA_MAPPING_INDEX =>
  DATA_MAPPING_IND : DATA_MAPPING_INDEX;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567220 389 ■


```
when SET_REFLECTANCE_INDEX =>
  REFLECTANCE_IND : REFLECTANCE_INDEX;

when SET_BACK_INTERIOR_INDEX =>
  BACK_INTERIOR_IND : INTERIOR_INDEX;

when SET_BACK_DATA_MAPPING_INDEX =>
  BACK_DATA_MAPPING_IND : DATA_MAPPING_INDEX;

when SET_BACK_REFLECTANCE_INDEX =>
  BACK_REFLECTANCE_IND : REFLECTANCE_INDEX;

when SET_PARAMETRIC_SURFACE_INDEX =>
  PARAMETRIC_SURFACE_IND : PARAMETRIC_SURFACE_INDEX;

-- PHIGS PLUS Individual Aspect Elements

when SET_POLYLINE_COLOUR =>
  GENERAL_POLYLINE_COLOUR : GENERAL_COLOUR;

when SET_POLYLINE_SHADING_METHOD =>
  POLYLINE_SHADING : POLYLINE_SHADING_METHOD;

when SET_POLYMARKER_COLOUR =>
  GENERAL_POLYMARKER_COLOUR : GENERAL_COLOUR;

when SET_TEXT_COLOUR =>
  GENERAL_TEXT_COLOUR : GENERAL_COLOUR;

when SET_FACET_DISTINGUISHING_MODE =>
  FACET_DISTINGUISHING : FACET_DISTINGUISHING_MODE;

when SET_FACET_CULLING_MODE =>
  FACET_CULLING : FACET_CULLING_MODE;

when SET_INTERIOR_COLOUR =>
  GENERAL_INTERIOR_COLOUR : GENERAL_COLOUR;

when SET_INTERIOR_SHADING_METHOD =>
  INTERIOR_SHADING : INTERIOR_SHADING_METHOD;

when SET_DATA_MAPPING_METHOD =>
  METHOD_OF_DATA_MAPPING : DATA_MAPPING_DATA_RECORD;

when SET_REFLECTANCE_PROPERTIES =>
  REFLECTANCE_PROPERTIES : REFLECTANCE_PROPERTIES_DATA_RECORD;

when SET_REFLECTANCE_MODEL =>
  MODEL_OF_REFLECTANCE : REFLECTANCE_MODEL;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567221 215 ■


```
when SET_BACK_INTERIOR_STYLE =>
  BACK_STYLE_OF_INTERIOR : INTERIOR_STYLE;

when SET_BACK_INTERIOR_STYLE_INDEX =>
  BACK_STYLE_IND : STYLE_INDEX;

when SET_BACK_INTERIOR_COLOUR =>
  BACK_GENERAL_INTERIOR_COLOUR : GENERAL_COLOUR;

when SET_BACK_INTERIOR_SHADING_METHOD =>
  BACK_INTERIOR_SHADING : INTERIOR_SHADING_METHOD;

when SET_BACK_DATA_MAPPING_METHOD =>
  BACK_METHOD_OF_DATA_MAPPING : DATA_MAPPING_METHOD;

when SET_BACK_REFLECTANCE_PROPERTIES =>
  BACK_REFLECTANCE_PROPERTIES
                        : REFLECTANCE_PROPERTIES_DATA_RECORD;

when SET_BACK_REFLECTANCE_MODEL =>
  BACK_REFLECTANCE : REFLECTANCE_MODEL;

when SET_LIGHT_SOURCE_STATE =>
  ACTIVATION_LIST : LIGHT_SOURCE_INDICES.LIST_OF;
  DEACTIVATION_LIST : LIGHT_SOURCE_INDICES.LIST_OF;

when SET_EDGE_COLOUR =>
  GENERAL_EDGE_COLOUR : GENERAL_COLOUR;

when SET_CURVE_APPROX_CRITERIA =>
  CRITERIA_FOR_CURVE_APPROX : CURVE_APPROX_DATA_RECORD;

when SET_SURFACE_APPROX_CRITERIA =>
  CRITERIA_FOR_SURFACE_APPROX
                        : SURFACE_APPROX_DATA_RECORD;

when SET_PARAMETRIC_SURFACE_CHARACTERISTICS =>
  PARAMETRIC_SURFACE_CHARACTERISTICS
                        : PARAMETRIC_SURFACE_DATA_RECORD;

when SET_RENDERING_COLOUR_MODEL =>
  RENDERING_COLOUR_MODEL : COLOUR_MODEL;

when SET_DEPTH_CUE_INDEX =>
  DEPTH_CUE_IND : DEPTH_CUE_INDEX;

when SET_COLOUR_MAPPING_INDEX =>
  COLOUR_MAPPING_IND : COLOUR_MAPPING_INDEX;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
        end case;
    end record;

-- This type defines the format of structure contents for PHIGS PLUS structures and is used to
-- return structure elements during inquiry.
```

6.1.6 Additions to alphabetical list of PHIGS type definitions

The following additional types should be added alphabetically to the alphabetical list of type definitions in 4.2.2:

ACCESS_ARRAY_OF_EDGE_FLAG_PAIR

```
type ACCESS_ARRAY_OF_EDGE_FLAG_PAIR is access ARRAY_OF_EDGE_FLAG_PAIR;

-- Provides for access variable to two-dimensional arrays of edge flag pairs. These will be
-- used when inquiring structure elements which are "Quadrilateral Mesh with Data"
-- primitives.
```

ACCESS_COLOUR_MAPPING_DATA_RECORD

```
type ACCESS_COLOUR_MAPPING_DATA_RECORD is
        access COLOUR_MAPPING_DATA_RECORD;

-- Provides for pointers to colour mapping data records.
```

ACCESS_COLOUR_VALUE_ARRAY

```
type ACCESS_COLOUR_VALUE_ARRAY is access COLOUR_VALUE_ARRAY;

-- Provides for pointers to arrays of general colour values.
```

ACCESS_COLOUR_VALUE_LIST

```
type ACCESS_COLOUR_VALUE_LIST is access COLOUR_VALUE_LIST;

-- Provides for pointers to lists of general colour values.
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567223 098 ■

ACCESS_CURVE_COLOURSPLINE

```
type ACCESS_CURVE_COLOURSPLINE is
        access CURVE_COLOURSPLINE;

-- Provides for pointers to colourspline definitions.
```

ACCESS_DATASPLINE_LIST

```
type ACCESS_DATASPLINE_LIST is
        access DATASPLINE_LIST;

-- Provides for pointers to lists of dataspline definitions.
```

ACCESS_DATA_VALUE_SET

```
type ACCESS_DATA_VALUE_SET is access DATA_VALUE_SET;

-- Provides for pointers to sets of data mapping data values.
```

ACCESS_EDGE_FLAG_LIST

```
type ACCESS_EDGE_FLAG_LIST is access EDGE_FLAG_LIST;

-- Provides for pointers to lists of edge flags.
```

ACCESS_LIST_OF_COLOUR_VALUE_LIST

```
type ACCESS_LIST_OF_COLOUR_VALUE_LIST is
        access LIST_OF_COLOUR_VALUE_LIST;

-- Provides for pointers to lists of lists of colour values values.
```

ACCESS_LIST_OF_EDGE_FLAG_LIST

```
type ACCESS_LIST_OF_EDGE_FLAG_LIST is access LIST_OF_EDGE_FLAG_LIST;

-- Provides for pointers to lists of lists of edge flags.
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

---

ACCESS_LIST_OF_EDGE_FLAG_TRIPLET

```
type ACCESS_LIST_OF_EDGE_FLAG_TRIPLET is
  access LIST_OF_EDGE_FLAG_TRIPLET;

-- Provides for pointers to lists of triplets of edge flags.
```

ACCESS_LIST_OF_LIST_OF_EDGE_FLAG_LIST

```
type ACCESS_LIST_OF_LIST_OF_EDGE_FLAG_LIST is
      access LIST_OF_LIST_OF_EDGE_FLAG_LIST;

-- Provides for pointers to lists of lists of edge flag lists.
```

ACCESS_LIST_OF_LIST_OF_VERTEX_INDEX_LIST

```
type ACCESS_LIST_OF_LIST_OF_VERTEX_INDEX_LIST is
      access LIST_OF_LIST_OF_VERTEX_INDEX_LIST;

-- Provides for pointers to lists of lists of lists of indices.
```

ACCESS_LIST_OF_TRIMCURVE_LIST

```
type ACCESS_LIST_OF_TRIMCURVE_LIST is access LIST_OF_TRIMCURVE_LIST;

-- Provides for pointers to lists of lists of edge flags.
```

ACCESS_LIST_OF_VERTEX_INDEX_LIST

```
type ACCESS_LIST_OF_VERTEX_INDEX_LIST is
  access LIST_OF_VERTEX_INDEX_LIST;

-- Provides for pointers to lists of lists of vertex indices.
```

ACCESS_LIST_OF_VERTEX_INDEX_TRIPLET

```
type ACCESS_LIST_OF_VERTEX_INDEX_TRIPLET is
  access LIST_OF_VERTEX_INDEX_TRIPLET;

-- Provides for pointers to lists of triplets of vertex indices.
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

---

ACCESS_SURFACE_COLOURSPLINE

```
type ACCESS_SURFACE_COLOURSPLINE is
        access SURFACE_COLOURSPLINE;

-- Provides for pointers to colourspline definitions.
```

ACCESS_TRIMCURVE_LIST

```
type ACCESS_TRIMCURVE_LIST is access TRIMCURVE_LIST;

-- Provides for pointers to lists of trimming curves.
```

ACCESS_WEIGHT_LIST

```
type ACCESS_WEIGHT_LIST is access WEIGHT_LIST;

-- Provides for pointers to lists of weights.
```

ACCESS_VERTEX_INDEX_LIST

```
type ACCESS_VERTEX_INDEX_LIST is access VERTEX_INDEX_LIST;

-- Provides for pointers to lists of vertex indices.
```

ARRAY_OF_DATA_VALUE_SET

```
type ARRAY_OF_DATA_VALUE_SET is
        array ( VERTEX_SET_DIMENSION range <>,
                VERTEX_SET_DIMENSION range <>,
                DATA_VALUE_INDEX range <>) of DATA_VALUE;

-- Provides for two-dimensional arrays of data mapping data value sets.  These
-- will be used with "Quadrilateral Mesh with Data" primitives.  The first index
-- refers to the width of the array, the second index to the height, and the third
-- index to the individual data values.
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

---

ARRAY_OF_EDGE_FLAG_PAIR

```
type ARRAY_OF_EDGE_FLAG_PAIR is
        array ( POSITIVE range <>,
                POSITIVE range <>) of EDGE_FLAG_PAIR;

-- Provides for two-dimensional arrays of edge flag pairs.  These will be used with
-- "Quadrilateral Mesh with Data" primitives.
```

ATTENUATION_COEFFICIENTS

```
type ATTENUATION_COEFFICIENTS is
  record
    C1 : COLOUR_COEFFICIENT;
    C2 : COLOUR_COEFFICIENT;
  end record;

-- Provides for specification of light source attenuation.
```

CIELUV_COLOUR_VALUE

```
type CIELUV_COLOUR_VALUE is
  record
    L_STAR_CIE : COLOUR_COEFFICIENT;
    U_STAR_CIE : COLOUR_COEFFICIENT;
    V_STAR_CIE : COLOUR_COEFFICIENT;
  end record;

-- Provides for specification of CIELUV colours.
```

CIELUV_HOMOGENEOUS_COLOUR_VALUE

```
type CIELUV_HOMOGENEOUS_COLOUR_VALUE is
  record
    L_STAR_CIE : COLOUR_COEFFICIENT;
    U_STAR_CIE : COLOUR_COEFFICIENT;
    V_STAR_CIE : COLOUR_COEFFICIENT;
    W_CIE      : COLOUR_COEFFICIENT;
  end record;

-- Provides for specification of homogeneous CIELUV colours for use in
-- rational coloursplines.
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

---

CIELUV_TYPES

```
package CIELUV_TYPES is
  new PHIGS_COLOUR_TYPES( CIELUV_COLOUR_VALUE,
                          CIELUV_HOMOGENEOUS_COLOUR_VALUE);

-- Provides for CIELUV colour types.
```

COLOUR_CONTROL_POINT_ARRAY

```
type COLOUR_CONTROL_POINT_ARRAY
  (MODEL          : COLOUR_MODEL := RGB;
   RATIONALITY    : SPLINE_RATIONALITY := RATIONAL;
   LENGTH_U       : COLOUR_VALUE_SET_DIMENSION := 1;
   LENGTH_V       : COLOUR_VALUE_SET_DIMENSION := 1)  is
    record
      case MODEL is

        when INDIRECT =>
          COLOURS_INDIRECT : INDIRECT_TYPES.CONTROL_POINT_ARRAY
                             (RATIONALITY, LENGTH_U, LENGTH_V);

        when RGB =>
          COLOURS_RGB : RGB_TYPES.CONTROL_POINT_ARRAY
                        (RATIONALITY, LENGTH_U, LENGTH_V);

        when CIELUV =>
          COLOURS_CIELUV : CIELUV_TYPES.CONTROL_POINT_ARRAY
                           (RATIONALITY, LENGTH_U, LENGTH_V);

        when HSV =>
          COLOURS_HSV  : HSV_TYPES.CONTROL_POINT_ARRAY
                         (RATIONALITY, LENGTH_U, LENGTH_V);

        when HLS =>
          COLOURS_HLS : HLS_TYPES.CONTROL_POINT_ARRAY
                        (RATIONALITY, LENGTH_U, LENGTH_V);

        when others =>
          COLOURS_GENERIC : GENERIC_COLOUR_TYPES.CONTROL_POINT_ARRAY
                            (RATIONALITY, LENGTH_U, LENGTH_V);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
      end case;
    end record;

-- Provides for specification of two-dimensional arrays of colour control point values
-- all of which are the same colour model.
```

COLOUR_CONTROL_POINT_LIST

```
type COLOUR_CONTROL_POINT_LIST
  (MODEL          : COLOUR_MODEL := RGB;
   RATIONALITY    : SPLINE_RATIONALITY := RATIONAL;
   LENGTH : COLOUR_VALUE_SET_DIMENSION := 0) is
    record
      case MODEL is

        when INDIRECT =>
          COLOURS_INDIRECT : INDIRECT_TYPES.CONTROL_POINT_LIST
                                  (RATIONALITY, LENGTH);

        when RGB =>
          COLOURS_RGB : RGB_TYPES.CONTROL_POINT_LIST
                                  (RATIONALITY, LENGTH);

        when CIELUV =>
          COLOURS_CIELUV  : CIELUV_TYPES.CONTROL_POINT_LIST
                                  (RATIONALITY, LENGTH);

        when HSV =>
          COLOURS_HSV : HSV_TYPES.CONTROL_POINT_LIST
                                  (RATIONALITY, LENGTH);

        when HLS =>
          COLOURS_HLS : HLS_TYPES.CONTROL_POINT_LIST
                                  (RATIONALITY, LENGTH);

        when others =>
          COLOURS_GENERIC : GENERIC_COLOUR_TYPES.CONTROL_POINT_LIST
                                  (RATIONALITY, LENGTH);

      end case;
    end record;

-- Provides for specification of lists of colour control point values
-- all of which are the same colour model.
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567229 506 ■

COLOUR_MAPPING_DATA_RECORD

```
type COLOUR_MAPPING_DATA_RECORD
  (METHOD : COLOUR_MAPPING_METHOD) is
    record
      case METHOD is

        when TRUE_COLOUR_MAPPING =>
          null;

        when PSEUDO_COLOUR_MAPPING =>
          PSEUDO_COLOUR_MODEL  : COLOUR_MODEL;
          PSEUDO_WEIGHTS       : ACCESS_WEIGHT_LIST;
          PSEUDO_COLOURS       : ACCESS_COLOUR_VALUE_LIST;

        when PSEUDO_N_COLOUR_MAPPING =>
          PSEUDO_N_COLOUR_MODEL  : COLOUR_MODEL;
          PSEUDO_N_COLOURS       : ACCESS_LIST_OF_COLOUR_VALUE_LIST;

        when others =>
          null;

      end case;
    end record;

-- Provides for specifying values for different methods of colour mapping.
-- Additional variants may be included when additional registered or unregistered colour
-- mapping methods are supported by an implementation.
```

COLOUR_MAPPING_INDEX

```
type COLOUR_MAPPING_INDEX is new PHIGS_NATURAL;

-- Provides for index values for colour mapping representations.
```

COLOUR_MAPPING_INDICES

```
package COLOUR_MAPPING_INDICES is
    new PHIGS_LIST_UTILITIES
        (COLOUR_MAPPING_INDEX,
         MAX_COLOUR_MAPPING_INDICES_SUPPORTED);

-- Provides for lists of index values for colour mapping representations.
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567230 228 ■

COLOUR_MAPPING_METHOD

```
type COLOUR_MAPPING_METHOD is new PHIGS_INTEGER;

-- Provides for specifying different methods for performing colour mapping.
```

COLOUR_MAPPING_METHOD_FACILITIES

```
type COLOUR_MAPPING_METHOD_FACILITIES
  (METHOD : COLOUR_MAPPING_METHOD := TRUE_COLOUR_MAPPING) is
    record
      case METHOD is

        when TRUE_COLOUR_MAPPING =>
          NUMBER_TRUE_COLOURS_AVAIL : PHIGS_INTEGER;

        when PSEUDO_COLOUR_MAPPING =>
          NUMBER_PSEUDO_COLORS_AVAIL : PHIGS_INTEGER;

        when PSEUDO_N_COLOUR_MAPPING =>
          null;

        when others =>
          null;

      end case;
    end record;

-- Provides for colour mapping availability information.  Additional variants may be included
-- when additional registered or unregistered colour mapping methods are supported by an
-- implementation.
```

COLOUR_MAPPING_METHODS

```
package COLOUR_MAPPING_METHODS is
        new PHIGS_LIST_UTILITIES
                (COLOUR_MAPPING_METHOD,
                  MAX_COLOUR_MAPPING_METHODS_SUPPORTED);

-- Provides for lists of colour mapping methods.
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567231 164 ■

COLOUR_MAPPING_STATE

```
type COLOUR_MAPPING_STATE
  (METHOD : COLOUR_MAPPING_METHOD := TRUE_COLOUR_MAPPING) is
    record
      case METHOD is

        when TRUE_COLOUR_MAPPING =>
          null;

        when PSEUDO_COLOUR_MAPPING =>
          null;

        when PSEUDO_N_COLOUR_MAPPING =>
          null;

        when others =>
          null;

      end case;
    end record;

-- Provides for colour mapping state information.  Additional variants may be included
-- when additional registered or unregistered colour mapping methods are supported by an
-- implementation.
```

COLOUR_VALUE_ARRAY

```
type COLOUR_VALUE_ARRAY
  (MODEL    : COLOUR_MODEL := RGB;
   LENGTH_M : COLOUR_VALUE_SET_DIMENSION := 1;
   LENGTH_N : COLOUR_VALUE_SET_DIMENSION := 1)  is
    record
      case MODEL is

        when INDIRECT =>
          COLOURS_INDIRECT : INDIRECT_TYPES.COLOUR_VALUE_ARRAY
                                (1..LENGTH_M, 1..LENGTH_N);

        when RGB =>
          COLOURS_RGB : RGB_TYPES.COLOUR_VALUE_ARRAY
                            (1..LENGTH_M, 1..LENGTH_N);

        when CIELUV =>
          COLOURS_CIELUV : CIELUV_TYPES.COLOUR_VALUE_ARRAY
                              (1..LENGTH_M, 1..LENGTH_N);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
            when HSV =>
              COLOURS_HSV  : HSV_TYPES.COLOUR_VALUE_ARRAY
                                (1..LENGTH_M, 1..LENGTH_N);

            when HLS =>
              COLOURS_HLS : HLS_TYPES.COLOUR_VALUE_ARRAY
                                (1..LENGTH_M, 1..LENGTH_N);

            when others =>
              COLOURS_GENERIC : GENERIC_COLOUR_TYPES.COLOUR_VALUE_ARRAY
                                      (1..LENGTH_M, 1..LENGTH_N);

          end case;
        end record;

    -- Provides for specification of two-dimensional arrays of colour values all of which
    -- are the same colour model.
```

COLOUR_VALUE_LIST

```
    type COLOUR_VALUE_LIST
      (MODEL   : COLOUR_MODEL := RGB;
       LENGTH  : COLOUR_VALUE_SET_DIMENSION := 0) is
        record
          case MODEL is

            when INDIRECT =>
              COLOURS_INDIRECT : INDIRECT_TYPES.COLOUR_VALUE_LIST(1..LENGTH);

            when RGB =>
              COLOURS_RGB : RGB_TYPES.COLOUR_VALUE_LIST(1..LENGTH);

            when CIELUV =>
              COLOURS_CIELUV  : CIELUV_TYPES.COLOUR_VALUE_LIST(1..LENGTH);

            when HSV =>
              COLOURS_HSV : HSV_TYPES.COLOUR_VALUE_LIST(1..LENGTH);

            when HLS =>
              COLOURS_HLS : HLS_TYPES.COLOUR_VALUE_LIST(1..LENGTH);

            when others =>
              COLOURS_GENERIC : GENERIC_COLOUR_TYPES.COLOUR_VALUE_LIST
                                      (1..LENGTH);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
        end case;
      end record;
```

-- Provides for specification of lists of colour values all of which are the same colour model.

COLOUR VALUE_SET_DIMENSION

```
subtype COLOUR_VALUE_SET_DIMENSION is
        NATURAL range 0..MAX_COLOUR_VALUES_SUPPORTED;
```

-- Provides for specifying the dimensions of colour value lists and arrays.

CURVE_APPROX_CRITERIA_TYPE

```
type CURVE_APPROX_CRITERIA_TYPE is new PHIGS_INTEGER;
```

-- Provides for differentiating types of curve approximation.

CURVE_APPROX_CRITERIA_TYPES

```
package CURVE_APPROX_CRITERIA_TYPES is
      new PHIGS_LIST_UTILITIES
            (CURVE_APPROX_CRITERIA_TYPE,
             MAX_CURVE_APPROX_CRITERIA_TYPES_SUPPORTED);
```

-- Provides for lists of curve approximation criteria.

CURVE_APPROX_DATA_RECORD

```
type CURVE_APPROX_DATA_RECORD
  (CRITERIA_TYPE : CURVE_APPROX_CRITERIA_TYPE := WS_DEPENDENT_CURVE) is
    record
      case CRITERIA_TYPE is

        when WS_DEPENDENT_CURVE =>
          null;

        when CONSTANT_SUBDIVISION_CURVE =>
          COUNT : PHIGS_INTEGER;

        when WC_CHORDAL_SIZE_CURVE =>
          WC_CHORDAL_SIZE_VALUE : WC.MAGNITUDE;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
            when NPC_CHORDAL_SIZE_CURVE =>
              NPC_CHORDAL_SIZE_VALUE : NPC.MAGNITUDE;

            when DC_CHORDAL_SIZE_CURVE =>
              DC_CHORDAL_SIZE_VALUE : DC.MAGNITUDE;

            when WC_CHORDAL_DEVIATION_CURVE =>
              WC_CHORDAL_DEVIATION_VALUE : WC.MAGNITUDE;

            when NPC_CHORDAL_DEVIATION_CURVE =>
              NPC_CHORDAL_DEVIATION_VALUE : NPC.MAGNITUDE;

            when DC_CHORDAL_DEVIATION_CURVE =>
              DC_CHORDAL_DEVIATION_VALUE : DC.MAGNITUDE;

            when WC_RELATIVE_CURVE =>
              WC_RELATIVE_VALUE : WC.RELATIVE_MAGNITUDE;

            when NPC_RELATIVE_CURVE =>
              NPC_RELATIVE_VALUE : NPC.RELATIVE_MAGNITUDE;

            when DC_RELATIVE_CURVE =>
              DC_RELATIVE_VALUE : DC.RELATIVE_MAGNITUDE;

            when others =>
              null;

          end case;
        end record;

    -- Provides for specifying values for different methods of curve approximation.
    -- Additional variants may be included when additional registered or unregistered curve
    -- approximation criteria types are supported by an implementation.
```

CURVE_COLOURSPLINE

```
    type CURVE_COLOURSPLINE( MODEL      : COLOUR_MODEL;
                             RATIONALITY : SPLINE_RATIONALITY;
                             KNOT_COUNT : SMALL_NATURAL;
                             LENGTH     : COLOUR_VALUE_SET_DIMENSION) is
      record
        ORDER          : SPLINE_ORDER;
        KNOTS          : KNOT_VECTOR(KNOT_COUNT);
        CONTROL_POINTS : COLOUR_CONTROL_POINT_LIST
                           (MODEL, RATIONALITY, LENGTH);
      end record;

    -- Provides for colour splines for use with Non-uniform B-Spline Curve primitives.
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

---

CURVE_PLACEMENT_TYPE

type CURVE_PLACEMENT_TYPE is (UNIFORM, NON_UNIFORM);

-- Provides for different ways of placing curves.

CURVE_VISIBILITY_FLAG

subtype CURVE_VISIBILITY_FLAG is EDGE_FLAG;

-- Provides control over display of curves on spline surfaces

---

DATA_MAPPING_DATA_RECORD

```
type DATA_MAPPING_DATA_RECORD
  (METHOD : DATA_MAPPING_METHOD := DATA_MAPPING_COLOUR) is
    record
      case METHOD is

        when DATA_MAPPING_COLOUR =>
          COLOUR_SELECTORS : SOURCE_SELECTOR_LIST;

        when SINGLE_VALUE_UNIFORM =>
          SINGLE_UNIFORM_SELECTORS          : SOURCE_SELECTOR_LIST;
          SINGLE_UNIFORM_DATA_VALUE_IND     : DATA_VALUE_INDEX;
          SINGLE_UNIFORM_RANGE              : DATA_VALUE_RANGE;
          SINGLE_UNIFORM_COLOURS            : ACCESS_COLOUR_VALUE_LIST;

        when SINGLE_VALUE_NON_UNIFORM =>
          SINGLE_NON_UNIFORM_SELECTORS      : SOURCE_SELECTOR_LIST;
          SINGLE_NON_UNIFORM_DATA_VALUE_IND : DATA_VALUE_INDEX;
          SINGLE_NON_UNIFORM_BOUNDARIES     : ACCESS_DATA_VALUE_SET;
          SINGLE_NON_UNIFORM_COLOURS        : ACCESS_COLOUR_VALUE_LIST;

        when BI_VALUE_UNIFORM =>
          BI_UNIFORM_SELECTORS : SOURCE_SELECTOR_LIST;
          BI_UNIFORM_DATA_VALUE_IND_PAIR : DATA_VALUE_INDEX_PAIR;
          BI_UNIFORM_A_RANGE   : DATA_VALUE_RANGE;
          BI_UNIFORM_B_RANGE   : DATA_VALUE_RANGE;
          BI_UNIFORM_COLOURS   : ACCESS_LIST_OF_COLOUR_VALUE_LIST;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
            when BI_VALUE_NON_UNIFORM =>
              BI_NON_UNIFORM_SELECTORS       : SOURCE_SELECTOR_LIST;
              BI_NON_UNIFORM_DATA_VALUE_IND_PAIR : DATA_VALUE_INDEX_PAIR;
              BI_NON_UNIFORM_A_BOUNDARIES : ACCESS_DATA_VALUE_SET;
              BI_NON_UNIFORM_B_BOUNDARIES : ACCESS_DATA_VALUE_SET;
              BI_NON_UNIFORM_COLOURS         : ACCESS_COLOUR_VALUE_ARRAY;

            when others =>
              null;

          end case;
        end record;

    -- Provides for specifying values for different methods of data_mapping.
    -- Additional variants may be included when additional registered or unregistered data
    -- mapping methods are supported by an implementation.
```

DATA_MAPPING_INDEX

```
    type DATA_MAPPING_INDEX is new PHIGS_NATURAL;

    -- Provides for index values for data mapping representations.
```

DATA_MAPPING_INDICES

```
    package DATA_MAPPING_INDICES is
        new PHIGS_LIST_UTILITIES
            (DATA_MAPPING_INDEX,
             MAX_DATA_MAPPING_INDICES_SUPPORTED);

    -- Provides for lists of data mapping indices.
```

DATA_MAPPING_METHOD

```
    type DATA_MAPPING_METHOD is new PHIGS_INTEGER;

    -- Provides for differentiating methods of performing data mapping.
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567237 682 ■

DATA_MAPPING_METHODS

```
package DATA_MAPPING_METHODS is
        new PHIGS_LIST_UTILITIES
                (DATA_MAPPING_METHOD,
                 MAX_DATA_MAPPING_METHODS_SUPPORTED);
```

-- Provides for lists of methods of performing data mapping.

DATASPLINE

```
type DATASPLINE( MODEL          : COLOUR_MODEL := RGB;
                 RATIONALITY    : SPLINE_RATIONALITY := RATIONAL;
                 KNOT_COUNT_U   : SMALL_NATURAL := 4;
                 KNOT_COUNT_V   : SMALL_NATURAL := 4;
                 WIDTH          : COLOUR_VALUE_SET_DIMENSION := 4;
                 HEIGHT         : COLOUR_VALUE_SET_DIMENSION := 4) is
  record
    U_ORDER          : SPLINE_ORDER;
    V_ORDER          : SPLINE_ORDER;
    U_KNOTS          : KNOT_VECTOR(KNOT_COUNT_U);
    V_KNOTS          : KNOT_VECTOR(KNOT_COUNT_V);
    CONTROL_POINTS : SPC.CONTROL_POINT_ARRAY_3
                        (RATIONALITY, WIDTH, HEIGHT);
  end record;
```

-- Provides for data splines for use with Non-uniform B-Spline primitives.

DATASPLINE_LIST

```
type DATASPLINE_LIST is
        array (DATA_VALUE_INDEX range <>) of DATASPLINE;
```

-- Provides for lists of data mapping data splines.

DATA_VALUE

```
type DATA_VALUE is digits PHIGS_PRECISION;
```

-- Provides for data mapping input values.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567238 519 ■

DATA_VALUE_INDEX

```
type DATA_VALUE_INDEX is
        new PHIGS_POSITIVE range 1 .. MAX_DATA_VALUES_PER_VERTEX;
```

-- Provides for selecting from lists of data mapping input values.

DATA_VALUE_INDEX_PAIR

```
type DATA_VALUE_INDEX_PAIR is
  record
    INDEX_A  : DATA_VALUE_INDEX;
    INDEX_B  : DATA_VALUE_INDEX;
  end record;
```

-- Provides for selecting from bi-valued sets of data mapping input values.

DATA_VALUE_RANGE

```
type DATA_VALUE_RANGE is
  record
    UPPER  : DATA_VALUE;
    LOWER  : DATA_VALUE;
  end record;
```

-- Provides for specifying a range of data mapping input values.

DATA_VALUE_SET

```
type DATA_VALUE_SET is
  array (DATA_VALUE_INDEX range <>) of DATA_VALUE;
```

-- Provides for arrays (ordered lists) of data mapping input values.

DEPTH_CUE_INDEX

```
type DEPTH_CUE_INDEX is new PHIGS_NATURAL;
```

-- Provides for index values for depth cue representations.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567239 455 ■

DEPTH_CUE_INDICES

```
package DEPTH_CUE_INDICES is
        new PHIGS_LIST_UTILITIES
                (DEPTH_CUE_INDEX,
                 MAX_DEPTH_CUE_INDICES_SUPPORTED);

-- Provides for lists of depth cue indices.
```

DEPTH_CUE_MODE

```
type DEPTH_CUE_MODE is (SUPPRESSED, ALLOWED);

-- Provides for activation and deactivation of depth cue operations.
```

DEPTH_CUE_MODES

```
package DEPTH_CUE_MODES is
        new PHIGS_LIST_UTILITIES (DEPTH_CUE_MODE, 2);

-- Provides for lists of depth cue indices.
```

DEPTH_CUE_SCALE_FACTOR

```
subtype DEPTH_CUE_SCALE_FACTOR is SCALE_FACTOR range 0.0 .. 1.0;

-- Provides for specifying scale factors for depth cue operations.
```

DEPTH_CUE_SCALE_FACTORS

```
type DEPTH_CUE_SCALE_FACTORS is
  record
    FRONT : DEPTH_CUE_SCALE_FACTOR;
    BACK  : DEPTH_CUE_SCALE_FACTOR;
  end record;

-- Provides for specifying a pair of scale factors for apportioning colours during depth cue
-- operations.
```


--`,,`-`-`,,`,,`,`,,`---
Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567240 177 ■

EDGE_FLAG_LIST

```
type EDGE_FLAG_LIST is array (POSITIVE range <>) of EDGE_FLAG;

-- Provides for arrays of edge flags. These will be used with various "with Data" primitives.
```

EDGE_FLAG_PAIR

```
type EDGE_FLAG_PAIR is array (1..2) of EDGE_FLAG;

-- Provides for pairs of edge flags. These will be used with "Quadrilateral Mesh with Data"
-- primitives.
```

EDGE_FLAG_TRIPLET

```
type EDGE_FLAG_TRIPLET is array (1..3) of EDGE_FLAG;

-- Provides for triplets of edge flags. These will be used with "Triangle Set with Data"
-- primitives.
```

FACET_CULLING_MODE

```
type FACET_CULLING_MODE is ( NO_FACET_CULLING,
                             BACKFACING,
                             FRONTFACING);

-- Provides for indicating the type of culling to be applied to facets.
```

FACET_DATA_FLAG

```
type FACET_DATA_FLAG is ( FACET_NONE,
                          FACET_COLOUR,
                          FACET_NORMAL,
                          FACET_DATA,
                          FACET_COLOUR_NORMAL,
                          FACET_COLOUR_DATA,
                          FACET_NORMAL_DATA,
                          FACET_COLOUR_NORMAL_DATA);

-- Provides for various types of facet data flags.
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

---

FACET_DISTINGUISHING_MODE

```
type FACET_DISTINGUISHING_MODE is new OFF_ON;

-- Provides for enabling and disabling facet distinguishing operations.
```

FACET_SET_DIMENSION

```
subtype FACET_SET_DIMENSION is
        NATURAL range 0..MAX_FACETS_SUPPORTED;

-- Provides for specifying the dimensions of facet lists and arrays.
```

GENERAL_COLOUR

```
type GENERAL_COLOUR
  (MODEL : COLOUR_MODEL := RGB) is
    record
      case MODEL is

        when INDIRECT =>
          INDEX              : COLOUR_INDEX;

        when RGB =>
          COLOUR_RGB         : RGB_COLOUR_VALUE;

        when CIELUV =>
          COLOUR_CIELUV      : CIELUV_COLOUR_VALUE;

        when HSV =>
          COLOUR_HSV         : HSV_COLOUR_VALUE;

        when HLS =>
          COLOUR_HLS         : HLS_COLOUR_VALUE;

        when others =>
          COLOUR_GENERIC     : COLOUR_COEFFICIENT_ARRAY;

      end case;
    end record;

-- Provides for specifying values for colours dependent upon colour models. Additional
-- variants may be included when additional registered or unregistered colour models are
-- supported by an implementation.
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

---

GENERIC_COLOUR_TYPES

```
package GENERIC_COLOUR_TYPES is
  new PHIGS_COLOUR_TYPES( COLOUR_COEFFICIENT_ARRAY,
                          COLOUR_COEFFICIENT_ARRAY);

-- Provides for generic (unsupported) colour types.
```

HLS_COLOUR_VALUE

```
type HLS_COLOUR_VALUE is
  record
    HUE_HLS        : INTENSITY;
    LIGHTNESS_HLS  : INTENSITY;
    SATURATION_HLS : INTENSITY;
  end record;

-- Provides for specification of HLS colours.
```

HLS_HOMOGENEOUS_COLOUR_VALUE

```
type HLS_HOMOGENEOUS_COLOUR_VALUE is
  record
    HUE_HLS        : COLOUR_COEFFICIENT;
    LIGHTNESS_HLS  : COLOUR_COEFFICIENT;
    SATURATION_HLS : COLOUR_COEFFICIENT;
    W_HLS          : COLOUR_COEFFICIENT;
  end record;

-- Provides for specification of homogeneous HLS colours for use in
-- rational coloursplines.
```

HLS_TYPES

```
package HLS_TYPES is
  new PHIGS_COLOUR_TYPES( HLS_COLOUR_VALUE,
                          HLS_HOMOGENEOUS_COLOUR_VALUE);

-- Provides for HLS colour types.
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

---

HSV_COLOUR_VALUE

```
type HSV_COLOUR_VALUE is
  record
    HUE_HSV          : INTENSITY;
    SATURATION_HSV   : INTENSITY;
    VALUE_HSV        : INTENSITY;
  end record;

-- Provides for specification of HSV colours.
```

HSV_HOMOGENEOUS_COLOUR_VALUE

```
type HSV_HOMOGENEOUS_COLOUR_VALUE is
  record
    HUE_HSV          : COLOUR_COEFFICIENT;
    SATURATION_HSV   : COLOUR_COEFFICIENT;
    VALUE_HSV        : COLOUR_COEFFICIENT;
    W_HSV            : COLOUR_COEFFICIENT;
  end record;

-- Provides for specification of homogeneous HSV colours for use in
-- rational coloursplines.
```

HSV_TYPES

```
package HSV_TYPES is
  new PHIGS_COLOUR_TYPES( HSV_COLOUR_VALUE,
                          HSV_HOMOGENEOUS_COLOUR_VALUE);

-- Provides for HSV colour types.
```

INDIRECT_HOMOGENEOUS_COLOUR_VALUE

```
type INDIRECT_HOMOGENEOUS_COLOUR_VALUE is
  record
    INDEX_INDIRECT : COLOUR_INDEX;
    W_INDIRECT     : COLOUR_COEFFICIENT;
  end record;

-- Provides for specification of homogeneous indirect colours for use in rational
-- coloursplines.
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

---

INDIRECT_TYPES

```
package INDIRECT_TYPES is
  new PHIGS_COLOUR_TYPES( COLOUR_INDEX,
                          INDIRECT_HOMOGENEOUS_COLOUR_VALUE);
```

-- Provides for INDIRECT colour types.

INTERIOR_SHADING_METHOD

```
type INTERIOR_SHADING_METHOD is new PHIGS_INTEGER;
```

-- Provides for differentiating methods of shading interiors of enclosed regions.

INTERIOR_SHADING_METHODS

```
package INTERIOR_SHADING_METHODS is
      new PHIGS_LIST_UTILITIES
            (INTERIOR_SHADING_METHOD,
             MAX_INTERIOR_SHADING_METHODS_SUPPORTED);
```

-- Provides for lists of methods of performing interior shading.

KNOT

```
subtype KNOT is SPC.MAGNITUDE;
```

-- Provides for specifying spline knots.

KNOT_VECTOR

```
subtype KNOT_VECTOR is SPC.MAGNITUDE_LIST;
```

-- Provides for specifying spline knot vectors.

LIGHTING_EXPONENT

```
type LIGHTING_EXPONENT is digits PHIGS_PRECISION;
```

-- Provides for specifying values for specular lighting exponents.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567245 759 ■

LIGHT_SOURCE_DATA_RECORD

```
type LIGHT_SOURCE_DATA_RECORD
  (TYPE_OF_LIGHT : LIGHT_SOURCE_TYPE := AMBIENT_LIGHT) is
    record
      case TYPE_OF_LIGHT is

        when AMBIENT_LIGHT =>
          AMBIENT_COLOUR            : GENERAL_COLOUR;

        when DIRECTIONAL_LIGHT =>
          DIRECTIONAL_COLOUR        : GENERAL_COLOUR;
          DIRECTIONAL_DIRECTION     : WC.VECTOR_3;

        when POSITIONAL_LIGHT =>
          POSITIONAL_COLOUR         : GENERAL_COLOUR;
          POSITIONAL_POSITION       : WC.POINT_3;
          POSITIONAL_ATTENUATION    : ATTENUATION_COEFFICIENTS;

        when SPOT_LIGHT =>
          SPOT_COLOUR               : GENERAL_COLOUR;
          SPOT_POSITION             : WC.POINT_3;
          SPOT_DIRECTION            : WC.VECTOR_3;
          SPOT_CONCENTRATION        : LIGHTING_EXPONENT;
          SPOT_ATTENUATION          : ATTENUATION_COEFFICIENTS;
          SPOT_SPREAD               : SPREAD_ANGLE;

        when others =>
          null;

      end case;
    end record;

-- Provides for descriptions of light sources.  Additional variants may be included when
-- additional registered or unregistered light source types are supported by an
-- implementation.
```

LIGHT_SOURCE_INDEX

```
type LIGHT_SOURCE_INDEX is new PHIGS_POSITIVE;

-- Provides for identifying light sources.
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

---

LIGHT_SOURCE_INDICES

```
package LIGHT_SOURCE_INDICES is
      new PHIGS_LIST_UTILITIES ( LIGHT_SOURCE_INDEX,
                                 MAX_LIGHT_SOURCE_INDICES_SUPPORTED);

-- Provides for lists of light sources.
```

LIGHT_SOURCE_TYPE

```
type LIGHT_SOURCE_TYPE is new PHIGS_INTEGER;

-- Provides for specifying kinds of light sources.
```

LIGHT_SOURCE_TYPES

```
package LIGHT_SOURCE_TYPES is
      new PHIGS_LIST_UTILITIES ( LIGHT_SOURCE_TYPE,
                                 MAX_LIGHT_SOURCE_TYPES_SUPPORTED);

-- Provides for lists of light source types.
```

LIST_OF_COLOUR_VALUE_LIST

```
type LIST_OF_COLOUR_VALUE_LIST
     (MODEL    : COLOUR_MODEL := RGB;
      LENGTH   : COLOUR_VALUE_SET_DIMENSION := 3) is
  record
    case MODEL is

      when INDIRECT =>
        COLOURS_INDIRECT
            : INDIRECT_TYPES.LIST_OF_COLOUR_VALUE_LIST(1..LENGTH);

      when RGB =>
        COLOURS_RGB
            : RGB_TYPES.LIST_OF_COLOUR_VALUE_LIST(1..LENGTH);

      when CIELUV =>
        COLOURS_CIELUV
            : CIELUV_TYPES.LIST_OF_COLOUR_VALUE_LIST(1..LENGTH);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
            when HSV =>
              COLOURS_HSV
                    : HSV_TYPES.LIST_OF_COLOUR_VALUE_LIST(1..LENGTH);

            when HLS =>
              COLOURS_HLS
                    : HLS_TYPES.LIST_OF_COLOUR_VALUE_LIST(1..LENGTH);

            when others =>
                COLOURS_GENERIC
                    : GENERIC_COLOUR_TYPES.LIST_OF_COLOUR_VALUE_LIST
                                                (1..LENGTH);

          end case;
        end record;

        -- Provides for specification of lists of lists of colour values all of which are the same
        -- colour model.
```

LIST_OF_DATA_VALUE_SET

```
        type LIST_OF_DATA_VALUE_SET is
                array ( VERTEX_SET_DIMENSION range <>,
                        DATA_VALUE_INDEX range <>) of DATA_VALUE;

        -- Provides for lists of data mapping data value sets. These will be used
        -- various output primitives. The first index refers to the number of data value lists
        -- and the second index to the individual data values.
```

LIST_OF_EDGE_FLAG_LIST

```
        type LIST_OF_EDGE_FLAG_LIST is
          array (POSITIVE range <>) of ACCESS_EDGE_FLAG_LIST;

        -- Provides for pointers to lists of lists of edge flags.
```

LIST_OF_EDGE_FLAG_TRIPLET

```
        type LIST_OF_EDGE_FLAG_TRIPLET is
          array (POSITIVE range <>) of EDGE_FLAG_TRIPLET;

        -- Provides for lists of triplets of edge flags.
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

---

LIST_OF_LIST_OF_EDGE_FLAG_LIST

```
type LIST_OF_LIST_OF_EDGE_FLAG_LIST is
  array (POSITIVE range <>) of ACCESS_LIST_OF_EDGE_FLAG_LIST;

-- Provides for pointers to lists of lists of edge flags.
```

LIST_OF_LIST_OF_VERTEX_INDEX_LIST

```
type LIST_OF_LIST_OF_VERTEX_INDEX_LIST is
  array (POSITIVE range <>) of ACCESS_LIST_OF_VERTEX_INDEX_LIST;

-- Provides for pointers to lists of lists of vertex indices.
```

LIST_OF_TRIMCURVE_LIST

```
type LIST_OF_TRIMCURVE_LIST is
  array (POSITIVE range <>) of ACCESS_TRIMCURVE_LIST;

-- Provides for pointers to lists of lists of trimming curves.
```

LIST_OF_VERTEX_INDEX_LIST

```
type LIST_OF_VERTEX_INDEX_LIST is
  array (POSITIVE range <>) of ACCESS_VERTEX_INDEX_LIST;

-- Provides for lists of lists of vertex indices.
```

LIST_OF_VERTEX_INDEX_TRIPLET

```
type LIST_OF_VERTEX_INDEX_TRIPLET is
  array (POSITIVE range <>) of VERTEX_INDEX_TRIPLET;

-- Provides for lists of triplets of vertex indices.
```

PARAMETRIC_SURFACE_CHARACTERISTIC

```
type PARAMETRIC_SURFACE_CHARACTERISTIC is new PHIGS_INTEGER;

-- Provides for differentiating representations of parametric surfaces.
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

---

PARAMETRIC_SURFACE_CHARACTERISTICS

```
package PARAMETRIC_SURFACE_CHARACTERISTICS is
        new PHIGS_LIST_UTILITIES
                (PARAMETRIC_SURFACE_CHARACTERISTIC,
                 MAX_PARAMETRIC_SURFACE_CHARACTERISTICS_SUPPORTED);

-- Provides for lists of parametric surface characteristics types.
```

PARAMETRIC_SURFACE_DATA_RECORD

```
type PARAMETRIC_SURFACE_DATA_RECORD
  (CHARACTERISTIC : PARAMETRIC_SURFACE_CHARACTERISTIC
                        := NO_PARAMETRIC_SURFACE) is
    record
      case CHARACTERISTIC is

        when NO_PARAMETRIC_SURFACE =>
          null;

        when WS_DEPENDENT_PARAMETRIC_SURFACE =>
          null;

        when ISOPARAMETRIC_CURVES =>
          CURVE_PLACEMENT : CURVE_PLACEMENT_TYPE;
          UCOUNT          : PHIGS_NATURAL;
          VCOUNT          : PHIGS_NATURAL;

        when MC_LEVEL_CURVES =>
          MC_LEVEL_ORIGIN      : MC.POINT_3;
          MC_LEVEL_DIRECTION   : MC.VECTOR_3;
          MC_LEVEL_PARAMETERS  : MC.MAGNITUDE_LIST;

        when WC_LEVEL_CURVES =>
          WC_LEVEL_ORIGIN      : WC.POINT_3;
          WC_LEVEL_DIRECTION   : WC.VECTOR_3;
          WC_LEVEL_PARAMETERS  : WC.MAGNITUDE_LIST;

        when others =>
          null;

      end case;
    end record;

-- Provides for specifying values for different parametric surface characteristics. Additional
-- variants may be included when additional registered or unregistered parametric surface
-- characteristics are supported by an implementation.
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

---

PARAMETRIC_SURFACE_INDEX

```
type PARAMETRIC_SURFACE_INDEX is new PHIGS_NATURAL;

-- Provides for index values for parametric surface representations.
```

PARAMETRIC_SURFACE_INDICES

```
package PARAMETRIC_SURFACE_INDICES is
        new PHIGS_LIST_UTILITIES
                (PARAMETRIC_SURFACE_INDEX,
                 MAX_PARAMETRIC_SURFACE_INDICES_SUPPORTED);

-- Provides for lists of parametric surface indices.
```

POLYLINE_SHADING_METHOD

```
type POLYLINE_SHADING_METHOD is new PHIGS_NATURAL;

-- Provides for values for distinguishing between various methods of polyline shading.
```

POLYLINE_SHADING_METHODS

```
package POLYLINE_SHADING_METHODS is
        new PHIGS_LIST_UTILITIES
                (POLYLINE_SHADING_METHOD,
                 MAX_POLYLINE_SHADING_METHODS_SUPPORTED);

-- Provides for lists of polyline shading methods.
```

REFLECTANCE_COEFFICIENT

```
type REFLECTANCE_COEFFICIENT is digits PHIGS_PRECISION range 0.0 .. 1.0;

-- Provides for specifying values for lighting reflectance coefficients.
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

---

REFLECTANCE_INDEX

```
type REFLECTANCE_INDEX is new PHIGS_NATURAL;

-- Provides for index values for reflectance representations.
```

REFLECTANCE_INDICES

```
package REFLECTANCE_INDICES is
      new PHIGS_LIST_UTILITIES ( REFLECTANCE_INDEX,
                                 MAX_REFLECTANCE_INDICES_SUPPORTED);

-- Provides for lists of reflectance indices.
```

REFLECTANCE_MODEL

```
type REFLECTANCE_MODEL is new PHIGS_INTEGER;

-- Provides for selecting the type of reflectance calculations.
```

REFLECTANCE_MODELS

```
package REFLECTANCE_MODELS is
      new PHIGS_LIST_UTILITIES
            (REFLECTANCE_MODEL,
             MAX_REFLECTANCE_MODELS_SUPPORTED);

-- Provides for lists of reflectance models.
```

REFLECTANCE_PROPERTIES_DATA_RECORD

```
type REFLECTANCE_PROPERTIES_DATA_RECORD
     (TYPE_OF_REFLECTANCE_PROPERTIES
           : REFLECTANCE_PROPERTIES_TYPE := SIMPLE_REFLECTANCE) is
   record
     case TYPE_OF_REFLECTANCE_PROPERTIES is
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
            when SIMPLE_REFLECTANCE =>
              AMBIENT_REFLECTANCE   : REFLECTANCE_COEFFICIENT;
              DIFFUSE_REFLECTANCE   : REFLECTANCE_COEFFICIENT;
              SPECULAR_REFLECTANCE  : REFLECTANCE_COEFFICIENT;
              SPECULAR_COLOUR       : GENERAL_COLOUR;
              SPECULAR_EXPONENT     : LIGHTING_EXPONENT;

            when others =>
              null;

          end case;
        end record;
```

-- Provides for specifying values for different types of reflectance properties. Additional
-- variants may be included when additional registered or unregistered reflectance
-- properties types are supported by an implementation.

REFLECTANCE_PROPERTIES_TYPE

```
type REFLECTANCE_PROPERTIES_TYPE is new PHIGS_INTEGER;
```

-- Provides for selecting the type of reflectance properties.

REFLECTANCE_PROPERTIES_TYPES

```
package REFLECTANCE_PROPERTIES_TYPES is
        new PHIGS_LIST_UTILITIES
              (REFLECTANCE_PROPERTIES_TYPE,
               MAX_REFLECTANCE_PROPERTIES_TYPES_SUPPORTED);
```

-- Provides for lists of reflectance properties types.

RETURN_DEPTH_CUE_INDEX_COUNT

```
subtype RETURN_DEPTH_CUE_INDEX_COUNT is
            PHIGS_POSITIVE range 2..MAX_DEPTH_CUE_INDICES_SUPPORTED;
```

-- Provides for restricting the returned count of predefined depth cue indices.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567253 825 ■

RETURN_LIGHT_SOURCE_COUNT

```
subtype RETURN_LIGHT_SOURCE_COUNT is
        PHIGS_POSITIVE range 2..PHIGS_POSITIVE'last;

-- Provides for restricting spline orders to those acceptable for returning values of the
-- facilities.
```

RETURN_SPLINE_ORDER

```
subtype RETURN_SPLINE_ORDER is
        SPLINE_ORDER range 6..SPLINE_ORDER'last;

-- Provides for restricting spline orders to those acceptable for returning values of the
-- facilities.
```

RETURN_TRIMCURVE_ORDER

```
subtype RETURN_TRIMCURVE_ORDER is
        TRIMCURVE_ORDER range 4..TRIMCURVE_ORDER'last;

-- Provides for restricting trimcurve orders to those acceptable for returning values of the
-- facilities.
```

RGB_COLOUR_VALUE

```
type RGB_COLOUR_VALUE is
  record
    RED_RGB   : INTENSITY;
    GREEN_RGB : INTENSITY;
    BLUE_RGB  : INTENSITY;
  end record;

-- Provides for specification of RGB colours.
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567254 761 ■

RGB_HOMOGENEOUS_COLOUR_VALUE

```
type RGB_HOMOGENEOUS_COLOUR_VALUE is
  record
    RED_RGB   : COLOUR_COEFFICIENT;
    GREEN_RGB : COLOUR_COEFFICIENT;
    BLUE_RGB  : COLOUR_COEFFICIENT;
    W_RGB     : COLOUR_COEFFICIENT;
  end record;
```

-- Provides for specification of homogeneous RGB colours for use in rational coloursplines.

RGB_TYPES

```
package RGB_TYPES is
  new PHIGS_COLOUR_TYPES( RGB_COLOUR_VALUE,
                          RGB_HOMOGENEOUS_COLOUR_VALUE);
```

-- Provides for RGB colour types.

SOURCE_SELECTOR

```
type SOURCE_SELECTOR is ( COLOUR_ASPECT,
                          VERTEX_COLOUR,
                          VERTEX_DATA,
                          FACET_COLOUR,
                          FACET_DATA);
```

-- Provides for specifying sources for colour information.

SOURCE_SELECTOR_LIST

```
type SOURCE_SELECTOR_LIST is
  array (SMALL_NATURAL) of SOURCE_SELECTOR;
```

-- Provides for specification of ordered lists of sources for colour information.

SPC

```
package SPC is new PHIGS_COORDINATE_SYSTEM(SPC_TYPE);
```

-- Defines the Spline Parameter Coordinate System.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

---

SPC_TYPE

```
type SPC_TYPE is digits PHIGS_PRECISION;

-- Defines the type of a coordinate in the Spline Parameter Coordinate System.
```

SPLINE_ORDER

```
type SPLINE_ORDER is new POSITIVE;

-- Provides for specifying orders for spline curves and surfaces.
```

SPLINE_RATIONALITY

```
type SPLINE_RATIONALITY is (RATIONAL, NON_RATIONAL);

-- Provides for specifying the rationality of splines.
```

SPREAD_ANGLE

```
subtype SPREAD_ANGLE is ANGLE range 0.0 .. PHIGS_PI;

-- Provides for specifying the amount of spread of spot light sources.
```

SURFACE_APPROX_CRITERIA_TYPE

```
type SURFACE_APPROX_CRITERIA_TYPE is new PHIGS_INTEGER;

-- Provides for differentiating methods of surface approximation.
```

SURFACE_APPROX_CRITERIA_TYPES

```
package SURFACE_APPROX_CRITERIA_TYPES is
        new PHIGS_LIST_UTILITIES
                (SURFACE_APPROX_CRITERIA_TYPE,
                 MAX_SURFACE_APPROX_CRITERIA_TYPES_SUPPORTED);

-- Provides for lists of surface approximation criteria types.
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567256 534 ■

SURFACE_APPROX_DATA_RECORD

```
type SURFACE_APPROX_DATA_RECORD
  (CRITERIA_TYPE : SURFACE_APPROX_CRITERIA_TYPE
                   := WS_DEPENDENT_SURFACE) is
    record
      case CRITERIA_TYPE is

        when WS_DEPENDENT_SURFACE =>
          null;

        when CONSTANT_SUBDIVISION_SURFACE_BETWEEN_KNOTS =>
          CONSTANT_SUBDIVISION_UCOUNT : PHIGS_INTEGER;
          CONSTANT_SUBDIVISION_VCOUNT : PHIGS_INTEGER;

        when WC_CHORDAL_SIZE_SURFACE =>
          WC_CHORDAL_SIZE_UVALUE : WC.MAGNITUDE;
          WC_CHORDAL_SIZE_VVALUE : WC.MAGNITUDE;

        when NPC_CHORDAL_SIZE_SURFACE =>
          NPC_CHORDAL_SIZE_UVALUE : NPC.MAGNITUDE;
          NPC_CHORDAL_SIZE_VVALUE : NPC.MAGNITUDE;

        when DC_CHORDAL_SIZE_SURFACE =>
          DC_CHORDAL_SIZE_UVALUE : DC.MAGNITUDE;
          DC_CHORDAL_SIZE_VVALUE : DC.MAGNITUDE;

        when WC_PLANAR_DEVIATION_SURFACE =>
          WC_PLANAR_DEVIATION_VALUE : WC.MAGNITUDE;

        when NPC_PLANAR_DEVIATION_SURFACE =>
          NPC_PLANAR_DEVIATION_VALUE : NPC.MAGNITUDE;

        when DC_PLANAR_DEVIATION_SURFACE =>
          DC_PLANAR_DEVIATION_VALUE : DC.MAGNITUDE;

        when WC_RELATIVE_SURFACE =>
          WC_RELATIVE_VALUE : WC.RELATIVE_MAGNITUDE;

        when NPC_RELATIVE_SURFACE =>
          NPC_RELATIVE_VALUE : NPC.RELATIVE_MAGNITUDE;

        when DC_RELATIVE_SURFACE =>
          DC_RELATIVE_VALUE : DC.RELATIVE_MAGNITUDE;

        when others =>
          null;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
        end case;
      end record;

    -- Provides for specifying values for different methods of surface approximation.
    -- Additional variants may be included when additional registered or unregistered surface
    -- approximation criteria types are supported by an implementation.
```

SURFACE_COLOURSPLINE

```
    type SURFACE_COLOURSPLINE
          (MODEL           : COLOUR_MODEL;
           RATIONALITY     : SPLINE_RATIONALITY;
           KNOT_COUNT_U    : SMALL_NATURAL;
           KNOT_COUNT_V    : SMALL_NATURAL;
           LENGTH_U        : COLOUR_VALUE_SET_DIMENSION;
           LENGTH_V        : COLOUR_VALUE_SET_DIMENSION) is
      record
        U_ORDER          : SPLINE_ORDER;
        V_ORDER          : SPLINE_ORDER;
        U_KNOTS          : KNOT_VECTOR(KNOT_COUNT_U);
        V_KNOTS          : KNOT_VECTOR(KNOT_COUNT_V);
        CONTROL_POINTS : COLOUR_CONTROL_POINT_ARRAY
                              (MODEL, RATIONALITY, LENGTH_U, LENGTH_V);
      end record;

    -- Provides for colour splines for use with Non-uniform B-Spline Surface primitives.
```

TRIMCURVE

```
    type TRIMCURVE( CRITERIA_TYPE : CURVE_APPROX_CRITERIA_TYPE
                                      := CONSTANT_SUBDIVISION_CURVE;
                    RATIONALITY   : SPLINE_RATIONALITY := RATIONAL;
                    KNOT_COUNT    : SMALL_NATURAL := 4;
                    LENGTH        : VERTEX_SET_DIMENSION := 2) is
      record
        CRITERIA_FOR_CURVE_APPROX
                          : CURVE_APPROX_DATA_RECORD (CRITERIA_TYPE);
        FLAG              : CURVE_VISIBILITY_FLAG;
        ORDER             : TRIMCURVE_ORDER;
        KNOTS             : KNOT_VECTOR(KNOT_COUNT);
        CONTROL_POINTS : SPC.CONTROL_POINT_LIST_2(RATIONALITY, LENGTH);
        LIMITS            : SPC.RANGE_OF_MAGNITUDES;
      end record;

    -- Provides for specifying minimum and maximum values of spline parameters.
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

---

TRIMCURVE_LIST

```
type TRIMCURVE_LIST is array (POSITIVE range <>) of TRIMCURVE;

-- Provides for arrays of trimming curves.  These will be used with NURB surface primitives.
```

TRIMCURVE_ORDER

```
subtype TRIMCURVE_ORDER is SPLINE_ORDER range 2..SPLINE_ORDER'last;

-- Provides for restricting spline orders to those acceptable for trimming curves.
```

VERTEX_COLOUR_FLAG

```
subtype VERTEX_COLOUR_FLAG is
        VERTEX_DATA_FLAG range COORDINATES .. COORDINATES_COLOUR;

-- Provides identifiers for various types of vertex information used in polyline sets.
```

VERTEX_DATA_FLAG

```
type VERTEX_DATA_FLAG is ( COORDINATES,
                           COORDINATES_COLOUR,
                           COORDINATES_NORMAL,
                           COORDINATES_DATA,
                           COORDINATES_COLOUR_NORMAL,
                           COORDINATES_COLOUR_DATA,
                           COORDINATES_NORMAL_DATA,
                           COORDINATES_COLOUR_NORMAL_DATA);

-- Provides identifiers for various types of vertex information.
```

VERTEX_INDEX

```
type VERTEX_INDEX is new PHIGS_NATURAL range 1..MAX_VERTICES_SUPPORTED;

-- Provides for indices into arrays of vertices.  These will be used with various "with Data"
-- primitives.
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

---

VERTEX_INDEX_LIST

```
type VERTEX_INDEX_LIST is array (POSITIVE range <>) of VERTEX_INDEX;

-- Provides for arrays of indices.  These will be used with various "with Data" primitives.
```

VERTEX_INDEX_TRIPLET

```
subtype VERTEX_INDEX_TRIPLET is VERTEX_INDEX_LIST(1..3);

-- Provides for triplets of indices.  These will be used with "Triangle Set with Data"
-- primitives.
```

VERTEX_SET_COUNT

```
subtype VERTEX_SET_COUNT is
        NATURAL range 0..MAX_VERTEX_SETS_SUPPORTED;

-- Provides for specifying the number of sets in a vertex set.
```

VERTEX_SET_DIMENSION

```
subtype VERTEX_SET_DIMENSION is
        NATURAL range 0..MAX_VERTICES_SUPPORTED;

-- Provides for specifying the dimensions of vertex lists and arrays.
```

WEIGHT_LIST

```
type WEIGHT_LIST
     (LENGTH : COLOUR_COMPONENTS := MAX_COLOUR_COEFFICIENTS) is
  record
    WEIGHTS : WEIGHT_LIST_ARRAY(1..LENGTH);
  end record;

-- Provides for a variable length list of weights to be applied during colour mapping.
```

WEIGHT_LIST_ARRAY

```
type WEIGHT_LIST_ARRAY is
     array (COLOUR_COMPONENTS range <>) of COLOUR_COEFFICIENT;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

-- Provides for a list of weights to be applied during colour mapping.

--`,,,`-`-`,,`,,`,`,,`---

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567261 9T1 ■

6.1.7 Additions to list of constant declarations

The following constant defines the PHIGS PLUS standard additional colour model:

```
INDIRECT                                : constant COLOUR_MODEL := 0;
```

The following constant is for use when specifying the rendering colour model:

```
WS_DEPENDENT_COLOUR_MODEL    : constant COLOUR_MODEL := 0;
```

The following constants define the PHIGS PLUS standard polyline shading methods:

```
NO_POLYLINE_SHADING              : constant POLYLINE_SHADING_METHOD := 1;
POLYLINE_COLOUR_SHADING          : constant POLYLINE_SHADING_METHOD := 2;
```

The following constants define the PHIGS PLUS standard interior shading methods:

```
NO_INTERIOR_SHADING                    : constant INTERIOR_SHADING_METHOD := 1;
INTERIOR_COLOUR_SHADING                : constant INTERIOR_SHADING_METHOD := 2;
INTERIOR_DATA_SHADING                  : constant INTERIOR_SHADING_METHOD := 3;
INTERIOR_DATA_AND_DOT_SHADING          : constant INTERIOR_SHADING_METHOD := 4;
INTERIOR_DATA_AND_NORMAL_SHADING       : constant INTERIOR_SHADING_METHOD := 5;
```

The following constants define the PHIGS PLUS standard data mapping methods:

```
DATA_MAPPING_COLOUR          : constant DATA_MAPPING_METHOD := 1;
SINGLE_VALUE_UNIFORM         : constant DATA_MAPPING_METHOD := 2;
SINGLE_VALUE_NON_UNIFORM     : constant DATA_MAPPING_METHOD := 3;
BI_VALUE_UNIFORM             : constant DATA_MAPPING_METHOD := 4;
BI_VALUE_NON_UNIFORM         : constant DATA_MAPPING_METHOD := 5;
```

The following constant defines the PHIGS PLUS standard reflectance properties type:

```
SIMPLE_REFLECTANCE  : constant REFLECTANCE_PROPERTIES_TYPE := 1;
```

The following constants define the PHIGS PLUS standard reflectance models:

```
NO_REFLECTANCE                          : constant REFLECTANCE_MODEL := 1;
AMBIENT_REFLECTANCE                     : constant REFLECTANCE_MODEL := 2;
AMBIENT_DIFFUSE_REFLECTANCE             : constant REFLECTANCE_MODEL := 3;
AMBIENT_DIFFUSE_SPECULAR_REFLECTANCE    : constant REFLECTANCE_MODEL := 4;
```

The following constants define the PHIGS PLUS standard curve approximation criteria types:

```
WS_DEPENDENT_CURVE           : constant CURVE_APPROX_CRITERIA_TYPE := 1;
CONSTANT_SUBDIVISION_CURVE   : constant CURVE_APPROX_CRITERIA_TYPE := 2;
WC_CHORDAL_SIZE_CURVE        : constant CURVE_APPROX_CRITERIA_TYPE := 3;
NPC_CHORDAL_SIZE_CURVE       : constant CURVE_APPROX_CRITERIA_TYPE := 4;
DC_CHORDAL_SIZE_CURVE        : constant CURVE_APPROX_CRITERIA_TYPE := 5;
WC_CHORDAL_DEVIATION_CURVE   : constant CURVE_APPROX_CRITERIA_TYPE := 6;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
NPC_CHORDAL_DEVIATION_CURVE : constant CURVE_APPROX_CRITERIA_TYPE := 7;
DC_CHORDAL_DEVIATION_CURVE  : constant CURVE_APPROX_CRITERIA_TYPE := 8;
WC_RELATIVE_CURVE           : constant CURVE_APPROX_CRITERIA_TYPE := 9;
NPC_RELATIVE_CURVE          : constant CURVE_APPROX_CRITERIA_TYPE := 10;
DC_RELATIVE_CURVE           : constant CURVE_APPROX_CRITERIA_TYPE := 11;
```

The following constants define the PHIGS PLUS standard surface approximation criteria types:

```
WS_DEPENDENT_SURFACE            : constant SURFACE_APPROX_CRITERIA_TYPE := 1;
CONSTANT_SUBDIVISION_SURFACE_BETWEEN_KNOTS
                                : constant SURFACE_APPROX_CRITERIA_TYPE := 2;
WC_CHORDAL_SIZE_SURFACE         : constant SURFACE_APPROX_CRITERIA_TYPE := 3;
NPC_CHORDAL_SIZE_SURFACE        : constant SURFACE_APPROX_CRITERIA_TYPE := 4;
DC_CHORDAL_SIZE_SURFACE         : constant SURFACE_APPROX_CRITERIA_TYPE := 5;
WC_PLANAR_DEVIATION_SURFACE     : constant SURFACE_APPROX_CRITERIA_TYPE := 6;
NPC_PLANAR_DEVIATION_SURFACE    : constant SURFACE_APPROX_CRITERIA_TYPE := 7;
DC_PLANAR_DEVIATION_SURFACE     : constant SURFACE_APPROX_CRITERIA_TYPE := 8;
WC_RELATIVE_SURFACE             : constant SURFACE_APPROX_CRITERIA_TYPE := 9;
NPC_RELATIVE_SURFACE            : constant SURFACE_APPROX_CRITERIA_TYPE := 10;
DC_RELATIVE_SURFACE             : constant SURFACE_APPROX_CRITERIA_TYPE := 11;
```

The following constants define the PHIGS PLUS standard parametric surface characteristic types:

```
NO_PARAMETRIC_SURFACE
                          : constant PARAMETRIC_SURFACE_CHARACTERISTIC := 1;
WS_DEPENDENT_PARAMETRIC_SURFACE
                          : constant PARAMETRIC_SURFACE_CHARACTERISTIC := 2;
ISOPARAMETRIC_CURVES      : constant PARAMETRIC_SURFACE_CHARACTERISTIC := 3;
MC_LEVEL_CURVES           : constant PARAMETRIC_SURFACE_CHARACTERISTIC := 4;
WC_LEVEL_CURVES           : constant PARAMETRIC_SURFACE_CHARACTERISTIC := 5;
```

The following constants define the PHIGS PLUS standard light source types:

```
AMBIENT_LIGHT             : constant LIGHT_SOURCE_TYPE := 1;
DIRECTIONAL_LIGHT         : constant LIGHT_SOURCE_TYPE := 2;
POSITIONAL_LIGHT          : constant LIGHT_SOURCE_TYPE := 3;
SPOT_LIGHT                : constant LIGHT_SOURCE_TYPE := 4;
```

The following constants define the PHIGS PLUS standard colour mapping methods:

```
TRUE_COLOUR_MAPPING       : constant COLOUR_MAPPING_METHOD := 1;
PSEUDO_COLOUR_MAPPING     : constant COLOUR_MAPPING_METHOD := 2;
PSEUDO_N_COLOUR_MAPPING   : constant COLOUR_MAPPING_METHOD := 3;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

6.1.8 PHIGS PLUS configuration values

The following configuration values are required by PHIGS PLUS to specify the maximum number of entities of the indicated type which are supported by the implementation:

MAX_COLOUR_MAPPING_INDICES_SUPPORTED
MAX_COLOUR_MAPPING_METHODS_SUPPORTED
MAX_COLOUR_VALUES_SUPPORTED
MAX_CURVE_APPROX_CRITERIA_TYPES_SUPPORTED
MAX_DATA_MAPPING_INDICES_SUPPORTED
MAX_DATA_MAPPING_METHODS_SUPPORTED
MAX_DATA_VALUES_PER_FACET
MAX_DATA_VALUES_PER_VERTEX
MAX_DEPTH_CUE_INDICES_SUPPORTED
MAX_FACETS_SUPPORTED
MAX_INTERIOR_SHADING_METHODS_SUPPORTED
MAX_LIGHT_SOURCE_INDICES_SUPPORTED
MAX_LIGHT_SOURCE_TYPES_SUPPORTED
MAX_PARAMETRIC_SURFACE_CHARACTERISTICS_SUPPORTED
MAX_PARAMETRIC_SURFACE_INDICES_SUPPORTED
MAX_POLYLINE_SHADING_METHODS_SUPPORTED
MAX_REFLECTANCE_INDICES_SUPPORTED
MAX_REFLECTANCE_MODELS_SUPPORTED
MAX_REFLECTANCE_PROPERTIES_TYPES_SUPPORTED
MAX_SURFACE_APPROX_CRITERIA_TYPES_SUPPORTED
MAX_VERTEX_SETS_SUPPORTED
MAX_VERTICES_SUPPORTED

The following constant defines the PHIGS PLUS standard implementation dependent value for π:

```
PHIGS_PI                    : constant ANGLE := implementation dependent precision;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567264 600 ■

7 Functions in the Ada Binding of PHIGS PLUS

7.1 Output primitive functions

POLYLINE SET 3 WITH COLOUR

```
procedure POLYLINE_SET
  (VERTICES: in MC.LIST_OF_VERTEX_COLOUR_LIST_SET_3);
```

FILL AREA SET 3 WITH DATA (without Edge Flags)

```
procedure FILL_AREA_SET_WITH_DATA
  (FACET      : in MC.FACET_DATA_SET;
   VERTICES   : in MC.LIST_OF_VERTEX_DATA_LIST_SET_3);
```

FILL AREA SET 3 WITH DATA (with Edge Flags)

```
procedure FILL_AREA_SET_WITH_DATA
  (FACET      : in MC.FACET_DATA_SET;
   EDGES      : in LIST_OF_EDGE_FLAG_LIST;
   VERTICES   : in MC.LIST_OF_VERTEX_DATA_LIST_SET_3);
```

FILL AREA SET WITH DATA (without Edge Flags)

```
procedure FILL_AREA_SET_WITH_DATA
  (FACET      : in MC.FACET_DATA_SET;
   VERTICES   : in MC.LIST_OF_VERTEX_DATA_LIST_SET_2);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567265 547 ■

FILL AREA SET WITH DATA (with Edge Flags)

```
procedure FILL_AREA_SET_WITH_DATA
  (FACET      : in MC.FACET_DATA_SET;
   EDGES      : in LIST_OF_EDGE_FLAG_LIST;
   VERTICES   : in MC.LIST_OF_VERTEX_DATA_LIST_SET_2);
```

CELL ARRAY 3 PLUS

```
procedure CELL_ARRAY
  (CORNER_P  : in MC.POINT_3;
   CORNER_Q  : in MC.POINT_3;
   CORNER_R  : in MC.POINT_3;
   CELLS     : in COLOUR_VALUE_ARRAY);
```

SET OF FILL AREA SETS 3 WITH DATA (without Edge Flags)

```
procedure SET_OF_FILL_AREA_SET_WITH_DATA
  (FACETS     : in MC.FACET_DATA_LIST_SET;
   VERTICES   : in MC.VERTEX_DATA_LIST_SET_3;
   INDICES    : in LIST_OF_LIST_OF_VERTEX_INDEX_LIST);
```

SET OF FILL AREA SETS 3 WITH DATA (with Edge Flags)

```
procedure SET_OF_FILL_AREA_SET_WITH_DATA
  (FACETS     : in MC.FACET_DATA_LIST_SET;
   EDGES      : in LIST_OF_LIST_OF_EDGE_FLAG_LIST;
   VERTICES   : in MC.VERTEX_DATA_LIST_SET_3;
   INDICES    : in LIST_OF_LIST_OF_VERTEX_INDEX_LIST);
```

SET OF FILL AREA SETS WITH DATA (without Edge Flags)

```
procedure SET_OF_FILL_AREA_SET_WITH_DATA
  (FACETS     : in MC.FACET_DATA_LIST_SET;
   VERTICES   : in MC.VERTEX_DATA_LIST_SET_2;
   INDICES    : in LIST_OF_LIST_OF_VERTEX_INDEX_LIST);
```


--`,,`--`,,`,,`,`,,`---

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567266 483 ■

SET OF FILL AREA SETS WITH DATA (with Edge Flags)

```
procedure SET_OF_FILL_AREA_SET_WITH_DATA
  (FACETS      : in MC.FACET_DATA_LIST_SET;
   EDGES       : in LIST_OF_LIST_OF_EDGE_FLAG_LIST;
   VERTICES    : in MC.VERTEX_DATA_LIST_SET_2;
   INDICES     : in LIST_OF_LIST_OF_VERTEX_INDEX_LIST);
```

TRIANGLE SET 3 WITH DATA (without Edge Flags)

```
procedure TRIANGLE_SET_WITH_DATA
  (FACETS      : in MC.FACET_DATA_LIST_SET;
   VERTICES    : in MC.VERTEX_DATA_LIST_SET_3;
   INDICES     : in LIST_OF_VERTEX_INDEX_TRIPLET);
```

TRIANGLE SET 3 WITH DATA (with Edge Flags)

```
procedure TRIANGLE_SET_WITH_DATA
  (FACETS      : in MC.FACET_DATA_LIST_SET;
   EDGES       : in LIST_OF_EDGE_FLAG_TRIPLET;
   VERTICES    : in MC.VERTEX_DATA_LIST_SET_3;
   INDICES     : in LIST_OF_VERTEX_INDEX_TRIPLET);
```

TRIANGLE SET WITH DATA (without Edge Flags)

```
procedure TRIANGLE_SET_WITH_DATA
  (FACETS      : in MC.FACET_DATA_LIST_SET;
   VERTICES    : in MC.VERTEX_DATA_LIST_SET_2;
   INDICES     : in LIST_OF_VERTEX_INDEX_TRIPLET);
```

TRIANGLE SET WITH DATA (with Edge Flags)

```
procedure TRIANGLE_SET_WITH_DATA
  (FACETS      : in MC.FACET_DATA_LIST_SET;
   EDGES       : in LIST_OF_EDGE_FLAG_TRIPLET;
   VERTICES    : in MC.VERTEX_DATA_LIST_SET_2;
   INDICES     : in LIST_OF_VERTEX_INDEX_TRIPLET);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567267 31T ■

TRIANGLE STRIP 3 WITH DATA (without Edge Flags)

```
procedure TRIANGLE_STRIP_WITH_DATA
  (FACETS      : in MC.FACET_DATA_LIST_SET;
   VERTICES    : in MC.VERTEX_DATA_LIST_SET_3);
```

TRIANGLE STRIP 3 WITH DATA (with Edge Flags)

```
procedure TRIANGLE_STRIP_WITH_DATA
  (FACETS      : in MC.FACET_DATA_LIST_SET;
   EDGES       : in EDGE_FLAG_LIST;
   VERTICES    : in MC.VERTEX_DATA_LIST_SET_3);
```

TRIANGLE STRIP WITH DATA (without Edge Flags)

```
procedure TRIANGLE_STRIP_WITH_DATA
  (FACETS      : in MC.FACET_DATA_LIST_SET;
   VERTICES    : in MC.VERTEX_DATA_LIST_SET_2);
```

TRIANGLE STRIP WITH DATA (with Edge Flags)

```
procedure TRIANGLE_STRIP_WITH_DATA
  (FACETS      : in MC.FACET_DATA_LIST_SET;
   EDGES       : in EDGE_FLAG_LIST;
   VERTICES    : in MC.VERTEX_DATA_LIST_SET_2);
```

QUADRILATERAL MESH 3 WITH DATA (without Edge Flags)

```
procedure QUADRILATERAL_MESH_WITH_DATA
  (FACETS      : in MC.FACET_DATA_ARRAY_SET;
   VERTICES    : in MC.VERTEX_DATA_ARRAY_SET_3);
```

The FACETS array must have dimension one less than the dimensions of the VERTICES array. If this relationship is violated, the following binding specific error must be generated:

2502 Ignoring function, the parameters have inconsistent dimensions.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567268 256 ■

QUADRILATERAL MESH 3 WITH DATA (with Edge Flags)

```
procedure QUADRILATERAL_MESH_WITH_DATA
  (FACETS     : in MC.FACET_DATA_ARRAY_SET;
   EDGES      : in ARRAY_OF_EDGE_FLAG_PAIR;
   VERTICES   : in MC.VERTEX_DATA_ARRAY_SET_3);
```

The EDGES and VECTICES arrays must have the same dimensions. The FACETS array must have dimension one less than the dimensions of the VERTICES array. If either of these two relationships is violated, the following binding specific error must be generated:

2502 Ignoring function, the parameters have inconsistent dimensions.

QUADRILATERAL MESH WITH DATA (without Edge Flags)

```
procedure QUADRILATERAL_MESH_WITH_DATA
  (FACETS     : in MC.FACET_DATA_ARRAY_SET;
   VERTICES   : in MC.VERTEX_DATA_ARRAY_SET_2);
```

The FACETS array must have dimension one less than the dimensions of the VERTICES array. If this relationship is violated, the following binding specific error must be generated:

2502 Ignoring function, the parameters have inconsistent dimensions.

QUADRILATERAL MESH WITH DATA (with Edge Flags)

```
procedure QUADRILATERAL_MESH_WITH_DATA
  (FACETS     : in MC.FACET_DATA_ARRAY_SET;
   EDGES      : in ARRAY_OF_EDGE_FLAG_PAIR;
   VERTICES   : in MC.VERTEX_DATA_ARRAY_SET_2);
```

The EDGES and VECTICES arrays must have the same dimensions. The FACETS array must have dimension one less than the dimensions of the VERTICES array. If either of these two relationships is violated, the following binding specific error must be generated:

2502 Ignoring function, the parameters have inconsistent dimensions.

NON-UNIFORM B-SPLINE CURVE

```
procedure NON_UNIFORM_B_SPLINE_CURVE
  (SPLINE    : in MC.CURVE_GEOMETRYSPLINE_3;
   LIMITS    : in SPC.RANGE_OF_MAGNITUDES);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567269 192 ■

NON-UNIFORM B-SPLINE CURVE WITH COLOUR

```
procedure NON_UNIFORM_B_SPLINE_CURVE
  (CURVE_SPLINE  : in MC.CURVE_GEOMETRYSPLINE_3;
   LIMITS        : in SPC.RANGE_OF_MAGNITUDES;
   COLOUR        : in CURVE_COLOURSPLINE);
```

NON-UNIFORM B-SPLINE SURFACE

```
procedure NON_UNIFORM_B_SPLINE_SURFACE
   (SURFACE_SPLINE : in MC.SURFACE_GEOMETRYSPLINE;
   TRIM_CURVES     : in LIST_OF_TRIMCURVE_LIST);
```

NON-UNIFORM B-SPLINE SURFACE WITH DATA

```
procedure NON_UNIFORM_B_SPLINE_SURFACE_WITH_DATA
  (SURFACE_SPLINE : in MC.SURFACE_GEOMETRYSPLINE;
   TRIM_CURVES    : in LIST_OF_TRIMCURVE_LIST;
   COLOUR_SPLINE  : in SURFACE_COLOURSPLINE;
   DATA_SPLINES   : in DATASPLINE_LIST);
```

7.2 Attribute specification functions

SET DATA MAPPING INDEX

```
procedure SET_DATA_MAPPING_INDEX
  (DATA_MAPPING_IND : in DATA_MAPPING_INDEX);
```

SET REFLECTANCE INDEX

```
procedure SET_REFLECTANCE_INDEX
  (REFLECTANCE_IND : in REFLECTANCE_INDEX);
```

SET BACK INTERIOR INDEX

```
procedure SET_BACK_INTERIOR_INDEX
  (BACK_INTERIOR_IND : in INTERIOR_INDEX);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

---

SET BACK DATA MAPPING INDEX

```
procedure SET_BACK_DATA_MAPPING_INDEX
  (BACK_DATA_MAPPING_IND : in DATA_MAPPING_INDEX);
```

SET BACK REFLECTANCE INDEX

```
procedure SET_BACK_REFLECTANCE_INDEX
  (BACK_REFLECTANCE_IND : in REFLECTANCE_INDEX);
```

SET PARAMETRIC SURFACE INDEX

```
procedure SET_PARAMETRIC_SURFACE_INDEX
  (PARAMETRIC_SURFACE_IND : in PARAMETRIC_SURFACE_INDEX);
```

SET POLYLINE COLOUR

```
procedure SET_POLYLINE_COLOUR
  (GENERAL_POLYLINE_COLOUR : in GENERAL_COLOUR);
```

SET POLYLINE SHADING METHOD

```
procedure SET_POLYLINE_SHADING_METHOD
  (POLYLINE_SHADING : in POLYLINE_SHADING_METHOD);
```

SET POLYMARKER COLOUR

```
procedure SET_POLYMARKER_COLOUR
  (GENERAL_POLYMARKER_COLOUR : in GENERAL_COLOUR);
```

SET TEXT COLOUR

```
procedure SET_TEXT_COLOUR
  (GENERAL_TEXT_COLOUR : in GENERAL_COLOUR);
```

SET FACET DISTINGUISHING MODE

```
procedure SET_FACET_DISTINGUISHING_MODE
  (FACET_DISTINGUISHING : in FACET_DISTINGUISHING_MODE);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

---

SET FACET CULLING MODE

```
procedure SET_FACET_CULLING_MODE
  (FACET_CULLING : in FACET_CULLING_MODE);
```

SET INTERIOR COLOUR

```
procedure SET_INTERIOR_COLOUR
  (GENERAL_INTERIOR_COLOUR : in GENERAL_COLOUR);
```

SET INTERIOR SHADING METHOD

```
procedure SET_INTERIOR_SHADING_METHOD
  (INTERIOR_SHADING : in INTERIOR_SHADING_METHOD);
```

SET DATA MAPPING METHOD

```
procedure SET_DATA_MAPPING_METHOD
  (METHOD_OF_DATA_MAPPING : in DATA_MAPPING_DATA_RECORD);
```

SET REFLECTANCE PROPERTIES

```
procedure SET_REFLECTANCE_PROPERTIES
   (REFLECTANCE_DATA : in REFLECTANCE_PROPERTIES_DATA_RECORD);
```

SET_REFLECTANCE_MODEL

```
procedure SET_REFLECTANCE_MODEL
  (REFLECTANCE : in REFLECTANCE_MODEL);
```

SET BACK INTERIOR STYLE

```
procedure SET_BACK_INTERIOR_STYLE
  (BACK_STYLE_OF_INTERIOR : in INTERIOR_STYLE);
```

SET BACK INTERIOR STYLE INDEX

```
procedure SET_BACK_INTERIOR_STYLE_INDEX
  (BACK_STYLE_IND : in STYLE_INDEX);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567272 787 ■

SET BACK INTERIOR COLOUR

```
procedure SET_BACK_INTERIOR_COLOUR
  (BACK_GENERAL_INTERIOR_COLOUR : in GENERAL_COLOUR);
```

SET BACK INTERIOR SHADING METHOD

```
procedure SET_BACK_INTERIOR_SHADING_METHOD
  (BACK_INTERIOR_SHADING : in INTERIOR_SHADING_METHOD);
```

SET BACK DATA MAPPING METHOD

```
procedure SET_BACK_DATA_MAPPING_METHOD
  (BACK_METHOD_OF_DATA_MAPPING : in DATA_MAPPING_DATA_RECORD);
```

SET_BACK_REFLECTANCE_PROPERTIES

```
procedure SET_BACK_REFLECTANCE_PROPERTIES
   (BACK_REFLECTANCE_DATA : in REFLECTANCE_PROPERTIES_DATA_RECORD);
```

SET_BACK_REFLECTANCE_MODEL

```
procedure SET_BACK_REFLECTANCE_MODEL
  (BACK_REFLECTANCE : in REFLECTANCE_MODEL);
```

SET LIGHT SOURCE STATE

```
procedure SET_LIGHT_SOURCE_STATE
  (ACTIVATION_LIST     : in LIGHT_SOURCE_INDICES.LIST_OF;
   DEACTIVATION_LIST   : in LIGHT_SOURCE_INDICES.LIST_OF);
```

SET EDGE COLOUR

```
procedure SET_EDGE_COLOUR
  (GENERAL_EDGE_COLOUR : GENERAL_COLOUR);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567273 613 ■

SET CURVE APPROXIMATION CRITERIA

```
procedure SET_CURVE_APPROXIMATION_CRITERIA
  (CRITERIA_FOR_CURVE_APPROX : in CURVE_APPROX_DATA_RECORD);
```

SET SURFACE APPROXIMATION CRITERIA

```
procedure SET_SURFACE_APPROXIMATION_CRITERIA
  (CRITERIA_FOR_SURFACE_APPROX : in SURFACE_APPROX_DATA_RECORD);
```

SET PARAMETRIC SURFACE CHARACTERISTICS

```
procedure SET_PARAMETRIC_SURFACE_CHARACTERISTICS
  (PARAMETRIC_SURFACE_CHARACTERISTICS
                                  : in PARAMETRIC_SURFACE_DATA_RECORD);
```

SET RENDERING COLOUR MODEL

```
procedure SET_RENDERING_COLOUR_MODEL
  (RENDERING_COLOUR_MODEL : in COLOUR_MODEL);
```

SET DEPTH CUE INDEX

```
procedure SET_DEPTH_CUE_INDEX
  (DEPTH_CUE_IND : in DEPTH_CUE_INDEX);
```

SET COLOUR MAPPING INDEX

```
procedure SET_COLOUR_MAPPING_INDEX
  (COLOUR_MAPPING_IND : in COLOUR_MAPPING_INDEX);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567274 55T ■

SET POLYLINE REPRESENTATION PLUS

```
procedure SET_POLYLINE_REPRESENTATION
  (WS                          : in WS_ID;
   POLYLINE_IND                : in POLYLINE_INDEX;
   TYPE_OF_LINE                : in LINETYPE;
   LINEWIDTH_SF                : in LINEWIDTH;
   GENERAL_LINE_COLOUR         : in GENERAL_COLOUR;
   LINE_SHADING                : in POLYLINE_SHADING_METHOD;
   CRITERIA_FOR_CURVE_APPROX   : in CURVE_APPROX_DATA_RECORD);
```

SET POLYMARKER REPRESENTATION PLUS

```
procedure SET_POLYMARKER_REPRESENTATION
  (WS                          : in WS_ID;
   POLYMARKER_IND              : in POLYMARKER_INDEX;
   TYPE_OF_MARKER              : in MARKER_TYPE;
   SIZE                        : in MARKER_SIZE;
   GENERAL_MARKER_COLOUR       : in GENERAL_COLOUR);
```

SET TEXT REPRESENTATION PLUS

```
procedure SET_TEXT_REPRESENTATION
  (WS                          : in WS_ID;
   TEXT_IND                    : in TEXT_INDEX;
   FONT                        : in TEXT_FONT;
   PRECISION                   : in TEXT_PRECISION;
   EXPANSION                   : in CHAR_EXPANSION;
   SPACING                     : in CHAR_SPACING;
   GENERAL_TEXT_COLOUR         : in GENERAL_COLOUR);
```

SET INTERIOR REPRESENTATION PLUS

```
procedure SET_INTERIOR_REPRESENTATION
  (WS                          : in WS_ID;
   INTERIOR_IND                : in INTERIOR_INDEX;
   STYLE_OF_INTERIOR           : in INTERIOR_STYLE;
   STYLE_IND                   : in STYLE_INDEX;
   GENERAL_INTERIOR_COLOUR     : in GENERAL_COLOUR;
   INTERIOR_SHADING            : in INTERIOR_SHADING_METHOD);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567275 496 ■

SET EDGE REPRESENTATION PLUS

```
procedure SET_EDGE_REPRESENTATION
  (WS                    : in WS_ID;
   EDGE_IND              : in EDGE_INDEX;
   FLAG                  : in EDGE_FLAG;
   TYPE_OF_EDGE          : in EDGETYPE;
   EDGEWIDTH_SF          : in EDGEWIDTH;
   GENERAL_EDGE_COLOUR   : in GENERAL_COLOUR);
```

SET DATA MAPPING REPRESENTATION

```
procedure SET_DATA_MAPPING_REPRESENTATION
  (WS                : in WS_ID;
   DATA_MAPPING_IND  : in DATA_MAPPING_INDEX;
   DATA_MAPPING      : in DATA_MAPPING_DATA_RECORD);
```

SET REFLECTANCE REPRESENTATION

```
procedure SET_REFLECTANCE_REPRESENTATION
  (WS                        : in WS_ID;
   REFLECTANCE_IND           : in REFLECTANCE_INDEX;
   MODEL_OF_REFLECTANCE      : in REFLECTANCE_MODEL;
   REFLECTANCE_PROPERTIES    : in REFLECTANCE_PROPERTIES_DATA_RECORD);
```

SET PARAMETRIC SURFACE REPRESENTATION

```
procedure SET_PARAMETRIC_SURFACE_REPRESENTATION
  (WS                                : in WS_ID;
   PARAMETRIC_SURFACE_IND            : in PARAMETRIC_SURFACE_INDEX;
   CRITERIA_FOR_SURFACE_APPROX       : in SURFACE_APPROX_DATA_RECORD;
   PARAMETRIC_SURFACE_CHARACTERISTICS
                                     : in PARAMETRIC_SURFACE_DATA_RECORD);
```

SET PATTERN REPRESENTATION PLUS

```
procedure SET_PATTERN_REPRESENTATION
  (WS                : in WS_ID;
   PATTERN_IND       : in PATTERN_INDEX;
   GENERAL_PATTERN   : in COLOUR_VALUE_ARRAY);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

---

SET LIGHT SOURCE REPRESENTATION

```
procedure SET_LIGHT_SOURCE_REPRESENTATION
  (WS                 : in WS_ID;
   LIGHT_SOURCE_IND   : in LIGHT_SOURCE_INDEX;
   LIGHT_SOURCE       : in LIGHT_SOURCE_DATA_RECORD);
```

SET DEPTH CUE REPRESENTATION

```
procedure SET_DEPTH_CUE_REPRESENTATION
  (WS                 : in WS_ID;
   DEPTH_CUE_IND      : in DEPTH_CUE_INDEX;
   MODE               : in DEPTH_CUE_MODE;
   REFERENCE_PLANES   : in NPC.RANGE_OF_MAGNITUDES;
   DEPTH_CUE_SF       : in DEPTH_CUE_SCALE_FACTORS;
   DEPTH_CUE_COLOUR   : in GENERAL_COLOUR);
```

SET COLOUR MAPPING REPRESENTATION

```
procedure SET_COLOUR_MAPPING_REPRESENTATION
  (WS                 : in WS_ID;
   COLOUR_MAPPING_IND : in COLOUR_MAPPING_INDEX;
   COLOUR_MAPPING     : in COLOUR_MAPPING_DATA_RECORD);
```

7.3 Inquiry functions

The following should be added:

INQUIRE POLYLINE REPRESENTATION PLUS

```
procedure INQ_POLYLINE_REPRESENTATION
  (WS                          : in  WS_ID;
   POLYLINE_IND                : in  POLYLINE_INDEX;
   RETURNED_VALUES             : in  RETURN_VALUE_TYPE;
   ERROR_INDICATOR             : out ERROR_NUMBER;
   TYPE_OF_LINE                : out LINETYPE;
   LINEWIDTH_SF                : out LINEWIDTH;
   GENERAL_LINE_COLOUR         : out GENERAL_COLOUR;
   LINE_SHADING                : out POLYLINE_SHADING_METHOD;
   CRITERIA_FOR_CURVE_APPROX   : out CURVE_APPROX_DATA_RECORD);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

---

INQUIRE POLYMARKER REPRESENTATION PLUS

```
procedure INQ_POLYMARKER_REPRESENTATION
  (WS                          : in  WS_ID;
   POLYMARKER_IND              : in  POLYMARKER_INDEX;
   RETURNED_VALUES             : in  RETURN_VALUE_TYPE;
   ERROR_INDICATOR             : out ERROR_NUMBER;
   TYPE_OF_MARKER              : out MARKER_TYPE;
   SIZE                        : out MARKER_SIZE;
   GENERAL_MARKER_COLOUR : out GENERAL_COLOUR);
```

INQUIRE TEXT REPRESENTATION PLUS

```
procedure INQ_TEXT_REPRESENTATION
  (WS                          : in  WS_ID;
   TEXT_IND                    : in  TEXT_INDEX;
   RETURNED_VALUES             : in  RETURN_VALUE_TYPE;
   ERROR_INDICATOR             : out ERROR_NUMBER;
   FONT                        : out TEXT_FONT;
   PRECISION                   : out TEXT_PRECISION;
   EXPANSION                   : out CHAR_EXPANSION;
   SPACING                     : out CHAR_SPACING;
   GENERAL_TEXT_COLOUR : out GENERAL_COLOUR);
```

INQUIRE INTERIOR REPRESENTATION PLUS

```
procedure INQ_INTERIOR_REPRESENTATION
  (WS                          : in  WS_ID;
   INTERIOR_IND                : in  INTERIOR_INDEX;
   RETURNED_VALUES             : in  RETURN_VALUE_TYPE;
   ERROR_INDICATOR             : out ERROR_NUMBER;
   STYLE_OF_INTERIOR           : out INTERIOR_STYLE;
   STYLE_IND                   : out STYLE_INDEX;
   GENERAL_INTERIOR_COLOUR : out GENERAL_COLOUR;
   INTERIOR_SHADING            : out INTERIOR_SHADING_METHOD);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

INQUIRE EDGE REPRESENTATION PLUS

```
procedure INQ_EDGE_REPRESENTATION
  (WS                    : in  WS_ID;
   EDGE_IND              : in  EDGE_INDEX;
   RETURNED_VALUES       : in  RETURN_VALUE_TYPE;
   ERROR_INDICATOR       : out ERROR_NUMBER;
   FLAG                  : out EDGE_FLAG;
   TYPE_OF_EDGE          : out EDGETYPE;
   EDGEWIDTH_SF          : out EDGEWIDTH;
   GENERAL_EDGE_COLOUR   : out GENERAL_COLOUR);
```

INQUIRE LIST OF DATA MAPPING INDICES

```
procedure INQ_LIST_OF_DATA_MAPPING_INDICES
  (WS                : in  WS_ID;
   ERROR_INDICATOR   : out ERROR_NUMBER;
   INDICES           : out DATA_MAPPING_INDICES.LIST_OF);
```

INQUIRE DATA MAPPING REPRESENTATION

```
procedure INQ_DATA_MAPPING_REPRESENTATION
  (WS                  : in  WS_ID;
   DATA_MAPPING_IND    : in  DATA_MAPPING_INDEX;
   RETURNED_VALUES     : in  RETURN_VALUE_TYPE;
   ERROR_INDICATOR     : out ERROR_NUMBER;
   DATA_MAPPING        : out DATA_MAPPING_DATA_RECORD);
```

INQUIRE LIST OF REFLECTANCE INDICES

```
procedure INQ_LIST_OF_REFLECTANCE_INDICES
  (WS                  : in  WS_ID;
   ERROR_INDICATOR     : out ERROR_NUMBER;
   INDICES             : out REFLECTANCE_INDICES.LIST_OF);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

---

INQUIRE REFLECTANCE REPRESENTATION

```
procedure INQ_REFLECTANCE_REPRESENTATION
  (WS                        : in  WS_ID;
   REFLECTANCE_IND           : in  REFLECTANCE_INDEX;
   RETURNED_VALUES           : in  RETURN_VALUE_TYPE;
   ERROR_INDICATOR           : out ERROR_NUMBER;
   MODEL                     : out REFLECTANCE_MODEL;
   REFLECTANCE_PROPERTIES    : out REFLECTANCE_PROPERTIES_DATA_RECORD);
```

INQUIRE LIST OF PARAMETRIC SURFACE INDICES

```
procedure INQ_LIST_OF_PARAMETRIC_SURFACE_INDICES
  (WS                : in  WS_ID;
   ERROR_INDICATOR   : out ERROR_NUMBER;
   INDICES           : out PARAMETRIC_SURFACE_INDICES.LIST_OF);
```

INQUIRE PARAMETRIC SURFACE REPRESENTATION

```
procedure INQ_PARAMETRIC_SURFACE_REPRESENTATION
  (WS                                : in  WS_ID;
   PARAMETRIC_SURFACE_IND            : in  PARAMETRIC_SURFACE_INDEX;
   RETURNED_VALUES                   : in  RETURN_VALUE_TYPE;
   ERROR_INDICATOR                   : out ERROR_NUMBER;
   CRITERIA_FOR_SURFACE_APPROX       : out SURFACE_APPROX_DATA_RECORD;
   PARAMETRIC_SURFACE_CHARACTERISTICS
                                     : out PARAMETRIC_SURFACE_DATA_RECORD);
```

INQUIRE PATTERN REPRESENTATION PLUS

```
procedure INQ_PATTERN_REPRESENTATION
  (WS                : in  WS_ID;
   PATTERN_IND       : in  PATTERN_INDEX;
   RETURNED_VALUES   : in  RETURN_VALUE_TYPE;
   ERROR_INDICATOR   : out ERROR_NUMBER;
   GENERAL_PATTERN   : out ACCESS_COLOUR_VALUE_ARRAY);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567280 853 ■

INQUIRE LIST OF LIGHT SOURCE INDICES

```
procedure INQ_LIST_OF_LIGHT_SOURCE_INDICES
  (WS                : in  WS_ID;
   ERROR_INDICATOR   : out ERROR_NUMBER;
   INDICES           : out LIGHT_SOURCE_INDICES.LIST_OF);
```

INQUIRE LIGHT SOURCE REPRESENTATION

```
procedure INQ_LIGHT_SOURCE_REPRESENTATION
  (WS                : in  WS_ID;
   LIGHT_SOURCE_IND  : in  LIGHT_SOURCE_INDEX;
   RETURNED_VALUES   : in  RETURN_VALUE_TYPE;
   ERROR_INDICATOR   : out ERROR_NUMBER;
   LIGHT_SOURCE      : out LIGHT_SOURCE_DATA_RECORD);
```

INQUIRE LIST OF DEPTH CUE INDICES

```
procedure INQ_LIST_OF_DEPTH_CUE_INDICES
  (WS                : in  WS_ID;
   ERROR_INDICATOR   : out ERROR_NUMBER;
   INDICES           : out DEPTH_CUE_INDICES.LIST_OF);
```

INQUIRE DEPTH CUE REPRESENTATION

```
procedure INQ_DEPTH_CUE_REPRESENTATION
  (WS                : in  WS_ID;
   DEPTH_CUE_IND     : in  DEPTH_CUE_INDEX;
   RETURNED_VALUES   : in  RETURN_VALUE_TYPE;
   ERROR_INDICATOR   : out ERROR_NUMBER;
   MODE              : out DEPTH_CUE_MODE;
   REFERENCE_PLANES  : out NPC.RANGE_OF_MAGNITUDES;
   DEPTH_CUE_SF      : out DEPTH_CUE_SCALE_FACTORS;
   DEPTH_CUE_COLOUR  : out GENERAL_COLOUR);
```

INQUIRE COLOUR MAPPING STATE

```
procedure INQ_COLOUR_MAPPING_STATE
  (WS                : in  WS_ID;
   METHOD            : in  COLOUR_MAPPING_METHOD;
   ERROR_INDICATOR   : out ERROR_NUMBER;
   STATE             : out COLOUR_MAPPING_STATE);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567281 79T ■

INQUIRE LIST OF COLOUR MAPPING INDICES

```
procedure INQ_LIST_OF_COLOUR_MAPPING_INDICES
  (WS                : in  WS_ID;
   ERROR_INDICATOR   : out ERROR_NUMBER;
   INDICES           : out COLOUR_MAPPING_INDICES.LIST_OF);
```

INQUIRE COLOUR MAPPING REPRESENTATION

```
procedure INQ_COLOUR_MAPPING_REPRESENTATION
  (WS                    : in  WS_ID;
   COLOUR_MAPPING_IND    : in  COLOUR_MAPPING_INDEX;
   RETURNED_VALUES       : in  RETURN_VALUE_TYPE;
   ERROR_INDICATOR       : out ERROR_NUMBER;
   COLOUR_MAPPING        : out ACCESS_COLOUR_MAPPING_DATA_RECORD);
```

INQUIRE DIRECT COLOUR MODEL FACILITIES

```
procedure INQ_DIRECT_COLOUR_MODEL_FACILITIES
  (TYPE_OF_WS        : in  WS_TYPE;
   ERROR_INDICATOR   : out ERROR_NUMBER;
   LIST_OF_MODELS    : out COLOUR_MODELS.LIST_OF);
```

INQUIRE RENDERING COLOUR MODEL FACILITIES

```
procedure INQ_RENDERING_COLOUR_MODEL_FACILITIES
  (TYPE_OF_WS        : in  WS_TYPE;
   ERROR_INDICATOR   : out ERROR_NUMBER;
   LIST_OF_MODELS    : out COLOUR_MODELS.LIST_OF);
```

INQUIRE DYNAMICS OF WORKSTATION ATTRIBUTES PLUS

```
procedure INQ_DYNAMICS_OF_WS_ATTRIBUTES
  (TYPE_OF_WS                          : in  WS_TYPE;
   ERROR_INDICATOR                     : out ERROR_NUMBER;
   DATA_MAPPING_REPRESENTATION         : out DYNAMIC_MODIFICATION;
   REFLECTANCE_REPRESENTATION          : out DYNAMIC_MODIFICATION;
   PARAMETRIC_SURFACE_REPRESENTATION   : out DYNAMIC_MODIFICATION;
   WS_LIGHT_SOURCE_REPRESENTATION      : out DYNAMIC_MODIFICATION;
   DEPTH_CUE_REPRESENTATION            : out DYNAMIC_MODIFICATION;
   COLOUR_MAPPING_REPRESENTATION       : out DYNAMIC_MODIFICATION);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567282 626 ■

INQUIRE POLYLINE FACILITIES PLUS

```
procedure INQ_POLYLINE_FACILITIES
  (TYPE_OF_WS                  : in  WS_TYPE;
   ERROR_INDICATOR             : out ERROR_NUMBER;
   NUMBER_OF_TYPES             : out PHIGS_INTEGER;
   LIST_OF_TYPES               : out LINETYPES.LIST_OF;
   NUMBER_OF_WIDTHS            : out PHIGS_NATURAL;
   NOMINAL_WIDTH               : out DC.MAGNITUDE;
   RANGE_OF_WIDTHS             : out DC.RANGE_OF_MAGNITUDES;
   LIST_OF_SHADING_METHODS     : out POLYLINE_SHADING_METHODS.LIST_OF;
   NUMBER_OF_INDICES           : out PHIGS_NATURAL);
```

INQUIRE PREDEFINED POLYLINE REPRESENTATION PLUS

```
procedure INQ_PREDEFINED_POLYLINE_REPRESENTATION
  (TYPE_OF_WS                  : in  WS_TYPE;
   POLYLINE_IND                : in  POLYLINE_INDEX;
   ERROR_INDICATOR             : out ERROR_NUMBER;
   TYPE_OF_LINE                : out LINETYPE;
   LINEWIDTH_SF                : out LINEWIDTH;
   GENERAL_LINE_COLOUR         : out GENERAL_COLOUR;
   LINE_SHADING                : out POLYLINE_SHADING_METHOD;
   CRITERIA_FOR_CURVE_APPROX   : out CURVE_APPROX_DATA_RECORD);
```

INQUIRE PREDEFINED POLYMARKER REPRESENTATION PLUS

```
procedure INQ_PREDEFINED_POLYMARKER_REPRESENTATION
  (TYPE_OF_WS                  : in  WS_TYPE;
   POLYMARKER_IND              : in  POLYMARKER_INDEX;
   ERROR_INDICATOR             : out ERROR_NUMBER;
   TYPE_OF_MARKER              : out MARKER_TYPE;
   SIZE                        : out MARKER_SIZE;
   GENERAL_MARKER_COLOUR       : out GENERAL_COLOUR);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567283 562 ■

INQUIRE PREDEFINED TEXT REPRESENTATION PLUS

```
procedure INQ_PREDEFINED_TEXT_REPRESENTATION
  (TYPE_OF_WS              : in  WS_TYPE;
   TEXT_IND                : in  TEXT_INDEX;
   ERROR_INDICATOR         : out ERROR_NUMBER;
   FONT                    : out TEXT_FONT;
   PRECISION               : out TEXT_PRECISION;
   EXPANSION               : out CHAR_EXPANSION;
   SPACING                 : out CHAR_SPACING;
   GENERAL_TEXT_COLOUR     : out GENERAL_COLOUR);
```

INQUIRE INTERIOR FACILITIES PLUS

```
procedure INQ_INTERIOR_FACILITIES
  (TYPE_OF_WS              : in  WS_TYPE;
   ERROR_INDICATOR         : out ERROR_NUMBER;
   LIST_OF_INTERIOR_STYLES : out INTERIOR_STYLES.LIST_OF;
   NUMBER_OF_HATCH_STYLES  : out PHIGS_NATURAL;
   LIST_OF_HATCH_STYLES    : out HATCH_STYLES.LIST_OF;
   LIST_OF_SHADING_METHODS : out INTERIOR_SHADING_METHODS.LIST_OF;
   NUMBER_OF_INDICES       : out PHIGS_NATURAL);
```

INQUIRE PREDEFINED INTERIOR REPRESENTATION PLUS

```
procedure INQ_PREDEFINED_INTERIOR_REPRESENTATION
  (TYPE_OF_WS              : in  WS_TYPE;
   INTERIOR_IND            : in  INTERIOR_INDEX;
   ERROR_INDICATOR         : out ERROR_NUMBER;
   STYLE_OF_INTERIOR       : out INTERIOR_STYLE;
   STYLE_IND               : out STYLE_INDEX;
   GENERAL_INTERIOR_COLOUR : out GENERAL_COLOUR;
   INTERIOR_SHADING        : out INTERIOR_SHADING_METHOD);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

INQUIRE PREDEFINED EDGE REPRESENTATION PLUS

```
procedure INQ_PREDEFINED_EDGE_REPRESENTATION
  (TYPE_OF_WS              : in  WS_TYPE;
   EDGE_IND                : in  EDGE_INDEX;
   ERROR_INDICATOR         : out ERROR_NUMBER;
   FLAG                    : out EDGE_FLAG;
   TYPE_OF_EDGE            : out EDGETYPE;
   EDGEWIDTH_SF            : out EDGEWIDTH;
   GENERAL_EDGE_COLOUR     : out GENERAL_COLOUR);
```

INQUIRE DATA MAPPING FACILITIES

```
procedure INQ_DATA_MAPPING_FACILITIES
  (TYPE_OF_WS              : in  WS_TYPE;
   ERROR_INDICATOR         : out ERROR_NUMBER;
   LIST_OF_METHODS         : out DATA_MAPPING_METHODS.LIST_OF;
   NUMBER_OF_INDICES       : out PHIGS_POSITIVE);
```

INQUIRE PREDEFINED DATA MAPPING REPRESENTATION

```
procedure INQ_PREDEFINED_DATA_MAPPING_REPRESENTATION
  (TYPE_OF_WS              : in  WS_TYPE;
   DATA_MAPPING_IND        : in  DATA_MAPPING_INDEX;
   ERROR_INDICATOR         : out ERROR_NUMBER;
   DATA_MAPPING            : out DATA_MAPPING_DATA_RECORD);
```

INQUIRE REFLECTANCE FACILITIES

```
procedure INQ_REFLECTANCE_FACILITIES
  (TYPE_OF_WS              : in  WS_TYPE;
   ERROR_INDICATOR         : out ERROR_NUMBER;
   MODELS                  : out REFLECTANCE_MODELS.LIST_OF;
   PROPERTIES              : out REFLECTANCE_PROPERTIES_TYPES.LIST_OF;
   NUMBER_OF_INDICES       : out PHIGS_POSITIVE);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567285 335 ■

INQUIRE PREDEFINED REFLECTANCE REPRESENTATION

```
procedure INQ_PREDEFINED_REFLECTANCE_REPRESENTATION
  (TYPE_OF_WS                 : in  WS_TYPE;
   REFLECTANCE_IND            : in  REFLECTANCE_INDEX;
   ERROR_INDICATOR            : out ERROR_NUMBER;
   MODEL                      : out REFLECTANCE_MODEL;
   REFLECTANCE_PROPERTIES     : out REFLECTANCE_PROPERTIES_DATA_RECORD);
```

INQUIRE CURVE AND SURFACE FACILITIES

```
procedure INQ_CURVE_AND_SURFACE_FACILITIES
  (TYPE_OF_WS                 : in  WS_TYPE;
   ERROR_INDICATOR            : out ERROR_NUMBER;
   MAX_CURVE_ORDER            : out RETURN_SPLINE_ORDER;
   MAX_SURFACE_ORDER          : out RETURN_SPLINE_ORDER;
   MAX_TRIM_ORDER             : out RETURN_TRIMCURVE_ORDER;
   LIST_OF_CURVE_APPROX_CRITERIA_TYPES
                              : out CURVE_APPROX_CRITERIA_TYPES.LIST_OF;
   LIST_OF_SURFACE_APPROX_CRITERIA_TYPES
                              : out SURFACE_APPROX_CRITERIA_TYPES.LIST_OF;
   LIST_OF_TRIM_APPROX_CRITERIA_TYPES
                              : out CURVE_APPROX_CRITERIA_TYPES.LIST_OF;
   LIST_OF_CHARACTERISTICS
                          : out PARAMETRIC_SURFACE_CHARACTERISTICS.LIST_OF;
   NUMBER_OF_INDICES          : out PHIGS_POSITIVE);
```

INQUIRE PREDEFINED PARAMETRIC SURFACE REPRESENTATION

```
procedure INQ_PREDEFINED_PARAMETRIC_SURFACE_REPRESENTATION
  (TYPE_OF_WS                   : in  WS_TYPE;
   PARAMETRIC_SURFACE_IND       : in  PARAMETRIC_SURFACE_INDEX;
   ERROR_INDICATOR              : out ERROR_NUMBER;
   CRITERIA_FOR_SURFACE_APPROX  : out SURFACE_APPROX_DATA_RECORD;
   PARAMETRIC_SURFACE_CHARACTERISTICS
                                : out PARAMETRIC_SURFACE_DATA_RECORD);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

INQUIRE PREDEFINED PATTERN REPRESENTATION PLUS

```
procedure INQ_PREDEFINED_PATTERN_REPRESENTATION
  (TYPE_OF_WS        : in  WS_TYPE;
   PATTERN_IND       : in  PATTERN_INDEX;
   ERROR_INDICATOR   : out ERROR_NUMBER;
   PATTERN           : out ACCESS_COLOUR_VALUE_ARRAY);
```

INQUIRE LIGHT SOURCE FACILITIES

```
procedure INQ_LIGHT_SOURCE_FACILITIES
  (TYPE_OF_WS                 : in  WS_TYPE;
   ERROR_INDICATOR            : out ERROR_NUMBER;
   LIST_OF_TYPES              : out LIGHT_SOURCE_TYPES.LIST_OF;
   MAX_SIMULTANEOUS_LIGHTS    : out PHIGS_POSITIVE;
   NUMBER_OF_INDICES          : out RETURN_LIGHT_SOURCE_COUNT);
```

INQUIRE PREDEFINED LIGHT SOURCE REPRESENTATION

```
procedure INQ_PREDEFINED_LIGHT_SOURCE_REPRESENTATION
  (TYPE_OF_WS        : in  WS_TYPE;
   LIGHT_SOURCE_IND  : in  LIGHT_SOURCE_INDEX;
   ERROR_INDICATOR   : out ERROR_NUMBER;
   LIGHT_SOURCE      : out LIGHT_SOURCE_DATA_RECORD);
```

INQUIRE DEPTH CUE FACILITIES

```
procedure INQ_DEPTH_CUE_FACILITIES
  (TYPE_OF_WS        : in  WS_TYPE;
   ERROR_INDICATOR   : out ERROR_NUMBER;
   NUMBER_OF_INDICES : out RETURN_DEPTH_CUE_INDEX_COUNT;
   AVAILABLE_MODES   : out DEPTH_CUE_MODES.LIST_OF);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

---

INQUIRE PREDEFINED DEPTH CUE REPRESENTATION

```
procedure INQ_PREDEFINED_DEPTH_CUE_REPRESENTATION
  (TYPE_OF_WS          : in  WS_TYPE;
   DEPTH_CUE_IND       : in  DEPTH_CUE_INDEX;
   ERROR_INDICATOR     : out ERROR_NUMBER;
   MODE                : out DEPTH_CUE_MODE;
   REFERENCE_PLANES    : out NPC.RANGE_OF_MAGNITUDES;
   DEPTH_CUE_SF        : out DEPTH_CUE_SCALE_FACTORS;
   DEPTH_CUE_COLOUR    : out GENERAL_COLOUR);
```

INQUIRE COLOUR MAPPING FACILITIES

```
procedure INQ_COLOUR_MAPPING_FACILITIES
  (TYPE_OF_WS          : in  WS_TYPE;
   ERROR_INDICATOR     : out ERROR_NUMBER;
   LIST_OF_METHODS     : out COLOUR_MAPPING_METHODS.LIST_OF;
   NUMBER_OF_INDICES   : out PHIGS_POSITIVE);
```

INQUIRE COLOUR MAPPING METHOD FACILITIES

```
procedure INQ_COLOUR_MAPPING_METHOD_FACILITIES
  (TYPE_OF_WS          : in  WS_TYPE;
   METHOD              : in  COLOUR_MAPPING_METHOD;
   ERROR_INDICATOR     : out ERROR_NUMBER;
   FACILITIES          : out COLOUR_MAPPING_METHOD_FACILITIES);
```

INQUIRE PREDEFINED COLOUR MAPPING REPRESENTATION

```
procedure INQ_PREDEFINED_COLOUR_MAPPING_REPRESENTATION
  (TYPE_OF_WS          : in  WS_TYPE;
   COLOUR_MAPPING_IND  : in  COLOUR_MAPPING_INDEX;
   ERROR_INDICATOR     : out ERROR_NUMBER;
   COLOUR_MAPPING      : out ACCESS_COLOUR_MAPPING_DATA_RECORD);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

INQUIRE WORKSTATION STATE TABLE LENGTHS PLUS

```
procedure INQ_WS_STATE_TABLE_LENGTHS
  (TYPE_OF_WS                          : in  WS_TYPE;
   ERROR_INDICATOR                     : out ERROR_NUMBER;
   MAX_DATA_MAPPING_ENTRIES            : out PHIGS_NATURAL;
   MAX_REFLECTANCE_ENTRIES             : out PHIGS_NATURAL;
   MAX_PARAMETRIC_SURFACE_ENTRIES      : out PHIGS_NATURAL;
   MAX_LIGHT_SOURCE_ENTRIES            : out PHIGS_NATURAL;
   MAX_DEPTH_CUE_ENTRIES               : out PHIGS_NATURAL;
   MAX_COLOUR_MAPPING_ENTRIES          : out PHIGS_NATURAL);
```

7.4 Additional functions

7.4.1 Changes to PHIGS generic coordinate system package

The following type declaration should replace the declaration for type MAGNITUDE in the PHIGS generic coordinate system package to support PHIGS PLUS.

```
subtype MAGNITUDE is MAGNITUDE_BASE_TYPE
   range 0.0 .. COORDINATE_COMPONENT_TYPE'SAFE_LARGE;
```

7.4.1 Additions to PHIGS generic coordinate system package

The following additional data types should be added to the PHIGS generic coordinate system package to support PHIGS PLUS:

```
type POINT_4 is
  record
    X : COORDINATE_COMPONENT_TYPE;
    Y : COORDINATE_COMPONENT_TYPE;
    Z : COORDINATE_COMPONENT_TYPE;
    W : COORDINATE_COMPONENT_TYPE;
  end record;

type POINT_LIST_4 is array (POSITIVE range <>) of POINT_4;

type ACCESS_POINT_LIST_4 is access POINT_LIST_4;

type LIST_OF_POINT_LIST_4 is
  array (POSITIVE range <>) of ACCESS_POINT_LIST_4;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
type ACCESS_LIST_OF_POINT_LIST_4 is access LIST_OF_POINT_LIST_4;

type POINT_ARRAY_4 is
  array (POSITIVE range <>, POSITIVE range <>) of POINT_4;

type ACCESS_POINT_ARRAY_4 is access POINT_ARRAY_4;

type POINT_ARRAY_3 is
  array (POSITIVE range <>, POSITIVE range <>) of POINT_3;

type ACCESS_POINT_ARRAY_3 is access POINT_ARRAY_3;

type POINT_ARRAY_2 is
  array (POSITIVE range <>, POSITIVE range <>) of POINT_2;

type ACCESS_POINT_ARRAY_2 is access POINT_ARRAY_2;

type CONTROL_POINT_LIST_3
        (TYPE_OF_RATIONALITY : SPLINE_RATIONALITY;
         LENGTH              : VERTEX_SET_DIMENSION) is
  record
    case TYPE_OF_RATIONALITY is

      when RATIONAL =>
        RATIONAL_POINTS : POINT_LIST_4(1..LENGTH);

      when NON_RATIONAL =>
        NON_RATIONAL_POINTS : POINT_LIST_3(1..LENGTH);

    end case;
  end record;

type ACCESS_CONTROL_POINT_LIST_3 is access CONTROL_POINT_LIST_3;

type CONTROL_POINT_LIST_2
        (TYPE_OF_RATIONALITY : SPLINE_RATIONALITY;
         LENGTH              : VERTEX_SET_DIMENSION) is
  record
    case TYPE_OF_RATIONALITY is

      when RATIONAL =>
        RATIONAL_POINTS : POINT_LIST_3(1..LENGTH);

      when NON_RATIONAL =>
        NON_RATIONAL_POINTS : POINT_LIST_2(1..LENGTH);

    end case;
  end record;

type ACCESS_CONTROL_POINT_LIST_2 is access CONTROL_POINT_LIST_2;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
type CONTROL_POINT_ARRAY_3
        (TYPE_OF_RATIONALITY : SPLINE_RATIONALITY;
         LENGTH_U            : VERTEX_SET_DIMENSION;
         LENGTH_V            : VERTEX_SET_DIMENSION) is
  record
    case TYPE_OF_RATIONALITY is

      when RATIONAL =>
        RATIONAL_POINTS : POINT_ARRAY_4(1..LENGTH_U, 1..LENGTH_V);

      when NON_RATIONAL =>
        NON_RATIONAL_POINTS : POINT_ARRAY_3(1..LENGTH_U, 1..LENGTH_V);

    end case;
  end record;

type ACCESS_CONTROL_POINT_ARRAY_3 is access CONTROL_POINT_ARRAY_3;

type VECTOR_ARRAY_3 is
  array (POSITIVE range <>, POSITIVE range <>) of VECTOR_3;

type ACCESS_VECTOR_ARRAY_3 is access VECTOR_ARRAY_3;

type VECTOR_LIST_3 is
  array (POSITIVE range <>) of VECTOR_3;

type ACCESS_VECTOR_LIST_3 is access VECTOR_LIST_3;

subtype RELATIVE_MAGNITUDE is MAGNITUDE range 0.0 .. 1.0;

type MAGNITUDE_LIST_ARRAY is
        array (SMALL_NATURAL range <>) of MAGNITUDE;

type MAGNITUDE_LIST
        (LENGTH : SMALL_NATURAL := 1) is
  record
    MAGNITUDES : MAGNITUDE_LIST_ARRAY(1..LENGTH);
  end record;

type ACCESS_MAGNITUDE_LIST is access MAGNITUDE_LIST;

type ARRAY_OF_MAGNITUDE_LISTS is
  array ( POSITIVE range <>,
        POSITIVE range <>) of ACCESS_MAGNITUDE_LIST;

type ACCESS_ARRAY_OF_MAGNITUDE_LISTS is
  access ARRAY_OF_MAGNITUDE_LISTS;

type CURVE_GEOMETRYSPLINE_3
       (RATIONALITY : SPLINE_RATIONALITY;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567291 639 ■


```
        KNOT_COUNT : SMALL_NATURAL;
        LENGTH     : VERTEX_SET_DIMENSION) is
  record
    ORDER          : SPLINE_ORDER;
    KNOTS          : MAGNITUDE_LIST(KNOT_COUNT);
    CONTROL_POINTS : CONTROL_POINT_LIST_3(RATIONALITY, LENGTH);
  end record;

type ACCESS_CURVE_GEOMETRYSPLINE_3 is
  access CURVE_GEOMETRYSPLINE_3;

type CURVE_GEOMETRYSPLINE_2
      (RATIONALITY : SPLINE_RATIONALITY;
       KNOT_COUNT : SMALL_NATURAL;
       LENGTH     : VERTEX_SET_DIMENSION) is
  record
    ORDER          : SPLINE_ORDER;
    KNOTS          : MAGNITUDE_LIST(KNOT_COUNT);
    CONTROL_POINTS : CONTROL_POINT_LIST_2(RATIONALITY, LENGTH);
  end record;

type ACCESS_CURVE_GEOMETRYSPLINE_2 is
  access CURVE_GEOMETRYSPLINE_2;

type SURFACE_GEOMETRYSPLINE
      (RATIONALITY   : SPLINE_RATIONALITY;
       KNOT_COUNT_U  : SMALL_NATURAL;
       KNOT_COUNT_V  : SMALL_NATURAL;
       LENGTH_U      : VERTEX_SET_DIMENSION;
       LENGTH_V      : VERTEX_SET_DIMENSION) is
  record
    U_ORDER        : SPLINE_ORDER;
    V_ORDER        : SPLINE_ORDER;
    U_KNOTS        : MAGNITUDE_LIST(KNOT_COUNT_U);
    V_KNOTS        : MAGNITUDE_LIST(KNOT_COUNT_V);
    CONTROL_POINTS : CONTROL_POINT_ARRAY_3( RATIONALITY,
                                            LENGTH_U,
                                            LENGTH_V);
  end record;

type ACCESS_SURFACE_GEOMETRYSPLINE is
  access SURFACE_GEOMETRYSPLINE;

type FACET_COLOUR_ITEM (FACET_DATA_PROVIDED : FACET_DATA_FLAG;
                        MODEL               : COLOUR_MODEL) is
  record
    case FACET_DATA_PROVIDED is
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
        when FACET_COLOUR |
             FACET_COLOUR_NORMAL |
             FACET_COLOUR_DATA |
             FACET_COLOUR_NORMAL_DATA =>
          COLOUR : GENERAL_COLOUR(MODEL);
        when others =>
          null;
      end case;
    end record;

  type FACET_NORMAL_ITEM (FACET_DATA_PROVIDED : FACET_DATA_FLAG) is
    record
      case FACET_DATA_PROVIDED is
        when FACET_NORMAL |
             FACET_COLOUR_NORMAL |
             FACET_NORMAL_DATA |
             FACET_COLOUR_NORMAL_DATA =>
          NORMAL : VECTOR_3;
        when others =>
          null;
      end case;
    end record;

  type FACET_DATA_ITEM
         (FACET_DATA_PROVIDED : FACET_DATA_FLAG;
          PER_FACET           : DATA_VALUE_INDEX) is
    record
      case FACET_DATA_PROVIDED is
        when FACET_DATA |
             FACET_COLOUR_DATA |
             FACET_NORMAL_DATA |
             FACET_COLOUR_NORMAL_DATA =>
          DATA_ITEMS : DATA_VALUE_SET(1..PER_FACET);
        when others =>
          null;
      end case;
    end record;

  type FACET_DATA_SET
    (FACET_DATA_PROVIDED  : FACET_DATA_FLAG;
     MODEL                : COLOUR_MODEL;
     PER_FACET            : DATA_VALUE_INDEX) is
      record
        COLOUR      : FACET_COLOUR_ITEM (FACET_DATA_PROVIDED, MODEL);
        NORMAL      : FACET_NORMAL_ITEM (FACET_DATA_PROVIDED);
        DATA_ITEMS  : FACET_DATA_ITEM (FACET_DATA_PROVIDED, PER_FACET);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
    end record;


type ACCESS_FACET_DATA_SET is access FACET_DATA_SET;

type FACET_COLOUR_ARRAY
      (FACET_DATA_PROVIDED : FACET_DATA_FLAG;
       WIDTH               : FACET_SET_DIMENSION;
       HEIGHT              : FACET_SET_DIMENSION;
       MODEL               : COLOUR_MODEL) is
  record
    case FACET_DATA_PROVIDED is
      when FACET_COLOUR |
           FACET_COLOUR_NORMAL |
           FACET_COLOUR_DATA |
           FACET_COLOUR_NORMAL_DATA =>
        COLOURS : COLOUR_VALUE_ARRAY( MODEL,
                                      WIDTH,
                                      HEIGHT);
      when others =>
        null;
    end case;
  end record;

type FACET_NORMAL_ARRAY
      (FACET_DATA_PROVIDED : FACET_DATA_FLAG;
       WIDTH               : FACET_SET_DIMENSION;
       HEIGHT              : FACET_SET_DIMENSION) is
  record
    case FACET_DATA_PROVIDED is
      when FACET_NORMAL |
           FACET_COLOUR_NORMAL |
           FACET_NORMAL_DATA |
           FACET_COLOUR_NORMAL_DATA =>
        NORMALS : VECTOR_ARRAY_3( 1..WIDTH,
                                  1..HEIGHT);
      when others =>
        null;
    end case;
  end record;

type FACET_DATA_ARRAY
      (FACET_DATA_PROVIDED : FACET_DATA_FLAG;
       WIDTH               : FACET_SET_DIMENSION;
       HEIGHT              : FACET_SET_DIMENSION;
       PER_FACET           : DATA_VALUE_INDEX) is
  record
    case FACET_DATA_PROVIDED is
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567294 348 ■


```
        when FACET_DATA |
             FACET_COLOUR_DATA |
             FACET_NORMAL_DATA |
             FACET_COLOUR_NORMAL_DATA =>
                  DATA_ITEMS : ARRAY_OF_DATA_VALUE_SET(1..WIDTH,
                                                       1..HEIGHT,
                                                       1..PER_FACET);
        when others =>
          null;
      end case;
    end record;

  type FACET_DATA_ARRAY_SET
        (FACET_DATA_PROVIDED : FACET_DATA_FLAG;
         WIDTH               : FACET_SET_DIMENSION;
         HEIGHT              : FACET_SET_DIMENSION;
         MODEL               : COLOUR_MODEL;
         PER_FACET           : DATA_VALUE_INDEX) is
      record
        COLOURS     : FACET_COLOUR_ARRAY( FACET_DATA_PROVIDED,
                                          WIDTH,
                                          HEIGHT,
                                          MODEL);
        NORMALS     : FACET_NORMAL_ARRAY( FACET_DATA_PROVIDED,
                                          WIDTH,
                                          HEIGHT);
        DATA_ITEMS : FACET_DATA_ARRAY( FACET_DATA_PROVIDED,
                                       WIDTH,
                                       HEIGHT,
                                       PER_FACET);
      end record;

  type ACCESS_FACET_DATA_ARRAY_SET is access FACET_DATA_ARRAY_SET;

  type FACET_COLOUR_LIST
        (FACET_DATA_PROVIDED : FACET_DATA_FLAG;
         LENGTH              : FACET_SET_DIMENSION;
         MODEL               : COLOUR_MODEL) is
    record
      case FACET_DATA_PROVIDED is
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567295 284 ■


```
        when FACET_COLOUR |
             FACET_COLOUR_NORMAL |
             FACET_COLOUR_DATA |
             FACET_COLOUR_NORMAL_DATA =>
          COLOURS : COLOUR_VALUE_LIST( MODEL,
                                       LENGTH);
        when others =>
          null;
      end case;
    end record;

  type FACET_NORMAL_LIST
        (FACET_DATA_PROVIDED : FACET_DATA_FLAG;
         LENGTH              : FACET_SET_DIMENSION) is
    record
      case FACET_DATA_PROVIDED is
        when FACET_NORMAL |
             FACET_COLOUR_NORMAL |
             FACET_NORMAL_DATA |
             FACET_COLOUR_NORMAL_DATA =>
          NORMALS : VECTOR_LIST_3(1..LENGTH);
        when others =>
          null;
      end case;
    end record;

  type FACET_DATA_LIST
        (FACET_DATA_PROVIDED : FACET_DATA_FLAG;
         LENGTH              : FACET_SET_DIMENSION;
         PER_FACET           : DATA_VALUE_INDEX) is
    record
      case FACET_DATA_PROVIDED is
        when FACET_DATA |
             FACET_COLOUR_DATA |
             FACET_NORMAL_DATA |
             FACET_COLOUR_NORMAL_DATA =>
          DATA_ITEMS : LIST_OF_DATA_VALUE_SET( 1..LENGTH,
                                               1..PER_FACET);
        when others =>
          null;
      end case;
    end record;

  type FACET_DATA_LIST_SET
        (FACET_DATA_PROVIDED : FACET_DATA_FLAG;
         LENGTH              : FACET_SET_DIMENSION;
         MODEL               : COLOUR_MODEL;
         PER_FACET           : DATA_VALUE_INDEX) is
      record
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567296 110 ■


```
        COLOURS     : FACET_COLOUR_LIST( FACET_DATA_PROVIDED,
                                         LENGTH,
                                         MODEL);
        NORMALS     : FACET_NORMAL_LIST( FACET_DATA_PROVIDED,
                                         LENGTH);
        DATA_ITEMS : FACET_DATA_LIST ( FACET_DATA_PROVIDED,
                                       LENGTH,
                                       PER_FACET);
      end record;

type ACCESS_FACET_DATA_LIST_SET is access FACET_DATA_LIST_SET;

type VERTEX_COLOUR_ARRAY
      (VERTEX_DATA_PROVIDED : VERTEX_DATA_FLAG;
       WIDTH                : VERTEX_SET_DIMENSION;
       HEIGHT               : VERTEX_SET_DIMENSION;
       MODEL                : COLOUR_MODEL) is
  record
    case VERTEX_DATA_PROVIDED is
      when COORDINATES_COLOUR |
           COORDINATES_COLOUR_NORMAL |
           COORDINATES_COLOUR_DATA |
           COORDINATES_COLOUR_NORMAL_DATA =>
        COLOURS : COLOUR_VALUE_ARRAY( MODEL,
                                      WIDTH,
                                      HEIGHT);
      when others =>
        null;
    end case;
  end record;

type VERTEX_NORMAL_ARRAY
      (VERTEX_DATA_PROVIDED : VERTEX_DATA_FLAG;
       WIDTH                : VERTEX_SET_DIMENSION;
       HEIGHT               : VERTEX_SET_DIMENSION) is
  record
    case VERTEX_DATA_PROVIDED is
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567297 057 ■


```
        when COORDINATES_NORMAL |
             COORDINATES_COLOUR_NORMAL |
             COORDINATES_NORMAL_DATA |
             COORDINATES_COLOUR_NORMAL_DATA =>
          NORMALS : VECTOR_ARRAY_3( 1..WIDTH,
                                    1..HEIGHT);
        when others =>
          null;
      end case;
    end record;

  type VERTEX_DATA_ARRAY
       (VERTEX_DATA_PROVIDED : VERTEX_DATA_FLAG;
        WIDTH                : VERTEX_SET_DIMENSION;
        HEIGHT               : VERTEX_SET_DIMENSION;
        PER_VERTEX           : DATA_VALUE_INDEX) is
    record
      case VERTEX_DATA_PROVIDED is
        when COORDINATES_DATA |
             COORDINATES_COLOUR_DATA |
             COORDINATES_NORMAL_DATA |
             COORDINATES_COLOUR_NORMAL_DATA =>
          DATA_ITEMS : ARRAY_OF_DATA_VALUE_SET( 1..WIDTH,
                                                1..HEIGHT,
                                                1..PER_VERTEX);
        when others =>
          null;
      end case;
    end record;

  type VERTEX_DATA_ARRAY_SET_3
       (VERTEX_DATA_PROVIDED : VERTEX_DATA_FLAG;
        WIDTH                : VERTEX_SET_DIMENSION;
        HEIGHT               : VERTEX_SET_DIMENSION;
        MODEL                : COLOUR_MODEL;
        PER_VERTEX           : DATA_VALUE_INDEX) is
    record
      POINTS      : POINT_ARRAY_3( 1..WIDTH,
                                   1..HEIGHT);

      COLOURS     : VERTEX_COLOUR_ARRAY( VERTEX_DATA_PROVIDED,
                                         WIDTH,
                                         HEIGHT,
                                         MODEL);
      NORMALS     : VERTEX_NORMAL_ARRAY( VERTEX_DATA_PROVIDED,
                                         WIDTH,
                                         HEIGHT);
      DATA_ITEMS : VERTEX_DATA_ARRAY( VERTEX_DATA_PROVIDED,
                                      WIDTH,
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567298 T93 ■


```
                                                HEIGHT,
                                                PER_VERTEX);
    end record;

type ACCESS_VERTEX_DATA_ARRAY_SET_3 is access VERTEX_DATA_ARRAY_SET_3;

type VERTEX_DATA_ARRAY_SET_2
      (VERTEX_DATA_PROVIDED : VERTEX_DATA_FLAG;
       WIDTH                : VERTEX_SET_DIMENSION;
       HEIGHT               : VERTEX_SET_DIMENSION;
       MODEL                : COLOUR_MODEL;
       PER_VERTEX           : DATA_VALUE_INDEX) is
   record
     POINTS       : POINT_ARRAY_2( 1..WIDTH,
                                   1..HEIGHT);

     COLOURS      : VERTEX_COLOUR_ARRAY( VERTEX_DATA_PROVIDED,
                                         WIDTH,
                                         HEIGHT,
                                         MODEL);
     NORMALS      : VERTEX_NORMAL_ARRAY( VERTEX_DATA_PROVIDED,
                                         WIDTH,
                                         HEIGHT);
     DATA_ITEMS : VERTEX_DATA_ARRAY( VERTEX_DATA_PROVIDED,
                                     WIDTH,
                                     HEIGHT,
                                     PER_VERTEX);
   end record;

type ACCESS_VERTEX_DATA_ARRAY_SET_2 is access VERTEX_DATA_ARRAY_SET_2;

type VERTEX_COLOUR_LIST
      (VERTEX_DATA_PROVIDED : VERTEX_DATA_FLAG;
       LENGTH               : VERTEX_SET_DIMENSION;
       MODEL                : COLOUR_MODEL) is
  record
    case VERTEX_DATA_PROVIDED is
      when COORDINATES_COLOUR |
           COORDINATES_COLOUR_NORMAL |
           COORDINATES_COLOUR_DATA |
           COORDINATES_COLOUR_NORMAL_DATA =>
        COLOURS : COLOUR_VALUE_LIST( MODEL,
                                     LENGTH);
      when others =>
        null;
    end case;
  end record;

type VERTEX_NORMAL_LIST
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567299 92T ■


```
      (VERTEX_DATA_PROVIDED : VERTEX_DATA_FLAG;
       LENGTH               : VERTEX_SET_DIMENSION) is
  record
    case VERTEX_DATA_PROVIDED is
      when COORDINATES_NORMAL |
           COORDINATES_COLOUR_NORMAL |
           COORDINATES_NORMAL_DATA |
           COORDINATES_COLOUR_NORMAL_DATA =>
        NORMALS : VECTOR_LIST_3(1..LENGTH);
      when others =>
        null;
    end case;
  end record;

type VERTEX_DATA_LIST
      (VERTEX_DATA_PROVIDED : VERTEX_DATA_FLAG;
       LENGTH               : VERTEX_SET_DIMENSION;
       PER_VERTEX           : DATA_VALUE_INDEX) is
  record
    case VERTEX_DATA_PROVIDED is
      when COORDINATES_DATA |
           COORDINATES_COLOUR_DATA |
           COORDINATES_NORMAL_DATA |
           COORDINATES_COLOUR_NORMAL_DATA =>
        DATA_ITEMS : LIST_OF_DATA_VALUE_SET( 1..LENGTH,
                                             1..PER_VERTEX);
      when others =>
        null;
    end case;
  end record;

type VERTEX_DATA_LIST_SET_3
       (VERTEX_DATA_PROVIDED : VERTEX_DATA_FLAG;
        LENGTH               : VERTEX_SET_DIMENSION;
        MODEL                : COLOUR_MODEL;
        PER_VERTEX           : DATA_VALUE_INDEX) is
    record
      POINTS      : POINT_LIST_3(1..LENGTH);

      COLOURS     : VERTEX_COLOUR_LIST( VERTEX_DATA_PROVIDED,
                                        LENGTH,
                                        MODEL);
      NORMALS     : VERTEX_NORMAL_LIST( VERTEX_DATA_PROVIDED,
                                        LENGTH);
      DATA_ITEMS : VERTEX_DATA_LIST( VERTEX_DATA_PROVIDED,
                                     LENGTH,
                                     PER_VERTEX);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
    end record;

type ACCESS_VERTEX_DATA_LIST_SET_3 is access VERTEX_DATA_LIST_SET_3;

type LIST_OF_VERTEX_DATA_LIST_SET_3 is array (VERTEX_SET_COUNT) of
        ACCESS_VERTEX_DATA_LIST_SET_3;

type ACCESS_LIST_OF_VERTEX_DATA_LIST_SET_3 is
        access LIST_OF_VERTEX_DATA_LIST_SET_3;

type VERTEX_DATA_LIST_SET_2
       (VERTEX_DATA_PROVIDED : VERTEX_DATA_FLAG;
        LENGTH               : VERTEX_SET_DIMENSION;
        MODEL                : COLOUR_MODEL;
        PER_VERTEX           : DATA_VALUE_INDEX) is
    record
      POINTS      : POINT_LIST_2(1..LENGTH);
      COLOURS     : VERTEX_COLOUR_LIST( VERTEX_DATA_PROVIDED,
                                        LENGTH,
                                        MODEL);
      NORMALS     : VERTEX_NORMAL_LIST( VERTEX_DATA_PROVIDED,
                                        LENGTH);
      DATA_ITEMS : VERTEX_DATA_LIST( VERTEX_DATA_PROVIDED,
                                     LENGTH,
                                     PER_VERTEX);
    end record;

type ACCESS_VERTEX_DATA_LIST_SET_2 is access VERTEX_DATA_LIST_SET_2;

type LIST_OF_VERTEX_DATA_LIST_SET_2 is array (VERTEX_SET_COUNT) of
        ACCESS_VERTEX_DATA_LIST_SET_2;

type ACCESS_LIST_OF_VERTEX_DATA_LIST_SET_2 is
        access LIST_OF_VERTEX_DATA_LIST_SET_2;

type VERTEX_COLOUR_LIST_SET_3
       (VERTEX_COLOUR_PROVIDED  : VERTEX_COLOUR_FLAG;
        LENGTH                  : VERTEX_SET_DIMENSION;
        MODEL                   : COLOUR_MODEL) is
    record
      POINTS      : POINT_LIST_3(1..LENGTH);
      COLOURS     : VERTEX_COLOUR_LIST( VERTEX_COLOUR_PROVIDED,
                                        LENGTH,
                                        MODEL);
    end record;

type ACCESS_VERTEX_COLOUR_LIST_SET_3 is access VERTEX_COLOUR_LIST_SET_3;

type LIST_OF_VERTEX_COLOUR_LIST_SET_3 is array (VERTEX_SET_COUNT) of
        ACCESS_VERTEX_COLOUR_LIST_SET_3;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
type ACCESS_LIST_OF_VERTEX_COLOUR_LIST_SET_3 is
        access LIST_OF_VERTEX_COLOUR_LIST_SET_3;

procedure DEALLOCATE (OBJECT: in out LIST_OF_VERTEX_DATA_LIST_SET_3);

procedure DEALLOCATE (OBJECT: in out LIST_OF_VERTEX_DATA_LIST_SET_2);

procedure DEALLOCATE (OBJECT: in out LIST_OF_VERTEX_COLOUR_LIST_SET_3);
```

7.4.2 PHIGS PLUS generic colour package

The generic package PHIGS_COLOUR_TYPES is instantiated in the PHIGS_TYPES package once for each colour model defined in PHIGS and once for a generic undefined colour model. It may also be instantiated once for each additional colour model supported by the implementation of this binding. The package defines the representation of the following data types which are to be replicated for each supported colour model:

COLOUR_VALUE_LIST
ACCESS_COLOUR_VALUE_LIST
HOMOGENEOUS_COLOUR_VALUE_LIST
ACCESS_HOMOGENEOUS_COLOUR_VALUE_LIST
LIST_OF_COLOUR_VALUE_LIST
ACCESS_LIST_OF_COLOUR_VALUE_LIST
COLOUR_VALUE_ARRAY
ACCESS_COLOUR_VALUE_ARRAY
HOMOGENEOUS_COLOUR_VALUE_ARRAY
ACCESS_HOMOGENEOUS_COLOUR_VALUE_ARRAY
CONTROL_POINT_LIST
ACCESS_CONTROL_POINT_LIST
CONTROL_POINT_ARRAY
ACCESS_CONTROL_POINT_ARRAY

The declaration for this generic package is shown below:

```
generic

    type COLOUR_VALUE is private;

    type HOMOGENEOUS_COLOUR_VALUE is private;

package PHIGS_COLOUR_TYPES is
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
type COLOUR_VALUE_LIST is
  array (COLOUR_VALUE_SET_DIMENSION range <>)
    of COLOUR_VALUE;

type ACCESS_COLOUR_VALUE_LIST is access COLOUR_VALUE_LIST;

type HOMOGENEOUS_COLOUR_VALUE_LIST is
  array (COLOUR_VALUE_SET_DIMENSION range <>)
    of HOMOGENEOUS_COLOUR_VALUE;

type ACCESS_HOMOGENEOUS_COLOUR_VALUE_LIST is
  access HOMOGENEOUS_COLOUR_VALUE_LIST;

type LIST_OF_COLOUR_VALUE_LIST is
  array (COLOUR_VALUE_SET_DIMENSION range <>)
    of ACCESS_COLOUR_VALUE_LIST;

type ACCESS_LIST_OF_COLOUR_VALUE_LIST is
  access LIST_OF_COLOUR_VALUE_LIST;

type COLOUR_VALUE_ARRAY is
  array ( COLOUR_VALUE_SET_DIMENSION range <>,
          COLOUR_VALUE_SET_DIMENSION range <> )
    of COLOUR_VALUE;

type ACCESS_COLOUR_VALUE_ARRAY is access COLOUR_VALUE_ARRAY;

type HOMOGENEOUS_COLOUR_VALUE_ARRAY is
  array ( COLOUR_VALUE_SET_DIMENSION range <>,
          COLOUR_VALUE_SET_DIMENSION range <> )
    of HOMOGENEOUS_COLOUR_VALUE;

type ACCESS_HOMOGENEOUS_COLOUR_VALUE_ARRAY is
  access HOMOGENEOUS_COLOUR_VALUE_ARRAY;

type CONTROL_POINT_LIST
       (TYPE_OF_RATIONALITY : SPLINE_RATIONALITY;
        LENGTH              : COLOUR_VALUE_SET_DIMENSION) is
  record
    case TYPE_OF_RATIONALITY is

      when RATIONAL =>
        RATIONAL_POINTS : HOMOGENEOUS_COLOUR_VALUE_LIST(1..LENGTH);

      when NON_RATIONAL =>
        NON_RATIONAL_POINTS : COLOUR_VALUE_LIST(1..LENGTH);

    end case;
  end record;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567303 180 ■


```
    type ACCESS_CONTROL_POINT_LIST is access CONTROL_POINT_LIST;

    type CONTROL_POINT_ARRAY
          (TYPE_OF_RATIONALITY : SPLINE_RATIONALITY;
           LENGTH_U            : COLOUR_VALUE_SET_DIMENSION;
           LENGTH_V            : COLOUR_VALUE_SET_DIMENSION) is
      record
        case TYPE_OF_RATIONALITY is

          when RATIONAL =>
            RATIONAL_POINTS : HOMOGENEOUS_COLOUR_VALUE_ARRAY
                                (1..LENGTH_U, 1..LENGTH_V);

          when NON_RATIONAL =>
            NON_RATIONAL_POINTS : COLOUR_VALUE_ARRAY
                                    (1..LENGTH_U, 1..LENGTH_V);

        end case;
      end record;

    type ACCESS_CONTROL_POINT_ARRAY is access CONTROL_POINT_ARRAY;

  end PHIGS_COLOUR_TYPES;
```

7.4.3 Deallocation of PHIGS PLUS structure element records

The following additional DEALLOCATE functions are defined:

DEALLOCATE

```
    procedure DEALLOCATE (OBJECT: in out COLOUR_MAPPING_DATA_RECORD);
```

DEALLOCATE

```
    procedure DEALLOCATE (OBJECT: in out COLOUR_VALUE_LIST);
```

DEALLOCATE

```
    procedure DEALLOCATE (OBJECT: in out CURVE_COLOURSPLINE);
```

DEALLOCATE

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

procedure DEALLOCATE (OBJECT: in out DATA_MAPPING_DATA_RECORD);

DEALLOCATE

procedure DEALLOCATE (OBJECT: in out DATASPLINE_LIST);

DEALLOCATE

procedure DEALLOCATE (OBJECT: in out DATA_VALUE_SET);

DEALLOCATE

procedure DEALLOCATE (OBJECT: in out LIST_OF_COLOUR_VALUE_LIST);

DEALLOCATE

procedure DEALLOCATE (OBJECT: in out LIST_OF_EDGE_FLAG_LIST);

DEALLOCATE

procedure DEALLOCATE (OBJECT: in out LIST_OF_LIST_OF_EDGE_FLAG_LIST);

DEALLOCATE

procedure DEALLOCATE (OBJECT: in out LIST_OF_LIST_OF_VERTEX_INDEX_LIST);

DEALLOCATE

procedure DEALLOCATE (OBJECT: in out LIST_OF_TRIMCURVE_LIST);

DEALLOCATE

procedure DEALLOCATE (OBJECT: in out LIST_OF_VERTEX_INDEX_LIST);

DEALLOCATE

procedure DEALLOCATE (OBJECT: in out SURFACE_COLOURSPLINE);

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

DEALLOCATE

```
procedure DEALLOCATE (OBJECT: in out TRIMCURVE_LIST);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

9593 PT*3 90 ■ 4851903 0567306 99T ■

Annex A

(informative)

Compilable PHIGS Specification

The following should replace the existing content:

This annex is intended to show one example of a compilable PHIGS PLUS binding to Ada. It is not the only possible such example. It is expected that implementations of a PHIGS PLUS binding to Ada will need to add additional constructs to the various packages or rearrange the constructs in a manner sutiable to the implementation. However, all names defined in this binding shall be visible as dictated in this standard.

```
-- PHIGS Configuration Package

    package PHIGS_CONFIGURATION is

-- The following define example values for the PHIGS configuration names.
-- Note that the values specified are implementation dependent. The values
-- shown below are arbitrary and have been chosen solely as examples. More
-- appropriate values should be chosen by an implementation. Note also that
-- only the technique of defining the names as constants was used in this
-- example. An implementation is free to choose the most appropriate means
-- of providing the value as discussed in 4.2.5.

  DEFAULT_ERROR_FILE                          : constant STRING := "ERROR.FIL";
  DEFAULT_LIST_SIZE                           : constant := 256;
  DEFAULT_MEMORY_UNITS                        : constant := 0;
  MAX_ANNOTATION_STYLES_SUPPORTED             : constant := 2;
  MAX_APPLICATION_DATA                        : constant := 256;
  MAX_ARCHIVE_IDS_SUPPORTED                   : constant := 4;
  MAX_CHAR_SETS_SUPPORTED                     : constant := 1;
  MAX_CHOICE_PROMPTS_SUPPORTED                : constant := 32;
  MAX_CHOICE_PROMPT_ECHO_TYPES_SUPPORTED      : constant := 10;
  MAX_CHOICE_SMALL_NATURAL                    : constant := 64;
  MAX_COLOUR_COEFFICIENTS                     : constant := 4;
  MAX_COLOUR_INDICES_SUPPORTED                : constant := 4096;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567307 826 ■


```
MAX_COLOUR_MATRIX_SMALL_NATURAL                 : constant := 128;
MAX_COLOUR_MODELS_SUPPORTED                     : constant := 2;
MAX_EDGE_INDICES_SUPPORTED                      : constant := 5;
MAX_EDGETYPES_SUPPORTED                         : constant := 1;
MAX_ESCAPE_IDS_SUPPORTED                        : constant := 16;
MAX_FILE_IDS_SUPPORTED                          : constant := 10;
MAX_FILE_SMALL_NATURAL                          : constant := 80;
MAX_GDP_3_IDS_SUPPORTED                         : constant := 24;
MAX_GDP_IDS_SUPPORTED                           : constant := 24;
MAX_GSE_IDS_SUPPORTED                           : constant := 8;
MAX_HATCH_STYLES_SUPPORTED                      : constant := 64;
MAX_HLHSR_IDS_SUPPORTED                         : constant := 128;
MAX_HLHSR_MODES_SUPPORTED                       : constant := 4;
MAX_INPUT_STRING_SMALL_NATURAL                  : constant := 80;
MAX_INTERIOR_INDICES_SUPPORTED                  : constant := 5;
MAX_LINETYPES_SUPPORTED                         : constant := 5;
MAX_LOCATOR_PROMPT_ECHO_TYPES_SUPPORTED         : constant := 8;
MAX_MARKER_TYPES_SUPPORTED                      : constant := 256;
MAX_MEMORY_UNITS                                : constant := 0;
MAX_METAFILE_ITEM_LENGTH                        : constant := 1024;
MAX_METAFILE_ITEM_TYPE                          : constant := 256;
MAX_MODELLING_CLIP_OPERATION_TYPES_SUPPORTED    : constant := 2;
MAX_MODELLING_CLIP_DISTINCT_PLANES_SUPPORTED    : constant := 6;
MAX_NAME_SUPPORTED                              : constant := 63;
MAX_NAME_SET_FILTER_LIST_LENGTH_SUPPORTED       : constant := 4;
MAX_OPEN_WS_SUPPORTED                           : constant := 4;
MAX_PATH_DEPTH_SUPPORTED                        : constant := 64;
MAX_PATTERN_INDICES_SUPPORTED                   : constant := 4;
MAX_PICK_IDS_SUPPORTED                          : constant := 64536;
MAX_PICK_PROMPT_ECHO_TYPES_SUPPORTED            : constant := 2;
MAX_POLYLINE_INDICES_SUPPORTED                  : constant := 5;
MAX_POLYMARKER_INDICES_SUPPORTED                : constant := 5;
MAX_POSTED_STRUCTURES_SUPPORTED                 : constant := 4096;
MAX_REFERENCE_PATHS_SUPPORTED                   : constant := 64;
MAX_SMALL_NATURAL                               : constant := 255;
MAX_STRING_PROMPT_ECHO_TYPES_SUPPORTED          : constant := 2;
MAX_STRING_SMALL_NATURAL                        : constant := 255;
MAX_STROKE_PROMPT_ECHO_TYPES_SUPPORTED          : constant := 2;
MAX_STRUCTURE_IDS_SUPPORTED                     : constant := 65535;
MAX_TEXT_FONT_PRECISION_PAIRS_SUPPORTED         : constant := 32;
MAX_TEXT_INDICES_SUPPORTED                      : constant := 5;
MAX_VALUATOR_PROMPT_ECHO_TYPES_SUPPORTED        : constant := 4;
MAX_VIEW_INDICES_SUPPORTED                      : constant := 20;
MAX_WS_IDS_SUPPORTED                            : constant := 16383;
MAX_WS_TYPES_SUPPORTED                          : constant := 64;
MIN_METAFILE_ITEM_TYPE                          : constant := 0;
MIN_INTEGER_VALUE                               : constant := -2147483648;
MAX_INTEGER_VALUE                               : constant := 2147483647;
PHIGS_PRECISION                                 : constant := 7;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567308 762 ■


```
  PHIGS_PI                                                    : constant := 3.1415926;

  MAX_COLOUR_MAPPING_INDICES_SUPPORTED                        : constant := 6;
  MAX_COLOUR_MAPPING_METHODS_SUPPORTED                        : constant := 3;
  MAX_COLOUR_VALUES_SUPPORTED                                 : constant := 4096;
  MAX_CURVE_APPROX_CRITERIA_TYPES_SUPPORTED                   : constant := 4;
  MAX_DATA_MAPPING_INDICES_SUPPORTED                          : constant := 20;
  MAX_DATA_MAPPING_METHODS_SUPPORTED                          : constant := 2;
  MAX_DATA_VALUES_PER_FACET                                   : constant := 10;
  MAX_DATA_VALUES_PER_VERTEX                                  : constant := 10;
  MAX_DEPTH_CUE_INDICES_SUPPORTED                             : constant := 20;
  MAX_FACETS_SUPPORTED                                        : constant := 2048;
  MAX_INTERIOR_SHADING_METHODS_SUPPORTED                      : constant := 5;
  MAX_LIGHT_SOURCE_INDICES_SUPPORTED                          : constant := 8;
  MAX_LIGHT_SOURCE_TYPES_SUPPORTED                            : constant := 4;
  MAX_PARAMETRIC_SURFACE_INDICES_SUPPORTED                    : constant := 6;
  MAX_PARAMETRIC_SURFACE_CHARACTERISTICS_SUPPORTED : constant := 2;
  MAX_POLYLINE_SHADING_METHODS_SUPPORTED                      : constant := 2;
  MAX_REFLECTANCE_MODELS_SUPPORTED                            : constant := 6;
  MAX_REFLECTANCE_INDICES_SUPPORTED                           : constant := 20;
  MAX_REFLECTANCE_PROPERTIES_TYPES_SUPPORTED                  : constant := 2;
  MAX_SURFACE_APPROX_CRITERIA_TYPES_SUPPORTED                 : constant := 4;
  MAX_VERTICES_SUPPORTED                                      : constant := 4096;

end PHIGS_CONFIGURATION;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
-- PHIGS List Utilities Generic Package

    with PHIGS_CONFIGURATION;

    generic

        type ELEMENT_TYPE is private;

        MAX_LIST_SIZE : NATURAL := PHIGS_CONFIGURATION.DEFAULT_LIST_SIZE;

    package PHIGS_LIST_UTILITIES is

        subtype LIST_SIZE is NATURAL range 0..MAX_LIST_SIZE;

        type LIST_OF (SIZE : LIST_SIZE := 0) is private;

        type LIST_VALUES is array (POSITIVE range <>) of ELEMENT_TYPE;

        function NULL_LIST return LIST_OF;

          -- This function returns an empty LIST_OF object.  This list is
          -- intended primarily for use by implementors.

        function SIZE_OF_LIST (LIST : in LIST_OF) return NATURAL;

          -- This function returns the number of element type values stored
          -- in the list object.

        function IS_IN_LIST ( ELEMENT : in ELEMENT_TYPE;
                              LIST    : in LIST_OF) return BOOLEAN;

        -- This function returns the value TRUE if the element parameter
        -- value is in the list object.  Otherwise it returns the value
        -- FALSE.

        function LIST_ELEMENT ( INDEX : in POSITIVE;
                                LIST  : in  LIST_OF) return ELEMENT_TYPE;

        -- This function returns the element value in the list object that
        -- has an associated index value equal to the index parameter.  The
        -- PHIGS_ERROR 2502 is generated if the index parameter exceeds the
        -- current size of the list parameter object.

        function LIST (VALUES : in LIST_VALUES) return LIST_OF;

        -- This function returns a valid LIST_OF object.  If the VALUES
        -- parameter is a null array, an empty LIST_OF object is returned.
        -- If the values parameter is not null, this function returns a
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567310 310 ■


```
-- LIST_OF object containing all the values in the VALUES parameter.
-- The PHIGS_ERROR 2502 is generated if the number of element values
-- exceeds the specified maximum size of the LIST_OF object.

procedure ADD_TO_LIST ( ELEMENT : in ELEMENT_TYPE;
                        LIST    : in out LIST_OF);

-- This procedure stores the element parameter value in the list
-- parameter object, and increases the size of the list object
-- by one. An index value equal to the incremented list size
-- is associated with the stored element value. The ADD_TO_LIST
-- procedure will generate PHIGS_ERROR 2502 if it is called when
-- the list parameter has a size equal to the specified maximum
-- size. If desired, the user can ensure duplicate values are not
-- stored. This is accomplised by calling ADD_TO_LIST with a
-- particular element value only if the function IS_IN_LIST returns
-- false for that element value.


procedure DELETE_FROM_LIST ( ELEMENT : in ELEMENT_TYPE;
                             LIST    : in out LIST_OF);

-- If the list parameter object does not contain the element
-- parameter value, this procedure has no effect. Otherwise, the
-- first occurrence of the element value is deleted. The size of
-- the list object is decreased by one, and the indices associated
-- with the remaining element values are adjusted so that the indices
-- begin at one and increment in steps of one. If desired, the user
-- can delete all occurrences of an element value. This is accomplished
-- by calling DELETE_FROM_LIST repeatedly with a particular element
-- value while the function IS_IN_LIST returns TRUE for that value.

private

-- The declaration of the LIST_OF type is implementation dependent.
-- However, the operations implicitly declared by the LIST_OF declaration,
-- including both assignment and the predefined comparison for equality
-- and inequality, shall produce the correct results. This requirement
-- precludes the use of access types for the implementation of the
-- LIST_OF type. The recommended implementation is given below:

type LIST_OF (SIZE : LIST_SIZE := 0) is
  record
    ELEMENTS : LIST_VALUES(1..SIZE);
  end record;
--
-- Note that declaring unconstrained LIST_OF objects by using the default
-- discriminant value allows dynamic modification of the size of the
-- element array.
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
end PHIGS_LIST_UTILITIES;

package body PHIGS_LIST_UTILITIES is

-- This package body contains stubs for each of the subprograms
-- contained in the PHIGS_LIST_UTILITIES package.  The purpose of
-- these stubs is to show compilability for the PHIGS/Ada binding.
-- An implementation would replace these stubs with code appropriate
-- to implement the functionality.


   function NULL_LIST return LIST_OF is
     EMPTY_LIST : LIST_OF;
   begin
     return EMPTY_LIST;
   end NULL_LIST;

   function SIZE_OF_LIST (LIST : in LIST_OF) return NATURAL is
   begin
     return 0;
   end SIZE_OF_LIST;

   function IS_IN_LIST ( ELEMENT : in ELEMENT_TYPE;
                         LIST    : in LIST_OF) return BOOLEAN is
   begin
     return FALSE;
   end IS_IN_LIST;

   function LIST_ELEMENT ( INDEX : in POSITIVE;
                           LIST  : in  LIST_OF) return ELEMENT_TYPE is
     EXAMPLE_ELEMENT : ELEMENT_TYPE;
   begin
     return EXAMPLE_ELEMENT;
   end LIST_ELEMENT;

   function LIST (VALUES : in LIST_VALUES) return LIST_OF is
   begin
     return NULL_LIST;
   end LIST;


   procedure ADD_TO_LIST ( ELEMENT : in ELEMENT_TYPE;
                           LIST    : in out LIST_OF) is
   begin
     null;
   end ADD_TO_LIST;

   procedure DELETE_FROM_LIST ( ELEMENT : in ELEMENT_TYPE;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567312 193 ■


```
                                        LIST        : in out LIST_OF) is
    begin
      null;
    end DELETE_FROM_LIST;

end PHIGS_LIST_UTILITIES;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567313 02T ■


```
-- PHIGS Name Set Facility Package

with PHIGS_CONFIGURATION, PHIGS_LIST_UTILITIES;

package PHIGS_NAME_SET_FACILITY is

   -- PHIGS_CONFIGURATION.MAX_NAME_SUPPORTED

   -- An implementation dependent constant which defines the maximum
   -- name supported. The minimum value allowed by the PHIGS
   -- specification is 63 thus supporting 64 classes. A larger
   -- value is strongly encouraged.

   type NAME_SET is private;

   -- Defines a name set data element.

   -- NAME_SET_ACCEPTANCE

   type NAME_SET_ACCEPTANCE is (REJECTED, ACCEPTED);

   -- Used to indicate the results of applying a name set
   -- against a filter pair.

   -- NAME

   type NAME is range 0..PHIGS_CONFIGURATION.MAX_NAME_SUPPORTED;

   -- Provides for names for each of the name set classes.

   -- NAMES

   package NAMES is new PHIGS_LIST_UTILITIES (NAME);

   -- Provides for lists of names.

   -- NAME_SET_FILTER

   type NAME_SET_FILTER is
     record
       INCLUSION  : NAME_SET;
       EXCLUSION  : NAME_SET;
     end record;

   -- Used to define name set pairs used as filters.

   -- NAME_SET_MEMBERSHIP
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567314 T66 ■


```
type NAME_SET_MEMBERSHIP is (NON_MEMBER, MEMBER);

-- Provides for indicating whether a name is a member of a
-- name set.

-- Subprograms for Manipulating Name Sets

procedure BUILD_NAME_SET
  (MEMBERS : in   NAMES.LIST_OF;
   SET     : out  NAME_SET);

-- Constructs a name set from a list of names.

procedure BUILD_NAME_SET_UNION
  (LEFT_SET  : in   NAME_SET;
   RIGHT_SET : in   NAME_SET;
   UNION     : out  NAME_SET);

-- Constructs the logical union of two name sets.

procedure BUILD_NAME_SET_INTERSECTION
  (LEFT_SET     : in   NAME_SET;
   RIGHT_SET    : in   NAME_SET;
   INTERSECTION : out  NAME_SET);

-- Constructs the logical intersection of two name sets.

procedure BUILD_NAME_SET_DIFFERENCE
  (LEFT_SET   : in   NAME_SET;
   RIGHT_SET  : in   NAME_SET;
   DIFFERENCE : out  NAME_SET);

-- Constructs the logical difference of two name sets.

procedure COPY_NAME_SET
  (ORIGINAL : in   NAME_SET;
   COPY     : out  NAME_SET);

-- Duplicates the provided name set.

function IS_EQUAL
  (LEFT_SET  : in NAME_SET;
   RIGHT_SET : in NAME_SET)
  return BOOLEAN;

-- Compares two name sets.
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567315 9T2 ■


```
    procedure EXTRACT_NAMES
      (SET         : in   NAME_SET;
       MEMBERS     : out  NAMES.LIST_OF);

    -- Extracts the names from the provided name set.

    procedure ADD_NAMES
      (MEMBERS     : in      NAMES.LIST_OF;
       SET         : in out  NAME_SET);

    -- Adds names to the name set if not already there.

    procedure REMOVE_NAMES
      (MEMBERS     : in      NAMES.LIST_OF;
       SET         : in out  NAME_SET);

    -- Removes names from the name set if they are there.

    function APPLY_FILTER
      (SET      : in NAME_SET;
       FILTER   : in NAME_SET_FILTER)
      return NAME_SET_ACCEPTANCE;

    -- Applies a filter to a name set and indicates whether the name
    -- set passes through the filter.

    function INQUIRE_MEMBERSHIP
      (MEMBER   : in NAME;
       SET      : in NAME_SET)
      return NAME_SET_MEMBERSHIP;

    -- Returns an indication of whether the indicated name is a member
    -- of the provided name set.

    procedure DEALLOCATE
      (OBJECT   : in out NAME_SET);

    -- Deallocates name set objects.

    -- PRIVATE TYPE DEFINITIONS

private

-- The following types define the specifications for private
-- types used in the PHIGS_NAME_SET_FACILITY.  Null records are
-- used soley for the purposes of demonstrating compilability.
-- An implementation would use some more appropriate declaration.
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567316 839 ■


```
        type NAME_SET is
          record
            null;
          end record;

    end PHIGS_NAME_SET_FACILITY;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567317 775 ■


```
-- Base types package

with PHIGS_CONFIGURATION;

  package PHIGS_BASE_TYPES is

    -- This package is used to encapsulate the base derived types of PHIGS
    -- since they are used as the parent of several other derived types.
    -- In Ada, if the parent of a derived type is itself a derived type,
    -- then this parent type cannot be declared immediately in the visible
    -- part of the same package.

    use PHIGS_CONFIGURATION;

    -- PHIGS_INTEGER

    type PHIGS_INTEGER is range MIN_INTEGER_VALUE .. MAX_INTEGER_VALUE;

    --  Base type for PHIGS integer types.

    -- PHIGS_NATURAL


    subtype PHIGS_NATURAL is
      PHIGS_INTEGER range 0..PHIGS_INTEGER'last;

    --  Base type for PHIGS natural types.

    -- PHIGS POSITIVE

    subtype PHIGS_POSITIVE is
      PHIGS_INTEGER range 1..PHIGS_INTEGER'last;

    --  Base type for PHIGS positive types.

    -- SCALE_FACTOR

    type SCALE_FACTOR is digits PHIGS_PRECISION;

    -- The type used for unitless scaling factors.

    -- PHIGS_STRING

    type PHIGS_STRING is
      array (PHIGS_INTEGER range <>) of CHARACTER;

    -- Base type for PHIGS string types.
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
-- SMALL_NATURAL

subtype SMALL_NATURAL is
  PHIGS_NATURAL range 0..MAX_SMALL_NATURAL;

-- This is a subtype declaration which allows for unconstrained record
-- objects for various record types without causing the exception
-- STORAGE_ERROR to be raised.

-- COLOUR_VALUE_SET_DIMENSION

subtype COLOUR_VALUE_SET_DIMENSION is
          NATURAL range 0..MAX_COLOUR_VALUES_SUPPORTED;

-- Provides for specifying the dimensions of colour value lists and arrays.

-- FACET_SET_DIMENSION

subtype FACET_SET_DIMENSION is
          NATURAL range 0..MAX_FACETS_SUPPORTED;

-- Provides for specifying the dimensions of facet lists and arrays.

-- VERTEX_SET_COUNT

subtype VERTEX_SET_COUNT is
          NATURAL range 0..MAX_VERTEX_SETS_SUPPORTED;

-- Provides for specifying the number of sets in a vertex set.

-- VERTEX_SET_DIMENSION

subtype VERTEX_SET_DIMENSION is
          NATURAL range 0..MAX_VERTICES_SUPPORTED;

-- Provides for specifying the dimensions of vertex lists and arrays.

-- SPLINE_ORDER

type SPLINE_ORDER is new POSITIVE;

-- Provides for specifying orders for spline curves and surfaces.

-- SPLINE_RATIONALITY

type SPLINE_RATIONALITY is (RATIONAL, NON_RATIONAL);

-- Provides for specifying the rationality of splines.
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567319 548 ■


```
end PHIGS_BASE_TYPES;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567320 26T ■


```
    -- The PHIGS_COLOUR_TYPES Package

  generic

    type COLOUR_VALUE is private;

    type HOMOGENEOUS_COLOUR_VALUE is private;

  package PHIGS_COLOUR_TYPES is


    type COLOUR_VALUE_LIST is
      array (COLOUR_VALUE_SET_DIMENSION range <>)
        of COLOUR_VALUE;

    type ACCESS_COLOUR_VALUE_LIST is access COLOUR_VALUE_LIST;

    type HOMOGENEOUS_COLOUR_VALUE_LIST is
      array (COLOUR_VALUE_SET_DIMENSION range <>)
        of HOMOGENEOUS_COLOUR_VALUE;

    type ACCESS_HOMOGENEOUS_COLOUR_VALUE_LIST is
      access HOMOGENEOUS_COLOUR_VALUE_LIST;

    type LIST_OF_COLOUR_VALUE_LIST is
      array (COLOUR_VALUE_SET_DIMENSION range <>)
        of ACCESS_COLOUR_VALUE_LIST;

    type ACCESS_LIST_OF_COLOUR_VALUE_LIST is
      access LIST_OF_COLOUR_VALUE_LIST;

    type COLOUR_VALUE_ARRAY is
      array ( COLOUR_VALUE_SET_DIMENSION range <>,
              COLOUR_VALUE_SET_DIMENSION range <> )
        of COLOUR_VALUE;

    type ACCESS_COLOUR_VALUE_ARRAY is access COLOUR_VALUE_ARRAY;

    type HOMOGENEOUS_COLOUR_VALUE_ARRAY is
      array ( COLOUR_VALUE_SET_DIMENSION range <>,
              COLOUR_VALUE_SET_DIMENSION range <> )
        of HOMOGENEOUS_COLOUR_VALUE;

    type ACCESS_HOMOGENEOUS_COLOUR_VALUE_ARRAY is
      access HOMOGENEOUS_COLOUR_VALUE_ARRAY;

    type CONTROL_POINT_LIST
          (TYPE_OF_RATIONALITY : SPLINE_RATIONALITY;
           LENGTH              : COLOUR_VALUE_SET_DIMENSION) is
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567321 1T6 ■


```
        record
          case TYPE_OF_RATIONALITY is

            when RATIONAL =>
              RATIONAL_POINTS : HOMOGENEOUS_COLOUR_VALUE_LIST(1..LENGTH);

            when NON_RATIONAL =>
              NON_RATIONAL_POINTS : COLOUR_VALUE_LIST(1..LENGTH);

          end case;
        end record;

      type ACCESS_CONTROL_POINT_LIST is access CONTROL_POINT_LIST;

      type CONTROL_POINT_ARRAY
            (TYPE_OF_RATIONALITY : SPLINE_RATIONALITY;
             LENGTH_U            : COLOUR_VALUE_SET_DIMENSION;
             LENGTH_V            : COLOUR_VALUE_SET_DIMENSION) is
        record
          case TYPE_OF_RATIONALITY is

            when RATIONAL =>
              RATIONAL_POINTS : HOMOGENEOUS_COLOUR_VALUE_ARRAY
                                  (1..LENGTH_U, 1..LENGTH_V);

            when NON_RATIONAL =>
              NON_RATIONAL_POINTS : COLOUR_VALUE_ARRAY
                                      (1..LENGTH_U, 1..LENGTH_V);

          end case;
        end record;

      type ACCESS_CONTROL_POINT_ARRAY is access CONTROL_POINT_ARRAY;

    end PHIGS_COLOUR_TYPES;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
-- The PHIGS_STANDARD_TYPES Package

with PHIGS_CONFIGURATION, PHIGS_BASE_TYPES;


package PHIGS_STANDARD_TYPES is


-- This package contains all the standard data type definitions required by
-- the PHIGS coordinate system generic package.

-- Configuration names

use PHIGS_CONFIGURATION, PHIGS_BASE_TYPES;

    -- Make PHIGS_BASE_TYPES visible externally.

    subtype PHIGS_INTEGER is PHIGS_BASE_TYPES.PHIGS_INTEGER;
    subtype PHIGS_NATURAL is PHIGS_BASE_TYPES.PHIGS_NATURAL;
    subtype PHIGS_POSITIVE is PHIGS_BASE_TYPES.PHIGS_POSITIVE;
    subtype PHIGS_STRING is PHIGS_BASE_TYPES.PHIGS_STRING;
    subtype SCALE_FACTOR is PHIGS_BASE_TYPES.SCALE_FACTOR;
    subtype SMALL_NATURAL is BASE_TYPE.SMALL_NATURAL;
    subtype COLOUR_VALUE_SET_DIMENSION is
            PHIGS_BASE_TYPES.COLOUR_VALUE_SET_DIMENSION;
    subtype FACET_SET_DIMENSION is PHIGS_BASE_TYPES.FACET_SET_DIMENSION;
    subtype VERTEX_SET_COUNT is PHIGS_BASE_TYPES.VERTEX_SET_COUNT;
    subtype VERTEX_SET_DIMENSION is
PHIGS_BASE_TYPES.VERTEX_SET_DIMENSION;
    subtype SPLINE_ORDER is PHIGS_BASE_TYPES.SPLINE_ORDER;
    subtype SPLINE_RATIONALITY is PHIGS_BASE_TYPES.SPLINE_RATIONALITY;

    -- DATA_VALUE

    type DATA_VALUE is digits PHIGS_PRECISION;

    -- Provides for data mapping input values.

    -- DATA_VALUE_INDEX

    type DATA_VALUE_INDEX is
        new PHIGS_POSITIVE range 1 .. MAX_DATA_VALUES_PER_VERTEX;

    -- Provides for selecting from lists of data mapping input values.

    -- DATA_VALUE_SET

    type DATA_VALUE_SET is
      array (DATA_VALUE_INDEX range <>) of DATA_VALUE;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567323 T79 ■


```
-- Provides for arrays (ordered lists) of data mapping input values.


-- ACCESS_DATA_VALUE_SET

type ACCESS_DATA_VALUE_SET is access DATA_VALUE_SET;

-- Provides for pointers to sets of data mapping data values.
-- DATA_MAPPING_INDEX

type DATA_MAPPING_INDEX is new PHIGS_NATURAL;

-- Provides for index values for data mapping representations.
-- LIST_OF_DATA_VALUE_SET

type LIST_OF_DATA_VALUE_SET is
        array (VERTEX_SET_DIMENSION range <>,
                DATA_VALUE_INDEX range <>) of DATA_VALUE;

-- Provides for lists of data mapping data value sets. These will be used
-- various output primitives. The first index refers to the number of data value lists
-- and the second index to the individual data values.

-- ARRAY_OF_DATA_VALUE_SET

type ARRAY_OF_DATA_VALUE_SET is
        array (VERTEX_SET_DIMENSION range <>,
                VERTEX_SET_DIMENSION range <>,
                DATA_VALUE_INDEX range <>) of DATA_VALUE;

-- Provides for two-dimensional arrays of data mapping data value sets. These
-- will be used with "Quadrilateral Mesh with Data" primitives. The first index
-- refers to the width of the array, the second index to the height, and the third
-- index to the individual data values.
-- COLOUR_COMPONENTS

type COLOUR_COMPONENTS is range 1..MAX_COLOUR_COEFFICIENTS;

-- Defines a type for the size of colour component arrays.

-- COLOUR_INDEX

type COLOUR_INDEX is new PHIGS_NATURAL;

-- Indices into colour tables are of this type.

-- COLOUR_MODEL
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
type COLOUR_MODEL is new PHIGS_INTEGER;

-- Indicates the colour models available in PHIGS.

-- The following constants define the colour models specified by PHIGS:

INDIRECT   : constant COLOUR_MODEL := 0;
RGB        : constant COLOUR_MODEL := 1;
CIELUV     : constant COLOUR_MODEL := 2;
HSV        : constant COLOUR_MODEL := 3;
HLS        : constant COLOUR_MODEL := 4;

-- COLOUR_COEFFICIENT

type COLOUR_COEFFICIENT is digits PHIGS_PRECISION;

-- Defines a general type for colour components.

-- COLOUR_COEFFICIENT_ARRAY

type COLOUR_COEFFICIENT_ARRAY is
  array (COLOUR_COMPONENTS) of COLOUR_COEFFICIENT;

-- Defines an array type for colour components.

-- INTENSITY

subtype INTENSITY is COLOUR_COEFFICIENT range 0.0..1.0;

-- Defines the restricted range of colour components required
-- by some colour models.

-- CIELUV COLOUR VALUE

type CIELUV_COLOUR_VALUE is
  record
    L_STAR_CIE : COLOUR_COEFFICIENT;
    U_STAR_CIE : COLOUR_COEFFICIENT;
    V_STAR_CIE : COLOUR_COEFFICIENT;
  end record;

-- Provides for specification of CIELUV colours.

-- CIELUV HOMOGENEOUS COLOUR VALUE
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
type CIELUV_HOMOGENEOUS_COLOUR_VALUE is
  record
    L_STAR_CIE : COLOUR_COEFFICIENT;
    U_STAR_CIE : COLOUR_COEFFICIENT;
    V_STAR_CIE : COLOUR_COEFFICIENT;
    W_CIE      : COLOUR_COEFFICIENT;
  end record;

-- Provides for specification of homogeneous CIELUV colours for use in
-- rational coloursplines.

-- CIELUV COLOUR PACKAGE

package CIELUV_TYPES is
  new PHIGS_COLOUR_TYPES( CIELUV_COLOUR_VALUE,
                          CIELUV_HOMOGENEOUS_COLOUR_VALUE);

-- Provides for CIELUV colour types.

-- GENERIC COLOUR PACKAGE

package GENERIC_COLOUR_TYPES is
  new PHIGS_COLOUR_TYPES( COLOUR_COEFFICIENT_ARRAY,
                          COLOUR_COEFFICIENT_ARRAY);

-- Provides for generic (unsupported) colour types.

-- HLS COLOUR VALUE

type HLS_COLOUR_VALUE is
  record
    HUE_HLS        : INTENSITY;
    LIGHTNESS_HLS  : INTENSITY;
    SATURATION_HLS : INTENSITY;
  end record;

-- Provides for specification of HLS colours.

-- HLS HOMOGENEOUS COLOUR VALUE
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
type HLS_HOMOGENEOUS_COLOUR_VALUE is
  record
    HUE_HLS          : COLOUR_COEFFICIENT;
    LIGHTNESS_HLS    : COLOUR_COEFFICIENT;
    SATURATION_HLS   : COLOUR_COEFFICIENT;
    W_HLS            : COLOUR_COEFFICIENT;
  end record;

-- Provides for specification of homogeneous HLS colours for use in
-- rational coloursplines.

-- HLS COLOUR PACKAGE

package HLS_TYPES is
  new PHIGS_COLOUR_TYPES( HLS_COLOUR_VALUE,
                          HLS_HOMOGENEOUS_COLOUR_VALUE);

-- Provides for HLS colour types.

-- HSV_COLOUR_VALUE

type HSV_COLOUR_VALUE is
  record
    HUE_HSV          : INTENSITY;
    SATURATION_HSV   : INTENSITY;
    VALUE_HSV        : INTENSITY;
  end record;

-- Provides for specification of HSV colours.

-- HSV HOMOGENEOUS COLOUR VALUE

type HSV_HOMOGENEOUS_COLOUR_VALUE is
  record
    HUE_HSV          : COLOUR_COEFFICIENT;
    SATURATION_HSV   : COLOUR_COEFFICIENT;
    VALUE_HSV        : COLOUR_COEFFICIENT;
    W_HSV            : COLOUR_COEFFICIENT;
  end record;

-- Provides for specification of homogeneous HSV colours for use in
-- rational coloursplines.

-- HSV COLOUR PACKAGE

package HSV_TYPES is
  new PHIGS_COLOUR_TYPES( HSV_COLOUR_VALUE,
                          HSV_HOMOGENEOUS_COLOUR_VALUE);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567327 614 ■


```
-- Provides for HSV colour types.

-- INDIRECT HOMOGENEOUS COLOUR VALUE

type INDIRECT_HOMOGENEOUS_COLOUR_VALUE is
  record
    INDEX_INDIRECT : COLOUR_INDEX;
    W_INDIRECT     : COLOUR_COEFFICIENT;
  end record;

-- Provides for specification of homogeneous indirect colours for use in rational
-- coloursplines.

-- INDIRECT COLOUR PACKAGE

package INDIRECT_TYPES is
  new PHIGS_COLOUR_TYPES( COLOUR_INDEX,
                          INDIRECT_HOMOGENEOUS_COLOUR_VALUE);

-- Provides for INDIRECT colour types.

-- RGB_COLOUR_VALUE

type RGB_COLOUR_VALUE is
  record
    RED_RGB   : INTENSITY;
    GREEN_RGB : INTENSITY;
    BLUE_RGB  : INTENSITY;
  end record;

-- Provides for specification of RGB colours.

-- RGB HOMOGENEOUS COLOUR VALUE

type RGB_HOMOGENEOUS_COLOUR_VALUE is
  record
    RED_RGB   : COLOUR_COEFFICIENT;
    GREEN_RGB : COLOUR_COEFFICIENT;
    BLUE_RGB  : COLOUR_COEFFICIENT;
    W_RGB     : COLOUR_COEFFICIENT;
  end record;

-- Provides for specification of homogeneous RGB colours for use in rational coloursplines.

-- RGB COLOUR PACKAGE
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567328 550 ■


```
package RGB_TYPES is
  new PHIGS_COLOUR_TYPES( RGB_COLOUR_VALUE,
                          RGB_HOMOGENEOUS_COLOUR_VALUE);

-- Provides for RGB colour types.

-- GENERAL_COLOUR

type GENERAL_COLOUR
  (MODEL : COLOUR_MODEL := RGB) is
    record
      case MODEL is

        when INDIRECT =>
          INDEX              : COLOUR_INDEX;

        when RGB =>
          COLOUR_RGB         : RGB_COLOUR_VALUE;

        when CIELUV =>
          COLOUR_CIELUV      : CIELUV_COLOUR_VALUE;

        when HSV =>
          COLOUR_HSV         : HSV_COLOUR_VALUE;

        when HLS =>
          COLOUR_HLS         : HLS_COLOUR_VALUE;

        when others =>
          COLOUR_GENERIC     : COLOUR_COEFFICIENT_ARRAY;

      end case;
    end record;

-- Provides for specifying values for colours dependent upon colour models.  Additional
-- variants may be included when additional registered or unregistered colour models are
-- supported by an implementation.

-- COLOUR_VALUE_ARRAY

type COLOUR_VALUE_ARRAY
  (MODEL    : COLOUR_MODEL := RGB;
   LENGTH_M : COLOUR_VALUE_SET_DIMENSION := 1;
   LENGTH_N : COLOUR_VALUE_SET_DIMENSION := 1)  is
    record
      case MODEL is
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
            when INDIRECT =>
              COLOURS_INDIRECT : INDIRECT_TYPES.COLOUR_VALUE_ARRAY
                                    (1..LENGTH_M, 1..LENGTH_N);

            when RGB =>
              COLOURS_RGB : RGB_TYPES.COLOUR_VALUE_ARRAY
                                (1..LENGTH_M, 1..LENGTH_N);

            when CIELUV =>
              COLOURS_CIELUV : CIELUV_TYPES.COLOUR_VALUE_ARRAY
                                    (1..LENGTH_M, 1..LENGTH_N);

            when HSV =>
              COLOURS_HSV  : HSV_TYPES.COLOUR_VALUE_ARRAY
                                (1..LENGTH_M, 1..LENGTH_N);

            when HLS =>
              COLOURS_HLS : HLS_TYPES.COLOUR_VALUE_ARRAY
                                (1..LENGTH_M, 1..LENGTH_N);

            when others =>
              COLOURS_GENERIC : GENERIC_COLOUR_TYPES.COLOUR_VALUE_ARRAY
                                    (1..LENGTH_M, 1..LENGTH_N);

          end case;
        end record;

-- Provides for specification of two-dimensional arrays of colour values all of which
-- are the same colour model.
-- COLOUR_VALUE_LIST

type COLOUR_VALUE_LIST
  (MODEL   : COLOUR_MODEL := RGB;
   LENGTH  : COLOUR_VALUE_SET_DIMENSION := 0) is
    record
      case MODEL is

        when INDIRECT =>
          COLOURS_INDIRECT : INDIRECT_TYPES.COLOUR_VALUE_LIST(1..LENGTH);

        when RGB =>
          COLOURS_RGB : RGB_TYPES.COLOUR_VALUE_LIST(1..LENGTH);

        when CIELUV =>
          COLOURS_CIELUV  : CIELUV_TYPES.COLOUR_VALUE_LIST(1..LENGTH);

        when HSV =>
          COLOURS_HSV : HSV_TYPES.COLOUR_VALUE_LIST(1..LENGTH);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
            when HLS =>
              COLOURS_HLS : HLS_TYPES.COLOUR_VALUE_LIST(1..LENGTH);

            when others =>
              COLOURS_GENERIC : GENERIC_COLOUR_TYPES.COLOUR_VALUE_LIST
                                        (1..LENGTH);

          end case;
        end record;

-- Provides for specification of lists of colour values all of which are the same colour model.
-- LIST_OF_COLOUR_VALUE_LIST

type LIST_OF_COLOUR_VALUE_LIST
        (MODEL    : COLOUR_MODEL := RGB;
         LENGTH   : COLOUR_VALUE_SET_DIMENSION := 3) is
      record
        case MODEL is

          when INDIRECT =>
            COLOURS_INDIRECT
                  : INDIRECT_TYPES.LIST_OF_COLOUR_VALUE_LIST(1..LENGTH);

          when RGB =>
            COLOURS_RGB
                  : RGB_TYPES.LIST_OF_COLOUR_VALUE_LIST(1..LENGTH);

          when CIELUV =>
            COLOURS_CIELUV
                  : CIELUV_TYPES.LIST_OF_COLOUR_VALUE_LIST(1..LENGTH);

          when HSV =>
            COLOURS_HSV
                  : HSV_TYPES.LIST_OF_COLOUR_VALUE_LIST(1..LENGTH);

          when HLS =>
            COLOURS_HLS
                  : HLS_TYPES.LIST_OF_COLOUR_VALUE_LIST(1..LENGTH);

          when others =>
              COLOURS_GENERIC
                  : GENERIC_COLOUR_TYPES.LIST_OF_COLOUR_VALUE_LIST
                                          (1..LENGTH);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567331 045 ■


```
    end case;
  end record;

  -- Provides for specification of lists of lists of colour values all of which are the same
  -- colour model.
  -- FACET_DATA_FLAG

  type FACET_DATA_FLAG is ( FACET_NONE,
                            FACET_COLOUR,
                            FACET_NORMAL,
                            FACET_DATA,
                            FACET_COLOUR_NORMAL,
                            FACET_COLOUR_DATA,
                            FACET_NORMAL_DATA,
                            FACET_COLOUR_NORMAL_DATA);

  -- Provides for various types of facet data flags.

  -- VERTEX_DATA_FLAG

  type VERTEX_DATA_FLAG is ( COORDINATES,
                             COORDINATES_COLOUR,
                             COORDINATES_NORMAL,
                             COORDINATES_DATA,
                             COORDINATES_COLOUR_NORMAL,
                             COORDINATES_COLOUR_DATA,
                             COORDINATES_NORMAL_DATA,
                             COORDINATES_COLOUR_NORMAL_DATA);

  -- Provides identifiers for various types of vertex information.

  -- VERTEX_COLOUR_FLAG

  subtype VERTEX_COLOUR_FLAG is
          VERTEX_DATA_FLAG range COORDINATES .. COORDINATES_COLOUR;

  -- Provides identifiers for various types of vertex information used in polyline sets.

end PHIGS_STANDARD_TYPES;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
-- The PHIGS_COORDINATE_SYSTEM Generic Package

with  PHIGS_CONFIGURATION,
      PHIGS_BASE_TYPES,
      PHIGS_STANDARD_TYPES;

generic

   type COORDINATE_COMPONENT_TYPE is digits <>;

package PHIGS_COORDINATE_SYSTEM is

   use PHIGS_CONFIGURATION, PHIGS_BASE_TYPES, PHIGS_STANDARD_TYPES;

   type POINT_4 is
     record
       X : COORDINATE_COMPONENT_TYPE;
       Y : COORDINATE_COMPONENT_TYPE;
       Z : COORDINATE_COMPONENT_TYPE;
       W : COORDINATE_COMPONENT_TYPE;
     end record;

   type POINT_3 is
     record
       X : COORDINATE_COMPONENT_TYPE;
       Y : COORDINATE_COMPONENT_TYPE;
       Z : COORDINATE_COMPONENT_TYPE;
     end record;

   type POINT_2 is
     record
       X : COORDINATE_COMPONENT_TYPE;
       Y : COORDINATE_COMPONENT_TYPE;
     end record;

   type POINT_LIST_4 is array (POSITIVE range <>) of POINT_4;

   type POINT_LIST_3 is array (POSITIVE range <>) of POINT_3;

   type POINT_LIST_2 is array (POSITIVE range <>) of POINT_2;

   type ACCESS_POINT_LIST_4 is access POINT_LIST_4;

   type ACCESS_POINT_LIST_3 is access POINT_LIST_3;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567333 918 ■


```
type ACCESS_POINT_LIST_2 is access POINT_LIST_2;

type LIST_OF_POINT_LIST_4 is
  array (POSITIVE range <>) of ACCESS_POINT_LIST_4;

type LIST_OF_POINT_LIST_3 is array (POSITIVE range <>) of
  ACCESS_POINT_LIST_3;

type LIST_OF_POINT_LIST_2 is array (POSITIVE range <>) of
  ACCESS_POINT_LIST_2;

type ACCESS_LIST_OF_POINT_LIST_4 is access LIST_OF_POINT_LIST_4;

type ACCESS_LIST_OF_POINT_LIST_3 is access LIST_OF_POINT_LIST_3;

type ACCESS_LIST_OF_POINT_LIST_2 is access LIST_OF_POINT_LIST_2;

type POINT_ARRAY_4 is
  array (POSITIVE range <>, POSITIVE range <>) of POINT_4;

type ACCESS_POINT_ARRAY_4 is access POINT_ARRAY_4;

type POINT_ARRAY_3 is
  array (POSITIVE range <>, POSITIVE range <>) of POINT_3;

type ACCESS_POINT_ARRAY_3 is access POINT_ARRAY_3;

type POINT_ARRAY_2 is
  array (POSITIVE range <>, POSITIVE range <>) of POINT_2;

type ACCESS_POINT_ARRAY_2 is access POINT_ARRAY_2;

type CONTROL_POINT_LIST_3
        (TYPE_OF_RATIONALITY : SPLINE_RATIONALITY;
         LENGTH              : VERTEX_SET_DIMENSION) is
  record
    case TYPE_OF_RATIONALITY is

      when RATIONAL =>
        RATIONAL_POINTS : POINT_LIST_4(1..LENGTH);

      when NON_RATIONAL =>
        NON_RATIONAL_POINTS : POINT_LIST_3(1..LENGTH);

    end case;
  end record;

type ACCESS_CONTROL_POINT_LIST_3 is access CONTROL_POINT_LIST_3;

type CONTROL_POINT_LIST_2
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
        (TYPE_OF_RATIONALITY : SPLINE_RATIONALITY;
         LENGTH              : VERTEX_SET_DIMENSION) is
    record
      case TYPE_OF_RATIONALITY is

        when RATIONAL =>
          RATIONAL_POINTS : POINT_LIST_3(1..LENGTH);

        when NON_RATIONAL =>
          NON_RATIONAL_POINTS : POINT_LIST_2(1..LENGTH);

      end case;
    end record;

  type ACCESS_CONTROL_POINT_LIST_2 is access CONTROL_POINT_LIST_2;

  type CONTROL_POINT_ARRAY_3
        (TYPE_OF_RATIONALITY : SPLINE_RATIONALITY;
         LENGTH_U            : VERTEX_SET_DIMENSION;
         LENGTH_V            : VERTEX_SET_DIMENSION) is
    record
      case TYPE_OF_RATIONALITY is

        when RATIONAL =>
          RATIONAL_POINTS : POINT_ARRAY_4(1..LENGTH_U, 1..LENGTH_V);

        when NON_RATIONAL =>
          NON_RATIONAL_POINTS : POINT_ARRAY_3(1..LENGTH_U, 1..LENGTH_V);

      end case;
    end record;

  type ACCESS_CONTROL_POINT_ARRAY_3 is access CONTROL_POINT_ARRAY_3;

  type VECTOR_3 is new POINT_3;

  type VECTOR_PAIR_3 is array (1..2) of VECTOR_3;

  type VECTOR_2 is new POINT_2;

  type VECTOR_PAIR_2 is array (1..2) of VECTOR_2;

  type VECTOR_ARRAY_3 is
    array (POSITIVE range <>, POSITIVE range <>) of VECTOR_3;

  type ACCESS_VECTOR_ARRAY_3 is access VECTOR_ARRAY_3;

  type VECTOR_LIST_3 is
    array (POSITIVE range <>) of VECTOR_3;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567335 790 ■


```
type ACCESS_VECTOR_LIST_3 is access VECTOR_LIST_3;

type RECTANGULAR_REGION_3 is
  record
    XMIN : COORDINATE_COMPONENT_TYPE;
    XMAX : COORDINATE_COMPONENT_TYPE;
    YMIN : COORDINATE_COMPONENT_TYPE;
    YMAX : COORDINATE_COMPONENT_TYPE;
    ZMIN : COORDINATE_COMPONENT_TYPE;
    ZMAX : COORDINATE_COMPONENT_TYPE;
  end record;

type RECTANGULAR_REGION_2 is
  record
    XMIN : COORDINATE_COMPONENT_TYPE;
    XMAX : COORDINATE_COMPONENT_TYPE;
    YMIN : COORDINATE_COMPONENT_TYPE;
    YMAX : COORDINATE_COMPONENT_TYPE;
  end record;

type HALF_SPACE_3 is
  record
    REFERENCE_POSITION  : POINT_3;
    ACCEPTANCE_NORMAL : VECTOR_3;
  end record;

type HALF_SPACE_2 is
  record
    REFERENCE_POSITION  : POINT_2;
    ACCEPTANCE_NORMAL : VECTOR_2;
  end record;

type HALF_SPACE_LIST_3 is
  array (POSITIVE range <>) of HALF_SPACE_3;

type HALF_SPACE_LIST_2 is
  array (POSITIVE range <>) of HALF_SPACE_2;

type ACCESS_HALF_SPACE_LIST_3 is access HALF_SPACE_LIST_3;

type ACCESS_HALF_SPACE_LIST_2 is access HALF_SPACE_LIST_2;

type MAGNITUDE_BASE_TYPE is digits PHIGS_PRECISION;

subtype MAGNITUDE is MAGNITUDE_BASE_TYPE
    range 0.0 .. COORDINATE_COMPONENT_TYPE'SAFE_LARGE;

subtype RELATIVE_MAGNITUDE is MAGNITUDE range 0.0 .. 1.0;

type SIZE_3 is
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
  record
    XAXIS : MAGNITUDE;
    YAXIS : MAGNITUDE;
    ZAXIS : MAGNITUDE;
  end record;

type SIZE_2 is
  record
    XAXIS : MAGNITUDE;
    YAXIS : MAGNITUDE;
  end record;

type MAGNITUDE_LIST_ARRAY is
        array (SMALL_NATURAL range <>) of MAGNITUDE;

type MAGNITUDE_LIST
         (LENGTH : SMALL_NATURAL := 1) is
  record
    MAGNITUDES : MAGNITUDE_LIST_ARRAY(1..LENGTH);
  end record;

type ACCESS_MAGNITUDE_LIST is access MAGNITUDE_LIST;

type ARRAY_OF_MAGNITUDE_LISTS is
  array ( POSITIVE range <>,
          POSITIVE range <>) of ACCESS_MAGNITUDE_LIST;

type ACCESS_ARRAY_OF_MAGNITUDE_LISTS is
  access ARRAY_OF_MAGNITUDE_LISTS;

type RANGE_OF_MAGNITUDES is
  record
    MIN  : MAGNITUDE;
    MAX : MAGNITUDE;
  end record;

type CURVE_GEOMETRYSPLINE_3
       (RATIONALITY : SPLINE_RATIONALITY;
        KNOT_COUNT : SMALL_NATURAL;
        LENGTH       : VERTEX_SET_DIMENSION) is
  record
    ORDER              : SPLINE_ORDER;
    KNOTS              : MAGNITUDE_LIST(KNOT_COUNT);
    CONTROL_POINTS : CONTROL_POINT_LIST_3(RATIONALITY, LENGTH);
  end record;

type ACCESS_CURVE_GEOMETRYSPLINE_3 is
  access CURVE_GEOMETRYSPLINE_3;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567337 563 ■


```
type CURVE_GEOMETRYSPLINE_2
      (RATIONALITY : SPLINE_RATIONALITY;
       KNOT_COUNT : SMALL_NATURAL;
       LENGTH     : VERTEX_SET_DIMENSION) is
  record
    ORDER          : SPLINE_ORDER;
    KNOTS          : MAGNITUDE_LIST(KNOT_COUNT);
    CONTROL_POINTS : CONTROL_POINT_LIST_2(RATIONALITY, LENGTH);
  end record;

type ACCESS_CURVE_GEOMETRYSPLINE_2 is
  access CURVE_GEOMETRYSPLINE_2;

type SURFACE_GEOMETRYSPLINE
      (RATIONALITY   : SPLINE_RATIONALITY;
       KNOT_COUNT_U  : SMALL_NATURAL;
       KNOT_COUNT_V  : SMALL_NATURAL;
       LENGTH_U      : VERTEX_SET_DIMENSION;
       LENGTH_V      : VERTEX_SET_DIMENSION) is
  record
    U_ORDER        : SPLINE_ORDER;
    V_ORDER        : SPLINE_ORDER;
    U_KNOTS        : MAGNITUDE_LIST(KNOT_COUNT_U);
    V_KNOTS        : MAGNITUDE_LIST(KNOT_COUNT_V);
    CONTROL_POINTS : CONTROL_POINT_ARRAY_3( RATIONALITY,
                                            LENGTH_U,
                                            LENGTH_V);
  end record;

type ACCESS_SURFACE_GEOMETRYSPLINE is
  access SURFACE_GEOMETRYSPLINE;

type FACET_COLOUR_ITEM (FACET_DATA_PROVIDED : FACET_DATA_FLAG;
                        MODEL               : COLOUR_MODEL) is
  record
    case FACET_DATA_PROVIDED is
      when FACET_COLOUR |
           FACET_COLOUR_NORMAL |
           FACET_COLOUR_DATA |
           FACET_COLOUR_NORMAL_DATA =>
        COLOUR : GENERAL_COLOUR(MODEL);
      when others =>
        null;
    end case;
  end record;

type FACET_NORMAL_ITEM (FACET_DATA_PROVIDED : FACET_DATA_FLAG) is
  record
    case FACET_DATA_PROVIDED is
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567338 4TT ■


```
        when FACET_NORMAL |
             FACET_COLOUR_NORMAL |
             FACET_NORMAL_DATA |
             FACET_COLOUR_NORMAL_DATA =>
          NORMAL : VECTOR_3;
        when others =>
          null;
      end case;
    end record;

  type FACET_DATA_ITEM
         (FACET_DATA_PROVIDED : FACET_DATA_FLAG;
          PER_FACET           : DATA_VALUE_INDEX) is
    record
      case FACET_DATA_PROVIDED is
        when FACET_DATA |
             FACET_COLOUR_DATA |
             FACET_NORMAL_DATA |
             FACET_COLOUR_NORMAL_DATA =>
          DATA_ITEMS : DATA_VALUE_SET(1..PER_FACET);
        when others =>
          null;
      end case;
    end record;

  type FACET_DATA_SET
    (FACET_DATA_PROVIDED  : FACET_DATA_FLAG;
     MODEL                : COLOUR_MODEL;
     PER_FACET            : DATA_VALUE_INDEX) is
      record
        COLOUR      : FACET_COLOUR_ITEM (FACET_DATA_PROVIDED, MODEL);
        NORMAL      : FACET_NORMAL_ITEM (FACET_DATA_PROVIDED);
        DATA_ITEMS  : FACET_DATA_ITEM (FACET_DATA_PROVIDED, PER_FACET);
      end record;

  type ACCESS_FACET_DATA_SET is access FACET_DATA_SET;

  type FACET_COLOUR_ARRAY
        (FACET_DATA_PROVIDED : FACET_DATA_FLAG;
         WIDTH               : FACET_SET_DIMENSION;
         HEIGHT              : FACET_SET_DIMENSION;
         MODEL               : COLOUR_MODEL) is
    record
      case FACET_DATA_PROVIDED is
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567339 336 ■


```
        when FACET_COLOUR |
             FACET_COLOUR_NORMAL |
             FACET_COLOUR_DATA |
             FACET_COLOUR_NORMAL_DATA =>
          COLOURS : COLOUR_VALUE_ARRAY( MODEL,
                                        WIDTH,
                                        HEIGHT);
        when others =>
          null;
      end case;
    end record;

  type FACET_NORMAL_ARRAY
        (FACET_DATA_PROVIDED : FACET_DATA_FLAG;
         WIDTH               : FACET_SET_DIMENSION;
         HEIGHT              : FACET_SET_DIMENSION) is
    record
      case FACET_DATA_PROVIDED is
        when FACET_NORMAL |
             FACET_COLOUR_NORMAL |
             FACET_NORMAL_DATA |
             FACET_COLOUR_NORMAL_DATA =>
          NORMALS : VECTOR_ARRAY_3( 1..WIDTH,
                                    1..HEIGHT);
        when others =>
          null;
      end case;
    end record;

  type FACET_DATA_ARRAY
        (FACET_DATA_PROVIDED : FACET_DATA_FLAG;
         WIDTH               : FACET_SET_DIMENSION;
         HEIGHT              : FACET_SET_DIMENSION;
         PER_FACET           : DATA_VALUE_INDEX) is
    record
      case FACET_DATA_PROVIDED is
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
        when FACET_DATA |
             FACET_COLOUR_DATA |
             FACET_NORMAL_DATA |
             FACET_COLOUR_NORMAL_DATA =>
                  DATA_ITEMS : ARRAY_OF_DATA_VALUE_SET(1..WIDTH,
                                                       1..HEIGHT,
                                                       1..PER_FACET);
        when others =>
          null;
      end case;
    end record;

  type FACET_DATA_ARRAY_SET
        (FACET_DATA_PROVIDED : FACET_DATA_FLAG;
         WIDTH               : FACET_SET_DIMENSION;
         HEIGHT              : FACET_SET_DIMENSION;
         MODEL               : COLOUR_MODEL;
         PER_FACET           : DATA_VALUE_INDEX) is
      record
        COLOURS     : FACET_COLOUR_ARRAY( FACET_DATA_PROVIDED,
                                          WIDTH,
                                          HEIGHT,
                                          MODEL);
        NORMALS     : FACET_NORMAL_ARRAY( FACET_DATA_PROVIDED,
                                          WIDTH,
                                          HEIGHT);
        DATA_ITEMS : FACET_DATA_ARRAY( FACET_DATA_PROVIDED,
                                       WIDTH,
                                       HEIGHT,
                                       PER_FACET);
      end record;

  type ACCESS_FACET_DATA_ARRAY_SET is access FACET_DATA_ARRAY_SET;

  type FACET_COLOUR_LIST
        (FACET_DATA_PROVIDED : FACET_DATA_FLAG;
         LENGTH              : FACET_SET_DIMENSION;
         MODEL               : COLOUR_MODEL) is
    record
      case FACET_DATA_PROVIDED is
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567341 T94 ■


```
        when FACET_COLOUR |
             FACET_COLOUR_NORMAL |
             FACET_COLOUR_DATA |
             FACET_COLOUR_NORMAL_DATA =>
          COLOURS : COLOUR_VALUE_LIST( MODEL,
                                       LENGTH);
        when others =>
          null;
      end case;
    end record;

  type FACET_NORMAL_LIST
        (FACET_DATA_PROVIDED : FACET_DATA_FLAG;
         LENGTH              : FACET_SET_DIMENSION) is
    record
      case FACET_DATA_PROVIDED is
        when FACET_NORMAL |
             FACET_COLOUR_NORMAL |
             FACET_NORMAL_DATA |
             FACET_COLOUR_NORMAL_DATA =>
          NORMALS : VECTOR_LIST_3(1..LENGTH);
        when others =>
          null;
      end case;
    end record;

  type FACET_DATA_LIST
        (FACET_DATA_PROVIDED : FACET_DATA_FLAG;
         LENGTH              : FACET_SET_DIMENSION;
         PER_FACET           : DATA_VALUE_INDEX) is
    record
      case FACET_DATA_PROVIDED is
        when FACET_DATA |
             FACET_COLOUR_DATA |
             FACET_NORMAL_DATA |
             FACET_COLOUR_NORMAL_DATA =>
          DATA_ITEMS : LIST_OF_DATA_VALUE_SET( 1..LENGTH,
                                               1..PER_FACET);
        when others =>
          null;
      end case;
    end record;

  type FACET_DATA_LIST_SET
         (FACET_DATA_PROVIDED : FACET_DATA_FLAG;
          LENGTH              : FACET_SET_DIMENSION;
          MODEL               : COLOUR_MODEL;
          PER_FACET           : DATA_VALUE_INDEX) is
      record
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567342 920 ■


```
        COLOURS     : FACET_COLOUR_LIST( FACET_DATA_PROVIDED,
                                         LENGTH,
                                         MODEL);
        NORMALS     : FACET_NORMAL_LIST( FACET_DATA_PROVIDED,
                                         LENGTH);
        DATA_ITEMS : FACET_DATA_LIST ( FACET_DATA_PROVIDED,
                                       LENGTH,
                                       PER_FACET);
    end record;

type ACCESS_FACET_DATA_LIST_SET is access FACET_DATA_LIST_SET;

type VERTEX_COLOUR_ARRAY
      (VERTEX_DATA_PROVIDED : VERTEX_DATA_FLAG;
       WIDTH                : VERTEX_SET_DIMENSION;
       HEIGHT               : VERTEX_SET_DIMENSION;
       MODEL                : COLOUR_MODEL) is
  record
    case VERTEX_DATA_PROVIDED is
      when COORDINATES_COLOUR |
           COORDINATES_COLOUR_NORMAL |
           COORDINATES_COLOUR_DATA |
           COORDINATES_COLOUR_NORMAL_DATA =>
        COLOURS : COLOUR_VALUE_ARRAY( MODEL,
                                      WIDTH,
                                      HEIGHT);
      when others =>
        null;
    end case;
  end record;

type VERTEX_NORMAL_ARRAY
      (VERTEX_DATA_PROVIDED : VERTEX_DATA_FLAG;
       WIDTH                : VERTEX_SET_DIMENSION;
       HEIGHT               : VERTEX_SET_DIMENSION) is
  record
    case VERTEX_DATA_PROVIDED is
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567343 867 ■


```
        when COORDINATES_NORMAL |
             COORDINATES_COLOUR_NORMAL |
             COORDINATES_NORMAL_DATA |
             COORDINATES_COLOUR_NORMAL_DATA =>
          NORMALS : VECTOR_ARRAY_3( 1..WIDTH,
                                    1..HEIGHT);
        when others =>
          null;
      end case;
    end record;

  type VERTEX_DATA_ARRAY
        (VERTEX_DATA_PROVIDED : VERTEX_DATA_FLAG;
         WIDTH                : VERTEX_SET_DIMENSION;
         HEIGHT               : VERTEX_SET_DIMENSION;
         PER_VERTEX           : DATA_VALUE_INDEX) is
    record
      case VERTEX_DATA_PROVIDED is
        when COORDINATES_DATA |
             COORDINATES_COLOUR_DATA |
             COORDINATES_NORMAL_DATA |
             COORDINATES_COLOUR_NORMAL_DATA =>
          DATA_ITEMS : ARRAY_OF_DATA_VALUE_SET( 1..WIDTH,
                                                1..HEIGHT,
                                                1..PER_VERTEX);
        when others =>
          null;
      end case;
    end record;

  type VERTEX_DATA_ARRAY_SET_3
         (VERTEX_DATA_PROVIDED : VERTEX_DATA_FLAG;
          WIDTH                : VERTEX_SET_DIMENSION;
          HEIGHT               : VERTEX_SET_DIMENSION;
          MODEL                : COLOUR_MODEL;
          PER_VERTEX           : DATA_VALUE_INDEX) is
      record
        POINTS      : POINT_ARRAY_3( 1..WIDTH,
                                     1..HEIGHT);

        COLOURS     : VERTEX_COLOUR_ARRAY( VERTEX_DATA_PROVIDED,
                                           WIDTH,
                                           HEIGHT,
                                           MODEL);
        NORMALS     : VERTEX_NORMAL_ARRAY( VERTEX_DATA_PROVIDED,
                                           WIDTH,
                                           HEIGHT);
        DATA_ITEMS : VERTEX_DATA_ARRAY( VERTEX_DATA_PROVIDED,
                                        WIDTH,
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567344 7T3 ■


```
                                        HEIGHT,
                                        PER_VERTEX);
    end record;

type ACCESS_VERTEX_DATA_ARRAY_SET_3 is access VERTEX_DATA_ARRAY_SET_3;

type VERTEX_DATA_ARRAY_SET_2
      (VERTEX_DATA_PROVIDED : VERTEX_DATA_FLAG;
       WIDTH                : VERTEX_SET_DIMENSION;
       HEIGHT               : VERTEX_SET_DIMENSION;
       MODEL                : COLOUR_MODEL;
       PER_VERTEX           : DATA_VALUE_INDEX) is
    record
      POINTS      : POINT_ARRAY_2( 1..WIDTH,
                                   1..HEIGHT);

      COLOURS     : VERTEX_COLOUR_ARRAY( VERTEX_DATA_PROVIDED,
                                         WIDTH,
                                         HEIGHT,
                                         MODEL);
      NORMALS     : VERTEX_NORMAL_ARRAY( VERTEX_DATA_PROVIDED,
                                         WIDTH,
                                         HEIGHT);
      DATA_ITEMS : VERTEX_DATA_ARRAY( VERTEX_DATA_PROVIDED,
                                      WIDTH,
                                      HEIGHT,
                                      PER_VERTEX);
    end record;

type ACCESS_VERTEX_DATA_ARRAY_SET_2 is access VERTEX_DATA_ARRAY_SET_2;

type VERTEX_COLOUR_LIST
      (VERTEX_DATA_PROVIDED : VERTEX_DATA_FLAG;
       LENGTH               : VERTEX_SET_DIMENSION;
       MODEL                : COLOUR_MODEL) is
  record
    case VERTEX_DATA_PROVIDED is
      when COORDINATES_COLOUR |
           COORDINATES_COLOUR_NORMAL |
           COORDINATES_COLOUR_DATA |
           COORDINATES_COLOUR_NORMAL_DATA =>
        COLOURS : COLOUR_VALUE_LIST( MODEL,
                                     LENGTH);
      when others =>
        null;
    end case;
  end record;

type VERTEX_NORMAL_LIST
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567345 63T ■


```
        (VERTEX_DATA_PROVIDED : VERTEX_DATA_FLAG;
         LENGTH               : VERTEX_SET_DIMENSION) is
    record
      case VERTEX_DATA_PROVIDED is
        when COORDINATES_NORMAL |
             COORDINATES_COLOUR_NORMAL |
             COORDINATES_NORMAL_DATA |
             COORDINATES_COLOUR_NORMAL_DATA =>
          NORMALS : VECTOR_LIST_3(1..LENGTH);
        when others =>
          null;
      end case;
    end record;

type VERTEX_DATA_LIST
        (VERTEX_DATA_PROVIDED : VERTEX_DATA_FLAG;
         LENGTH               : VERTEX_SET_DIMENSION;
         PER_VERTEX           : DATA_VALUE_INDEX) is
    record
      case VERTEX_DATA_PROVIDED is
        when COORDINATES_DATA |
             COORDINATES_COLOUR_DATA |
             COORDINATES_NORMAL_DATA |
             COORDINATES_COLOUR_NORMAL_DATA =>
          DATA_ITEMS : LIST_OF_DATA_VALUE_SET( 1..LENGTH,
                                               1..PER_VERTEX);
        when others =>
          null;
      end case;
    end record;

type VERTEX_DATA_LIST_SET_3
        (VERTEX_DATA_PROVIDED : VERTEX_DATA_FLAG;
         LENGTH               : VERTEX_SET_DIMENSION;
         MODEL                : COLOUR_MODEL;
         PER_VERTEX           : DATA_VALUE_INDEX) is
      record
        POINTS      : POINT_LIST_3(1..LENGTH);

        COLOURS     : VERTEX_COLOUR_LIST( VERTEX_DATA_PROVIDED,
                                          LENGTH,
                                          MODEL);
        NORMALS     : VERTEX_NORMAL_LIST( VERTEX_DATA_PROVIDED,
                                          LENGTH);
        DATA_ITEMS : VERTEX_DATA_LIST( VERTEX_DATA_PROVIDED,
                                       LENGTH,
                                       PER_VERTEX);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
    end record;

type ACCESS_VERTEX_DATA_LIST_SET_3 is access VERTEX_DATA_LIST_SET_3;

type LIST_OF_VERTEX_DATA_LIST_SET_3 is array (VERTEX_SET_COUNT) of
        ACCESS_VERTEX_DATA_LIST_SET_3;

type ACCESS_LIST_OF_VERTEX_DATA_LIST_SET_3 is
        access LIST_OF_VERTEX_DATA_LIST_SET_3;

type VERTEX_DATA_LIST_SET_2
       (VERTEX_DATA_PROVIDED : VERTEX_DATA_FLAG;
        LENGTH               : VERTEX_SET_DIMENSION;
        MODEL                : COLOUR_MODEL;
        PER_VERTEX           : DATA_VALUE_INDEX) is
    record
      POINTS      : POINT_LIST_2(1..LENGTH);
      COLOURS     : VERTEX_COLOUR_LIST( VERTEX_DATA_PROVIDED,
                                        LENGTH,
                                        MODEL);
      NORMALS     : VERTEX_NORMAL_LIST( VERTEX_DATA_PROVIDED,
                                        LENGTH);
      DATA_ITEMS : VERTEX_DATA_LIST( VERTEX_DATA_PROVIDED,
                                     LENGTH,
                                     PER_VERTEX);
    end record;

type ACCESS_VERTEX_DATA_LIST_SET_2 is access VERTEX_DATA_LIST_SET_2;

type LIST_OF_VERTEX_DATA_LIST_SET_2 is array (VERTEX_SET_COUNT) of
        ACCESS_VERTEX_DATA_LIST_SET_2;

type ACCESS_LIST_OF_VERTEX_DATA_LIST_SET_2 is
        access LIST_OF_VERTEX_DATA_LIST_SET_2;

type VERTEX_COLOUR_LIST_SET_3
       (VERTEX_COLOUR_PROVIDED  : VERTEX_COLOUR_FLAG;
        LENGTH                  : VERTEX_SET_DIMENSION;
        MODEL                   : COLOUR_MODEL) is
    record
      POINTS      : POINT_LIST_3(1..LENGTH);
      COLOURS     : VERTEX_COLOUR_LIST( VERTEX_COLOUR_PROVIDED,
                                        LENGTH,
                                        MODEL);
    end record;

type ACCESS_VERTEX_COLOUR_LIST_SET_3 is access VERTEX_COLOUR_LIST_SET_3;

type LIST_OF_VERTEX_COLOUR_LIST_SET_3 is array (VERTEX_SET_COUNT) of
        ACCESS_VERTEX_COLOUR_LIST_SET_3;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567347 402 ■


```
    type ACCESS_LIST_OF_VERTEX_COLOUR_LIST_SET_3 is
            access LIST_OF_VERTEX_COLOUR_LIST_SET_3;

    procedure DEALLOCATE (OBJECT: in out LIST_OF_VERTEX_DATA_LIST_SET_3);

    procedure DEALLOCATE (OBJECT: in out LIST_OF_VERTEX_DATA_LIST_SET_2);

    procedure DEALLOCATE (OBJECT: in out LIST_OF_VERTEX_COLOUR_LIST_SET_3);

end PHIGS_COORDINATE_SYSTEM;


package body PHIGS_COORDINATE_SYSTEM is

-- This package body contains stubs for each of the subprograms contained in the
-- PHIGS_COORDINATE_SYSTEM package.  The purpose of these stubs is to show
-- compilability for the PHIGS/Ada binding.  An implementation would replace these
-- stubs with code appropriate to implement the functionality.

    procedure DEALLOCATE (OBJECT   : in out LIST_OF_VERTEX_DATA_LIST_SET_3) is
    begin
      null;
    end DEALLOCATE;

    procedure DEALLOCATE (OBJECT   : in out LIST_OF_VERTEX_DATA_LIST_SET_2) is
    begin
      null;
    end DEALLOCATE;

    procedure DEALLOCATE (OBJECT   : in out LIST_OF_VERTEX_COLOUR_LIST_SET_3) is
    begin
      null;
    end DEALLOCATE;

end PHIGS_COORDINATE_SYSTEM;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
-- The PHIGS Types Package

with  PHIGS_CONFIGURATION,
      PHIGS_BASE_TYPES,
      PHIGS_COLOUR_TYPES,
      PHIGS_STANDARD_TYPES,
      PHIGS_COORDINATE_SYSTEM,
      PHIGS_LIST_UTILITIES,
      PHIGS_NAME_SET_FACILITY;


package PHIGS_TYPES is

-- This package contains all the data type definitions used to define
-- the Ada binding to PHIGS.

-- Configuration names

use PHIGS_CONFIGURATION;

   -- Make PHIGS_BASE_TYPES visible externally.

   subtype PHIGS_INTEGER is PHIGS_BASE_TYPES.PHIGS_INTEGER;
   subtype PHIGS_NATURAL is PHIGS_BASE_TYPES.PHIGS_NATURAL;
   subtype PHIGS_POSITIVE is PHIGS_BASE_TYPES.PHIGS_POSITIVE;
   subtype PHIGS_STRING is PHIGS_BASE_TYPES.PHIGS_STRING;
   subtype SCALE_FACTOR is PHIGS_BASE_TYPES.SCALE_FACTOR;
   subtype SMALL_NATURAL is PHIGS_BASE_TYPES.SMALL_NATURAL;
   subtype COLOUR_VALUE_SET_DIMENSION is
                                 PHIGS_BASE_TYPES.COLOUR_VALUE_SET_DIMENSION;
   subtype FACET_SET_DIMENSION is PHIGS_BASE_TYPES.FACET_SET_DIMENSION;
   subtype VERTEX_SET_COUNT is PHIGS_BASE_TYPES.VERTEX_SET_COUNT;
   subtype VERTEX_SET_DIMENSION is PHIGS_BASE_TYPES.VERTEX_SET_DIMENSION;
   subtype SPLINE_ORDER is PHIGS_BASE_TYPES.SPLINE_ORDER;
   subtype SPLINE_RATIONALITY is PHIGS_BASE_TYPES.SPLINE_RATIONALITY;

   -- Make PHIGS_STANDARD_TYPES visible externally.

   subtype DATA_VALUE is PHIGS_STANDARD_TYPES.DATA_VALUE;
   subtype DATA_VALUE_INDEX is PHIGS_STANDARD_TYPES.DATA_VALUE_INDEX;
   subtype DATA_VALUE_SET is PHIGS_STANDARD_TYPES.DATA_VALUE_SET;
   subtype ACCESS_DATA_VALUE_SET is
                                  PHIGS_STANDARD_TYPES.ACCESS_DATA_VALUE_SET;
   subtype DATA_MAPPING_INDEX is PHIGS_STANDARD_TYPES.DATA_MAPPING_INDEX;
   subtype LIST_OF_DATA_VALUE_SET is
                                 PHIGS_STANDARD_TYPES.LIST_OF_DATA_VALUE_SET;
   subtype ARRAY_OF_DATA_VALUE_SET is
                               PHIGS_STANDARD_TYPES.ARRAY_OF_DATA_VALUE_SET;
   subtype COLOUR_COMPONENTS is PHIGS_STANDARD_TYPES.COLOUR_COMPONENTS;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
subtype COLOUR_INDEX is PHIGS_STANDARD_TYPES.COLOUR_INDEX;
subtype COLOUR_MODEL is PHIGS_STANDARD_TYPES.COLOUR_MODEL;

-- The following constants define the colour models specified by PHIGS:

INDIRECT   : constant COLOUR_MODEL := 0;
RGB        : constant COLOUR_MODEL := 1;
CIELUV     : constant COLOUR_MODEL := 2;
HSV        : constant COLOUR_MODEL := 3;
HLS        : constant COLOUR_MODEL := 4;

subtype COLOUR_COEFFICIENT is PHIGS_STANDARD_TYPES.COLOUR_COEFFICIENT;
subtype COLOUR_COEFFICIENT_ARRAY is
                        PHIGS_STANDARD_TYPES.COLOUR_COEFFICIENT_ARRAY;
subtype INTENSITY is PHIGS_STANDARD_TYPES.INTENSITY;

-- Types for CIELUV Colour Model

subtype CIELUV_COLOUR_VALUE is
                              PHIGS_STANDARD_TYPES.CIELUV_COLOUR_VALUE:
subtype CIELUV_HOMOGENEOUS_COLOUR_VALUE is
            PHIGS_STANDARD_TYPES.CIELUV_HOMOGENEOUS_COLOUR_VALUE;
package CIELUV_TYPES renames PHIGS_STANDARD_TYPES.CIELUV_TYPES;

-- Type for Generic Colour Models

package GENERIC_COLOUR_TYPES renames
                              PHIGS_STANDARD_TYPES.GENERIC_COLOUR_TYPES;

-- Types for HLS Colour Model

subtype HLS_COLOUR_VALUE is PHIGS_STANDARD_TYPES.HLS_COLOUR_VALUE;
subtype HLS_HOMOGENEOUS_COLOUR_VALUE is
                  PHIGS_STANDARD_TYPES.HLS_HOMOGENEOUS_COLOUR_VALUE;
package HLS_TYPES renames PHIGS_STANDARD_TYPES.HLS_TYPES;

-- Types for HSV Colour Model

subtype HSV_COLOUR_VALUE is PHIGS_STANDARD_TYPES.HSV_COLOUR_VALUE;
subtype HSV_HOMOGENEOUS_COLOUR_VALUE is
                  PHIGS_STANDARD_TYPES.HSV_HOMOGENEOUS_COLOUR_VALUE;
package HSV_TYPES renames PHIGS_STANDARD_TYPES.HSV_TYPES;

-- Types for Indirect Colour Model

subtype INDIRECT_HOMOGENEOUS_COLOUR_VALUE is
            PHIGS_STANDARD_TYPES.INDIRECT_HOMOGENEOUS_COLOUR_VALUE;
package INDIRECT_TYPES renames PHIGS_STANDARD_TYPES.INDIRECT_TYPES;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
-- Types for RGB Colour Model

subtype RGB_COLOUR_VALUE is PHIGS_STANDARD_TYPES.RGB_COLOUR_VALUE;
subtype RGB_HOMOGENEOUS_COLOUR_VALUE is
                    PHIGS_STANDARD_TYPES.RGB_HOMOGENOUS_COLOUR_VALUE;
package RGB_TYPES renames PHIGS_STANDARD_TYPES.RGB_TYPES;

-- GENERAL_COLOUR

subtype GENERAL_COLOUR is PHIGS_STANDARD_TYPES.GENERAL_COLOUR;

-- COLOUR_VALUE_ARRAY

subtype COLOUR_VALUE_ARRAY is
                                  PHIGS_STANDARD_TYPES.COLOUR_VALUE_ARRAY;

-- COLOUR_VALUE_LIST

subtype COLOUR_VALUE_LIST is PHIGS_STANDARD_TYPES.COLOUR_VALUE_LIST;
subtype LIST_OF_COLOUR_VALUE_LIST is
                              PHIGS_STANDARD_TYPES.LIST_OF_COLOUR_VALUE_LIST;

-- FACET_DATA_FLAG

subtype FACET_DATA_FLAG is PHIGS_STANDARD_TYPES.FACET_DATA_FLAG;

-- VERTEX_DATA_FLAG

subtype VERTEX_DATA_FLAG is PHIGS_STANDARD_TYPES.VERTEX_DATA_FLAG;

-- VERTEX_COLOUR_FLAG

subtype VERTEX_COLOUR_FLAG is PHIGS_STANDARD_TYPES.VERTEX_COLOUR_FLAG;

-- CHOICE_SMALL_NATURAL

subtype CHOICE_SMALL_NATURAL is
  PHIGS_NATURAL range 0..MAX_CHOICE_SMALL_NATURAL;

-- This is a subtype declaration which allows for
-- unconstrained record objects for CHOICE_PROMPT_STRING_LIST
-- type without causing the exception STORAGE_ERROR to be raised.

-- COLOUR_MATRIX_SMALL_NATURAL

subtype COLOUR_MATRIX_SMALL_NATURAL is
  PHIGS_NATURAL range 0..MAX_COLOUR_MATRIX_SMALL_NATURAL;

-- This is a subtype declaration which allows for unconstrained record
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567351 933 ■


```
-- objects of VARIABLE_COLOUR_MATRIX type without causing the
-- exception STORAGE_ERROR to be raised.

-- FILE_SMALL_NATURAL

subtype FILE_SMALL_NATURAL is
  PHIGS_NATURAL range 0..MAX_FILE_SMALL_NATURAL;

-- This is a subtype declaration which allows for
-- unconstrained record objects for VARIABLE_FILE_ID type
-- without causing the exception STORAGE_ERROR to be raised.

-- INPUT_STRING_SMALL_NATURAL

subtype INPUT_STRING_SMALL_NATURAL is
  PHIGS_NATURAL range 0..MAX_INPUT_STRING_SMALL_NATURAL;

-- This is a subtype declaration which allows for
-- unconstrained record objects for INPUT_STRING type
-- without causing the exception STORAGE_ERROR to be raised.

-- STRING_SMALL_NATURAL

subtype STRING_SMALL_NATURAL is
  PHIGS_NATURAL range 0..MAX_STRING_SMALL_NATURAL;

-- This is a subtype declaration which allows for
-- unconstrained record objects for various string record
-- types without causing the exception STORAGE_ERROR to be
-- raised.


-- DATA_VALUE_INDEX_PAIR

type DATA_VALUE_INDEX_PAIR is
  record
    INDEX_A  : DATA_VALUE_INDEX;
    INDEX_B  : DATA_VALUE_INDEX;
  end record;

-- Provides for selecting from bi-valued sets of data mapping input values.

-- DATA_VALUE_RANGE
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
type DATA_VALUE_RANGE is
  record
    UPPER  : DATA_VALUE;
    LOWER  : DATA_VALUE;
  end record;

-- Provides for specifying a range of data mapping input values.

-- SOURCE_SELECTOR

type SOURCE_SELECTOR is ( COLOUR_ASPECT,
                          VERTEX_COLOUR,
                          VERTEX_DATA,
                          FACET_COLOUR,
                          FACET_DATA);

-- Provides for specifying sources for colour information.

-- SOURCE_SELECTOR_LIST

type SOURCE_SELECTOR_LIST is
  array (SMALL_NATURAL) of SOURCE_SELECTOR;

-- Provides for specification of ordered lists of sources for colour information.

-- DATA_MAPPING_METHOD

type DATA_MAPPING_METHOD is new PHIGS_INTEGER;

-- Provides for differentiating methods of performing data mapping.

-- DATA_MAPPING_METHODS

package DATA_MAPPING_METHODS is
        new PHIGS_LIST_UTILITIES
                (DATA_MAPPING_METHOD,
                 MAX_DATA_MAPPING_METHODS_SUPPORTED);

-- Provides for lists of methods of performing data mapping.

DATA_MAPPING_COLOUR          : constant DATA_MAPPING_METHOD := 1;
SINGLE_VALUE_UNIFORM         : constant DATA_MAPPING_METHOD := 2;
SINGLE_VALUE_NON_UNIFORM     : constant DATA_MAPPING_METHOD := 3;
BI_VALUE_UNIFORM             : constant DATA_MAPPING_METHOD := 4;
BI_VALUE_NON_UNIFORM         : constant DATA_MAPPING_METHOD := 5;

-- DATA_MAPPING_INDICES
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567353 706 ■


```
package DATA_MAPPING_INDICES is
        new PHIGS_LIST_UTILITIES
                (DATA_MAPPING_INDEX,
                 MAX_DATA_MAPPING_INDICES_SUPPORTED);

-- Provides for lists of data mapping indices.

-- RETURN_SPLINE_ORDER

subtype RETURN_SPLINE_ORDER is
            SPLINE_ORDER range 6..SPLINE_ORDER'last;

-- Provides for restricting spline orders to those acceptable for returning values of the
-- facilities.

-- COLOUR_AVAILABLE

type COLOUR_AVAILABLE is (  COLOUR,
                            MONOCHROME);

-- Indicates whether colour output is available on a
-- workstation.

-- COLOUR_INDICES

package COLOUR_INDICES is
  new PHIGS_LIST_UTILITIES ( COLOUR_INDEX,
                             MAX_COLOUR_INDICES_SUPPORTED);

-- Provides for a set of colour indices which are available
-- on a particular workstation.

-- COLOUR_MATRIX

type COLOUR_MATRIX is array ( COLOUR_MATRIX_SMALL_NATURAL range <>,
                              COLOUR_MATRIX_SMALL_NATURAL range <>)
  of COLOUR_INDEX;

-- Provides for matrices containing colour indices corresponding
-- to a cell array or pattern array.

-- COLOUR_MODELS

package COLOUR_MODELS is
  new PHIGS_LIST_UTILITIES ( COLOUR_MODEL,
                             MAX_COLOUR_MODELS_SUPPORTED);

-- Provides for lists of colour models.
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
-- The following constant is for use when specifying the rendering colour model:

WS_DEPENDENT_COLOUR_MODEL  : constant COLOUR_MODEL := 0;

-- COLOUR_COEFFICIENTS

package COLOUR_COEFFICIENTS is
  new PHIGS_LIST_UTILITIES ( COLOUR_COEFFICIENT,
                             MAX_COLOUR_COEFFICIENTS);

-- Provides for a lists of colour coefficients for the generalized
-- colour model.


-- COLOUR_REPRESENTATION

type COLOUR_REPRESENTATION (MODEL : COLOUR_MODEL := RGB) is
  record
    case MODEL is

      when RGB =>
        RED_RGB              : INTENSITY;
        GREEN_RGB            : INTENSITY;
        BLUE_RGB             : INTENSITY;

      when CIELUV =>
        L_STAR_CIE           : COLOUR_COEFFICIENT;
        U_STAR_CIE           : COLOUR_COEFFICIENT;
        V_STAR_CIE           : COLOUR_COEFFICIENT;

      when HSV =>
        HUE_HSV              : INTENSITY;
        SATURATION_HSV       : INTENSITY;
        VALUE_HSV            : INTENSITY;

      when HLS =>
        HUE_HLS              : INTENSITY;
        LIGHTNESS_HLS        : INTENSITY;
        SATURATION_HLS       : INTENSITY;

      when others =>
        GENERIC_COMPONENTS : COLOUR_COEFFICIENT_ARRAY;
    end case;
  end record;

-- Defines the representation of a colour as a combination of three
-- components whose meaning depends on the colour model.  Additional
-- variants may be included when additional registered or unregistered
-- colour models are supported by an implementation.
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567355 589 ■


```
-- CHROMATICITY_COEFFICIENT

type CHROMATICITY_COEFFICIENT is
  record
    U_PRIME  : COLOUR_COEFFICIENT;
    V_PRIME  : COLOUR_COEFFICIENT;
    Y        : COLOUR_COEFFICIENT;
  end record;

-- Defines a CIE colour coefficient value.

-- CHROMATICITY_COEFFICIENT_SET

type CHROMATICITY_COEFFICIENT_SET is
  record
    R : CHROMATICITY_COEFFICIENT;
    G : CHROMATICITY_COEFFICIENT;
    B : CHROMATICITY_COEFFICIENT;
  end record;

-- Defines a set of CIE primary colour chromaticity coefficients.

-- ACCESS_COLOUR_VALUE_ARRAY

type ACCESS_COLOUR_VALUE_ARRAY is access COLOUR_VALUE_ARRAY;

-- Provides for pointers to arrays of general colour values.


-- ACCESS_COLOUR_VALUE_LIST

type ACCESS_COLOUR_VALUE_LIST is access COLOUR_VALUE_LIST;

-- Provides for pointers to lists of general colour values.

-- ACCESS_LIST_OF_COLOUR_VALUE_LIST

type ACCESS_LIST_OF_COLOUR_VALUE_LIST is
        access LIST_OF_COLOUR_VALUE_LIST;

-- Provides for pointers to lists of lists of colour values values.

-- COLOUR CONTROL POINT ARRAY

type COLOUR_CONTROL_POINT_ARRAY
  (MODEL          : COLOUR_MODEL := RGB;
   RATIONALITY    : SPLINE_RATIONALITY := RATIONAL;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
    LENGTH_U : COLOUR_VALUE_SET_DIMENSION := 1;
    LENGTH_V : COLOUR_VALUE_SET_DIMENSION := 1) is
     record
       case MODEL is

         when INDIRECT =>
           COLOURS_INDIRECT : INDIRECT_TYPES.CONTROL_POINT_ARRAY
                                (RATIONALITY, LENGTH_U, LENGTH_V);

         when RGB =>
           COLOURS_RGB : RGB_TYPES.CONTROL_POINT_ARRAY
                           (RATIONALITY, LENGTH_U, LENGTH_V);

         when CIELUV =>
           COLOURS_CIELUV : CIELUV_TYPES.CONTROL_POINT_ARRAY
                              (RATIONALITY, LENGTH_U, LENGTH_V);

         when HSV =>
           COLOURS_HSV  : HSV_TYPES.CONTROL_POINT_ARRAY
                            (RATIONALITY, LENGTH_U, LENGTH_V);

         when HLS =>
           COLOURS_HLS : HLS_TYPES.CONTROL_POINT_ARRAY
                           (RATIONALITY, LENGTH_U, LENGTH_V);

         when others =>
           COLOURS_GENERIC : GENERIC_COLOUR_TYPES.CONTROL_POINT_ARRAY
                               (RATIONALITY, LENGTH_U, LENGTH_V);

       end case;
     end record;

-- Provides for specification of two-dimensional arrays of colour control point values
-- all of which are the same colour model.


-- COLOUR CONTROL POINT LIST

type COLOUR_CONTROL_POINT_LIST
  (MODEL          : COLOUR_MODEL := RGB;
   RATIONALITY  : SPLINE_RATIONALITY := RATIONAL;
   LENGTH  : COLOUR_VALUE_SET_DIMENSION := 0) is
    record
      case MODEL is

        when INDIRECT =>
          COLOURS_INDIRECT : INDIRECT_TYPES.CONTROL_POINT_LIST
                               (RATIONALITY, LENGTH);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567357 351 ■


```
            when RGB =>
              COLOURS_RGB : RGB_TYPES.CONTROL_POINT_LIST
                                    (RATIONALITY, LENGTH);

            when CIELUV =>
              COLOURS_CIELUV  : CIELUV_TYPES.CONTROL_POINT_LIST
                                    (RATIONALITY, LENGTH);

            when HSV =>
              COLOURS_HSV : HSV_TYPES.CONTROL_POINT_LIST
                                    (RATIONALITY, LENGTH);

            when HLS =>
              COLOURS_HLS : HLS_TYPES.CONTROL_POINT_LIST
                                    (RATIONALITY, LENGTH);

            when others =>
              COLOURS_GENERIC : GENERIC_COLOUR_TYPES.CONTROL_POINT_LIST
                                    (RATIONALITY, LENGTH);

          end case;
        end record;

    -- Provides for specification of lists of colour control point values
    -- all of which are the same colour model.

    -- DATA_MAPPING_DATA_RECORD

    type DATA_MAPPING_DATA_RECORD
      (METHOD : DATA_MAPPING_METHOD := DATA_MAPPING_COLOUR) is
        record
          case METHOD is

            when DATA_MAPPING_COLOUR =>
              COLOUR_SELECTORS : SOURCE_SELECTOR_LIST;

            when SINGLE_VALUE_UNIFORM =>
              SINGLE_UNIFORM_SELECTORS          : SOURCE_SELECTOR_LIST;
              SINGLE_UNIFORM_DATA_VALUE_IND     : DATA_VALUE_INDEX;
              SINGLE_UNIFORM_RANGE              : DATA_VALUE_RANGE;
              SINGLE_UNIFORM_COLOURS            : ACCESS_COLOUR_VALUE_LIST;

            when SINGLE_VALUE_NON_UNIFORM =>
              SINGLE_NON_UNIFORM_SELECTORS     : SOURCE_SELECTOR_LIST;
              SINGLE_NON_UNIFORM_DATA_VALUE_IND : DATA_VALUE_INDEX;
              SINGLE_NON_UNIFORM_BOUNDARIES    : ACCESS_DATA_VALUE_SET;
              SINGLE_NON_UNIFORM_COLOURS       : ACCESS_COLOUR_VALUE_LIST;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
            when BI_VALUE_UNIFORM =>
              BI_UNIFORM_SELECTORS : SOURCE_SELECTOR_LIST;
              BI_UNIFORM_DATA_VALUE_IND_PAIR : DATA_VALUE_INDEX_PAIR;
              BI_UNIFORM_A_RANGE   : DATA_VALUE_RANGE;
              BI_UNIFORM_B_RANGE   : DATA_VALUE_RANGE;
              BI_UNIFORM_COLOURS   : ACCESS_LIST_OF_COLOUR_VALUE_LIST;

            when BI_VALUE_NON_UNIFORM =>
              BI_NON_UNIFORM_SELECTORS      : SOURCE_SELECTOR_LIST;
              BI_NON_UNIFORM_DATA_VALUE_IND_PAIR : DATA_VALUE_INDEX_PAIR;
              BI_NON_UNIFORM_A_BOUNDARIES : ACCESS_DATA_VALUE_SET;
              BI_NON_UNIFORM_B_BOUNDARIES : ACCESS_DATA_VALUE_SET;
              BI_NON_UNIFORM_COLOURS       : ACCESS_COLOUR_VALUE_ARRAY;

            when others =>
              null;

          end case;
        end record;

    -- Provides for specifying values for different methods of data_mapping.
    -- Additional variants may be included when additional registered or unregistered data
    -- mapping methods are supported by an implementation.

    -- MC_TYPE

    type MC_TYPE is digits PHIGS_PRECISION;

    -- Defines the type of coordinate in the Modeling Coordinate
    -- System.

    -- MC

    package MC is new PHIGS_COORDINATE_SYSTEM (MC_TYPE);

    -- Defines the Modeling Coordinate System.

    -- WC_TYPE

    type WC_TYPE is digits PHIGS_PRECISION;

    -- Defines the type of coordinate in the World Coordinate
    -- System.

    -- WC

    package WC is new PHIGS_COORDINATE_SYSTEM (WC_TYPE);

    -- Defines the World Coordinate System.
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567359 124 ■


```
-- VRC_TYPE

type VRC_TYPE is digits PHIGS_PRECISION;

-- Defines the type of coordinate in the View Reference
-- Coordinate System.

-- VRC

package VRC is new PHIGS_COORDINATE_SYSTEM (VRC_TYPE);

-- Defines the viewing coordinate system.

-- NPC_TYPE

type NPC_TYPE is digits PHIGS_PRECISION;

-- Defines the type of a coordinate in the Normalized
-- Projection Coordinate System.

-- NPC

package NPC is new PHIGS_COORDINATE_SYSTEM (NPC_TYPE);

-- Defines the Normalized Projection Coordinate System.

-- DC_TYPE

type DC_TYPE is digits PHIGS_PRECISION;

-- The type of a coordinate component in the Device Coordinate
-- System.

-- DC

package DC is new PHIGS_COORDINATE_SYSTEM (DC_TYPE);

-- Defines the Device Coordinate System.


-- SPC_TYPE

type SPC_TYPE is digits PHIGS_PRECISION;

-- Defines the type of a coordinate in the Spline Parameter Coordinate System.


-- SPC
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
package SPC is new PHIGS_COORDINATE_SYSTEM(SPC_TYPE);

-- Defines the Spline Parameter Coordinate System.

-- KNOT_VECTOR

subtype KNOT_VECTOR is SPC.MAGNITUDE_LIST;

-- Provides for specifying spline knot vectors.


-- CURVE COLOURSPLINE

type CURVE_COLOURSPLINE( MODEL        : COLOUR_MODEL;
                         RATIONALITY  : SPLINE_RATIONALITY;
                         KNOT_COUNT   : SMALL_NATURAL;
                         LENGTH       : COLOUR_VALUE_SET_DIMENSION) is
  record
    ORDER          : SPLINE_ORDER;
    KNOTS          : KNOT_VECTOR(KNOT_COUNT);
    CONTROL_POINTS : COLOUR_CONTROL_POINT_LIST
                       (MODEL, RATIONALITY, LENGTH);
  end record;

-- Provides for colour splines for use with Non-uniform B-Spline Curve primitives.


-- ACCESS CURVE COLOURSPLINE

type ACCESS_CURVE_COLOURSPLINE is
        access CURVE_COLOURSPLINE;

-- Provides for pointers to colourspline definitions.


-- SURFACE COLOURSPLINE

type SURFACE_COLOURSPLINE
       (MODEL          : COLOUR_MODEL;
        RATIONALITY    : SPLINE_RATIONALITY;
        KNOT_COUNT_U   : SMALL_NATURAL;
        KNOT_COUNT_V   : SMALL_NATURAL;
        LENGTH_U       : COLOUR_VALUE_SET_DIMENSION;
        LENGTH_V       : COLOUR_VALUE_SET_DIMENSION) is
  record
    U_ORDER        : SPLINE_ORDER;
    V_ORDER        : SPLINE_ORDER;
    U_KNOTS        : KNOT_VECTOR(KNOT_COUNT_U);
    V_KNOTS        : KNOT_VECTOR(KNOT_COUNT_V);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
    CONTROL_POINTS : COLOUR_CONTROL_POINT_ARRAY
                        (MODEL, RATIONALITY, LENGTH_U, LENGTH_V);
  end record;

-- Provides for colour splines for use with Non-uniform B-Spline Surface primitives.


-- ACCESS SURFACE COLOURSPLINE

type ACCESS_SURFACE_COLOURSPLINE is
        access SURFACE_COLOURSPLINE;

-- Provides for pointers to colourspline definitions.


-- DATASPLINE

type DATASPLINE( MODEL          : COLOUR_MODEL := RGB;
                 RATIONALITY    : SPLINE_RATIONALITY := RATIONAL;
                 KNOT_COUNT_U   : SMALL_NATURAL := 4;
                 KNOT_COUNT_V   : SMALL_NATURAL := 4;
                 WIDTH          : COLOUR_VALUE_SET_DIMENSION := 4;
                 HEIGHT         : COLOUR_VALUE_SET_DIMENSION := 4) is
  record
    U_ORDER          : SPLINE_ORDER;
    V_ORDER          : SPLINE_ORDER;
    U_KNOTS          : KNOT_VECTOR(KNOT_COUNT_U);
    V_KNOTS          : KNOT_VECTOR(KNOT_COUNT_V);
    CONTROL_POINTS : SPC.CONTROL_POINT_ARRAY_3
                        (RATIONALITY, WIDTH, HEIGHT);
  end record;

-- Provides for data splines for use with Non-uniform B-Spline primitives.

-- DATASPLINE_LIST

type DATASPLINE_LIST is
        array (DATA_VALUE_INDEX range <>) of DATASPLINE;

-- Provides for lists of data mapping data splines.

-- ACCESS_DATASPLINE_LIST

type ACCESS_DATASPLINE_LIST is
        access DATASPLINE_LIST;

-- Provides for pointers to lists of dataspline definitions.

-- GENERAL_FLOAT_PARAMETER
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
type GENERAL_FLOAT_PARAMETER is digits PHIGS_PRECISION;

-- Defines a type for specifying implementation-specific real
-- data in GDP, GSE, and ESCAPE data records.

-- OFF_ON

type OFF_ON is ( OFF,
                 ON);

-- Generic type for off/on switches.

-- ANGLE

type ANGLE is digits PHIGS_PRECISION;

-- Values used in the modeling transformation utility functions
-- for specifying angles of rotation in radians. Positive
-- indicates a counterclockwise direction.

-- ANNOTATION_STYLE

type ANNOTATION_STYLE is new PHIGS_INTEGER;

-- Defines the annotation styles for annotation primitives.

-- ANNOTATION_STYLES

package ANNOTATION_STYLES is
  new PHIGS_LIST_UTILITIES ( ANNOTATION_STYLE,
                             MAX_ANNOTATION_STYLES_SUPPORTED);

-- Provides for lists of annotation styles.

-- APPLICATION_DATA_RANGE

subtype APPLICATION_DATA_RANGE is
  PHIGS_NATURAL range 0..MAX_APPLICATION_DATA;

-- Defines range of lengths for application data objects.

-- APPLICATION DATA RECORD

type APPLICATION_DATA_RECORD
  (LENGTH : APPLICATION_DATA_RANGE := 0) is
    record
      DATA : PHIGS_STRING (1..LENGTH);
    end record;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567363 655 ■


```
-- Defines type for defining application data objects.

-- ARCHIVE_ID

type ARCHIVE_ID is new PHIGS_NATURAL;

-- Provides for an application specified pointer to archive
-- files separate from the file identifier.

-- ARCHIVE_IDS

package ARCHIVE_IDS is
  new PHIGS_LIST_UTILITIES ( ARCHIVE_ID,
                             MAX_ARCHIVE_IDS_SUPPORTED);

-- Provides for lists of archive identifiers.

-- ARCHIVE_STATE

type ARCHIVE_STATE is ( ARCL,
                        AROP);

-- The type used to return the archive state.

-- ASF

type ASF is ( BUNDLED,
              INDIVIDUAL);

-- This type defines an aspect source flag whose
-- value indicates whether an aspect of a primitive
-- should be set from a bundle table or from an
-- individual attribute.

-- ASPECT

type ASPECT is ( TYPE_OF_LINE,
                 LINEWIDTH_SF,
                 LINE_COLOUR,

                 TYPE_OF_MARKER,
                 SIZE,
                 MARKER_COLOUR,
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

FONT,
PRECISION,
EXPANSION,
SPACING,
TEXT_COLOUR,

STYLE_OF_INTERIOR,
STYLE_IND,
INTERIOR_COLOUR,

FLAG,
TYPE_OF_EDGE,
EDGEWIDTH_SF,
EDGE_COLOUR,

GENERAL_POLYLINE_COLOUR,
GENERAL_POLYMARKER_COLOUR,
GENERAL_TEXT_COLOUR,
GENERAL_INTERIOR_COLOUR,
GENERAL_EDGE_COLOUR,

METHOD_OF_POLYLINE_SHADING,
METHOD_OF_INTERIOR_SHADING,
METHOD_OF_DATA_MAPPING,
REFLECTANCE_PROPERTIES,
MODEL_OF_REFLECTANCE,

BACK_STYLE_OF_INTERIOR,
BACK_STYLE_IND,
BACK_INTERIOR_COLOUR_DIR,
BACK_METHOD_OF_INTERIOR_SHADING,
BACK_METHOD_OF_DATA_MAPPING,
BACK_REFLECTANCE_PROPERTIES,
BACK_REFLECTANCE_MODEL,

CRITERIA_FOR_CURVE_APPROX,
CRITERIA_FOR_SURFACE_APPROX,
CHARACTERISTICS_OF_PARAMETRIC_SURFACE);

-- This type lists the aspects for which an aspect source flag exists in PHIGS PLUS.

-- ATTRIBUTES_USED_TYPE

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
type ATTRIBUTES_USED_TYPE is ( POLYLINE_ATTRIBUTES,
                               POLYMARKER_ATTRIBUTES,
                               TEXT_ATTRIBUTES,
                               INTERIOR_ATTRIBUTES,
                               EDGE_ATTRIBUTES,
                               REFLECTANCE_ATTRIBUTES,
                               PARAMETRIC_SURFACE_ATTRIBUTES);

-- The types of attributes which may be used in generating output for a GDP and in generating
-- prompt and echo information for certain prompt and echo types of certain classes of input
-- devices.


-- ATTENUATION_COEFFICIENT

type ATTENUATION_COEFFICIENTS is
  record
    C1 : COLOUR_COEFFICIENT;
    C2 : COLOUR_COEFFICIENT;
  end record;

-- Provides for specification of light source attenuation.


-- ATTRIBUTES_USED

package ATTRIBUTES_USED is
  new PHIGS_LIST_UTILITIES
    (ATTRIBUTES_USED_TYPE,
     ATTRIBUTES_USED_TYPE'POS(ATTRIBUTES_USED_TYPE'LAST)+1);

-- Provides for a list of the attributes used which is returned
-- by the INQ_GDP and LOCATOR_ATTRIBUTES_USED functions.

-- CHAR_EXPANSION

type CHAR_EXPANSION is new SCALE_FACTOR range
  SCALE_FACTOR'SAFE_SMALL..SCALE_FACTOR'LAST;

-- Defines a character expansion factor.

-- CHAR_SET

type CHAR_SET is new PHIGS_NATURAL;

-- Provides identifications for the available character
-- sets.  ISO 646 character set maps to value zero.
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567366 364 ■


```
-- CHAR_SETS

package CHAR_SETS is new PHIGS_LIST_UTILITIES (       CHAR_SET,
MAX_CHAR_SETS_SUPPORTED);

-- Provides for lists of character set identifications.

-- CHAR_SPACING

type CHAR_SPACING is new SCALE_FACTOR;

-- Defines a character spacing factor.  The factors are
-- unitless.  A positive value indicates the amount of extra
-- space between character boxes in a text string, and a
-- negative value indicates the amount of overlap between
-- character boxes in a text string.

-- CHOICE_PROMPT_ECHO_TYPE

type CHOICE_PROMPT_ECHO_TYPE is new PHIGS_INTEGER;

-- Defines the choice prompt and echo type.

-- CHOICE_PROMPT_ECHO_TYPES

package CHOICE_PROMPT_ECHO_TYPES is
  new PHIGS_LIST_UTILITIES
    (CHOICE_PROMPT_ECHO_TYPE,
     MAX_CHOICE_PROMPT_ECHO_TYPES_SUPPORTED);

-- Provides for lists of choice prompt and echo types.

-- DEVICE_NUMBER

subtype DEVICE_NUMBER is PHIGS_POSITIVE;

--  Defines the base type for input device numbers.

-- CHOICE_DEVICE_NUMBER

type CHOICE_DEVICE_NUMBER is new DEVICE_NUMBER;

-- Provides for choice device identifiers.

-- CHOICE_NUMBER

type CHOICE_NUMBER is new PHIGS_POSITIVE;

-- Defines the choice numbers available on an implementation.
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567367 2T0 ■


```
-- CHOICE_PROMPT

type CHOICE_PROMPT is new OFF_ON;

-- Indicates whether a specified choice prompt is to be displayed
-- or not.

-- CHOICE_PROMPTS

package CHOICE_PROMPTS is
  new PHIGS_LIST_UTILITIES ( CHOICE_PROMPT,
                             MAX_CHOICE_PROMPTS_SUPPORTED);

-- Provides for lists of choice prompts.

-- CHOICE_PROMPT_STRING

type CHOICE_PROMPT_STRING (LENGTH : STRING_SMALL_NATURAL := 0) is
  record
    CONTENTS : PHIGS_STRING (1..LENGTH);
  end record;

-- Provides for a variable length prompt.  Objects of this type
-- should be declared unconstrained to allow for dynamic
-- modification of the length.

-- CHOICE_PROMPT_STRING_ARRAY

type CHOICE_PROMPT_STRING_ARRAY is array (PHIGS_POSITIVE range <>)
  of CHOICE_PROMPT_STRING;

-- Provides for an array of prompt strings.

-- CHOICE_PROMPT_STRING_LIST

type CHOICE_PROMPT_STRING_LIST
  (LENGTH : CHOICE_SMALL_NATURAL := 0) is
    record
      LIST : CHOICE_PROMPT_STRING_ARRAY (1..LENGTH);
    end record;

-- Provides for lists of prompt strings.

-- CHOICE_REQUEST_STATUS

type CHOICE_REQUEST_STATUS is ( OK,
                                NOCHOICE,
                                NONE);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
-- Defines the status of a choice input operation for the
-- request function.

-- CHOICE_STATUS

subtype CHOICE_STATUS is CHOICE_REQUEST_STATUS range OK..NOCHOICE;

-- Indicates if a choice was made by the operator for the
-- sample, get, and inquiry functions.

-- CLIPPING_INDICATOR

type CLIPPING_INDICATOR is ( CLIP,
                             NOCLIP);

-- Indicates whether or not clipping is to be performed.


-- WEIGHT_LIST_ARRAY

type WEIGHT_LIST_ARRAY is
      array (COLOUR_COMPONENTS range <>) of COLOUR_COEFFICIENT;

-- Provides for a list of weights to be applied during colour mapping.


-- WEIGHT_LIST

type WEIGHT_LIST
      (LENGTH : COLOUR_COMPONENTS := MAX_COLOUR_COEFFICIENTS) is
  record
    WEIGHTS : WEIGHT_LIST_ARRAY(1..LENGTH);
  end record;

-- Provides for a variable length list of weights to be applied during colour mapping.

type ACCESS_WEIGHT_LIST is access WEIGHT_LIST;

-- Provides for pointers to lists of weights.

-- COLOUR_MAPPING_METHOD

type COLOUR_MAPPING_METHOD is new PHIGS_INTEGER;

-- Provides for specifying different methods for performing colour mapping.
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567369 073 ■


```
TRUE_COLOUR_MAPPING          : constant COLOUR_MAPPING_METHOD := 1;
PSEUDO_COLOUR_MAPPING        : constant COLOUR_MAPPING_METHOD := 2;
PSEUDO_N_COLOUR_MAPPING      : constant COLOUR_MAPPING_METHOD := 3;

    -- COLOUR_MAPPING_METHODS

    package COLOUR_MAPPING_METHODS is
            new PHIGS_LIST_UTILITIES
                    (COLOUR_MAPPING_METHOD,
                     MAX_COLOUR_MAPPING_METHODS_SUPPORTED);

    -- Provides for lists of colour mapping methods.

    -- COLOUR_MAPPING_DATA_RECORD

    type COLOUR_MAPPING_DATA_RECORD
      (METHOD : COLOUR_MAPPING_METHOD) is
        record
          case METHOD is

            when TRUE_COLOUR_MAPPING =>
              null;

            when PSEUDO_COLOUR_MAPPING =>
              PSEUDO_COLOUR_MODEL  : COLOUR_MODEL;
              PSEUDO_WEIGHTS       : ACCESS_WEIGHT_LIST;
              PSEUDO_COLOURS       : ACCESS_COLOUR_VALUE_LIST;

            when PSEUDO_N_COLOUR_MAPPING =>
              PSEUDO_N_COLOUR_MODEL  : COLOUR_MODEL;
              PSEUDO_N_COLOURS       : ACCESS_LIST_OF_COLOUR_VALUE_LIST;

            when others =>
              null;

          end case;
        end record;

    -- Provides for specifying values for different methods of colour mapping.
    -- Additional variants may be included when additional registered or unregistered colour
    -- mapping methods are supported by an implementation.

    -- ACCESS_COLOUR_MAPPING_DATA_RECORD

    type ACCESS_COLOUR_MAPPING_DATA_RECORD is
            access COLOUR_MAPPING_DATA_RECORD;

    -- Provides for pointers to colour mapping data records.
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
-- COLOUR_MAPPING_INDEX

type COLOUR_MAPPING_INDEX is new PHIGS_NATURAL;

-- Provides for index values for colour mapping representations.

-- COLOUR_MAPPING_INDICES

package COLOUR_MAPPING_INDICES is
        new PHIGS_LIST_UTILITIES
                (COLOUR_MAPPING_INDEX,
                 MAX_COLOUR_MAPPING_INDICES_SUPPORTED);

-- Provides for lists of index values for colour mapping representations.

-- COLOUR_MAPPING_METHOD_FACILITIES

type COLOUR_MAPPING_METHOD_FACILITIES
  (METHOD : COLOUR_MAPPING_METHOD := TRUE_COLOUR_MAPPING) is
    record
      case METHOD is

        when TRUE_COLOUR_MAPPING =>
          NUMBER_TRUE_COLOURS_AVAIL : PHIGS_INTEGER;

        when PSEUDO_COLOUR_MAPPING =>
          NUMBER_PSEUDO_COLORS_AVAIL : PHIGS_INTEGER;

        when PSEUDO_N_COLOUR_MAPPING =>
          null;

        when others =>
          null;

      end case;
    end record;

-- Provides for colour mapping availability information. Additional variants may be included
-- when additional registered or unregistered colour mapping methods are supported by an
-- implementation.

-- COLOUR_MAPPING_STATE

type COLOUR_MAPPING_STATE
  (METHOD : COLOUR_MAPPING_METHOD := TRUE_COLOUR_MAPPING) is
    record
      case METHOD is
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567371 721 ■


```
            when TRUE_COLOUR_MAPPING =>
              null;

            when PSEUDO_COLOUR_MAPPING =>
              null;

            when PSEUDO_N_COLOUR_MAPPING =>
              null;

            when others =>
              null;

          end case;
        end record;

-- Provides for colour mapping state information.  Additional variants may be included
-- when additional registered or unregistered colour mapping methods are supported by an
-- implementation.

-- COMPOSITION_TYPE

type COMPOSITION_TYPE is (  PRECONCATENATE,
                            POSTCONCATENATE,
                            REPLACE);

-- Defines the type of matrix composition allowed between
-- modeling transformations.

-- CONFLICT_RESOLUTION

type CONFLICT_RESOLUTION is ( MAINTAIN,
                              ABANDON,
                              UPDATE);

-- Defines a type for specifying the conflict resolution mode.

-- CONNECTION_ID

type CONNECTION_ID is new PHIGS_STRING;

--  Provides for specifying connection identifiers.

-- CONTROL_FLAG

type CONTROL_FLAG is (  CONDITIONALLY,
                        ALWAYS);

-- The control flag is used to indicate the conditions under
-- which the display surface should be cleared.
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567372 668 ■


```
        -- CURVE_APPROX_CRITERIA_TYPE

        type CURVE_APPROX_CRITERIA_TYPE is new PHIGS_INTEGER;

        -- Provides for differentiating types of curve approximation.

        -- CURVE_APPROX_CRITERIA_TYPES

        package CURVE_APPROX_CRITERIA_TYPES is
                new PHIGS_LIST_UTILITIES
                        (CURVE_APPROX_CRITERIA_TYPE,
                         MAX_CURVE_APPROX_CRITERIA_TYPES_SUPPORTED);

        -- Provides for lists of curve approximation criteria.

WS_DEPENDENT_CURVE          : constant CURVE_APPROX_CRITERIA_TYPE := 1;
CONSTANT_SUBDIVISION_CURVE  : constant CURVE_APPROX_CRITERIA_TYPE := 2;
WC_CHORDAL_SIZE_CURVE       : constant CURVE_APPROX_CRITERIA_TYPE := 3;
NPC_CHORDAL_SIZE_CURVE      : constant CURVE_APPROX_CRITERIA_TYPE := 4;
DC_CHORDAL_SIZE_CURVE       : constant CURVE_APPROX_CRITERIA_TYPE := 5;
WC_CHORDAL_DEVIATION_CURVE  : constant CURVE_APPROX_CRITERIA_TYPE := 6;
NPC_CHORDAL_DEVIATION_CURVE : constant CURVE_APPROX_CRITERIA_TYPE := 7;
DC_CHORDAL_DEVIATION_CURVE  : constant CURVE_APPROX_CRITERIA_TYPE := 8;
WC_RELATIVE_CURVE           : constant CURVE_APPROX_CRITERIA_TYPE := 9;
NPC_RELATIVE_CURVE          : constant CURVE_APPROX_CRITERIA_TYPE := 10;
DC_RELATIVE_CURVE           : constant CURVE_APPROX_CRITERIA_TYPE := 11;

        -- CURVE_APPROXIMATION_DATA_RECORD

        type CURVE_APPROX_DATA_RECORD
          (CRITERIA_TYPE : CURVE_APPROX_CRITERIA_TYPE := WS_DEPENDENT_CURVE) is
            record
              case CRITERIA_TYPE is

                when WS_DEPENDENT_CURVE =>
                  null;

                when CONSTANT_SUBDIVISION_CURVE =>
                  COUNT : PHIGS_INTEGER;

                when WC_CHORDAL_SIZE_CURVE =>
                  WC_CHORDAL_SIZE_VALUE : WC.MAGNITUDE;

                when NPC_CHORDAL_SIZE_CURVE =>
                  NPC_CHORDAL_SIZE_VALUE : NPC.MAGNITUDE;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
            when DC_CHORDAL_SIZE_CURVE =>
              DC_CHORDAL_SIZE_VALUE : DC.MAGNITUDE;

            when WC_CHORDAL_DEVIATION_CURVE =>
              WC_CHORDAL_DEVIATION_VALUE : WC.MAGNITUDE;

            when NPC_CHORDAL_DEVIATION_CURVE =>
              NPC_CHORDAL_DEVIATION_VALUE : NPC.MAGNITUDE;

            when DC_CHORDAL_DEVIATION_CURVE =>
              DC_CHORDAL_DEVIATION_VALUE : DC.MAGNITUDE;

            when WC_RELATIVE_CURVE =>
              WC_RELATIVE_VALUE : WC.RELATIVE_MAGNITUDE;

            when NPC_RELATIVE_CURVE =>
              NPC_RELATIVE_VALUE : NPC.RELATIVE_MAGNITUDE;

            when DC_RELATIVE_CURVE =>
              DC_RELATIVE_VALUE : DC.RELATIVE_MAGNITUDE;

            when others =>
              null;

          end case;
        end record;

-- Provides for specifying values for different methods of curve approximation.
-- Additional variants may be included when additional registered or unregistered curve
-- approximation criteria types are supported by an implementation.


-- CURVE_PLACEMENT_TYPE

type CURVE_PLACEMENT_TYPE is (UNIFORM, NON_UNIFORM);

-- Provides for different ways of placing curves.

-- DC_UNITS

type DC_UNITS is ( METRES,
                   OTHER);

-- Device coordinate units for a particular workstation
-- may be in metres, or some other unit (such as inches).

-- DEFERRAL_MODE

type DEFERRAL_MODE is (  ASAP,
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
                    BNIG,
                    BNIL,
                    ASTI,
                    WAIT);

-- Defines the five PHIGS deferral modes.

-- DEPTH_CUE_INDEX

type DEPTH_CUE_INDEX is new PHIGS_NATURAL;

-- Provides for index values for depth cue representations.


-- DEPTH_CUE_INDICES

package DEPTH_CUE_INDICES is
        new PHIGS_LIST_UTILITIES
                (DEPTH_CUE_INDEX,
                 MAX_DEPTH_CUE_INDICES_SUPPORTED);

-- Provides for lists of depth cue indices.

-- DEPTH_CUE_MODE

type DEPTH_CUE_MODE is (SUPPRESSED, ALLOWED);

-- Provides for activation and deactivation of depth cue operations.

--DEPTH_CUE_MODES

package DEPTH_CUE_MODES is
        new PHIGS_LIST_UTILITIES (DEPTH_CUE_MODE, 2);

-- Provides for lists of depth cue indices.


-- DEPTH_CUE_SCALE_FACTOR

subtype DEPTH_CUE_SCALE_FACTOR is SCALE_FACTOR range 0.0 .. 1.0;

-- Provides for specifying scale factors for depth cue operations.


-- DEPTH_CUE_SCALE_FACTORS
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567375 377 ■


```
type DEPTH_CUE_SCALE_FACTORS is
  record
    FRONT : DEPTH_CUE_SCALE_FACTOR;
    BACK  : DEPTH_CUE_SCALE_FACTOR;
  end record;

-- Provides for specifying a pair of scale factors for apportioning colours during depth cue
-- operations.


-- RETURN_DEPTH_CUE_INDEX_COUNT

subtype RETURN_DEPTH_CUE_INDEX_COUNT is
          PHIGS_POSITIVE range 2..MAX_DEPTH_CUE_INDICES_SUPPORTED;

-- Provides for restricting the returned count of predefined depth cue indices.


-- DISPLAY_CLASS

type DISPLAY_CLASS is ( VECTOR_DISPLAY,
                        RASTER_DISPLAY,
                        OTHER_DISPLAY);

-- The classification of a workstation of category OUTPUT
-- or OUTIN.

-- DISPLAY_SURFACE_EMPTY

type DISPLAY_SURFACE_EMPTY is ( EMPTY,
                                NOTEMPTY);

-- Indicates whether there is graphical output on the display
-- surface.

-- DYNAMIC_MODIFICATION

type DYNAMIC_MODIFICATION is ( IRG,
                               IMM,
                               CBS);

-- Indicates whether an update to the state list is performed
-- immediately (IMM), requires implicit regeneration (IRG), or
-- can be simulated (CBS).

-- ECHO_SWITCH

type ECHO_SWITCH is ( NOECHO,
                      ECHO);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
-- Indicates whether or not echoing of the measure is
-- performed.

-- EDIT_MODE

type EDIT_MODE is ( INSERT,
                    REPLACE);

-- Defines a type for specifying the mode in which editing
-- takes place.

-- ELEMENT_POSITION

type ELEMENT_POSITION is new PHIGS_INTEGER;

-- Base type for element pointer values and offsets.

-- RETURNED_ELEMENT_POSITION

subtype RETURNED_ELEMENT_POSITION is
  ELEMENT_POSITION range 0..ELEMENT_POSITION'LAST;

-- Defines a type for returning element pointer values.

-- ELEMENT_TYPE

type ELEMENT_TYPE is
       (ALL_ELEMENT_TYPES,
        NIL,

        POLYLINE_3,
        POLYLINE,

        POLYMARKER_3,
        POLYMARKER,

        TEXT_3,
        TEXT,

        ANNOTATION_TEXT_RELATIVE_3,
        ANNOTATION_TEXT_RELATIVE,

        FILL_AREA_3,
        FILL_AREA,
        FILL_AREA_SET_3,
        FILL_AREA_SET,
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567377 14T ■

CELL_ARRAY_3,
CELL_ARRAY,

GDP_3,
GDP,

SET_POLYLINE_INDEX,
SET_POLYMARKER_INDEX,
SET_TEXT_INDEX,
SET_INTERIOR_INDEX,
SET_EDGE_INDEX,

SET_LINETYPE,
SET_LINEWIDTH_SCALE_FACTOR,
SET_POLYLINE_COLOUR_INDEX,

SET_MARKER_TYPE,
SET_MARKER_SIZE_SCALE_FACTOR,
SET_POLYMARKER_COLOUR_INDEX,

SET_TEXT_FONT,
SET_TEXT_PRECISION,
SET_CHAR_EXPANSION_FACTOR,
SET_CHAR_SPACING,
SET_TEXT_COLOUR_INDEX,
SET_CHAR_HEIGHT,
SET_CHAR_UP_VECTOR,
SET_TEXT_PATH,
SET_TEXT_ALIGNMENT,

SET_ANNOTATION_TEXT_CHAR_HEIGHT,
SET_ANNOTATION_TEXT_CHAR_UP_VECTOR,
SET_ANNOTATION_TEXT_PATH,
SET_ANNOTATION_TEXT_ALIGNMENT,
SET_ANNOTATION_STYLE,

SET_INTERIOR_STYLE,
SET_INTERIOR_STYLE_INDEX,
SET_INTERIOR_COLOUR_INDEX,

SET_EDGE_FLAG,
SET_EDGETYPE,
SET_EDGEWIDTH_SCALE_FACTOR,
SET_EDGE_COLOUR_INDEX,

SET_PATTERN_SIZE,
SET_PATTERN_REFERENCE_POINT_AND_VECTORS,
SET_PATTERN_REFERENCE_POINT,

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567378 086 ■

ADD_NAMES_TO_SET,
REMOVE_NAMES_FROM_SET,

SET_INDIVIDUAL_ASF,
SET_HLHSR_IDENTIFIER,

SET_LOCAL_TRANSFORMATION_3,
SET_LOCAL_TRANSFORMATION,
SET_GLOBAL_TRANSFORMATION_3,
SET_GLOBAL_TRANSFORMATION,
SET_MODELLING_CLIPPING_VOLUME_3,
SET_MODELLING_CLIPPING_VOLUME,
SET_MODELLING_CLIPPING_INDICATOR,
RESTORE_MODELLING_CLIPPING_VOLUME,
SET_VIEW_INDEX,

EXECUTE_STRUCTURE,

LABEL,
APPLICATION_DATA,
GSE,
SET_PICK_IDENTIFIER,

POLYLINE_SET_3_WITH_COLOUR,
FILL_AREA_SET_3_WITH_DATA,
FILL_AREA_SET_WITH_DATA,
CELL_ARRAY_3_PLUS,
SET_OF_FILL_AREA_SETS_3_WITH_DATA,
SET_OF_FILL_AREA_SETS_WITH_DATA,
TRIANGLE_SET_3_WITH_DATA,
TRIANGLE_SET_WITH_DATA,
TRIANGLE_STRIP_3_WITH_DATA,
TRIANGLE_STRIP_WITH_DATA,
QUADRILATERAL_MESH_3_WITH_DATA,
QUADRILATERAL_MESH_WITH_DATA,
NON_UNIFORM_B_SPLINE_CURVE,
NON_UNIFORM_B_SPLINE_CURVE_WITH_COLOUR,
NON_UNIFORM_B_SPLINE_SURFACE,
NON_UNIFORM_B_SPLINE_SURFACE_WITH_DATA,

SET_DATA_MAPPING_INDEX,
SET_REFLECTANCE_INDEX,

SET_BACK_INTERIOR_INDEX,
SET_BACK_DATA_MAPPING_INDEX,
SET_BACK_REFLECTANCE_INDEX,

SET_PARAMETRIC_SURFACE_INDEX,

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567379 T12 ■


```
                SET_POLYLINE_COLOUR,
                SET_POLYLINE_SHADING_METHOD,

                SET_POLYMARKER_COLOUR,

                SET_TEXT_COLOUR,

                SET_FACET_DISTINGUISHING_MODE,
                SET_FACET_CULLING_MODE,

                SET_INTERIOR_COLOUR,
                SET_INTERIOR_SHADING_METHOD,

                SET_DATA_MAPPING_METHOD,
                SET_REFLECTANCE_PROPERTIES,
                SET_REFLECTANCE_MODEL,

                SET_BACK_INTERIOR_STYLE,
                SET_BACK_INTERIOR_STYLE_INDEX,
                SET_BACK_INTERIOR_COLOUR,
                SET_BACK_INTERIOR_SHADING_METHOD,
                SET_BACK_DATA_MAPPING_METHOD,
                SET_BACK_REFLECTANCE_PROPERTIES,
                SET_BACK_REFLECTANCE_MODEL,

                SET_LIGHT_SOURCE_STATE,

                SET_EDGE_COLOUR,

                SET_CURVE_APPROX_CRITERIA,
                SET_SURFACE_APPROX_CRITERIA,
                SET_PARAMETRIC_SURFACE_CHARACTERISTICS,

                SET_RENDERING_COLOUR_MODEL,
                SET_DEPTH_CUE_INDEX,
                SET_COLOUR_MAPPING_INDEX);

        -- This type lists the element types which exist in PHIGS PLUS.


        -- ELEMENT_TYPES

        package ELEMENT_TYPES is
          new PHIGS_LIST_UTILITIES ( ELEMENT_TYPE,
                                     ELEMENT_TYPE'POS(ELEMENT_TYPE'LAST)+1);

        -- Provides for lists of element types.

        -- ERROR_HANDLING_MODE
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567380 734 ■


```
type ERROR_HANDLING_MODE is new OFF_ON;

-- Type for inquiring the error handling mode.

-- ERROR_NUMBER

type ERROR_NUMBER is new PHIGS_INTEGER;

-- Defines the type for error indicator values.

-- ESCAPE_ID

type ESCAPE_ID is new PHIGS_INTEGER;

-- Defines a type for identifying different escapes.

-- ESCAPE_IDS

package ESCAPE_IDS is
  new PHIGS_LIST_UTILITIES ( ESCAPE_ID,
                             MAX_ESCAPE_IDS_SUPPORTED);

-- Provides for lists of Escape ID's.

-- ESCAPE_DATA_RECORD

type ESCAPE_DATA_RECORD (ESCAPE : ESCAPE_ID := 0) is
  record
    case ESCAPE is

      -- For each ESCAPE which needs a record, an implementation
      -- dependent entry is defined.  For others, a null record
      -- is defined.

      when others =>
        null;
    end case;
  end record;

-- Provides for the definition of ESCAPE data records.

-- FACET_CULLING_MODE

type FACET_CULLING_MODE is ( NO_FACET_CULLING,
                             BACKFACING,
                             FRONTFACING);

-- Provides for indicating the type of culling to be applied to facets.
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
-- FACET_DISTINGUISHING_MODE

type FACET_DISTINGUISHING_MODE is new OFF_ON;

-- Provides for enabling and disabling facet distinguishing operations.


-- FILE_ID

subtype FILE_ID is PHIGS_STRING;

-- Provides for file identifiers.

-- VARIABLE_FILE_ID

type VARIABLE_FILE_ID (LENGTH : FILE_SMALL_NATURAL := 0) is
  record
    NAME : FILE_ID (1..LENGTH);
  end record;

-- Provides for variable length file identifiers.

-- VARIABLE_FILE_IDS

package VARIABLE_FILE_IDS is
  new PHIGS_LIST_UTILITIES ( VARIABLE_FILE_ID,
                             MAX_FILE_IDS_SUPPORTED);

-- This type is used to define lists of file identifiers.

-- GDP_3_ID

type GDP_3_ID is new PHIGS_INTEGER;

-- Defines a type for selecting a Generalized Drawing Primitive.

-- GDP_3_IDS

package GDP_3_IDS is
  new PHIGS_LIST_UTILITIES ( GDP_3_ID,
                             MAX_GDP_3_IDS_SUPPORTED);

-- Provides for lists of 3D Generalized Drawing Primitive ID's.

-- GDP_3_RECORD

type GDP_3_RECORD (GDP : GDP_3_ID := 0) is
  record
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567382 507 ■


```
      case GDP is

        -- For each supported GDP_3 which needs a record, an implementation
        -- dependent entry is defined.  For others, a generic record
        -- is defined.

        when others =>
          GDP_INFO : APPLICATION_DATA_RECORD;
      end case;
    end record;

-- Provides for the definition of GDP_3 data records.

-- GDP_ID

type GDP_ID is new PHIGS_INTEGER;

-- Defines a type for selecting a Generalized Drawing Primitive.

-- GDP_IDS

package GDP_IDS is
  new PHIGS_LIST_UTILITIES ( GDP_ID,
                             MAX_GDP_IDS_SUPPORTED);

-- Provides for lists of Generalized Drawing Primitive ID's.

-- GDP_RECORD

type GDP_RECORD (GDP : GDP_ID := 0) is
  record
    case GDP is

      -- For each supported GDP which needs a record, an implementation
      -- dependent entry is defined.  For others, a generic record
      -- is defined.

      when others =>
        GDP_INFO : APPLICATION_DATA_RECORD;
    end case;
  end record;

-- Provides for the definition of GDP data records.

-- GSE_ID

type GSE_ID is new PHIGS_INTEGER;

--  Defines a type for selecting a Generalized Structure Element.
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
-- GSE_IDS

package GSE_IDS is new PHIGS_LIST_UTILITIES ( GSE_ID,
                                              MAX_GSE_IDS_SUPPORTED);

-- Provides for lists of Generalized Structure Element ID's.

-- GSE_RECORD

type GSE_RECORD (GSE : GSE_ID := 0) is
  record
    case GSE is

      -- For each supported GSE which needs a record, an implementation
      -- dependent entry is defined. For others, a generic record
      -- is defined.

      when others =>
        GSE_INFO : APPLICATION_DATA_RECORD;
    end case;
  end record;

-- Provides for the definition of GSE data records.

-- STYLE_INDEX

type STYLE_INDEX is new PHIGS_INTEGER;

-- Defines an interior style index.

-- HATCH_STYLE

subtype HATCH_STYLE is STYLE_INDEX;

-- Defines the interior hatch styles type.

-- HATCH_STYLES

package HATCH_STYLES is
  new PHIGS_LIST_UTILITIES ( HATCH_STYLE,
                             MAX_HATCH_STYLES_SUPPORTED);

-- Provides for lists of hatch styles.

-- HLHSR_ID

type HLHSR_ID is new PHIGS_INTEGER;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567384 38T ■


```
-- A number used as an attribute for workstation independent
-- HLHSR factors.

-- HLHSR_IDS

package HLHSR_IDS is new PHIGS_LIST_UTILITIES ( HLHSR_ID,
                                                MAX_HLHSR_IDS_SUPPORTED);

-- Provides for lists of HLHSR identifiers.

-- HLHSR_MODE

type HLHSR_MODE is new PHIGS_INTEGER;

-- A number used for controlling workstation dependent
-- HLHSR factors.

-- HLHSR_MODES

package HLHSR_MODES is
  new PHIGS_LIST_UTILITIES ( HLHSR_MODE,
                             MAX_HLHSR_MODES_SUPPORTED);

-- Provides for lists of HLHSR modes.

-- HORIZONTAL_ALIGNMENT

type HORIZONTAL_ALIGNMENT is ( NORMAL,
                               LEFT,
                               CENTRE,
                               RIGHT);

-- The alignment of the text extent rectangle with respect to
-- horizontal positioning of the text.

-- INPUT_CLASS

type INPUT_CLASS is ( NONE,
                      LOCATOR_INPUT,
                      STROKE_INPUT,
                      VALUATOR_INPUT,
                      CHOICE_INPUT,
                      PICK_INPUT,
                      STRING_INPUT);

-- Defines the input device classifications for workstations
-- of category INPUT or OUTIN.

-- INPUT_QUEUE_CLASS
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567385 216 ■


```
subtype INPUT_QUEUE_CLASS is
  INPUT_CLASS range LOCATOR_INPUT .. STRING_INPUT;

-- Defines the input device classifications for workstations
-- of category INPUT and OUTIN that do not return a NONE
-- classification.

-- INPUT_STATUS

type INPUT_STATUS is ( OK,
                       NONE);

-- Defines the status of a locator, stroke, valuator, or
-- string operation.

-- INPUT_STRING

type INPUT_STRING (LENGTH : INPUT_STRING_SMALL_NATURAL := 0) is
  record
    CONTENTS : PHIGS_STRING (1..LENGTH);
  end record;

-- Provides a variable length input string.  Objects of this type
-- should be declared unconstrained to allow for dynamic modification
-- of the length.

-- INTERIOR_INDEX

type INTERIOR_INDEX is new PHIGS_POSITIVE;

-- Defines interior bundle table indices.

-- INTERIOR_INDICES

package INTERIOR_INDICES is
  new PHIGS_LIST_UTILITIES ( INTERIOR_INDEX,
                             MAX_INTERIOR_INDICES_SUPPORTED);

-- Provides for lists of interior bundle table indices.

-- INTERIOR_SHADING_METHOD

type INTERIOR_SHADING_METHOD is new PHIGS_INTEGER;

-- Provides for differentiating methods of shading interiors of enclosed regions.


-- INTERIOR_SHADING_METHODS
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567386 152 ■


```
package INTERIOR_SHADING_METHODS is
        new PHIGS_LIST_UTILITIES
                (INTERIOR_SHADING_METHOD,
                 MAX_INTERIOR_SHADING_METHODS_SUPPORTED);

-- Provides for lists of methods of performing interior shading.

NO_INTERIOR_SHADING                 : constant INTERIOR_SHADING_METHOD := 1;
INTERIOR_COLOUR_SHADING             : constant INTERIOR_SHADING_METHOD := 2;
INTERIOR_DATA_SHADING               : constant INTERIOR_SHADING_METHOD := 3;
INTERIOR_DATA_AND_DOT_SHADING       : constant INTERIOR_SHADING_METHOD := 4;
INTERIOR_DATA_AND_NORMAL_SHADING    : constant INTERIOR_SHADING_METHOD := 5;

-- INTERIOR_STYLE

type INTERIOR_STYLE is ( HOLLOW,
                         SOLID,
                         PATTERN,
                         HATCH,
                         EMPTY);

-- Defines the interior styles for filled areas.

-- INTERIOR_STYLES

package INTERIOR_STYLES is
  new PHIGS_LIST_UTILITIES ( INTERIOR_STYLE,
                             INTERIOR_STYLE'POS(INTERIOR_STYLE'LAST)+1);

-- Provides for lists of interior styles.

-- INTERIOR_DATA

type INTERIOR_DATA is
  record
    STYLE_OF_INTERIOR_ASF  : ASF;
    STYLE_IND_ASF          : ASF;
    INTERIOR_COLOUR_ASF    : ASF;
    INTERIOR_IND           : INTERIOR_INDEX;
    STYLE_OF_INTERIOR      : INTERIOR_STYLE;
    STYLE_IND              : STYLE_INDEX;
    INTERIOR_COLOUR        : COLOUR_INDEX;
  end record;

-- A record containing information needed to specify the
-- appearance of a filled area interior.

-- INVALID_VALUES_INDICATOR
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567387 099 ■


```
type INVALID_VALUES_INDICATOR is ( ABSENT,
                                   PRESENT);

-- Indicates whether invalid values are contained in a colour
-- array or matrix.

-- KNOT

subtype KNOT is SPC.MAGNITUDE;

-- Provides for specifying spline knots.


-- LABEL_ID

type LABEL_ID is new PHIGS_INTEGER;

-- Defines a type for specifying structure element labels.

-- LIGHTING_EXPONENT

type LIGHTING_EXPONENT is digits PHIGS_PRECISION;

-- Provides for specifying values for specular lighting exponents.


-- SPREAD_ANGLE

subtype SPREAD_ANGLE is ANGLE range 0.0 .. PHIGS_PI;

-- Provides for specifying the amount of spread of spot light sources.


-- LIGHT_SOURCE_TYPE

type LIGHT_SOURCE_TYPE is new PHIGS_INTEGER;

-- Provides for specifying kinds of light sources.


-- LIGHT_SOURCE_TYPES

package LIGHT_SOURCE_TYPES is
      new PHIGS_LIST_UTILITIES ( LIGHT_SOURCE_TYPE,
                                 MAX_LIGHT_SOURCE_TYPES_SUPPORTED);

-- Provides for lists of light source types.
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
AMBIENT_LIGHT                : constant LIGHT_SOURCE_TYPE := 1;
DIRECTIONAL_LIGHT            : constant LIGHT_SOURCE_TYPE := 2;
POSITIONAL_LIGHT             : constant LIGHT_SOURCE_TYPE := 3;
SPOT_LIGHT                   : constant LIGHT_SOURCE_TYPE := 4;

    -- LIGHT_SOURCE_DATA_RECORD

    type LIGHT_SOURCE_DATA_RECORD
      (TYPE_OF_LIGHT : LIGHT_SOURCE_TYPE := AMBIENT_LIGHT) is
        record
          case TYPE_OF_LIGHT is

            when AMBIENT_LIGHT =>
              AMBIENT_COLOUR          : GENERAL_COLOUR;

            when DIRECTIONAL_LIGHT =>
              DIRECTIONAL_COLOUR      : GENERAL_COLOUR;
              DIRECTIONAL_DIRECTION   : WC.VECTOR_3;

            when POSITIONAL_LIGHT =>
              POSITIONAL_COLOUR       : GENERAL_COLOUR;
              POSITIONAL_POSITION     : WC.POINT_3;
              POSITIONAL_ATTENUATION  : ATTENUATION_COEFFICIENTS;

            when SPOT_LIGHT =>
              SPOT_COLOUR             : GENERAL_COLOUR;
              SPOT_POSITION           : WC.POINT_3;
              SPOT_DIRECTION          : WC.VECTOR_3;
              SPOT_CONCENTRATION      : LIGHTING_EXPONENT;
              SPOT_ATTENUATION        : ATTENUATION_COEFFICIENTS;
              SPOT_SPREAD             : SPREAD_ANGLE;

            when others =>
              null;

          end case;
        end record;

    -- Provides for descriptions of light sources.  Additional variants may be included when
    -- additional registered or unregistered light source types are supported by an
    -- implementation.


    -- LIGHT_SOURCE_INDEX

    type LIGHT_SOURCE_INDEX is new PHIGS_POSITIVE;

    -- Provides for identifying light sources.
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567389 961 ■


```
-- LIGHT_SOURCE_INDICES

package LIGHT_SOURCE_INDICES is
      new PHIGS_LIST_UTILITIES ( LIGHT_SOURCE_INDEX,
                                 MAX_LIGHT_SOURCE_INDICES_SUPPORTED);

-- Provides for lists of light sources.

-- RETURN_LIGHT_SOURCE_COUNT

subtype RETURN_LIGHT_SOURCE_COUNT is
          PHIGS_POSITIVE range 2..PHIGS_POSITIVE'last;

-- Provides for restricting spline orders to those acceptable for returning values of the
-- facilities.

-- POLYLINE_INDEX

type POLYLINE_INDEX is new PHIGS_POSITIVE;

-- Defines the range of polyline indices.

-- POLYLINE_INDICES

package POLYLINE_INDICES is
  new PHIGS_LIST_UTILITIES ( POLYLINE_INDEX,
                             MAX_POLYLINE_INDICES_SUPPORTED);

-- Provides for lists of polyline indices.

-- LINETYPE

type LINETYPE is new PHIGS_INTEGER;

-- Defines the types of line styles provided by PHIGS.

-- LINETYPES

package LINETYPES is new PHIGS_LIST_UTILITIES (  LINETYPE,
                                                 MAX_LINETYPES_SUPPORTED);

-- Provides for lists of linetypes.

-- LINEWIDTH

type LINEWIDTH is new SCALE_FACTOR range 0.0..SCALE_FACTOR'LAST;

-- The width of a line is indicated by a scale factor.
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
-- LINE_DATA

type LINE_DATA is
  record
    TYPE_OF_LINE_ASF  : ASF;
    WIDTH_ASF         : ASF;
    LINE_COLOUR_ASF   : ASF;
    POLYLINE_IND      : POLYLINE_INDEX;
    TYPE_OF_LINE      : LINETYPE;
    WIDTH             : LINEWIDTH;
    LINE_COLOUR       : COLOUR_INDEX;
  end record;

-- A record containing information needed to specify the
-- appearance of a line for input data records.

-- EDGE_FLAG

type EDGE_FLAG is new OFF_ON;

-- Defines a type for indicating whether or not to depict edges
-- on fill area set and related primitives.

-- CURVE VISIBILITY FLAG

subtype CURVE_VISIBILITY_FLAG is EDGE_FLAG;

-- Provides control over display of curves on spline surfaces


-- EDGE_FLAG_LIST

type EDGE_FLAG_LIST is array (POSITIVE range <>) of EDGE_FLAG;

-- Provides for arrays of edge flags.  These will be used with various "with Data" primitives.


-- ACCESS_EDGE_FLAG_LIST

type ACCESS_EDGE_FLAG_LIST is access EDGE_FLAG_LIST;

-- Provides for pointers to lists of edge flags.


-- LIST_OF_EDGE_FLAG_LIST

type LIST_OF_EDGE_FLAG_LIST is
  array (POSITIVE range <>) of ACCESS_EDGE_FLAG_LIST;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
-- Provides for pointers to lists of lists of edge flags.

-- ACCESS_LIST_OF_EDGE_FLAG_LIST

type ACCESS_LIST_OF_EDGE_FLAG_LIST is access LIST_OF_EDGE_FLAG_LIST;

-- Provides for pointers to lists of lists of edge flags.

-- LIST_OF_LIST_OF_EDGE_FLAG_LIST

type LIST_OF_LIST_OF_EDGE_FLAG_LIST is
  array (POSITIVE range <>) of ACCESS_LIST_OF_EDGE_FLAG_LIST;

-- Provides for pointers to lists of lists of edge flags.

-- ACCESS_LIST_OF_LIST_OF_EDGE_FLAG_LIST

type ACCESS_LIST_OF_LIST_OF_EDGE_FLAG_LIST is
        access LIST_OF_LIST_OF_EDGE_FLAG_LIST;

-- Provides for pointers to lists of lists of edge flag lists.

-- EDGE_FLAG_PAIR

type EDGE_FLAG_PAIR is array (1..2) of EDGE_FLAG;

-- Provides for pairs of edge flags.  These will be used with "Quadrilateral Mesh with Data"
-- primitives.

-- ARRAY_OF_EDGE_FLAG_PAIR

type ARRAY_OF_EDGE_FLAG_PAIR is
        array ( POSITIVE range <>,
                POSITIVE range <>) of EDGE_FLAG_PAIR;

-- Provides for two-dimensional arrays of edge flag pairs.  These will be used with
-- "Quadrilateral Mesh with Data" primitives.

-- ACCESS_ARRAY_OF_EDGE_FLAG_PAIR
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
type ACCESS_ARRAY_OF_EDGE_FLAG_PAIR is access ARRAY_OF_EDGE_FLAG_PAIR;

-- Provides for access variable to two-dimensional arrays of edge flag pairs.  These will be
-- used when inquiring structure elements which are "Quadrilateral Mesh with Data"
-- primitives.


-- EDGE_FLAG_TRIPLET

type EDGE_FLAG_TRIPLET is array (1..3) of EDGE_FLAG;

-- Provides for triplets of edge flags.  These will be used with "Triangle Set with Data"
-- primitives.


-- LIST_OF_EDGE_FLAG_TRIPLET

type LIST_OF_EDGE_FLAG_TRIPLET is
  array (POSITIVE range <>) of EDGE_FLAG_TRIPLET;

-- Provides for lists of triplets of edge flags.


-- ACCESS_LIST_OF_EDGE_FLAG_TRIPLET

type ACCESS_LIST_OF_EDGE_FLAG_TRIPLET is
  access LIST_OF_EDGE_FLAG_TRIPLET;

-- Provides for pointers to lists of triplets of edge flags.


-- EDGE_INDEX

type EDGE_INDEX is new PHIGS_POSITIVE;

-- Defines the edge bundle table indices.

-- EDGE_INDICES

package EDGE_INDICES is
  new PHIGS_LIST_UTILITIES ( EDGE_INDEX,
                             MAX_EDGE_INDICES_SUPPORTED);

-- Provides for a list of edge indices.

-- EDGETYPE

type EDGETYPE is new PHIGS_INTEGER;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
-- Defines a type for describing the form of edges.

-- EDGETYPES

package EDGETYPES is
  new PHIGS_LIST_UTILITIES ( EDGETYPE,
                             MAX_EDGETYPES_SUPPORTED);

-- Provides for lists of edgetypes.

-- EDGEWIDTH

type EDGEWIDTH is new SCALE_FACTOR range 0.0..SCALE_FACTOR'LAST;

-- Defines a type for describing the width scale factor of edges.

-- EDGE_DATA

type EDGE_DATA is
  record
    FLAG_ASF            : ASF;
    TYPE_OF_EDGE_ASF    : ASF;
    EDGEWIDTH_SF_ASF    : ASF;
    EDGE_COLOUR_ASF     : ASF;
    FLAG                : EDGE_FLAG;
    TYPE_OF_EDGE        : EDGETYPE;
    EDGEWIDTH_SF        : EDGEWIDTH;
    EDGE_COLOUR         : COLOUR_INDEX;
  end record;

-- A record containing information needed to specify the
-- appearance of a fill area set edge.

-- LOCATOR_DEVICE_NUMBER

type LOCATOR_DEVICE_NUMBER is new DEVICE_NUMBER;

-- Provides for locator device identifiers.

-- LOCATOR_PROMPT_ECHO_TYPE

type LOCATOR_PROMPT_ECHO_TYPE is new PHIGS_INTEGER;

-- Defines the locator prompt and echo types supported by
-- the implementation.

-- LOCATOR_PROMPT_ECHO_TYPES

package LOCATOR_PROMPT_ECHO_TYPES is
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
    new PHIGS_LIST_UTILITIES
      (LOCATOR_PROMPT_ECHO_TYPE,
       MAX_LOCATOR_PROMPT_ECHO_TYPES_SUPPORTED);

-- Provides for lists of locator prompt and echo types.

-- POLYMARKER_INDEX

type POLYMARKER_INDEX is new PHIGS_POSITIVE;

-- Defines the range of polymarker bundle table indices.

-- POLYMARKER_INDICES

package POLYMARKER_INDICES is
  new PHIGS_LIST_UTILITIES ( POLYMARKER_INDEX,
                             MAX_POLYMARKER_INDICES_SUPPORTED);

-- Provides for lists of polymarker indices.

-- MARKER_SIZE

type MARKER_SIZE is new SCALE_FACTOR;

-- The size of a marker is indicated by a scale factor.

-- MARKER_TYPE

type MARKER_TYPE is new PHIGS_INTEGER;

-- Defines the type for markers provided by PHIGS.

-- MARKER_TYPES

package MARKER_TYPES is
  new PHIGS_LIST_UTILITIES ( MARKER_TYPE,
                             MAX_MARKER_TYPES_SUPPORTED);

-- Provides for lists of marker types.

-- MARKER_DATA

type MARKER_DATA is
  record
    TYPE_OF_MARKER_ASF  : ASF;
    SIZE_ASF            : ASF;
    MARKER_COLOUR_ASF   : ASF;
    POLYMARKER_IND      : POLYMARKER_INDEX;
    TYPE_OF_MARKER      : MARKER_TYPE;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567395 165 ■


```
    SIZE                    : MARKER_SIZE;
    MARKER_COLOUR           : COLOUR_INDEX;
  end record;

-- A record containing information needed to specify the
-- appearance of a marker.

-- MEMORY_UNITS

type MEMORY_UNITS is range 0..MAX_MEMORY_UNITS;

-- Defines the type for units of memory that may be allocated
-- by PHIGS.

-- METAFILE_ITEM_TYPE

type METAFILE_ITEM_TYPE is
  new PHIGS_NATURAL
    range MIN_METAFILE_ITEM_TYPE..MAX_METAFILE_ITEM_TYPE;

-- The type of an item contained in a metafile.

-- METAFILE_ITEM_LENGTH

type METAFILE_ITEM_LENGTH is
  new PHIGS_NATURAL range 0..MAX_METAFILE_ITEM_LENGTH;

-- The length of an item contained in a metafile.

-- MODELLING_CLIP_OPERATION_TYPE

type MODELLING_CLIP_OPERATION_TYPE is new PHIGS_INTEGER;

-- This type provides for specifying modelling clip operation
-- types.

-- MODELLING_CLIP_OPERATION_TYPES

package MODELLING_CLIP_OPERATION_TYPES is
  new PHIGS_LIST_UTILITIES
    (MODELLING_CLIP_OPERATION_TYPE,
     MAX_MODELLING_CLIP_OPERATION_TYPES_SUPPORTED);

-- This type provides for lists of modelling clip operation types.

-- MODIFICATION_MODE

type MODIFICATION_MODE is ( NIVE,
                            UWOR,
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567396 0T1 ■


```
                                        UQUM);

-- Defines type for specifying modification modes.

-- MORE_EVENTS

type MORE_EVENTS is ( NOMORE,
                      MORE);

-- Indicates whether more simultaneous events are contained in
-- the input queue.

-- NAME_SET

subtype NAME_SET is PHIGS_NAME_SET_FACILITY.NAME_SET;

-- Used to define name sets. The internal structure of
-- a name set is implementation-dependent. "Build" functions
-- contained in the package PHIGS_NAME_SET_FACILITIES shall be
-- used to create and manipulate name sets (See 5.14.4).

-- NAME_SET_ACCEPTANCE

subtype NAME_SET_ACCEPTANCE is
  PHIGS_NAME_SET_FACILITY.NAME_SET_ACCEPTANCE;

-- Used to indicate the results of applying a name set
-- against a filter pair. See 5.14.4.

-- NAME

subtype NAME is PHIGS_NAME_SET_FACILITY.NAME;

-- Provides for names for each of the name set classes.
-- See 5.14.4.

-- NAMES

package NAMES renames PHIGS_NAME_SET_FACILITY.NAMES;

-- Provides for lists of name set classes. See 5.14.4.

-- NAME_SET_FILTER

subtype NAME_SET_FILTER is PHIGS_NAME_SET_FACILITY.NAME_SET_FILTER;

-- Used to define name set pairs used as filters.
-- See 5.14.4.
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567397 T38 ■


```
-- NAME_SET_FILTERS

package NAME_SET_FILTERS is
  new PHIGS_LIST_UTILITIES
    (NAME_SET_FILTER,
     MAX_NAME_SET_FILTER_LIST_LENGTH_SUPPORTED);

-- Used to provide lists of name set filters.
-- See 5.14.4.

-- NAME_SET_MEMBERSHIP

subtype NAME_SET_MEMBERSHIP is
  PHIGS_NAME_SET_FACILITY.NAME_SET_MEMBERSHIP;

-- Provides for indicating whether a class is contained in a
-- name set. See 5.14.4.

-- OPEN_STRUCTURE_STATUS

type OPEN_STRUCTURE_STATUS is ( NONE,
                                OPEN);

-- Returns the status of the open structure.

-- OPERATING_MODE

type OPERATING_MODE is ( REQUEST_MODE,
                         SAMPLE_MODE,
                         EVENT_MODE);

-- Defines the operating modes of an input device.

-- PARAMETRIC_SURFACE_CHARACTERISTIC

type PARAMETRIC_SURFACE_CHARACTERISTIC is new PHIGS_INTEGER;

-- Provides for differentiating representations of parametric surfaces.


-- PARAMETRIC_SURFACE_CHARACTERISTICS

package PARAMETRIC_SURFACE_CHARACTERISTICS is
        new PHIGS_LIST_UTILITIES
              (PARAMETRIC_SURFACE_CHARACTERISTIC,
               MAX_PARAMETRIC_SURFACE_CHARACTERISTICS_SUPPORTED);

-- Provides for lists of parametric surface characteristics types.
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
NO_PARAMETRIC_SURFACE
                          : constant PARAMETRIC_SURFACE_CHARACTERISTIC := 1;
WS_DEPENDENT_PARAMETRIC_SURFACE
                          : constant PARAMETRIC_SURFACE_CHARACTERISTIC := 2;
ISOPARAMETRIC_CURVES      : constant PARAMETRIC_SURFACE_CHARACTERISTIC := 3;
MC_LEVEL_CURVES           : constant PARAMETRIC_SURFACE_CHARACTERISTIC := 4;
WC_LEVEL_CURVES           : constant PARAMETRIC_SURFACE_CHARACTERISTIC := 5;

    -- PARAMETRIC_SURFACE_DATA_RECORD

    type PARAMETRIC_SURFACE_DATA_RECORD
      (CHARACTERISTIC : PARAMETRIC_SURFACE_CHARACTERISTIC
                          := NO_PARAMETRIC_SURFACE) is
        record
          case CHARACTERISTIC is

            when NO_PARAMETRIC_SURFACE =>
              null;

            when WS_DEPENDENT_PARAMETRIC_SURFACE =>
              null;

            when ISOPARAMETRIC_CURVES =>
              CURVE_PLACEMENT : CURVE_PLACEMENT_TYPE;
              UCOUNT          : PHIGS_NATURAL;
              VCOUNT          : PHIGS_NATURAL;

            when MC_LEVEL_CURVES =>
              MC_LEVEL_ORIGIN      : MC.POINT_3;
              MC_LEVEL_DIRECTION   : MC.VECTOR_3;
              MC_LEVEL_PARAMETERS  : MC.MAGNITUDE_LIST;

            when WC_LEVEL_CURVES =>
              WC_LEVEL_ORIGIN      : WC.POINT_3;
              WC_LEVEL_DIRECTION   : WC.VECTOR_3;
              WC_LEVEL_PARAMETERS  : WC.MAGNITUDE_LIST;

            when others =>
              null;

          end case;
        end record;

    -- Provides for specifying values for different parametric surface characteristics.  Additional
    -- variants may be included when additional registered or unregistered parametric surface
    -- characteristics are supported by an implementation.


    -- PARAMETRIC_SURFACE_INDEX
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
type PARAMETRIC_SURFACE_INDEX is new PHIGS_NATURAL;

-- Provides for index values for parametric surface representations.


-- PARAMETRIC_SURFACE_INDICES

package PARAMETRIC_SURFACE_INDICES is
        new PHIGS_LIST_UTILITIES
                (PARAMETRIC_SURFACE_INDEX,
                 MAX_PARAMETRIC_SURFACE_INDICES_SUPPORTED);

-- Provides for lists of parametric surface indices.


-- PATH_DEPTH

subtype PATH_DEPTH is
  PHIGS_NATURAL range 0..MAX_PATH_DEPTH_SUPPORTED;

-- Defines a type for specifying the depth of a traversal path.

-- PATH_ORDER

type PATH_ORDER is (  TOPFIRST,
                      BOTTOMFIRST);

-- Provides for specifying the order of pick paths.

-- PATTERN_INDEX

subtype PATTERN_INDEX is STYLE_INDEX range 1..STYLE_INDEX'LAST;

-- Defines the range of pattern table indices.

-- PATTERN_INDICES

package PATTERN_INDICES is
  new PHIGS_LIST_UTILITIES ( PATTERN_INDEX,
                             MAX_PATTERN_INDICES_SUPPORTED);

-- Provides for lists of pattern table indices.

-- PICK_DEVICE_NUMBER

type PICK_DEVICE_NUMBER is new DEVICE_NUMBER;

-- Provides for pick device identifiers.
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
-- PICK_ID

type PICK_ID is new PHIGS_INTEGER;

-- Defines the range of pick identifiers available on an
-- implementation.

-- PICK_IDS

package PICK_IDS is
  new PHIGS_LIST_UTILITIES ( PICK_ID,
                             MAX_PICK_IDS_SUPPORTED);

-- Provides for a list of pick identifiers.

-- PICK_PROMPT_ECHO_TYPE

type PICK_PROMPT_ECHO_TYPE is new PHIGS_INTEGER;

-- Defines the pick prompt and echo type.

-- PICK_PROMPT_ECHO_TYPES

package PICK_PROMPT_ECHO_TYPES is
  new PHIGS_LIST_UTILITIES ( PICK_PROMPT_ECHO_TYPE,
                             MAX_PICK_PROMPT_ECHO_TYPES_SUPPORTED);

-- Provides for lists of pick prompt and echo types.

-- PICK_REQUEST_STATUS

type PICK_REQUEST_STATUS is ( OK,
                              NOPICK,
                              NONE);

-- Defines the status of a pick input operation for the
-- request function.

-- PICK_STATUS

subtype PICK_STATUS is PICK_REQUEST_STATUS range OK..NOPICK;

-- Defines the status of pick input operation for the sample
-- get, and inquiry functions.

-- REFLECTANCE_MODEL
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567401 299 ■


```
        type REFLECTANCE_MODEL is new PHIGS_INTEGER;

        -- Provides for selecting the type of reflectance calculations.


        -- REFLECTANCE_MODELS

        package REFLECTANCE_MODELS is
                new PHIGS_LIST_UTILITIES
                        (REFLECTANCE_MODEL,
                         MAX_REFLECTANCE_MODELS_SUPPORTED);

        -- Provides for lists of reflectance models.

NO_REFLECTANCE                          : constant REFLECTANCE_MODEL := 1;
AMBIENT_REFLECTANCE                     : constant REFLECTANCE_MODEL := 2;
AMBIENT_DIFFUSE_REFLECTANCE             : constant REFLECTANCE_MODEL := 3;
AMBIENT_DIFFUSE_SPECULAR_REFLECTANCE    : constant REFLECTANCE_MODEL := 4;

        -- REFLECTANCE_COEFFICIENT

        type REFLECTANCE_COEFFICIENT is digits PHIGS_PRECISION range 0.0 .. 1.0;

        -- Provides for specifying values for lighting reflectance coefficients.


        -- REFLECTANCE_INDEX

        type REFLECTANCE_INDEX is new PHIGS_NATURAL;

        -- Provides for index values for reflectance representations.


        -- REFLECTANCE_INDICES

        package REFLECTANCE_INDICES is
                new PHIGS_LIST_UTILITIES ( REFLECTANCE_INDEX,
                                           MAX_REFLECTANCE_INDICES_SUPPORTED);

        -- Provides for lists of reflectance indices.


        -- REFLECTANCE_PROPERTIES_TYPE
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567402 125 ■


```
        type REFLECTANCE_PROPERTIES_TYPE is new PHIGS_INTEGER;

        -- Provides for selecting the type of reflectance properties.

        -- REFLECTANCE_PROPERTIES_TYPES

        package REFLECTANCE_PROPERTIES_TYPES is
                new PHIGS_LIST_UTILITIES
                        (REFLECTANCE_PROPERTIES_TYPE,
                         MAX_REFLECTANCE_PROPERTIES_TYPES_SUPPORTED);

        -- Provides for lists of reflectance properties types.

    SIMPLE_REFLECTANCE  : constant REFLECTANCE_PROPERTIES_TYPE := 1;

        -- REFLECTANCE_PROPERTIES_DATA_RECORD

        type REFLECTANCE_PROPERTIES_DATA_RECORD
              (TYPE_OF_REFLECTANCE_PROPERTIES
                      : REFLECTANCE_PROPERTIES_TYPE := SIMPLE_REFLECTANCE) is
            record
              case TYPE_OF_REFLECTANCE_PROPERTIES is

                when SIMPLE_REFLECTANCE =>
                  AMBIENT_REFLECTANCE   : REFLECTANCE_COEFFICIENT;
                  DIFFUSE_REFLECTANCE   : REFLECTANCE_COEFFICIENT;
                  SPECULAR_REFLECTANCE  : REFLECTANCE_COEFFICIENT;
                  SPECULAR_COLOUR       : GENERAL_COLOUR;
                  SPECULAR_EXPONENT     : LIGHTING_EXPONENT;

                when others =>
                  null;

              end case;
            end record;

        -- Provides for specifying values for different types of reflectance properties.  Additional
        -- variants may be included when additional registered or unregistered reflectance
        -- properties types are supported by an implementation.


        -- STRUCTURE_ID

        type STRUCTURE_ID is new PHIGS_NATURAL;

        -- Defines a type for structure identifiers.

        -- STRUCTURE_IDS
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
package STRUCTURE_IDS is
  new PHIGS_LIST_UTILITIES ( STRUCTURE_ID,
                             MAX_STRUCTURE_IDS_SUPPORTED);

-- Provides for lists of structure identifiers.

-- PICK_PATH_ENTRY

type PICK_PATH_ENTRY is
  record
    STRUCTURE_IDENTIFIER : STRUCTURE_ID;
    PICK_IDENTIFIER      : PICK_ID;
    ELEMENT_NUMBER       : ELEMENT_POSITION;
  end record;

-- Defines the entries of a pick path.

-- PICK_PATH_ARRAY

type PICK_PATH_ARRAY is array (PATH_DEPTH range <>) of
  PICK_PATH_ENTRY;

-- Provides for the specification of an unconstrained array of pick
-- path entries.

-- PICK_PATH

type PICK_PATH (DEPTH : PATH_DEPTH := 0) is
  record
    PATH : PICK_PATH_ARRAY (0..DEPTH);
  end record;

-- Provides for the specification of a pick path.

-- POLYLINE_FILL_AREA_CONTROL_FLAG

type POLYLINE_FILL_AREA_CONTROL_FLAG is ( POLYLINE,
                                          FILL_AREA,
                                          FILL_AREA_SET);

-- Provides for indicating the type of primitive used
-- to produce locator prompt and echo type 5.

-- POLYLINE_SHADING_METHOD

type POLYLINE_SHADING_METHOD is new PHIGS_NATURAL;

-- Provides for values for distinguishing between various methods of polyline shading.
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
        -- POLYLINE_SHADING_METHODS

        package POLYLINE_SHADING_METHODS is
                new PHIGS_LIST_UTILITIES
                        (POLYLINE_SHADING_METHOD,
                         MAX_POLYLINE_SHADING_METHODS_SUPPORTED);

        -- Provides for lists of polyline shading methods.

NO_POLYLINE_SHADING             : constant POLYLINE_SHADING_METHOD := 1;
POLYLINE_COLOUR_SHADING         : constant POLYLINE_SHADING_METHOD := 2;

        -- PROJECTION_TYPE

        type PROJECTION_TYPE is ( PARALLEL,
                                  PERSPECTIVE);

        -- Defines the type of projections supported by PHIGS.

        -- RASTER_UNITS

        type RASTER_UNITS is new PHIGS_POSITIVE;

        -- Defines the range of raster units.

        -- RASTER_UNIT_SIZE_3

        type RASTER_UNIT_SIZE_3 is
          record
            X : RASTER_UNITS;
            Y : RASTER_UNITS;
            Z : RASTER_UNITS;
          end record;

        -- Defines the size of an object in raster units on a raster
        -- device.

        -- RASTER_UNIT_SIZE_2

        type RASTER_UNIT_SIZE_2 is
          record
            X : RASTER_UNITS;
            Y : RASTER_UNITS;
          end record;

        -- Defines the size of an object in raster units on a raster
        -- device.
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567405 934 ■


```
-- REFERENCE_HANDLING_FLAG

type REFERENCE_HANDLING_FLAG is (  DELETE,
                                   KEEP);

-- Provides for specification of the handling of deleted
-- structure networks.

-- REFERENCE_PATH_ENTRY

type REFERENCE_PATH_ENTRY is
  record
    STRUCTURE_IDENTIFIER : STRUCTURE_ID;
    ELEMENT_NUMBER       : ELEMENT_POSITION;
  end record;

-- Defines the entries of a reference path.

-- REFERENCE_PATH_ARRAY

type REFERENCE_PATH_ARRAY is array (PATH_DEPTH range <>) of
  REFERENCE_PATH_ENTRY;

-- Provides for the specification of a basic reference path.

-- REFERENCE_PATH

type REFERENCE_PATH (DEPTH : PATH_DEPTH := 0) is
  record
    PATH : REFERENCE_PATH_ARRAY (0..DEPTH);
  end record;

-- Provides for the specification of a variable size
-- reference path.

-- REFERENCE_PATHS

package REFERENCE_PATHS is
  new PHIGS_LIST_UTILITIES ( REFERENCE_PATH,
                             MAX_REFERENCE_PATHS_SUPPORTED);

-- Provides for the specification of lists of reference
-- paths.

-- RELATIVE_PRIORITY

type RELATIVE_PRIORITY is ( HIGHER,
                            LOWER);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
-- Indicates the relative input priority between two views.

-- RETURN_VALUE_TYPE

type RETURN_VALUE_TYPE is ( SET,
                            REALIZED);

-- Indicates whether the returned values should be as they
-- were set by the program or as they were actually realized
-- on the device.

-- SEARCH_DIRECTION

type SEARCH_DIRECTION is ( BACKWARD,
                           FORWARD);

-- This type provides for indicating search direction.

-- SEARCH_STATUS_INDICATOR

type SEARCH_STATUS_INDICATOR is ( FAILURE,
                                  SUCCESS);

-- This type provides for indicating the result of a search.

-- STRING_DEVICE_NUMBER

type STRING_DEVICE_NUMBER is new DEVICE_NUMBER;

-- Provides for string device identifiers.

-- STRING_PROMPT_ECHO_TYPE

type STRING_PROMPT_ECHO_TYPE is new PHIGS_INTEGER;

-- Defines the string prompt and echo types.

-- STRING_PROMPT_ECHO_TYPES

package STRING_PROMPT_ECHO_TYPES is
  new PHIGS_LIST_UTILITIES ( STRING_PROMPT_ECHO_TYPE,
                             MAX_STRING_PROMPT_ECHO_TYPES_SUPPORTED);

-- Provides for lists of string prompt and echo types.

-- STROKE DEVICE NUMBER

type STROKE_DEVICE_NUMBER is new DEVICE_NUMBER;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
-- Provides for stroke device identifiers.

-- STROKE_PROMPT_ECHO_TYPE

type STROKE_PROMPT_ECHO_TYPE is new PHIGS_INTEGER;

-- Defines the stroke prompt and echo types.

-- STROKE_PROMPT_ECHO_TYPES

package STROKE_PROMPT_ECHO_TYPES is
  new PHIGS_LIST_UTILITIES ( STROKE_PROMPT_ECHO_TYPE,
                             MAX_STROKE_PROMPT_ECHO_TYPES_SUPPORTED);

-- Provides for lists of stroke prompt and echo types

-- STRUCTURE_ELEMENT_TYPE

subtype STRUCTURE_ELEMENT_TYPE is
  ELEMENT_TYPE range NIL..ELEMENT_TYPE'LAST;

-- Provides for identifying the types of individual
-- structure elements.

-- STRUCTURE NETWORK SOURCE

type STRUCTURE_NETWORK_SOURCE is ( CSS,
                                   ARCHIVE);

-- Provides for indicating source for check of conflicting
-- structure identifiers.

-- SURFACE_APPROX_CRITERIA_TYPE

type SURFACE_APPROX_CRITERIA_TYPE is new PHIGS_INTEGER;

-- Provides for differentiating methods of surface approximation.


-- SURFACE_APPROX_CRITERIA_TYPES

package SURFACE_APPROX_CRITERIA_TYPES is
        new PHIGS_LIST_UTILITIES
                (SURFACE_APPROX_CRITERIA_TYPE,
                 MAX_SURFACE_APPROX_CRITERIA_TYPES_SUPPORTED);

-- Provides for lists of surface approximation criteria types.
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567408 643 ■


```
WS_DEPENDENT_SURFACE              : constant SURFACE_APPROX_CRITERIA_TYPE := 1;
CONSTANT_SUBDIVISION_SURFACE_BETWEEN_KNOTS
                                  : constant SURFACE_APPROX_CRITERIA_TYPE := 2;
WC_CHORDAL_SIZE_SURFACE           : constant SURFACE_APPROX_CRITERIA_TYPE := 3;
NPC_CHORDAL_SIZE_SURFACE          : constant SURFACE_APPROX_CRITERIA_TYPE := 4;
DC_CHORDAL_SIZE_SURFACE           : constant SURFACE_APPROX_CRITERIA_TYPE := 5;
WC_PLANAR_DEVIATION_SURFACE       : constant SURFACE_APPROX_CRITERIA_TYPE := 6;
NPC_PLANAR_DEVIATION_SURFACE      : constant SURFACE_APPROX_CRITERIA_TYPE := 7;
DC_PLANAR_DEVIATION_SURFACE       : constant SURFACE_APPROX_CRITERIA_TYPE := 8;
WC_RELATIVE_SURFACE               : constant SURFACE_APPROX_CRITERIA_TYPE := 9;
NPC_RELATIVE_SURFACE              : constant SURFACE_APPROX_CRITERIA_TYPE := 10;
DC_RELATIVE_SURFACE               : constant SURFACE_APPROX_CRITERIA_TYPE := 11;

    -- SURFACE_APPROX_DATA_RECORD

    type SURFACE_APPROX_DATA_RECORD
      (CRITERIA_TYPE : SURFACE_APPROX_CRITERIA_TYPE
                     := WS_DEPENDENT_SURFACE) is
        record
          case CRITERIA_TYPE is

            when WS_DEPENDENT_SURFACE =>
              null;

            when CONSTANT_SUBDIVISION_SURFACE_BETWEEN_KNOTS =>
              CONSTANT_SUBDIVISION_UCOUNT : PHIGS_INTEGER;
              CONSTANT_SUBDIVISION_VCOUNT : PHIGS_INTEGER;

            when WC_CHORDAL_SIZE_SURFACE =>
              WC_CHORDAL_SIZE_UVALUE : WC.MAGNITUDE;
              WC_CHORDAL_SIZE_VVALUE : WC.MAGNITUDE;

            when NPC_CHORDAL_SIZE_SURFACE =>
              NPC_CHORDAL_SIZE_UVALUE : NPC.MAGNITUDE;
              NPC_CHORDAL_SIZE_VVALUE : NPC.MAGNITUDE;

            when DC_CHORDAL_SIZE_SURFACE =>
              DC_CHORDAL_SIZE_UVALUE : DC.MAGNITUDE;
              DC_CHORDAL_SIZE_VVALUE : DC.MAGNITUDE;

            when WC_PLANAR_DEVIATION_SURFACE =>
              WC_PLANAR_DEVIATION_VALUE : WC.MAGNITUDE;

            when NPC_PLANAR_DEVIATION_SURFACE =>
              NPC_PLANAR_DEVIATION_VALUE : NPC.MAGNITUDE;

            when DC_PLANAR_DEVIATION_SURFACE =>
              DC_PLANAR_DEVIATION_VALUE : DC.MAGNITUDE;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567409 58T ■


```
            when WC_RELATIVE_SURFACE =>
              WC_RELATIVE_VALUE : WC.RELATIVE_MAGNITUDE;

            when NPC_RELATIVE_SURFACE =>
              NPC_RELATIVE_VALUE : NPC.RELATIVE_MAGNITUDE;

            when DC_RELATIVE_SURFACE =>
              DC_RELATIVE_VALUE : DC.RELATIVE_MAGNITUDE;

            when others =>
              null;

          end case;
        end record;

-- Provides for specifying values for different methods of surface approximation.
-- Additional variants may be included when additional registered or unregistered surface
-- approximation criteria types are supported by an implementation.


-- DISPLAY_PRIORITY

type DISPLAY_PRIORITY is digits PHIGS_PRECISION range 0.0..1.0;

-- Defines type for specifying structure display priority.

-- STRUCTURE_STATE

type STRUCTURE_STATE is ( STCL,
                          STOP);

-- The type used to return the structure state value.

-- STRUCTURE_STATUS

type STRUCTURE_STATUS is ( NON_EXISTENT,
                           EMPTY,
                           NOTEMPTY);

--  The type used to return the structure status.

-- POSTED_STRUCTURE

type POSTED_STRUCTURE is
  record
    STRUCTURE_IDENTIFIER : STRUCTURE_ID;
    PRIORITY             : DISPLAY_PRIORITY;
  end record;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
-- Defines a data type for describing posted structure networks.

-- POSTED_STRUCTURES

package POSTED_STRUCTURES is
  new PHIGS_LIST_UTILITIES ( POSTED_STRUCTURE,
                             MAX_POSTED_STRUCTURES_SUPPORTED);

-- Provides for the specification of lists of posted structure
-- networks.

-- SUBPROGRAM_NAME

subtype SUBPROGRAM_NAME is PHIGS_STRING;

-- Defines the name of a PHIGS function detecting an error.

-- SYSTEM_STATE

type SYSTEM_STATE is ( PHCL,
                       PHOP);

-- The type used to return the system state value.

-- VERTICAL_ALIGNMENT

type VERTICAL_ALIGNMENT is ( NORMAL,
                             TOP,
                             CAP,
                             HALF,
                             BASE,
                             BOTTOM);

-- The alignment of the text extent parallelogram with respect
-- to the vertical positioning of the text.

-- TEXT_ALIGNMENT

type TEXT_ALIGNMENT is
  record
    HORIZONTAL : HORIZONTAL_ALIGNMENT;
    VERTICAL   : VERTICAL_ALIGNMENT;
  end record;

-- The type of the attribute controlling the positioning of
-- the text extent parallelogram in relation to the text
-- position, having horizontal and vertical components as
-- defined above.
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
-- TEXT_FONT

type TEXT_FONT is new PHIGS_INTEGER;

-- Defines the types of fonts provided by the implementation.

-- TEXT_PRECISION

type TEXT_PRECISION is ( STRING_PRECISION,
                         CHAR_PRECISION,
                         STROKE_PRECISION);

-- The precision with which text appears.

-- TEXT_FONT_PRECISION

type TEXT_FONT_PRECISION is
  record
    FONT      : TEXT_FONT;
    PRECISION : TEXT_PRECISION;
  end record;

-- This type defines a record describing the text font and
-- precision supported.

-- TEXT_FONT_PRECISIONS

package TEXT_FONT_PRECISIONS is
  new PHIGS_LIST_UTILITIES ( TEXT_FONT_PRECISION,
                             MAX_TEXT_FONT_PRECISION_PAIRS_SUPPORTED);

-- Provides for lists of text font and precision pairs.

-- TEXT_INDEX

type TEXT_INDEX is new PHIGS_POSITIVE;

-- Defines the range of text bundle table indices.

-- TEXT_INDICES

package TEXT_INDICES is
  new PHIGS_LIST_UTILITIES ( TEXT_INDEX,
                             MAX_TEXT_INDICES_SUPPORTED);

-- Provides for lists of text indices.

-- TEXT_PATH
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567412 074 ■


```
type TEXT_PATH is ( RIGHT,
                    LEFT,
                    UP,
                    DOWN);

-- The direction taken by a text string.

-- TRANSFORMATION_MATRIX_2

type TRANSFORMATION_MATRIX_2 is array (1..3, 1..3) of MC_TYPE;

-- For 2D modeling and view transformation matrices.

-- TRANSFORMATION_MATRIX_3

type TRANSFORMATION_MATRIX_3 is array (1..4, 1..4) of MC_TYPE;

-- For 3D modeling and view transformation matrices.

-- TRIMCURVE_ORDER

subtype TRIMCURVE_ORDER is SPLINE_ORDER range 2..SPLINE_ORDER'last;

-- Provides for restricting spline orders to those acceptable for trimming curves.


-- TRIMCURVE

type TRIMCURVE( CRITERIA_TYPE  : CURVE_APPROX_CRITERIA_TYPE
                                   := CONSTANT_SUBDIVISION_CURVE;
                RATIONALITY    : SPLINE_RATIONALITY := RATIONAL;
                KNOT_COUNT     : SMALL_NATURAL := 4;
                LENGTH         : VERTEX_SET_DIMENSION := 2) is
  record
    CRITERIA_FOR_CURVE_APPROX
                   :  CURVE_APPROX_DATA_RECORD (CRITERIA_TYPE);
    FLAG           : CURVE_VISIBILITY_FLAG;
    ORDER          : TRIMCURVE_ORDER;
    KNOTS          : KNOT_VECTOR(KNOT_COUNT);
    CONTROL_POINTS : SPC.CONTROL_POINT_LIST_2(RATIONALITY, LENGTH);
    LIMITS         : SPC.RANGE_OF_MAGNITUDES;
  end record;

-- Provides for specifying minimum and maximum values of spline parameters.


-- TRIMCURVE_LIST
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567413 T00 ■


```
type TRIMCURVE_LIST is array (POSITIVE range <>) of TRIMCURVE;

-- Provides for arrays of trimming curves.  These will be used with NURB surface primitives.


-- ACCESS_TRIMCURVE_LIST

type ACCESS_TRIMCURVE_LIST is access TRIMCURVE_LIST;

-- Provides for pointers to lists of trimming curves.


-- LIST_OF_TRIMCURVE_LIST

type LIST_OF_TRIMCURVE_LIST is
  array (POSITIVE range <>) of ACCESS_TRIMCURVE_LIST;

-- Provides for pointers to lists of lists of trimming curves.


-- ACCESS_LIST_OF_TRIMCURVE_LIST

type ACCESS_LIST_OF_TRIMCURVE_LIST is access LIST_OF_TRIMCURVE_LIST;

-- Provides for pointers to lists of lists of edge flags.


-- RETURN_TRIMCURVE_ORDER

subtype RETURN_TRIMCURVE_ORDER is
          TRIMCURVE_ORDER range 4..TRIMCURVE_ORDER'last;

-- Provides for restricting trimcurve orders to those acceptable for returning values of the
-- facilities.


-- TRUNCATION_METHOD

type TRUNCATION_METHOD is ( HEAD,
                            TAIL);

-- Provides for specifying if the head or tail of a reference path should be truncated.

-- UPDATE_REGENERATION_FLAG

type UPDATE_REGENERATION_FLAG is ( PERFORM,
                                   POSTPONE);

-- Flag indicating regeneration action on display.
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
-- UPDATE_STATE

type UPDATE_STATE is ( NOTPENDING,
                       PENDING);

-- Indicates whether or not a workstation transformation
-- change has been requested and not yet provided.

-- VALUATOR_DEVICE_NUMBER

type VALUATOR_DEVICE_NUMBER is new DEVICE_NUMBER;

-- Provides for valuator device identifiers.

-- VALUATOR_PROMPT_ECHO_TYPE

type VALUATOR_PROMPT_ECHO_TYPE is new PHIGS_INTEGER;

-- Defines the possible range of valuator prompt and
-- echo types.

-- VALUATOR_PROMPT_ECHO_TYPES

package VALUATOR_PROMPT_ECHO_TYPES is
  new PHIGS_LIST_UTILITIES
    (VALUATOR_PROMPT_ECHO_TYPE,
     MAX_VALUATOR_PROMPT_ECHO_TYPES_SUPPORTED);

-- Provides for lists of valuator prompt and echo types.

-- VALUATOR_VALUE

type VALUATOR_VALUE is digits PHIGS_PRECISION;

-- Defines the range of accuracy of valuator input values on
-- an implementation.

-- EVENT_DEVICE_NUMBER

type EVENT_DEVICE_NUMBER (CLASS : INPUT_CLASS := NONE) is
  record
    case CLASS is

      when NONE =>
        null;

      when LOCATOR_INPUT =>
        LOCATOR_EVENT_DEVICE : LOCATOR_DEVICE_NUMBER;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
        when STROKE_INPUT =>
          STROKE_EVENT_DEVICE : STROKE_DEVICE_NUMBER;

        when VALUATOR_INPUT =>
          VALUATOR_EVENT_DEVICE : VALUATOR_DEVICE_NUMBER;

        when CHOICE_INPUT =>
          CHOICE_EVENT_DEVICE : CHOICE_DEVICE_NUMBER;

        when PICK_INPUT =>
          PICK_EVENT_DEVICE : PICK_DEVICE_NUMBER;

        when STRING_INPUT =>
          STRING_EVENT_DEVICE : STRING_DEVICE_NUMBER;

      end case;
    end record;

-- Provides for returning any class of device number from the
-- event queue.

-- EVENT_QUEUE_DEVICE_NUMBER

type EVENT_QUEUE_DEVICE_NUMBER
  (CLASS : INPUT_QUEUE_CLASS := LOCATOR_INPUT) is
    record
      case CLASS is

        when LOCATOR_INPUT =>
          LOCATOR_EVENT_DEVICE : LOCATOR_DEVICE_NUMBER;

        when STROKE_INPUT =>
          STROKE_EVENT_DEVICE : STROKE_DEVICE_NUMBER;

        when VALUATOR_INPUT =>
          VALUATOR_EVENT_DEVICE : VALUATOR_DEVICE_NUMBER;

        when CHOICE_INPUT =>
          CHOICE_EVENT_DEVICE : CHOICE_DEVICE_NUMBER;

        when PICK_INPUT =>
          PICK_EVENT_DEVICE : PICK_DEVICE_NUMBER;

        when STRING_INPUT =>
          STRING_EVENT_DEVICE : STRING_DEVICE_NUMBER;

      end case;
    end record;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
-- Allows EVENT_DEVICE_NUMBERs which cannot have value NONE.  This is
-- used when returning only real input classes as causes of event
-- queue overflow and for flushing the queue.

-- ACCESS_COLOUR_MATRIX

type ACCESS_COLOUR_MATRIX is access COLOUR_MATRIX;

-- Provides for returning variable sized matrices containing colour
-- indices corresponding to a cell array or pattern array.

-- VARIABLE_CONNECTION_ID

type VARIABLE_CONNECTION_ID (LENGTH : STRING_SMALL_NATURAL := 0) is
  record
    CONNECT : CONNECTION_ID (1..LENGTH);
  end record;

-- Defines a variable length connection identifier for

-- ACCESS_STRING

type ACCESS_STRING is access PHIGS_STRING;

--  Defines a variable length string needed to store strings in
--  structure element records.

-- VERTEX_INDEX

type VERTEX_INDEX is new PHIGS_NATURAL range 1..MAX_VERTICES_SUPPORTED;

-- Provides for indices into arrays of vertices.  These will be used with various "with Data"
-- primitives.

-- VERTEX_INDEX_LIST

type VERTEX_INDEX_LIST is array (POSITIVE range <>) of VERTEX_INDEX;

-- Provides for arrays of indices.  These will be used with various "with Data" primitives.


-- ACCESS_VERTEX_INDEX_LIST

type ACCESS_VERTEX_INDEX_LIST is access VERTEX_INDEX_LIST;

-- Provides for pointers to lists of vertex indices.

-- LIST_OF_VERTEX_INDEX_LIST
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567417 656 ■


```
type LIST_OF_VERTEX_INDEX_LIST is
  array (POSITIVE range <>) of ACCESS_VERTEX_INDEX_LIST;

-- Provides for lists of lists of vertex indices.

-- ACCESS_LIST_OF_VERTEX_INDEX_LIST

type ACCESS_LIST_OF_VERTEX_INDEX_LIST is
  access LIST_OF_VERTEX_INDEX_LIST;

-- Provides for pointers to lists of lists of vertex indices.

-- LIST_OF_LIST_OF_VERTEX_INDEX_LIST

type LIST_OF_LIST_OF_VERTEX_INDEX_LIST is
  array (POSITIVE range <>) of ACCESS_LIST_OF_VERTEX_INDEX_LIST;

-- Provides for pointers to lists of lists of vertex indices.

-- ACCESS_LIST_OF_LIST_OF_VERTEX_INDEX_LIST

type ACCESS_LIST_OF_LIST_OF_VERTEX_INDEX_LIST is
        access LIST_OF_LIST_OF_VERTEX_INDEX_LIST;

-- Provides for pointers to lists of lists of lists of indices.

-- VERTEX_INDEX_TRIPLET

subtype VERTEX_INDEX_TRIPLET is VERTEX_INDEX_LIST(1..3);

-- Provides for triplets of indices.  These will be used with "Triangle Set with Data"
-- primitives.

-- LIST_OF_VERTEX_INDEX_TRIPLET

type LIST_OF_VERTEX_INDEX_TRIPLET is
  array (POSITIVE range <>) of VERTEX_INDEX_TRIPLET;

-- Provides for lists of triplets of vertex indices.


-- ACCESS_LIST_OF_VERTEX_INDEX_TRIPLET

type ACCESS_LIST_OF_VERTEX_INDEX_TRIPLET is
  access LIST_OF_VERTEX_INDEX_TRIPLET;

-- Provides for pointers to lists of triplets of vertex indices.

-- VIEW_INDEX
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
type VIEW_INDEX is new PHIGS_NATURAL;

-- Defines a type for specifying view indices.

-- VIEW_INDICES

package VIEW_INDICES is
  new PHIGS_LIST_UTILITIES ( VIEW_INDEX,
                             MAX_VIEW_INDICES_SUPPORTED);

-- Provides for lists of view indices.

-- VISUAL REPRESENTATION STATE

type VISUAL_REPRESENTATION_STATE is ( CORRECT,
                                      DEFERRED,
                                      SIMULATED);

-- Specifies state of image on display surface.

-- WS_CATEGORY

type WS_CATEGORY is ( OUTPUT,
                      INPUT,
                      OUTIN,
                      MO,
                      MI);

-- Type for PHIGS workstation categories.

-- WS_DEPENDENCY

type WS_DEPENDENCY is ( WS_INDEPENDENT,
                        WS_DEPENDENT);

-- Type for indications of workstation dependency.

-- WS_DEPENDENCIES

package WS_DEPENDENCIES is
  new PHIGS_LIST_UTILITIES ( WS_DEPENDENCY,
                             MAX_GSE_IDS_SUPPORTED);

-- Provides for lists of workstation dependencies.

-- WS_ID

type WS_ID is new PHIGS_POSITIVE;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567419 429 ■


```
-- Defines the range of workstation identifiers.

-- WS_IDS

package WS_IDS is new PHIGS_LIST_UTILITIES ( WS_ID,
                                             MAX_OPEN_WS_SUPPORTED);

-- Provides for lists of workstation identifiers.

-- WS_STATE

type WS_STATE is ( WSCL,
                   WSOP);

-- The type used to return the workstation state value.

-- WS_TYPE

type WS_TYPE is new PHIGS_POSITIVE;

-- Range of values corresponding to valid workstation types.
-- Constants specifying names for the various types of
-- workstations should be provided by an implementation.

-- WS_TYPES

package WS_TYPES is new PHIGS_LIST_UTILITIES ( WS_TYPE,
                                               MAX_WS_TYPES_SUPPORTED);

-- Provides for lists of workstation types.

-- STRUCTURE_ELEMENT_RECORD

type STRUCTURE_ELEMENT_RECORD
  (ELEMENT_TYPE : STRUCTURE_ELEMENT_TYPE := NIL) is
    record
      case ELEMENT_TYPE is

        -- The empty element

        when NIL =>
          null;

        -- PHIGS Primitive Elements

        when POLYLINE_3 =>
          POLYLINE_3_POINTS : MC.ACCESS_POINT_LIST_3;

        when POLYLINE =>
          POLYLINE_POINTS : MC.ACCESS_POINT_LIST_2;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567420 140 ■


```
when POLYMARKER_3 =>
  POLYMARKER_3_POINTS : MC.ACCESS_POINT_LIST_3;

when POLYMARKER =>
  POLYMARKER_POINTS : MC.ACCESS_POINT_LIST_2;

when TEXT_3 =>
  TEXT_3_POINT : MC.POINT_3;
  TEXT_DIRECTION_VECTORS : MC.VECTOR_PAIR_3;
  TEXT_3_CHAR_STRING : ACCESS_STRING;

when TEXT =>
  TEXT_POINT : MC.POINT_2;
  TEXT_CHAR_STRING : ACCESS_STRING;

when ANNOTATION_TEXT_RELATIVE_3 =>
  ANNOTATION_TEXT_RELATIVE_3_REF_POINT : MC.POINT_3;
  ANNOTATION_TEXT_RELATIVE_3_OFFSET : NPC.POINT_3;
  ANNOTATION_TEXT_3_CHAR_STRING : ACCESS_STRING;

when ANNOTATION_TEXT_RELATIVE =>
  ANNOTATION_TEXT_RELATIVE_REF_POINT : MC.POINT_2;
  ANNOTATION_TEXT_RELATIVE_OFFSET : NPC.POINT_2;
  ANNOTATION_TEXT_CHAR_STRING : ACCESS_STRING;

when FILL_AREA_3 =>
  FILL_AREA_3_POINTS : MC.ACCESS_POINT_LIST_3;

when FILL_AREA =>
  FILL_AREA_POINTS : MC.ACCESS_POINT_LIST_2;

when FILL_AREA_SET_3 =>
  FILL_AREA_SET_3_POINTS : MC.ACCESS_LIST_OF_POINT_LIST_3;

when FILL_AREA_SET =>
  FILL_AREA_SET_POINTS : MC.ACCESS_LIST_OF_POINT_LIST_2;

when CELL_ARRAY_3 =>
  CORNER_P_3 : MC.POINT_3;
  CORNER_Q_3 : MC.POINT_3;
  CORNER_R_3 : MC.POINT_3;
  CELL_ARRAY_3_CELLS : ACCESS_COLOUR_MATRIX;

when CELL_ARRAY =>
  CORNER_P : MC.POINT_2;
  CORNER_Q : MC.POINT_2;
  CELL_ARRAY_CELLS : ACCESS_COLOUR_MATRIX;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567421 087 ■


```
when GDP_3 =>
  GDP_3_POINTS : MC.ACCESS_POINT_LIST_3;
  GDP_3_DATA : GDP_3_RECORD;

when GDP =>
  GDP_POINTS : MC.ACCESS_POINT_LIST_2;
  GDP_DATA : GDP_RECORD;

-- PHIGS Bundle Index Elements

when SET_POLYLINE_INDEX =>
  POLYLINE_IND : POLYLINE_INDEX;

when SET_POLYMARKER_INDEX =>
  POLYMARKER_IND : POLYMARKER_INDEX;

when SET_TEXT_INDEX =>
  TEXT_IND : TEXT_INDEX;

when SET_INTERIOR_INDEX =>
  INTERIOR_IND : INTERIOR_INDEX;

when SET_EDGE_INDEX =>
  EDGE_IND : EDGE_INDEX;

-- PHIGS Individual Aspect Elements

when SET_LINETYPE =>
  TYPE_OF_LINE : LINETYPE;

when SET_LINEWIDTH_SCALE_FACTOR =>
  LINEWIDTH_SF : LINEWIDTH;

when SET_POLYLINE_COLOUR_INDEX =>
  LINE_COLOUR : COLOUR_INDEX;

when SET_MARKER_TYPE =>
  TYPE_OF_MARKER : MARKER_TYPE;

when SET_MARKER_SIZE_SCALE_FACTOR =>
  SIZE : MARKER_SIZE;

when SET_POLYMARKER_COLOUR_INDEX =>
  MARKER_COLOUR : COLOUR_INDEX;

when SET_TEXT_FONT =>
  FONT : TEXT_FONT;

when SET_TEXT_PRECISION =>
  PRECISION : TEXT_PRECISION;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
when SET_CHAR_EXPANSION_FACTOR =>
  EXPANSION : CHAR_EXPANSION;

when SET_CHAR_SPACING =>
  SPACING : CHAR_SPACING;

when SET_TEXT_COLOUR_INDEX =>
  TEXT_COLOUR : COLOUR_INDEX;

when SET_CHAR_HEIGHT =>
  HEIGHT : MC.MAGNITUDE;

when SET_CHAR_UP_VECTOR =>
  CHAR_UP_VECTOR : MC.VECTOR_2;

when SET_TEXT_PATH =>
  PATH : TEXT_PATH;

when SET_TEXT_ALIGNMENT =>
  ALIGNMENT : TEXT_ALIGNMENT;

when SET_ANNOTATION_TEXT_CHAR_HEIGHT =>
  ANNOTATION_HEIGHT : NPC.MAGNITUDE;

when SET_ANNOTATION_TEXT_CHAR_UP_VECTOR =>
  ANNOTATION_CHAR_UP_VECTOR : NPC.VECTOR_2;

when SET_ANNOTATION_TEXT_PATH =>
  ANNOTATION_PATH : TEXT_PATH;

when SET_ANNOTATION_TEXT_ALIGNMENT =>
  ANNOTATION_ALIGNMENT : TEXT_ALIGNMENT;

when SET_ANNOTATION_STYLE =>
  STYLE_OF_ANNOTATION : ANNOTATION_STYLE;

when SET_INTERIOR_STYLE =>
  STYLE_OF_INTERIOR : INTERIOR_STYLE;

when SET_INTERIOR_STYLE_INDEX =>
  STYLE_IND : STYLE_INDEX;

when SET_INTERIOR_COLOUR_INDEX =>
  INTERIOR_COLOUR : COLOUR_INDEX;

when SET_EDGE_FLAG =>
  FLAG : EDGE_FLAG;

when SET_EDGETYPE =>
  TYPE_OF_EDGE : EDGETYPE;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
when SET_EDGEWIDTH_SCALE_FACTOR =>
  EDGEWIDTH_SF : EDGEWIDTH;

when SET_EDGE_COLOUR_INDEX =>
  EDGE_COLOUR : COLOUR_INDEX;

-- PHIGS Pattern Attribute Elements

when SET_PATTERN_SIZE =>
  PATTERN_SIZE : MC.SIZE_2;

when SET_PATTERN_REFERENCE_POINT_AND_VECTORS =>
  PATTERN_REFERENCE_POINT_3 : MC.POINT_3;
  PATTERN_REFERENCE_VECTORS : MC.VECTOR_PAIR_3;

when SET_PATTERN_REFERENCE_POINT =>
  PATTERN_REFERENCE_POINT : MC.POINT_2;

-- PHIGS Name Set Elements

when ADD_NAMES_TO_SET =>
  NAMES_TO_ADD : NAME_SET;

when REMOVE_NAMES_FROM_SET =>
  NAMES_TO_REMOVE : NAME_SET;

-- PHIGS ASF Elements

when SET_INDIVIDUAL_ASF =>
  ATTRIBUTE_ID : ASPECT;
  SOURCE_FLAG : ASF;

-- PHIGS HLHSR Elements

when SET_HLHSR_IDENTIFIER =>
  HLHSR_IDENTIFIER : HLHSR_ID;

-- PHIGS Transformation Elements

when SET_LOCAL_TRANSFORMATION_3 =>
  LOCAL_MATRIX_3 : TRANSFORMATION_MATRIX_3;
  HOW_APPLIED_3 : COMPOSITION_TYPE;

when SET_LOCAL_TRANSFORMATION =>
  LOCAL_MATRIX : TRANSFORMATION_MATRIX_2;
  HOW_APPLIED : COMPOSITION_TYPE;

when SET_GLOBAL_TRANSFORMATION_3 =>
  GLOBAL_MATRIX_3 : TRANSFORMATION_MATRIX_3;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567424 896 ■


```
when SET_GLOBAL_TRANSFORMATION =>
  GLOBAL_MATRIX : TRANSFORMATION_MATRIX_2;

when SET_MODELLING_CLIPPING_VOLUME_3 =>
  MODELLING_CLIPPING_OPERATOR_3
                        : MODELLING_CLIP_OPERATION_TYPE;
  MODELLING_CLIPPING_LIMITS_3
                        : MC.ACCESS_HALF_SPACE_LIST_3;

when SET_MODELLING_CLIPPING_VOLUME =>
  MODELLING_CLIPPING_OPERATOR
                        : MODELLING_CLIP_OPERATION_TYPE;
  MODELLING_CLIPPING_LIMITS : MC.ACCESS_HALF_SPACE_LIST_2;

when SET_MODELLING_CLIPPING_INDICATOR =>
  MODELLING_CLIPPING_INDICATOR : CLIPPING_INDICATOR;

when RESTORE_MODELLING_CLIPPING_VOLUME =>
  null;

when SET_VIEW_INDEX =>
  VIEW_IND : VIEW_INDEX;

-- PHIGS Invocation Elements

when EXECUTE_STRUCTURE =>
  STRUCTURE_IDENTIFIER : STRUCTURE_ID;

-- PHIGS Structure Content Identification Elements

when LABEL =>
  LABEL_IDENTIFIER : LABEL_ID;

when APPLICATION_DATA =>
  DATA : APPLICATION_DATA_RECORD;

when GSE =>
  GSE_DATA : GSE_RECORD;

when SET_PICK_IDENTIFIER =>
  PICK_IDENTIFIER : PICK_ID;

-- PHIGS PLUS Primitive Elements

when POLYLINE_SET_3_WITH_COLOUR =>
  POLYLINE_SET_3_VERTICES
                        : MC.ACCESS_LIST_OF_VERTEX_COLOUR_LIST_SET_3;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567425 722 ■


```
when FILL_AREA_SET_3_WITH_DATA =>
  FILL_AREA_SET_3_FACET : MC.ACCESS_FACET_DATA_SET;
  FILL_AREA_SET_3_EDGES : ACCESS_LIST_OF_EDGE_FLAG_LIST;
  FILL_AREA_SET_3_VERTICES
                        : MC.ACCESS_LIST_OF_VERTEX_DATA_LIST_SET_3;

when FILL_AREA_SET_WITH_DATA =>
  FILL_AREA_SET_FACET : MC.ACCESS_FACET_DATA_SET;
  FILL_AREA_SET_EDGES : ACCESS_LIST_OF_EDGE_FLAG_LIST;
  FILL_AREA_SET_VERTICES
                        : MC.ACCESS_LIST_OF_VERTEX_DATA_LIST_SET_2;

when CELL_ARRAY_3_PLUS =>
  CORNER_P_3_PLUS : MC.POINT_3;
  CORNER_Q_3_PLUS : MC.POINT_3;
  CORNER_R_3_PLUS : MC.POINT_3;
  CELL_ARRAY_3_PLUS_CELLS : ACCESS_COLOUR_VALUE_ARRAY;

when SET_OF_FILL_AREA_SETS_3_WITH_DATA =>
  SET_OF_FILL_AREA_SETS_3_FACETS
                        : MC.ACCESS_FACET_DATA_LIST_SET;
  SET_OF_FILL_AREA_SETS_3_EDGES
                        : ACCESS_LIST_OF_LIST_OF_EDGE_FLAG_LIST;
  SET_OF_FILL_AREA_SETS_3_VERTICES
                        : MC.ACCESS_VERTEX_DATA_LIST_SET_3;
  SET_OF_FILL_AREA_SETS_3_INDICES
                        : ACCESS_LIST_OF_LIST_OF_VERTEX_INDEX_LIST;

when SET_OF_FILL_AREA_SETS_WITH_DATA =>
  SET_OF_FILL_AREA_SETS_FACETS : MC.ACCESS_FACET_DATA_LIST_SET;
  SET_OF_FILL_AREA_SETS_EDGES
                        : ACCESS_LIST_OF_LIST_OF_EDGE_FLAG_LIST;
  SET_OF_FILL_AREA_SETS_VERTICES
                        : MC.ACCESS_VERTEX_DATA_LIST_SET_2;
  SET_OF_FILL_AREA_SETS_INDICES
                        : ACCESS_LIST_OF_LIST_OF_VERTEX_INDEX_LIST;

when TRIANGLE_SET_3_WITH_DATA =>
  TRIANGLE_SET_3_FACETS : MC.ACCESS_FACET_DATA_LIST_SET;
  TRIANGLE_SET_3_EDGES : ACCESS_LIST_OF_EDGE_FLAG_TRIPLET;
  TRIANGLE_SET_3_VERTICES : MC.ACCESS_VERTEX_DATA_LIST_SET_3;
  TRIANGLE_SET_3_INDICES
                        : ACCESS_LIST_OF_VERTEX_INDEX_TRIPLET;

when TRIANGLE_SET_WITH_DATA =>
  TRIANGLE_SET_FACETS : MC.ACCESS_FACET_DATA_LIST_SET;
  TRIANGLE_SET_EDGES : ACCESS_LIST_OF_EDGE_FLAG_TRIPLET;
  TRIANGLE_SET_VERTICES : MC.ACCESS_VERTEX_DATA_LIST_SET_2;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

9593 PT*3 90 ■ 4851903 0567426 669 ■


```
    TRIANGLE_SET_INDICES
                        : ACCESS_LIST_OF_VERTEX_INDEX_TRIPLET;

  when TRIANGLE_STRIP_3_WITH_DATA =>
    TRIANGLE_STRIP_3_FACETS : MC.ACCESS_FACET_DATA_LIST_SET;
    TRIANGLE_STRIP_3_EDGES : ACCESS_EDGE_FLAG_LIST;
    TRIANGLE_STRIP_3_VERTICES
                        : MC.ACCESS_VERTEX_DATA_LIST_SET_3;

  when TRIANGLE_STRIP_WITH_DATA =>
    TRIANGLE_STRIP_FACETS : MC.ACCESS_FACET_DATA_LIST_SET;
    TRIANGLE_STRIP_EDGES : ACCESS_EDGE_FLAG_LIST;
    TRIANGLE_STRIP_VERTICES : MC.ACCESS_VERTEX_DATA_LIST_SET_2;

  when QUADRILATERAL_MESH_3_WITH_DATA =>
    QUADRILATERAL_MESH_3_FACETS
                        : MC.ACCESS_FACET_DATA_ARRAY_SET;
    QUADRILATERAL_MESH_3_EDGES
                        : ACCESS_ARRAY_OF_EDGE_FLAG_PAIR;
    QUADRILATERAL_MESH_3_VERTICES
                        : MC.ACCESS_VERTEX_DATA_ARRAY_SET_3;

  when QUADRILATERAL_MESH_WITH_DATA =>
    QUADRILATERAL_MESH_FACETS
                        : MC.ACCESS_FACET_DATA_ARRAY_SET;
    QUADRILATERAL_MESH_EDGES
                        : ACCESS_ARRAY_OF_EDGE_FLAG_PAIR;
    QUADRILATERAL_MESH_VERTICES
                        : MC.ACCESS_VERTEX_DATA_ARRAY_SET_2;

  when NON_UNIFORM_B_SPLINE_CURVE =>
    NURB_CURVE_SPLINE : MC.ACCESS_CURVE_GEOMETRYSPLINE_3;
    NURB_CURVE_LIMITS : SPC.RANGE_OF_MAGNITUDES;

  when NON_UNIFORM_B_SPLINE_CURVE_WITH_COLOUR =>
    NURB_CURVE_COLOUR_GEOMETRYSPLINE
                        : MC.ACCESS_CURVE_GEOMETRYSPLINE_3;
    NURB_CURVE_COLOUR_LIMITS : SPC.RANGE_OF_MAGNITUDES;
    NURB_CURVE_COLOUR_COLOURSPLINE
                        : ACCESS_CURVE_COLOURSPLINE;

  when NON_UNIFORM_B_SPLINE_SURFACE =>
    NURB_SURFACE_GEOMETRYSPLINE
                        : MC.ACCESS_SURFACE_GEOMETRYSPLINE;
    NURB_SURFACE_TRIM_CURVES
                        : ACCESS_LIST_OF_TRIMCURVE_LIST;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567427 5T5 ■


```
when NON_UNIFORM_B_SPLINE_SURFACE_WITH_DATA =>
  NURB_SURFACE_DATA_GEOMETRYSPLINE
                        : MC.ACCESS_SURFACE_GEOMETRYSPLINE;
  NURB_SURFACE_DATA_TRIM_CURVES
                        : ACCESS_LIST_OF_TRIMCURVE_LIST;
  NURB_SURFACE_DATA_COLOURSPLINE
                        : ACCESS_SURFACE_COLOURSPLINE;
  NURB_SURFACE_DATA_DATASPLINES
                              : ACCESS_DATASPLINE_LIST;

-- PHIGS PLUS Bundle Index Elements

when SET_DATA_MAPPING_INDEX =>
  DATA_MAPPING_IND : DATA_MAPPING_INDEX;

when SET_REFLECTANCE_INDEX =>
  REFLECTANCE_IND : REFLECTANCE_INDEX;

when SET_BACK_INTERIOR_INDEX =>
  BACK_INTERIOR_IND : INTERIOR_INDEX;

when SET_BACK_DATA_MAPPING_INDEX =>
  BACK_DATA_MAPPING_IND : DATA_MAPPING_INDEX;

when SET_BACK_REFLECTANCE_INDEX =>
  BACK_REFLECTANCE_IND : REFLECTANCE_INDEX;

when SET_PARAMETRIC_SURFACE_INDEX =>
  PARAMETRIC_SURFACE_IND : PARAMETRIC_SURFACE_INDEX;

-- PHIGS PLUS Individual Aspect Elements

when SET_POLYLINE_COLOUR =>
  GENERAL_POLYLINE_COLOUR : GENERAL_COLOUR;

when SET_POLYLINE_SHADING_METHOD =>
  POLYLINE_SHADING : POLYLINE_SHADING_METHOD;

when SET_POLYMARKER_COLOUR =>
  GENERAL_POLYMARKER_COLOUR : GENERAL_COLOUR;

when SET_TEXT_COLOUR =>
  GENERAL_TEXT_COLOUR : GENERAL_COLOUR;

when SET_FACET_DISTINGUISHING_MODE =>
  FACET_DISTINGUISHING : FACET_DISTINGUISHING_MODE;

when SET_FACET_CULLING_MODE =>
  FACET_CULLING : FACET_CULLING_MODE;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
when SET_INTERIOR_COLOUR =>
  GENERAL_INTERIOR_COLOUR : GENERAL_COLOUR;

when SET_INTERIOR_SHADING_METHOD =>
  INTERIOR_SHADING : INTERIOR_SHADING_METHOD;

when SET_DATA_MAPPING_METHOD =>
  METHOD_OF_DATA_MAPPING : DATA_MAPPING_DATA_RECORD;

when SET_REFLECTANCE_PROPERTIES =>
  REFLECTANCE_PROPERTIES : REFLECTANCE_PROPERTIES_DATA_RECORD;

when SET_REFLECTANCE_MODEL =>
  MODEL_OF_REFLECTANCE : REFLECTANCE_MODEL;

when SET_BACK_INTERIOR_STYLE =>
  BACK_STYLE_OF_INTERIOR : INTERIOR_STYLE;

when SET_BACK_INTERIOR_STYLE_INDEX =>
  BACK_STYLE_IND : STYLE_INDEX;

when SET_BACK_INTERIOR_COLOUR =>
  BACK_GENERAL_INTERIOR_COLOUR : GENERAL_COLOUR;

when SET_BACK_INTERIOR_SHADING_METHOD =>
  BACK_INTERIOR_SHADING : INTERIOR_SHADING_METHOD;

when SET_BACK_DATA_MAPPING_METHOD =>
  BACK_METHOD_OF_DATA_MAPPING : DATA_MAPPING_METHOD;

when SET_BACK_REFLECTANCE_PROPERTIES =>
  BACK_REFLECTANCE_PROPERTIES
                    : REFLECTANCE_PROPERTIES_DATA_RECORD;

when SET_BACK_REFLECTANCE_MODEL =>
  BACK_REFLECTANCE : REFLECTANCE_MODEL;

when SET_LIGHT_SOURCE_STATE =>
  ACTIVATION_LIST : LIGHT_SOURCE_INDICES.LIST_OF;
  DEACTIVATION_LIST : LIGHT_SOURCE_INDICES.LIST_OF;

when SET_EDGE_COLOUR =>
  GENERAL_EDGE_COLOUR : GENERAL_COLOUR;

when SET_CURVE_APPROX_CRITERIA =>
  CRITERIA_FOR_CURVE_APPROX : CURVE_APPROX_DATA_RECORD;

when SET_SURFACE_APPROX_CRITERIA =>
  CRITERIA_FOR_SURFACE_APPROX
                    : SURFACE_APPROX_DATA_RECORD;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
            when SET_PARAMETRIC_SURFACE_CHARACTERISTICS =>
              PARAMETRIC_SURFACE_CHARACTERISTICS
                                    : PARAMETRIC_SURFACE_DATA_RECORD;

            when SET_RENDERING_COLOUR_MODEL =>
              RENDERING_COLOUR_MODEL : COLOUR_MODEL;

            when SET_DEPTH_CUE_INDEX =>
              DEPTH_CUE_IND : DEPTH_CUE_INDEX;

            when SET_COLOUR_MAPPING_INDEX =>
              COLOUR_MAPPING_IND : COLOUR_MAPPING_INDEX;

          end case;
        end record;

    -- This type defines the format of structure contents for PHIGS PLUS structures and is used to
    -- return structure elements during inquiry.

    PHIGS_ERROR : exception;

    -- Exception to notify the application program of an error in a
    -- PHIGS procedure should the version of ERROR_HANDLING which raises
    -- an exception be used.

    -- This section contains the constants that provide the PHIGS standard
    -- values defined for some PHIGS/Ada types.  The constants for colour
    -- models are defined earlier.

    -- The following constants define the PHIGS standard line types:

  SOLID_LINE                : constant LINETYPE := 1;
  DASHED_LINE               : constant LINETYPE := 2;
  DOTTED_LINE               : constant LINETYPE := 3;
  DASHED_DOTTED_LINE        : constant LINETYPE := 4;

-- The following constants define the PHIGS standard marker types:

  DOT_MARKER                : constant MARKER_TYPE := 1;
  PLUS_MARKER               : constant MARKER_TYPE := 2;
  STAR_MARKER               : constant MARKER_TYPE := 3;
  ZERO_MARKER               : constant MARKER_TYPE := 4;
  X_MARKER                  : constant MARKER_TYPE := 5;

-- The following constants define the PHIGS standard edgetypes:
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567430 09T ■


```
  SOLID_EDGE                    : constant EDGETYPE := 1;
  DASHED_EDGE                   : constant EDGETYPE := 2;
  DOTTED_EDGE                   : constant EDGETYPE := 3;
  DASHED_DOTTED_EDGE            : constant EDGETYPE := 4;

-- The following constants define the PHIGS standard annotation styles:

  UNCONNECTED_ANNOTATION        : constant ANNOTATION_STYLE := 1;
  LEAD_LINE_ANNOTATION          : constant ANNOTATION_STYLE := 2;

-- The following constants are used for defining the modelling clipping
-- operators specified by PHIGS:

  REPLACE_VOLUME                : constant MODELLING_CLIP_OPERATION_TYPE := 1;
  INTERSECT_VOLUME              : constant MODELLING_CLIP_OPERATION_TYPE := 2;

-- The following constants define the prompt and echo types supported
-- by PHIGS:

  DEFAULT_LOCATOR               : constant LOCATOR_PROMPT_ECHO_TYPE := 1;
  CROSS_HAIR_LOCATOR            : constant LOCATOR_PROMPT_ECHO_TYPE := 2;
  TRACKING_CROSS_LOCATOR        : constant LOCATOR_PROMPT_ECHO_TYPE := 3;
  RUBBER_BAND_LINE_LOCATOR      : constant LOCATOR_PROMPT_ECHO_TYPE := 4;
  RECTANGLE_LOCATOR             : constant LOCATOR_PROMPT_ECHO_TYPE := 5;
  DIGITAL_LOCATOR               : constant LOCATOR_PROMPT_ECHO_TYPE := 6;

  DEFAULT_STROKE                : constant STROKE_PROMPT_ECHO_TYPE := 1;
  DIGITAL_STROKE                : constant STROKE_PROMPT_ECHO_TYPE := 2;
  MARKER_STROKE                 : constant STROKE_PROMPT_ECHO_TYPE := 3;
  LINE_STROKE                   : constant STROKE_PROMPT_ECHO_TYPE := 4;

  DEFAULT_VALUATOR              : constant VALUATOR_PROMPT_ECHO_TYPE := 1;
  GRAPHICAL_VALUATOR            : constant VALUATOR_PROMPT_ECHO_TYPE := 2;
  DIGITAL_VALUATOR              : constant VALUATOR_PROMPT_ECHO_TYPE := 3;

  DEFAULT_CHOICE                : constant CHOICE_PROMPT_ECHO_TYPE := 1;
  PROMPT_ECHO_CHOICE            : constant CHOICE_PROMPT_ECHO_TYPE := 2;
  STRING_PROMPT_CHOICE          : constant CHOICE_PROMPT_ECHO_TYPE := 3;
  STRING_INPUT_CHOICE           : constant CHOICE_PROMPT_ECHO_TYPE := 4;
  STRUCTURE_CHOICE              : constant CHOICE_PROMPT_ECHO_TYPE := 5;

  DEFAULT_PICK                  : constant PICK_PROMPT_ECHO_TYPE := 1;
  GROUP_HIGHLIGHT_PICK          : constant PICK_PROMPT_ECHO_TYPE := 2;
  STRUCTURE_HIGHLIGHT_PICK      : constant PICK_PROMPT_ECHO_TYPE := 3;

  end PHIGS_TYPES;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567431 T26 ■


```
-- The PHIGS Package

with PHIGS_CONFIGURATION, PHIGS_TYPES, PHIGS_NAME_SET_FACILITY;
use PHIGS_CONFIGURATION, PHIGS_TYPES;

package PHIGS is

-- The package PHIGS contains all of the procedures that are required
-- to implement PHIGS.

   -- Private Type Declarations

   -- The following data types are the PHIGS private types and are
   -- included in package PHIGS for ease of manipulation.

   -- METAFILE_DATA_RECORD

   type METAFILE_DATA_RECORD (TYPE_OF_ITEM : METAFILE_ITEM_TYPE := 0) is
     private;

   --  A data record for metafiles.

   -- CHOICE_DATA_RECORD

   type CHOICE_DATA_RECORD
     (PROMPT_ECHO_TYPE : CHOICE_PROMPT_ECHO_TYPE := 1) is private;

   -- Defines a choice input data record.

   -- LOCATOR_DATA_RECORD

   type LOCATOR_DATA_RECORD
     (PROMPT_ECHO_TYPE : LOCATOR_PROMPT_ECHO_TYPE := 1) is private;

   -- Defines a locator input data record.

   -- PICK_DATA_RECORD

   type PICK_DATA_RECORD
     (PROMPT_ECHO_TYPE : PICK_PROMPT_ECHO_TYPE := 1) is private;

   -- Defines a pick input data record.

   -- STRING_DATA_RECORD

   type STRING_DATA_RECORD
     (PROMPT_ECHO_TYPE : STRING_PROMPT_ECHO_TYPE := 1) is private;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
        -- Defines a string input data record.

        -- STROKE_DATA_RECORD

        type STROKE_DATA_RECORD
          (PROMPT_ECHO_TYPE : STROKE_PROMPT_ECHO_TYPE := 1) is private;

        -- Defines a stroke input data record.

        -- VALUATOR_DATA_RECORD

        type VALUATOR_DATA_RECORD
          (PROMPT_ECHO_TYPE : VALUATOR_PROMPT_ECHO_TYPE := 1) is private;

        -- Defines a valuator data record.

    -- PHIGS Procedures

    -- CONTROL FUNCTIONS

        procedure OPEN_PHIGS
          (ERROR_FILE              : in FILE_ID := PHIGS_STRING(DEFAULT_ERROR_FILE);
           AMOUNT_OF_MEMORY : in PHIGS_NATURAL := DEFAULT_MEMORY_UNITS);

        procedure CLOSE_PHIGS;

        procedure OPEN_WS
          (WS          : in WS_ID;
           CONNECTION  : in CONNECTION_ID;
           TYPE_OF_WS  : in WS_TYPE);

        procedure CLOSE_WS
          (WS : in WS_ID);

        procedure REDRAW_ALL_STRUCTURES
          (WS     : in WS_ID;
           FLAG   : in CONTROL_FLAG);

        procedure UPDATE_WS
          (WS                  : in WS_ID;
           REGENERATION  : in UPDATE_REGENERATION_FLAG);

        procedure SET_DISPLAY_UPDATE_STATE
          (WS                  : in WS_ID;
           DEFERRAL        : in DEFERRAL_MODE;
           MODIFICATION  : in MODIFICATION_MODE);

        procedure MESSAGE
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567433 8T9 ■


```
      (WS          : in WS_ID;
       CONTENTS    : in PHIGS_STRING);

  -- OUTPUT FUNCTIONS

    procedure POLYLINE
      (POINTS : in MC.POINT_LIST_3);

    procedure POLYLINE
      (POINTS : in MC.POINT_LIST_2);

    procedure POLYMARKER
      (POINTS : in MC.POINT_LIST_3);

    procedure POLYMARKER
      (POINTS : in MC.POINT_LIST_2);

    procedure TEXT
      (POSITION           : in MC.POINT_3;
       DIRECTION_VECTORS  : in MC.VECTOR_PAIR_3;
       CHAR_STRING        : in PHIGS_STRING);

    procedure TEXT
      (POSITION      : in MC.POINT_2;
       CHAR_STRING   : in PHIGS_STRING);

    procedure ANNOTATION_TEXT_RELATIVE
      (REFERENCE_POINT    : in MC.POINT_3;
       ANNOTATION_OFFSET  : in NPC.POINT_3;
       CHAR_STRING        : in PHIGS_STRING);

    procedure ANNOTATION_TEXT_RELATIVE
      (REFERENCE_POINT    : in MC.POINT_2;
       ANNOTATION_OFFSET  : in NPC.POINT_2;
       CHAR_STRING        : in PHIGS_STRING);

    procedure FILL_AREA
      (POINTS : in MC.POINT_LIST_3);

    procedure FILL_AREA
      (POINTS : in MC.POINT_LIST_2);

    procedure FILL_AREA_SET
      (POINT_LISTS : in MC.LIST_OF_POINT_LIST_3);

    procedure FILL_AREA_SET
      (POINT_LISTS : in MC.LIST_OF_POINT_LIST_2);

    procedure CELL_ARRAY
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567434 735 ■


```
    (CORNER_P   : in MC.POINT_3;
     CORNER_Q   : in MC.POINT_3;
     CORNER_R   : in MC.POINT_3;
     CELLS      : in COLOUR_MATRIX);

  procedure CELL_ARRAY
    (CORNER_P   : in MC.POINT_2;
     CORNER_Q   : in MC.POINT_2;
     CELLS      : in COLOUR_MATRIX);


  procedure POLYLINE_SET
    (VERTICES: in MC.LIST_OF_VERTEX_COLOUR_LIST_SET_3);

  procedure FILL_AREA_SET_WITH_DATA
    (FACET      : in MC.FACET_DATA_SET;
     VERTICES   : in MC.LIST_OF_VERTEX_DATA_LIST_SET_3);

  procedure FILL_AREA_SET_WITH_DATA
    (FACET      : in MC.FACET_DATA_SET;
     EDGES      : in LIST_OF_EDGE_FLAG_LIST;
     VERTICES   : in MC.LIST_OF_VERTEX_DATA_LIST_SET_3);

  procedure FILL_AREA_SET_WITH_DATA
    (FACET      : in MC.FACET_DATA_SET;
     VERTICES   : in MC.LIST_OF_VERTEX_DATA_LIST_SET_2);

  procedure FILL_AREA_SET_WITH_DATA
    (FACET      : in MC.FACET_DATA_SET;
     EDGES      : in LIST_OF_EDGE_FLAG_LIST;
     VERTICES   : in MC.LIST_OF_VERTEX_DATA_LIST_SET_2);

  procedure CELL_ARRAY
    (CORNER_P   : in MC.POINT_3;
     CORNER_Q   : in MC.POINT_3;
     CORNER_R   : in MC.POINT_3;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567435 671 ■


```
    CELLS       : in COLOUR_VALUE_ARRAY);

procedure SET_OF_FILL_AREA_SET_WITH_DATA
  (FACETS     : in MC.FACET_DATA_LIST_SET;
   VERTICES   : in MC.VERTEX_DATA_LIST_SET_3;
   INDICES    : in LIST_OF_LIST_OF_VERTEX_INDEX_LIST);

procedure SET_OF_FILL_AREA_SET_WITH_DATA
  (FACETS     : in MC.FACET_DATA_LIST_SET;
   EDGES      : in LIST_OF_LIST_OF_EDGE_FLAG_LIST;
   VERTICES   : in MC.VERTEX_DATA_LIST_SET_3;
   INDICES    : in LIST_OF_LIST_OF_VERTEX_INDEX_LIST);

procedure SET_OF_FILL_AREA_SET_WITH_DATA
  (FACETS     : in MC.FACET_DATA_LIST_SET;
   VERTICES   : in MC.VERTEX_DATA_LIST_SET_2;
   INDICES    : in LIST_OF_LIST_OF_VERTEX_INDEX_LIST);

procedure SET_OF_FILL_AREA_SET_WITH_DATA
  (FACETS     : in MC.FACET_DATA_LIST_SET;
   EDGES      : in LIST_OF_LIST_OF_EDGE_FLAG_LIST;
   VERTICES   : in MC.VERTEX_DATA_LIST_SET_2;
   INDICES    : in LIST_OF_LIST_OF_VERTEX_INDEX_LIST);

procedure TRIANGLE_SET_WITH_DATA
  (FACETS     : in MC.FACET_DATA_LIST_SET;
   VERTICES   : in MC.VERTEX_DATA_LIST_SET_3;
   INDICES    : in LIST_OF_VERTEX_INDEX_TRIPLET);

procedure TRIANGLE_SET_WITH_DATA
  (FACETS     : in MC.FACET_DATA_LIST_SET;
   EDGES      : in LIST_OF_EDGE_FLAG_TRIPLET;
   VERTICES   : in MC.VERTEX_DATA_LIST_SET_3;
   INDICES    : in LIST_OF_VERTEX_INDEX_TRIPLET);

procedure TRIANGLE_SET_WITH_DATA
  (FACETS     : in MC.FACET_DATA_LIST_SET;
   VERTICES   : in MC.VERTEX_DATA_LIST_SET_2;
   INDICES    : in LIST_OF_VERTEX_INDEX_TRIPLET);

procedure TRIANGLE_SET_WITH_DATA
  (FACETS     : in MC.FACET_DATA_LIST_SET;
   EDGES      : in LIST_OF_EDGE_FLAG_TRIPLET;
   VERTICES   : in MC.VERTEX_DATA_LIST_SET_2;
   INDICES    : in LIST_OF_VERTEX_INDEX_TRIPLET);

procedure TRIANGLE_STRIP_WITH_DATA
  (FACETS     : in MC.FACET_DATA_LIST_SET;
   VERTICES   : in MC.VERTEX_DATA_LIST_SET_3);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567436 508 ■


```
procedure TRIANGLE_STRIP_WITH_DATA
  (FACETS      : in MC.FACET_DATA_LIST_SET;
   EDGES       : in EDGE_FLAG_LIST;
   VERTICES    : in MC.VERTEX_DATA_LIST_SET_3);

procedure TRIANGLE_STRIP_WITH_DATA
  (FACETS      : in MC.FACET_DATA_LIST_SET;
   VERTICES    : in MC.VERTEX_DATA_LIST_SET_2);

procedure TRIANGLE_STRIP_WITH_DATA
  (FACETS      : in MC.FACET_DATA_LIST_SET;
   EDGES       : in EDGE_FLAG_LIST;
   VERTICES    : in MC.VERTEX_DATA_LIST_SET_2);

procedure QUADRILATERAL_MESH_WITH_DATA
  (FACETS      : in MC.FACET_DATA_ARRAY_SET;
   VERTICES    : in MC.VERTEX_DATA_ARRAY_SET_3);

procedure QUADRILATERAL_MESH_WITH_DATA
  (FACETS      : in MC.FACET_DATA_ARRAY_SET;
   EDGES       : in ARRAY_OF_EDGE_FLAG_PAIR;
   VERTICES    : in MC.VERTEX_DATA_ARRAY_SET_3);

procedure QUADRILATERAL_MESH_WITH_DATA
  (FACETS      : in MC.FACET_DATA_ARRAY_SET;
   VERTICES    : in MC.VERTEX_DATA_ARRAY_SET_2);

procedure QUADRILATERAL_MESH_WITH_DATA
  (FACETS      : in MC.FACET_DATA_ARRAY_SET;
   EDGES       : in ARRAY_OF_EDGE_FLAG_PAIR;
   VERTICES    : in MC.VERTEX_DATA_ARRAY_SET_2);

procedure NON_UNIFORM_B_SPLINE_CURVE
  (SPLINE    : in MC.CURVE_GEOMETRYSPLINE_3;
   LIMITS    : in SPC.RANGE_OF_MAGNITUDES);

procedure NON_UNIFORM_B_SPLINE_CURVE
  (CURVE_SPLINE : in MC.CURVE_GEOMETRYSPLINE_3;
   LIMITS       : in SPC.RANGE_OF_MAGNITUDES;
   COLOUR       : in CURVE_COLOURSPLINE);

procedure NON_UNIFORM_B_SPLINE_SURFACE
   (SURFACE_SPLINE : in MC.SURFACE_GEOMETRYSPLINE;
   TRIM_CURVES     : in LIST_OF_TRIMCURVE_LIST);

procedure NON_UNIFORM_B_SPLINE_SURFACE_WITH_DATA
  (SURFACE_SPLINE : in MC.SURFACE_GEOMETRYSPLINE;
   TRIM_CURVES    : in LIST_OF_TRIMCURVE_LIST;
   COLOUR_SPLINE  : in SURFACE_COLOURSPLINE;
   DATA_SPLINES   : in DATASPLINE_LIST);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
-- OUTPUT ATTRIBUTE FUNCTIONS

   procedure SET_POLYLINE_INDEX
     (POLYLINE_IND : in POLYLINE_INDEX);

   procedure SET_POLYMARKER_INDEX
     (POLYMARKER_IND : in POLYMARKER_INDEX);

   procedure SET_TEXT_INDEX
     (TEXT_IND : in TEXT_INDEX);

   procedure SET_INTERIOR_INDEX
     (INTERIOR_IND : in INTERIOR_INDEX);

   procedure SET_EDGE_INDEX
     (EDGE_IND : in EDGE_INDEX);

   procedure SET_DATA_MAPPING_INDEX
     (DATA_MAPPING_IND : in DATA_MAPPING_INDEX);

   procedure SET_REFLECTANCE_INDEX
     (REFLECTANCE_IND : in REFLECTANCE_INDEX);

   procedure SET_BACK_INTERIOR_INDEX
     (BACK_INTERIOR_IND : in INTERIOR_INDEX);

   procedure SET_BACK_DATA_MAPPING_INDEX
     (BACK_DATA_MAPPING_IND : in DATA_MAPPING_INDEX);

   procedure SET_BACK_REFLECTANCE_INDEX
     (BACK_REFLECTANCE_IND : in REFLECTANCE_INDEX);

   procedure SET_PARAMETRIC_SURFACE_INDEX
     (PARAMETRIC_SURFACE_IND : in PARAMETRIC_SURFACE_INDEX);

   procedure SET_LINETYPE
     (TYPE_OF_LINE : in LINETYPE);

   procedure SET_LINEWIDTH_SCALE_FACTOR
     (LINEWIDTH_SF : in LINEWIDTH);

   procedure SET_POLYLINE_COLOUR_INDEX
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567438 380 ■


```
    (LINE_COLOUR : in COLOUR_INDEX);

  procedure SET_POLYLINE_COLOUR
    (GENERAL_POLYLINE_COLOUR : in GENERAL_COLOUR);

  procedure SET_POLYLINE_SHADING_METHOD
    (POLYLINE_SHADING : in POLYLINE_SHADING_METHOD);

  procedure SET_MARKER_TYPE
    (TYPE_OF_MARKER : in MARKER_TYPE);

  procedure SET_MARKER_SIZE_SCALE_FACTOR
    (SIZE : in MARKER_SIZE);

  procedure SET_POLYMARKER_COLOUR_INDEX
    (MARKER_COLOUR : in COLOUR_INDEX);

  procedure SET_POLYMARKER_COLOUR
    (GENERAL_POLYMARKER_COLOUR : in GENERAL_COLOUR);

  procedure SET_TEXT_FONT
    (FONT : in TEXT_FONT);

  procedure SET_TEXT_PRECISION
    (PRECISION : in TEXT_PRECISION);

  procedure SET_CHAR_EXPANSION_FACTOR
    (EXPANSION : in CHAR_EXPANSION);

  procedure SET_CHAR_SPACING
    (SPACING : in CHAR_SPACING);

  procedure SET_TEXT_COLOUR_INDEX
    (TEXT_COLOUR : in COLOUR_INDEX);

  procedure SET_TEXT_COLOUR
    (GENERAL_TEXT_COLOUR : in GENERAL_COLOUR);

  procedure SET_CHAR_HEIGHT
    (HEIGHT : in MC.MAGNITUDE);

  procedure SET_CHAR_UP_VECTOR
    (CHAR_UP_VECTOR : in MC.VECTOR_2);

  procedure SET_TEXT_PATH
    (PATH : in TEXT_PATH);

  procedure SET_TEXT_ALIGNMENT
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567439 217 ■


```
    (ALIGNMENT : in TEXT_ALIGNMENT);

  procedure SET_ANNOTATION_TEXT_CHAR_HEIGHT
    (ANNOTATION_HEIGHT : in NPC.MAGNITUDE);

  procedure SET_ANNOTATION_TEXT_CHAR_UP_VECTOR
    (ANNOTATION_CHAR_UP_VECTOR : in NPC.VECTOR_2);

  procedure SET_ANNOTATION_TEXT_PATH
    (ANNOTATION_PATH : in TEXT_PATH);

  procedure SET_ANNOTATION_TEXT_ALIGNMENT
    (ANNOTATION_ALIGNMENT : in TEXT_ALIGNMENT);

  procedure SET_ANNOTATION_STYLE
    (STYLE_OF_ANNOTATION : in ANNOTATION_STYLE);

  procedure SET_INTERIOR_STYLE
    (STYLE_OF_INTERIOR : in INTERIOR_STYLE);

  procedure SET_INTERIOR_STYLE_INDEX
    (STYLE_IND : in STYLE_INDEX);

  procedure SET_FACET_DISTINGUISHING_MODE
    (FACET_DISTINGUISHING : in FACET_DISTINGUISHING_MODE);

  procedure SET_FACET_CULLING_MODE
    (FACET_CULLING : in FACET_CULLING_MODE);

  procedure SET_INTERIOR_COLOUR_INDEX
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567440 T39 ■


```
    (INTERIOR_COLOUR : in COLOUR_INDEX);

  procedure SET_INTERIOR_COLOUR
    (GENERAL_INTERIOR_COLOUR : in GENERAL_COLOUR);

  procedure SET_INTERIOR_SHADING_METHOD
    (INTERIOR_SHADING : in INTERIOR_SHADING_METHOD);

  procedure SET_DATA_MAPPING_METHOD
    (METHOD_OF_DATA_MAPPING : in DATA_MAPPING_DATA_RECORD);

  procedure SET_REFLECTANCE_PROPERTIES
     (REFLECTANCE_DATA : in REFLECTANCE_PROPERTIES_DATA_RECORD);

  procedure SET_REFLECTANCE_MODEL
    (REFLECTANCE : in REFLECTANCE_MODEL);

  procedure SET_BACK_INTERIOR_STYLE
    (BACK_STYLE_OF_INTERIOR : in INTERIOR_STYLE);

  procedure SET_BACK_INTERIOR_STYLE_INDEX
    (BACK_STYLE_IND : in STYLE_INDEX);

  procedure SET_BACK_INTERIOR_COLOUR
    (BACK_GENERAL_INTERIOR_COLOUR : in GENERAL_COLOUR);

  procedure SET_BACK_INTERIOR_SHADING_METHOD
    (BACK_INTERIOR_SHADING : in INTERIOR_SHADING_METHOD);

  procedure SET_BACK_DATA_MAPPING_METHOD
    (BACK_METHOD_OF_DATA_MAPPING : in DATA_MAPPING_DATA_RECORD);

  procedure SET_BACK_REFLECTANCE_PROPERTIES
     (BACK_REFLECTANCE_DATA : in REFLECTANCE_PROPERTIES_DATA_RECORD);

  procedure SET_BACK_REFLECTANCE_MODEL
    (BACK_REFLECTANCE : in REFLECTANCE_MODEL);

  procedure SET_LIGHT_SOURCE_STATE
    (ACTIVATION_LIST      : in LIGHT_SOURCE_INDICES.LIST_OF;
     DEACTIVATION_LIST  : in LIGHT_SOURCE_INDICES.LIST_OF);

  procedure SET_EDGE_FLAG
    (FLAG : in EDGE_FLAG);

  procedure SET_EDGETYPE
    (TYPE_OF_EDGE : in EDGETYPE);

  procedure SET_EDGEWIDTH_SCALE_FACTOR
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
    (EDGEWIDTH_SF : in EDGEWIDTH);

  procedure SET_EDGE_COLOUR_INDEX
    (EDGE_COLOUR : in COLOUR_INDEX);

  procedure SET_EDGE_COLOUR
    (GENERAL_EDGE_COLOUR : GENERAL_COLOUR);

  procedure SET_PATTERN_SIZE
    (SIZE : in MC.SIZE_2);

  procedure SET_PATTERN_REFERENCE_POINT_AND_VECTORS
    (REFERENCE_POINT      : in MC.POINT_3;
     REFERENCE_VECTORS    : in MC.VECTOR_PAIR_3);

  procedure SET_PATTERN_REFERENCE_POINT
    (REFERENCE_POINT : MC.POINT_2);

  procedure SET_CURVE_APPROXIMATION_CRITERIA
    (CRITERIA_FOR_CURVE_APPROX : in CURVE_APPROX_DATA_RECORD);

  procedure SET_SURFACE_APPROXIMATION_CRITERIA
    (CRITERIA_FOR_SURFACE_APPROX :  in  SURFACE_APPROX_DATA_RECORD);

  procedure SET_PARAMETRIC_SURFACE_CHARACTERISTICS
    (PARAMETRIC_SURFACE_CHARACTERISTICS
                                    : in PARAMETRIC_SURFACE_DATA_RECORD);

  procedure SET_RENDERING_COLOUR_MODEL
    (RENDERING_COLOUR_MODEL : in COLOUR_MODEL);

  procedure SET_DEPTH_CUE_INDEX
    (DEPTH_CUE_IND : in DEPTH_CUE_INDEX);

  procedure SET_COLOUR_MAPPING_INDEX
    (COLOUR_MAPPING_IND : in COLOUR_MAPPING_INDEX);

  procedure ADD_NAMES_TO_SET
    (NAMES_TO_ADD : in NAME_SET);

  procedure REMOVE_NAMES_FROM_SET
    (NAMES_TO_REMOVE : in NAME_SET);

  procedure SET_INDIVIDUAL_ASF
    (ASPECT_ID       : in ASPECT;
     SOURCE_FLAG     : in ASF);

  procedure SET_POLYLINE_REPRESENTATION
    (WS              : in WS_ID;
     POLYLINE_IND    : in POLYLINE_INDEX;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567442 801 ■


```
    TYPE_OF_LINE    : in LINETYPE;
    LINEWIDTH_SF    : in LINEWIDTH;
    LINE_COLOUR     : in COLOUR_INDEX);

procedure SET_POLYLINE_REPRESENTATION
  (WS                              : in WS_ID;
   POLYLINE_IND                    : in POLYLINE_INDEX;
   TYPE_OF_LINE                    : in LINETYPE;
   LINEWIDTH_SF                    : in LINEWIDTH;
   GENERAL_LINE_COLOUR             : in GENERAL_COLOUR;
   LINE_SHADING                    : in POLYLINE_SHADING_METHOD;
   CRITERIA_FOR_CURVE_APPROX       : in CURVE_APPROX_DATA_RECORD);

procedure SET_POLYMARKER_REPRESENTATION
  (WS                  : in WS_ID;
   POLYMARKER_IND      : in POLYMARKER_INDEX;
   TYPE_OF_MARKER      : in MARKER_TYPE;
   SIZE                : in MARKER_SIZE;
   MARKER_COLOUR       : in COLOUR_INDEX);

procedure SET_POLYMARKER_REPRESENTATION
  (WS                              : in WS_ID;
   POLYMARKER_IND                  : in POLYMARKER_INDEX;
   TYPE_OF_MARKER                  : in MARKER_TYPE;
   SIZE                            : in MARKER_SIZE;
   GENERAL_MARKER_COLOUR           : in GENERAL_COLOUR);

procedure SET_TEXT_REPRESENTATION
  (WS                  : in WS_ID;
   TEXT_IND            : in TEXT_INDEX;
   FONT                : in TEXT_FONT;
   PRECISION           : in TEXT_PRECISION;
   EXPANSION           : in CHAR_EXPANSION;
   SPACING             : in CHAR_SPACING;
   TEXT_COLOUR         : in COLOUR_INDEX);

procedure SET_TEXT_REPRESENTATION
  (WS                              : in WS_ID;
   TEXT_IND                        : in TEXT_INDEX;
   FONT                            : in TEXT_FONT;
   PRECISION                       : in TEXT_PRECISION;
   EXPANSION                       : in CHAR_EXPANSION;
   SPACING                         : in CHAR_SPACING;
   GENERAL_TEXT_COLOUR             : in GENERAL_COLOUR);

procedure SET_INTERIOR_REPRESENTATION
  (WS                    : in WS_ID;
   INTERIOR_IND          : in INTERIOR_INDEX;
   STYLE_OF_INTERIOR     : in INTERIOR_STYLE;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
    STYLE_IND              : in STYLE_INDEX;
    INTERIOR_COLOUR        : in COLOUR_INDEX);

procedure SET_INTERIOR_REPRESENTATION
  (WS                              : in WS_ID;
   INTERIOR_IND                    : in INTERIOR_INDEX;
   STYLE_OF_INTERIOR               : in INTERIOR_STYLE;
   STYLE_IND                       : in STYLE_INDEX;
   GENERAL_INTERIOR_COLOUR         : in GENERAL_COLOUR;
   INTERIOR_SHADING                : in INTERIOR_SHADING_METHOD);

procedure SET_EDGE_REPRESENTATION
  (WS                 : in WS_ID;
   EDGE_IND           : in EDGE_INDEX;
   FLAG               : in EDGE_FLAG;
   TYPE_OF_EDGE       : in EDGETYPE;
   EDGEWIDTH_SF       : in EDGEWIDTH;
   EDGE_COLOUR        : in COLOUR_INDEX);

procedure SET_EDGE_REPRESENTATION
  (WS                           : in WS_ID;
   EDGE_IND                     : in EDGE_INDEX;
   FLAG                         : in EDGE_FLAG;
   TYPE_OF_EDGE                 : in EDGETYPE;
   EDGEWIDTH_SF                 : in EDGEWIDTH;
   GENERAL_EDGE_COLOUR          : in GENERAL_COLOUR);

procedure SET_DATA_MAPPING_REPRESENTATION
  (WS                    : in WS_ID;
   DATA_MAPPING_IND      : in DATA_MAPPING_INDEX;
   DATA_MAPPING          : in DATA_MAPPING_DATA_RECORD);

procedure SET_REFLECTANCE_REPRESENTATION
  (WS                              : in WS_ID;
   REFLECTANCE_IND                 : in REFLECTANCE_INDEX;
   MODEL_OF_REFLECTANCE            : in REFLECTANCE_MODEL;
   REFLECTANCE_PROPERTIES          : in REFLECTANCE_PROPERTIES_DATA_RECORD);

procedure SET_PARAMETRIC_SURFACE_REPRESENTATION
  (WS                                    : in WS_ID;
   PARAMETRIC_SURFACE_IND                : in PARAMETRIC_SURFACE_INDEX;
   CRITERIA_FOR_SURFACE_APPROX           : in SURFACE_APPROX_DATA_RECORD;
   PARAMETRIC_SURFACE_CHARACTERISTICS
                                         : in PARAMETRIC_SURFACE_DATA_RECORD);

procedure SET_PATTERN_REPRESENTATION
  (WS                 : in WS_ID;
   PATTERN_IND        : in PATTERN_INDEX;
   PATTERN            : in COLOUR_MATRIX);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
    procedure SET_PATTERN_REPRESENTATION
      (WS                  : in WS_ID;
       PATTERN_IND         : in PATTERN_INDEX;
       GENERAL_PATTERN     : in COLOUR_VALUE_ARRAY);

    procedure SET_LIGHT_SOURCE_REPRESENTATION
      (WS                  : in WS_ID;
       LIGHT_SOURCE_IND    : in LIGHT_SOURCE_INDEX;
       LIGHT_SOURCE        : in LIGHT_SOURCE_DATA_RECORD);

    procedure SET_DEPTH_CUE_REPRESENTATION
      (WS                  : in WS_ID;
       DEPTH_CUE_IND       : in DEPTH_CUE_INDEX;
       MODE                : in DEPTH_CUE_MODE;
       REFERENCE_PLANES    : in NPC.RANGE_OF_MAGNITUDES;
       DEPTH_CUE_SF        : in DEPTH_CUE_SCALE_FACTORS;
       DEPTH_CUE_COLOUR    : in GENERAL_COLOUR);

    procedure SET_COLOUR_MAPPING_REPRESENTATION
      (WS                  : in WS_ID;
       COLOUR_MAPPING_IND  : in COLOUR_MAPPING_INDEX;
       COLOUR_MAPPING      : in COLOUR_MAPPING_DATA_RECORD);

    procedure SET_COLOUR_REPRESENTATION
      (WS                  : in WS_ID;
       COLOUR_IND          : in COLOUR_INDEX;
       COLOUR              : in COLOUR_REPRESENTATION);

    procedure SET_HIGHLIGHTING_FILTER
      (WS                  : in WS_ID;
       HIGHLIGHTING        : in NAME_SET_FILTER);


    procedure SET_INVISIBILITY_FILTER
      (WS                  : in WS_ID;
       INVISIBILITY        : in NAME_SET_FILTER);

    procedure SET_COLOUR_MODEL
      (WS                  : in WS_ID;
       MODEL               : in COLOUR_MODEL);

    procedure SET_HLHSR_IDENTIFIER
      (HLHSR_IDENTIFIER    : in HLHSR_ID);

    procedure SET_HLHSR_MODE
      (WS                  : in WS_ID;
       MODE                : in HLHSR_MODE);

  -- TRANSFORMATION FUNCTIONS
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567445 510 ■


```
procedure SET_LOCAL_TRANSFORMATION
  (MATRIX          : in TRANSFORMATION_MATRIX_3;
   HOW_APPLIED     : in COMPOSITION_TYPE);

procedure SET_LOCAL_TRANSFORMATION
  (MATRIX          : in TRANSFORMATION_MATRIX_2;
   HOW_APPLIED     : in COMPOSITION_TYPE);

procedure SET_GLOBAL_TRANSFORMATION
  (MATRIX : in TRANSFORMATION_MATRIX_3);

procedure SET_GLOBAL_TRANSFORMATION
  (MATRIX : in TRANSFORMATION_MATRIX_2);

procedure SET_MODELLING_CLIPPING_VOLUME
  (OPERATOR        : in MODELLING_CLIP_OPERATION_TYPE;
   HALF_SPACES     : in MC.HALF_SPACE_LIST_3);

procedure SET_MODELLING_CLIPPING_VOLUME
  (OPERATOR        : in MODELLING_CLIP_OPERATION_TYPE;
   HALF_SPACES     : in MC.HALF_SPACE_LIST_2);

procedure SET_MODELLING_CLIPPING_INDICATOR
  (MODELLING_CLIPPING : in CLIPPING_INDICATOR);

procedure RESTORE_MODELLING_CLIPPING_VOLUME;

procedure SET_VIEW_INDEX
  (VIEW_IND : in VIEW_INDEX);

procedure SET_VIEW_REPRESENTATION
  (WS                    : in WS_ID;
   VIEW_IND              : in VIEW_INDEX;
   ORIENTATION_MATRIX    : in TRANSFORMATION_MATRIX_3;
   MAPPING_MATRIX        : in TRANSFORMATION_MATRIX_3;
   CLIPPING_LIMITS       : in NPC.RECTANGULAR_REGION_3;
   XY_CLIPPING           : in CLIPPING_INDICATOR;
   BACK_CLIPPING         : in CLIPPING_INDICATOR;
   FRONT_CLIPPING        : in CLIPPING_INDICATOR);

procedure SET_VIEW_REPRESENTATION
  (WS                    : in WS_ID;
   VIEW_IND              : in VIEW_INDEX;
   ORIENTATION_MATRIX    : in TRANSFORMATION_MATRIX_2;
   MAPPING_MATRIX        : in TRANSFORMATION_MATRIX_2;
   CLIPPING_LIMITS       : in NPC.RECTANGULAR_REGION_2;
   XY_CLIPPING           : in CLIPPING_INDICATOR);

procedure SET_VIEW_TRANSFORMATION_INPUT_PRIORITY
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567446 457 ■


```
    (WS                  : in WS_ID;
     VIEW_IND            : in VIEW_INDEX;
     REFERENCE_INDEX     : in VIEW_INDEX;
     PRIORITY            : in RELATIVE_PRIORITY);

  procedure SET_WS_WINDOW
    (WS                  : in WS_ID;
     WINDOW_LIMITS       : in NPC.RECTANGULAR_REGION_3);

  procedure SET_WS_WINDOW
    (WS                  : in WS_ID;
     WINDOW_LIMITS       : in NPC.RECTANGULAR_REGION_2);

  procedure SET_WS_VIEWPORT
    (WS                  : in WS_ID;
     VIEWPORT_LIMITS     : in DC.RECTANGULAR_REGION_3);

  procedure SET_WS_VIEWPORT
    (WS                  : in WS_ID;
     VIEWPORT_LIMITS     : in DC.RECTANGULAR_REGION_2);

-- TRANSFORMATION UTILITY FUNCTIONS

  procedure TRANSLATE
    (VECTOR              : in  MC.VECTOR_3;
     ERROR_INDICATOR     : out ERROR_NUMBER;
     MATRIX              : out TRANSFORMATION_MATRIX_3);

  procedure TRANSLATE
    (VECTOR              : in  MC.VECTOR_2;
     ERROR_INDICATOR     : out ERROR_NUMBER;
     MATRIX              : out TRANSFORMATION_MATRIX_2);

  procedure SCALE
    (FACTOR              : in  MC.VECTOR_3;
     ERROR_INDICATOR     : out ERROR_NUMBER;
     MATRIX              : out TRANSFORMATION_MATRIX_3);

  procedure SCALE
    (FACTOR              : in  MC.VECTOR_2;
     ERROR_INDICATOR     : out ERROR_NUMBER;
     MATRIX              : out TRANSFORMATION_MATRIX_2);

  procedure ROTATE_X
    (ANGLE_X             : in  ANGLE;
     ERROR_INDICATOR     : out ERROR_NUMBER;
     MATRIX              : out TRANSFORMATION_MATRIX_3);

  procedure ROTATE_Y
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567447 393 ■


```
  (ANGLE_Y           : in  ANGLE;
   ERROR_INDICATOR   : out ERROR_NUMBER;
   MATRIX            : out TRANSFORMATION_MATRIX_3);

procedure ROTATE_Z
  (ANGLE_Z           : in  ANGLE;
   ERROR_INDICATOR   : out ERROR_NUMBER;
   MATRIX            : out TRANSFORMATION_MATRIX_3);

procedure ROTATE
  (ANGLE_Z           : in  ANGLE;
   ERROR_INDICATOR   : out ERROR_NUMBER;
   MATRIX            : out TRANSFORMATION_MATRIX_2);

procedure COMPOSE_MATRIX
  (MATRIX_A          : in  TRANSFORMATION_MATRIX_3;
   MATRIX_B          : in  TRANSFORMATION_MATRIX_3;
   ERROR_INDICATOR   : out ERROR_NUMBER;
   COMPOSED_MATRIX   : out TRANSFORMATION_MATRIX_3);

procedure COMPOSE_MATRIX
  (MATRIX_A          : in  TRANSFORMATION_MATRIX_2;
   MATRIX_B          : in  TRANSFORMATION_MATRIX_2;
   ERROR_INDICATOR   : out ERROR_NUMBER;
   COMPOSED_MATRIX   : out TRANSFORMATION_MATRIX_2);

procedure TRANSFORM_POINT
  (POINT             : in  MC.POINT_3;
   MATRIX            : in  TRANSFORMATION_MATRIX_3;
   ERROR_INDICATOR   : out ERROR_NUMBER;
   TRANSFORMED_POINT : out MC.POINT_3);

procedure TRANSFORM_POINT
  (POINT             : in  MC.POINT_2;
   MATRIX            : in  TRANSFORMATION_MATRIX_2;
   ERROR_INDICATOR   : out ERROR_NUMBER;
   TRANSFORMED_POINT : out MC.POINT_2);

procedure BUILD_TRANSFORMATION_MATRIX
  (FIXED_POINT       : in  MC.POINT_3;
   SHIFT_VECTOR      : in  MC.VECTOR_3;
   ANGLE_X           : in  ANGLE;
   ANGLE_Y           : in  ANGLE;
   ANGLE_Z           : in  ANGLE;
   SCALE_FACTORS     : in  MC.VECTOR_3;
   ERROR_INDICATOR   : out ERROR_NUMBER;
   MATRIX            : out TRANSFORMATION_MATRIX_3);

procedure BUILD_TRANSFORMATION_MATRIX
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567448 22T ■


```
    (FIXED_POINT          : in  MC.POINT_2;
     SHIFT_VECTOR         : in  MC.VECTOR_2;
     ANGLE_Z              : in  ANGLE;
     SCALE_FACTORS        : in  MC.VECTOR_2;
     ERROR_INDICATOR      : out ERROR_NUMBER;
     MATRIX               : out TRANSFORMATION_MATRIX_2);

  procedure COMPOSE_TRANSFORMATION_MATRIX
    (MATRIX               : in  TRANSFORMATION_MATRIX_3;
     FIXED_POINT          : in  MC.POINT_3;
     SHIFT_VECTOR         : in  MC.VECTOR_3;
     ANGLE_X              : in  ANGLE;
     ANGLE_Y              : in  ANGLE;
     ANGLE_Z              : in  ANGLE;
     SCALE_FACTORS        : in  MC.VECTOR_3;
     ERROR_INDICATOR      : out ERROR_NUMBER;
     COMPOSED_MATRIX      : out TRANSFORMATION_MATRIX_3);

  procedure COMPOSE_TRANSFORMATION_MATRIX
    (MATRIX               : in  TRANSFORMATION_MATRIX_2;
     FIXED_POINT          : in  MC.POINT_2;
     SHIFT_VECTOR         : in  MC.VECTOR_2;
     ANGLE_Z              : in  ANGLE;
     SCALE_FACTORS        : in  MC.VECTOR_2;
     ERROR_INDICATOR      : out ERROR_NUMBER;
     COMPOSED_MATRIX      : out TRANSFORMATION_MATRIX_2);

  procedure EVALUATE_VIEW_ORIENTATION_MATRIX
    (REFERENCE_POINT      : in  WC.POINT_3;
     NORMAL_VECTOR        : in  WC.VECTOR_3;
     UP_VECTOR            : in  WC.VECTOR_3;
     ERROR_INDICATOR      : out ERROR_NUMBER;
     ORIENTATION_MATRIX   : out TRANSFORMATION_MATRIX_3);

  procedure EVALUATE_VIEW_ORIENTATION_MATRIX
    (REFERENCE_POINT      : in  WC.POINT_2;
     UP_VECTOR            : in  WC.VECTOR_2;
     ERROR_INDICATOR      : out ERROR_NUMBER;
     ORIENTATION_MATRIX   : out TRANSFORMATION_MATRIX_2);

  procedure EVALUATE_VIEW_MAPPING_MATRIX
    (WINDOW_LIMITS        : in  VRC.RECTANGULAR_REGION_2;
     VIEWPORT_LIMITS      : in  NPC.RECTANGULAR_REGION_3;
     TYPE_OF_PROJECTION   : in  PROJECTION_TYPE;
     REFERENCE_POINT      : in  VRC.POINT_3;
     VIEW_DISTANCE        : in  VRC_TYPE;
     BACK_DISTANCE        : in  VRC_TYPE;
     FRONT_DISTANCE       : in  VRC_TYPE;
     ERROR_INDICATOR      : out ERROR_NUMBER;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
      MAPPING_MATRIX       : out  TRANSFORMATION_MATRIX_3);

    procedure EVALUATE_VIEW_MAPPING_MATRIX
      (WINDOW_LIMITS     : in   VRC.RECTANGULAR_REGION_2;
       VIEWPORT_LIMITS   : in   NPC.RECTANGULAR_REGION_2;
       ERROR_INDICATOR   : out  ERROR_NUMBER;
       MAPPING_MATRIX    : out  TRANSFORMATION_MATRIX_2);

  -- STRUCTURE MANIPULATION FUNCTIONS

    procedure OPEN_STRUCTURE
      (STRUCTURE_IDENTIFIER : in STRUCTURE_ID);

    procedure CLOSE_STRUCTURE;

    procedure EXECUTE_STRUCTURE
      (STRUCTURE_IDENTIFIER : in STRUCTURE_ID);

    procedure LABEL
      (LABEL_IDENTIFIER : in LABEL_ID);

    procedure APPLICATION_DATA
      (DATA : in APPLICATION_DATA_RECORD);

    procedure SET_EDIT_MODE
      (MODE : in EDIT_MODE);

    procedure COPY_ALL_ELEMENTS_FROM_STRUCTURE
      (STRUCTURE_IDENTIFIER : in STRUCTURE_ID);

    procedure SET_ELEMENT_POINTER
      (POSITION : in ELEMENT_POSITION);

    procedure OFFSET_ELEMENT_POINTER
      (OFFSET : in ELEMENT_POSITION);

    procedure SET_ELEMENT_POINTER_AT_LABEL
      (LABEL_IDENTIFIER : in LABEL_ID);

    procedure DELETE_ELEMENT;

    procedure DELETE_ELEMENT_RANGE
      (POSITION_1  : in ELEMENT_POSITION;
       POSITION_2  : in ELEMENT_POSITION);

    procedure DELETE_ELEMENTS_BETWEEN_LABELS
      (LABEL_IDENTIFIER_1   : in LABEL_ID;
       LABEL_IDENTIFIER_2   : in LABEL_ID);

    procedure EMPTY_STRUCTURE
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567450 988 ■


```
    (STRUCTURE_IDENTIFIER : in STRUCTURE_ID);

  procedure DELETE_STRUCTURE
    (STRUCTURE_IDENTIFIER : in STRUCTURE_ID);

  procedure DELETE_STRUCTURE_NETWORK
    (STRUCTURE_IDENTIFIER  : in STRUCTURE_ID;
     HANDLING_FLAG         : in REFERENCE_HANDLING_FLAG);

  procedure DELETE_ALL_STRUCTURES;

  procedure CHANGE_STRUCTURE_IDENTIFIER
    (ORIGINAL_IDENTIFIER   : in STRUCTURE_ID;
     RESULTING_IDENTIFIER  : in STRUCTURE_ID);

  procedure CHANGE_STRUCTURE_REFERENCES
    (ORIGINAL_IDENTIFIER   : in STRUCTURE_ID;
     RESULTING_IDENTIFIER  : in STRUCTURE_ID);

  procedure CHANGE_STRUCTURE_IDENTIFIER_AND_REFERENCES
    (ORIGINAL_IDENTIFIER   : in STRUCTURE_ID;
     RESULTING_IDENTIFIER  : in STRUCTURE_ID);

  procedure POST_STRUCTURE
    (WS                    : in WS_ID;
     STRUCTURE_IDENTIFIER  : in STRUCTURE_ID;
     PRIORITY              : in DISPLAY_PRIORITY);

  procedure UNPOST_STRUCTURE
    (WS                    : in WS_ID;
     STRUCTURE_IDENTIFIER  : in STRUCTURE_ID);

  procedure UNPOST_ALL_STRUCTURES
    (WS : in WS_ID);

-- STRUCTURE ARCHIVING FUNCTIONS

  procedure OPEN_ARCHIVE_FILE
    (ARCHIVE_IDENTIFIER  : in ARCHIVE_ID;
     ARCHIVE_FILE        : in FILE_ID);

  procedure CLOSE_ARCHIVE_FILE
    (ARCHIVE_IDENTIFIER : in ARCHIVE_ID);

  procedure ARCHIVE_STRUCTURES
    (ARCHIVE_IDENTIFIER       : in ARCHIVE_ID;
     STRUCTURE_IDENTIFIERS    : in STRUCTURE_IDS.LIST_OF);

  procedure ARCHIVE_STRUCTURE_NETWORKS
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567451 814 ■


```
  (ARCHIVE_IDENTIFIER       : in ARCHIVE_ID;
   STRUCTURE_IDENTIFIERS    : in STRUCTURE_IDS.LIST_OF);

procedure ARCHIVE_ALL_STRUCTURES
  (ARCHIVE_IDENTIFIER : in ARCHIVE_ID);

procedure SET_CONFLICT_RESOLUTION
  (ARCHIVAL_CONFLICT    : in CONFLICT_RESOLUTION;
   RETRIEVAL_CONFLICT   : in CONFLICT_RESOLUTION);

procedure RETRIEVE_STRUCTURE_IDENTIFIERS
  (ARCHIVE_IDENTIFIER       : in  ARCHIVE_ID;
   STRUCTURE_IDENTIFIERS    : out STRUCTURE_IDS.LIST_OF);

procedure RETRIEVE_STRUCTURES
  (ARCHIVE_IDENTIFIER       : in ARCHIVE_ID;
   STRUCTURE_IDENTIFIERS    : in STRUCTURE_IDS.LIST_OF);

procedure RETRIEVE_STRUCTURE_NETWORKS
  (ARCHIVE_IDENTIFIER       : in ARCHIVE_ID;
   STRUCTURE_IDENTIFIERS    : in STRUCTURE_IDS.LIST_OF);

procedure RETRIEVE_ALL_STRUCTURES
  (ARCHIVE_IDENTIFIER : in ARCHIVE_ID);

procedure RETRIEVE_PATHS_TO_ANCESTORS
  (ARCHIVE_IDENTIFIER       : in  ARCHIVE_ID;
   STRUCTURE_IDENTIFIER     : in  STRUCTURE_ID;
   TRUNCATION               : in  TRUNCATION_METHOD;
   DEPTH                    : in  PATH_DEPTH;
   LIST_OF_PATHS            : out REFERENCE_PATHS.LIST_OF);

procedure RETRIEVE_PATHS_TO_DESCENDANTS
  (ARCHIVE_IDENTIFIER       : in  ARCHIVE_ID;
   STRUCTURE_IDENTIFIER     : in  STRUCTURE_ID;
   TRUNCATION               : in  TRUNCATION_METHOD;
   DEPTH                    : in  PATH_DEPTH;
   LIST_OF_PATHS            : out REFERENCE_PATHS.LIST_OF);

procedure DELETE_STRUCTURES_FROM_ARCHIVE
  (ARCHIVE_IDENTIFIER       : in ARCHIVE_ID;
   STRUCTURE_IDENTIFIERS    : in STRUCTURE_IDS.LIST_OF);

procedure DELETE_STRUCTURE_NETWORKS_FROM_ARCHIVE
  (ARCHIVE_IDENTIFIER       : in ARCHIVE_ID;
   STRUCTURE_IDENTIFIERS    : in STRUCTURE_IDS.LIST_OF);

procedure DELETE_ALL_STRUCTURES_FROM_ARCHIVE
  (ARCHIVE_IDENTIFIER : in ARCHIVE_ID);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
-- INPUT FUNCTIONS

   procedure SET_PICK_IDENTIFIER
     (PICK_IDENTIFIER : in PICK_ID);

   procedure SET_PICK_FILTER
     (WS              : in WS_ID;
      PICK_DEVICE     : in PICK_DEVICE_NUMBER;
      PICKABILITY     : in NAME_SET_FILTER);

   procedure INITIALIZE_LOCATOR
     (WS              : in WS_ID;
      DEVICE          : in LOCATOR_DEVICE_NUMBER;
      INITIAL_VIEW_IND : in VIEW_INDEX;
      INITIAL_POSITION : in WC.POINT_3;
      ECHO_VOLUME     : in DC.RECTANGULAR_REGION_3;
      DATA_RECORD     : in LOCATOR_DATA_RECORD);

   procedure INITIALIZE_LOCATOR
     (WS              : in WS_ID;
      DEVICE          : in LOCATOR_DEVICE_NUMBER;
      INITIAL_VIEW_IND : in VIEW_INDEX;
      INITIAL_POSITION : in WC.POINT_2;
      ECHO_AREA       : in DC.RECTANGULAR_REGION_2;
      DATA_RECORD     : in LOCATOR_DATA_RECORD);

   procedure INITIALIZE_STROKE
     (WS              : in WS_ID;
      DEVICE          : in STROKE_DEVICE_NUMBER;
      INITIAL_VIEW_IND : in VIEW_INDEX;
      INITIAL_STROKE  : in WC.POINT_LIST_3;
      ECHO_VOLUME     : in DC.RECTANGULAR_REGION_3;
      DATA_RECORD     : in STROKE_DATA_RECORD);

   procedure INITIALIZE_STROKE
     (WS              : in WS_ID;
      DEVICE          : in STROKE_DEVICE_NUMBER;
      INITIAL_VIEW_IND : in VIEW_INDEX;
      INITIAL_STROKE  : in WC.POINT_LIST_2;
      ECHO_AREA       : in DC.RECTANGULAR_REGION_2;
      DATA_RECORD     : in STROKE_DATA_RECORD);

   procedure INITIALIZE_VALUATOR
     (WS              : in WS_ID;
      DEVICE          : in VALUATOR_DEVICE_NUMBER;
      INITIAL_VALUE   : in VALUATOR_VALUE;
      ECHO_VOLUME     : in DC.RECTANGULAR_REGION_3;
      DATA_RECORD     : in VALUATOR_DATA_RECORD);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567453 697 ■


```
procedure INITIALIZE_VALUATOR
  (WS              : in WS_ID;
   DEVICE          : in VALUATOR_DEVICE_NUMBER;
   INITIAL_VALUE   : in VALUATOR_VALUE;
   ECHO_AREA       : in DC.RECTANGULAR_REGION_2;
   DATA_RECORD     : in VALUATOR_DATA_RECORD);

procedure INITIALIZE_CHOICE
  (WS              : in WS_ID;
   DEVICE          : in CHOICE_DEVICE_NUMBER;
   INITIAL_STATUS  : in CHOICE_STATUS;
   INITIAL_CHOICE  : in CHOICE_NUMBER;
   ECHO_VOLUME     : in DC.RECTANGULAR_REGION_3;
   DATA_RECORD     : in CHOICE_DATA_RECORD);

procedure INITIALIZE_CHOICE
  (WS              : in WS_ID;
   DEVICE          : in CHOICE_DEVICE_NUMBER;
   INITIAL_STATUS  : in CHOICE_STATUS;
   INITIAL_CHOICE  : in CHOICE_NUMBER;
   ECHO_AREA       : in DC.RECTANGULAR_REGION_2;
   DATA_RECORD     : in CHOICE_DATA_RECORD);

procedure INITIALIZE_PICK
  (WS              : in WS_ID;
   DEVICE          : in PICK_DEVICE_NUMBER;
   INITIAL_STATUS  : in PICK_STATUS;
   INITIAL_PATH    : in PICK_PATH;
   ECHO_VOLUME     : in DC.RECTANGULAR_REGION_3;
   DATA_RECORD     : in PICK_DATA_RECORD;
   ORDER           : in PATH_ORDER);

procedure INITIALIZE_PICK
  (WS              : in WS_ID;
   DEVICE          : in PICK_DEVICE_NUMBER;
   INITIAL_STATUS  : in PICK_STATUS;
   INITIAL_PATH    : in PICK_PATH;
   ECHO_AREA       : in DC.RECTANGULAR_REGION_2;
   DATA_RECORD     : in PICK_DATA_RECORD;
   ORDER           : in PATH_ORDER);

procedure INITIALIZE_STRING
  (WS              : in WS_ID;
   DEVICE          : in STRING_DEVICE_NUMBER;
   INITIAL_STRING  : in PHIGS_STRING;
   ECHO_VOLUME     : in DC.RECTANGULAR_REGION_3;
   DATA_RECORD     : in STRING_DATA_RECORD);

procedure INITIALIZE_STRING
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567454 523 ■


```
    (WS                : in WS_ID;
     DEVICE            : in STRING_DEVICE_NUMBER;
     INITIAL_STRING    : in PHIGS_STRING;
     ECHO_AREA         : in DC.RECTANGULAR_REGION_2;
     DATA_RECORD       : in STRING_DATA_RECORD);

  procedure SET_LOCATOR_MODE
    (WS       : in WS_ID;
     DEVICE   : in LOCATOR_DEVICE_NUMBER;
     MODE     : in OPERATING_MODE;
     SWITCH   : in ECHO_SWITCH);

  procedure SET_STROKE_MODE
    (WS       : in WS_ID;
     DEVICE   : in STROKE_DEVICE_NUMBER;
     MODE     : in OPERATING_MODE;
     SWITCH   : in ECHO_SWITCH);

  procedure SET_VALUATOR_MODE
    (WS       : in WS_ID;
     DEVICE   : in VALUATOR_DEVICE_NUMBER;
     MODE     : in OPERATING_MODE;
     SWITCH   : in ECHO_SWITCH);

  procedure SET_CHOICE_MODE
    (WS       : in WS_ID;
     DEVICE   : in CHOICE_DEVICE_NUMBER;
     MODE     : in OPERATING_MODE;
     SWITCH   : in ECHO_SWITCH);

  procedure SET_PICK_MODE
    (WS       : in WS_ID;
     DEVICE   : in PICK_DEVICE_NUMBER;
     MODE     : in OPERATING_MODE;
     SWITCH   : in ECHO_SWITCH);

  procedure SET_STRING_MODE
    (WS       : in WS_ID;
     DEVICE   : in STRING_DEVICE_NUMBER;
     MODE     : in OPERATING_MODE;
     SWITCH   : in ECHO_SWITCH);

  procedure REQUEST_LOCATOR
    (WS         : in  WS_ID;
     DEVICE     : in  LOCATOR_DEVICE_NUMBER;
     STATUS     : out INPUT_STATUS;
     VIEW_IND   : out VIEW_INDEX;
     POSITION   : out WC.POINT_3);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567455 46T ■


```
procedure REQUEST_LOCATOR
  (WS          : in   WS_ID;
   DEVICE      : in   LOCATOR_DEVICE_NUMBER;
   STATUS      : out  INPUT_STATUS;
   VIEW_IND    : out  VIEW_INDEX;
   POSITION    : out  WC.POINT_2);

procedure REQUEST_STROKE
  (WS                : in   WS_ID;
   DEVICE            : in   STROKE_DEVICE_NUMBER;
   STATUS            : out  INPUT_STATUS;
   VIEW_IND          : out  VIEW_INDEX;
   STROKE_POINTS     : out  WC.ACCESS_POINT_LIST_3);

procedure REQUEST_STROKE
  (WS                : in   WS_ID;
   DEVICE            : in   STROKE_DEVICE_NUMBER;
   STATUS            : out  INPUT_STATUS;
   VIEW_IND          : out  VIEW_INDEX;
   STROKE_POINTS     : out  WC.ACCESS_POINT_LIST_2);

procedure REQUEST_VALUATOR
  (WS       : in   WS_ID;
   DEVICE   : in   VALUATOR_DEVICE_NUMBER;
   STATUS   : out  INPUT_STATUS;
   VALUE    : out  VALUATOR_VALUE);

procedure REQUEST_CHOICE
  (WS       : in   WS_ID;
   DEVICE   : in   CHOICE_DEVICE_NUMBER;
   STATUS   : out  CHOICE_REQUEST_STATUS;
   CHOICE   : out  CHOICE_NUMBER);

procedure REQUEST_PICK
  (WS                  : in   WS_ID;
   DEVICE              : in   PICK_DEVICE_NUMBER;
   DEPTH_TO_RETURN     : in   PATH_DEPTH;
   STATUS              : out  PICK_REQUEST_STATUS;
   PATH                : out  PICK_PATH);

procedure REQUEST_STRING
  (WS               : in   WS_ID;
   DEVICE           : in   STRING_DEVICE_NUMBER;
   STATUS           : out  INPUT_STATUS;
   CHAR_STRING      : out  INPUT_STRING);

procedure SAMPLE_LOCATOR
  (WS          : in   WS_ID;
   DEVICE      : in   LOCATOR_DEVICE_NUMBER;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567456 3T6 ■


```
    VIEW_IND : out  VIEW_INDEX;
    POSITION : out  WC.POINT_3);

procedure SAMPLE_LOCATOR
  (WS        : in   WS_ID;
   DEVICE    : in   LOCATOR_DEVICE_NUMBER;
   VIEW_IND  : out  VIEW_INDEX;
   POSITION  : out  WC.POINT_2);

procedure SAMPLE_STROKE
  (WS                : in   WS_ID;
   DEVICE            : in   STROKE_DEVICE_NUMBER;
   VIEW_IND          : out  VIEW_INDEX;
   STROKE_POINTS     : out  WC.ACCESS_POINT_LIST_3);

procedure SAMPLE_STROKE
  (WS                : in   WS_ID;
   DEVICE            : in   STROKE_DEVICE_NUMBER;
   VIEW_IND          : out  VIEW_INDEX;
   STROKE_POINTS     : out  WC.ACCESS_POINT_LIST_2);

procedure SAMPLE_VALUATOR
  (WS       : in   WS_ID;
   DEVICE   : in   VALUATOR_DEVICE_NUMBER;
   VALUE    : out  VALUATOR_VALUE);

procedure SAMPLE_CHOICE
  (WS       : in   WS_ID;
   DEVICE   : in   CHOICE_DEVICE_NUMBER;
   STATUS   : out  CHOICE_STATUS;
   CHOICE   : out  CHOICE_NUMBER);

procedure SAMPLE_PICK
  (WS                : in   WS_ID;
   DEVICE            : in   PICK_DEVICE_NUMBER;
   DEPTH_TO_RETURN   : in   PATH_DEPTH;
   STATUS            : out  PICK_STATUS;
   PATH              : out  PICK_PATH);

procedure SAMPLE_STRING
  (WS                : in   WS_ID;
   DEVICE            : in   STRING_DEVICE_NUMBER;
   CHAR_STRING       : out  INPUT_STRING);

procedure AWAIT_EVENT
  (TIMEOUT  : in   DURATION;
   WS       : out  WS_ID;
   DEVICE   : out  EVENT_DEVICE_NUMBER);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
      procedure FLUSH_DEVICE_EVENTS
        (WS     : in WS_ID;
         DEVICE : in EVENT_QUEUE_DEVICE_NUMBER);

      procedure GET_LOCATOR
        (VIEW_IND : out VIEW_INDEX;
         POSITION : out WC.POINT_3);

      procedure GET_LOCATOR
        (VIEW_IND : out VIEW_INDEX;
         POSITION : out WC.POINT_2);

      procedure GET_STROKE
        (VIEW_IND      : out VIEW_INDEX;
         STROKE_POINTS : out WC.ACCESS_POINT_LIST_3);

      procedure GET_STROKE
        (VIEW_IND      : out VIEW_INDEX;
         STROKE_POINTS : out WC.ACCESS_POINT_LIST_2);

      procedure GET_VALUATOR
        (VALUE : out VALUATOR_VALUE);

      procedure GET_CHOICE
        (STATUS : out CHOICE_STATUS;
         CHOICE : out CHOICE_NUMBER);

      procedure GET_PICK
        (DEPTH_TO_RETURN : in  PATH_DEPTH;
         STATUS          : out PICK_STATUS;
         PATH            : out PICK_PATH);

      procedure GET_STRING
        (CHAR_STRING : out INPUT_STRING);

    -- METAFILE FUNCTIONS

      procedure WRITE_ITEM_TO_METAFILE
        (WS   : in WS_ID;
         ITEM : in METAFILE_DATA_RECORD);

      procedure GET_ITEM_TYPE_FROM_METAFILE
        (WS           : in  WS_ID;
         TYPE_OF_ITEM : out METAFILE_ITEM_TYPE;
         LENGTH       : out METAFILE_ITEM_LENGTH);

      procedure READ_ITEM_FROM_METAFILE
        (WS         : in  WS_ID;
         MAX_LENGTH : in  METAFILE_ITEM_LENGTH;
         ITEM       : out METAFILE_DATA_RECORD);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567458 179 ■


```
    procedure INTERPRET_ITEM
      (ITEM : in METAFILE_DATA_RECORD);

  -- INQUIRY FUNCTIONS

    procedure INQ_SYSTEM_STATE_VALUE
      (STATE_VALUE : out SYSTEM_STATE);

    procedure INQ_WS_STATE_VALUE
      (STATE_VALUE : out WS_STATE);

    procedure INQ_STRUCTURE_STATE_VALUE
      (STATE_VALUE : out STRUCTURE_STATE);

    procedure INQ_ARCHIVE_STATE_VALUE
      (STATE_VALUE : out ARCHIVE_STATE);

    procedure INQ_LIST_OF_AVAILABLE_WS_TYPES
      (ERROR_INDICATOR : out ERROR_NUMBER;
       LIST_OF_TYPES   : out WS_TYPES.LIST_OF);

    procedure INQ_PHIGS_FACILITIES
      (ERROR_INDICATOR                : out ERROR_NUMBER;
       MAX_SIMUL_OPEN_WS              : out PHIGS_POSITIVE;
       MAX_SIMUL_OPEN_ARCHIVES        : out PHIGS_POSITIVE;
       NUMBER_AVAIL_NAMES             : out PHIGS_POSITIVE;
       AVAIL_CHAR_SETS                : out CHAR_SETS.LIST_OF;
       MAX_ISS_NORMAL_FILTER_LIST     : out PHIGS_POSITIVE;
       MAX_ISS_INVERTED_FILTER_LIST   : out PHIGS_POSITIVE);

    procedure INQ_GSE_FACILITIES
      (ERROR_INDICATOR      : out ERROR_NUMBER;
       LIST_AVAIL_GSES      : out GSE_IDS.LIST_OF;
       LIST_WS_DEPENDENCIES : out WS_DEPENDENCIES.LIST_OF);

    procedure INQ_MODELLING_CLIPPING_FACILITIES
      (ERROR_INDICATOR     : out ERROR_NUMBER;
       NUMBER_DISTINCT_PLANES
                           : out PHIGS_POSITIVE;
       LIST_OF_OPERATIONS  : out MODELLING_CLIP_OPERATION_TYPES.LIST_OF);

    procedure INQ_EDIT_MODE
      (ERROR_INDICATOR : out ERROR_NUMBER;
       MODE            : out EDIT_MODE);

    procedure INQ_SET_OF_OPEN_WS
      (ERROR_INDICATOR : out ERROR_NUMBER;
       OPEN_WS         : out WS_IDS.LIST_OF);

    procedure INQ_STRUCTURE_IDENTIFIERS
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567459 005 ■


```
  (ERROR_INDICATOR     : out ERROR_NUMBER;
   LIST_OF_STRUCTURES : out STRUCTURE_IDS.LIST_OF);

procedure INQ_ARCHIVE_FILES
  (ERROR_INDICATOR           : out ERROR_NUMBER;
   LIST_ARCHIVE_IDENTIFIERS : out ARCHIVE_IDS.LIST_OF;
   LIST_ARCHIVE_FILES        : out VARIABLE_FILE_IDS.LIST_OF);

procedure INQ_CONFLICT_RESOLUTION
  (ERROR_INDICATOR     : out ERROR_NUMBER;
   ARCHIVAL_CONFLICT   : out CONFLICT_RESOLUTION;
   RETRIEVAL_CONFLICT  : out CONFLICT_RESOLUTION);

procedure INQ_ALL_CONFLICTING_STRUCTURES
  (ARCHIVE_IDENTIFIER  : in  ARCHIVE_ID;
   ERROR_INDICATOR     : out ERROR_NUMBER;
   LIST_OF_STRUCTURES  : out STRUCTURE_IDS.LIST_OF);

procedure INQ_CONFLICTING_STRUCTURES_IN_NETWORK
  (ARCHIVE_IDENTIFIER    : in  ARCHIVE_ID;
   STRUCTURE_IDENTIFIER : in  STRUCTURE_ID;
   SOURCE                : in  STRUCTURE_NETWORK_SOURCE;
   ERROR_INDICATOR       : out ERROR_NUMBER;
   LIST_OF_STRUCTURES    : out STRUCTURE_IDS.LIST_OF);

procedure INQ_MORE_SIMULTANEOUS_EVENTS
  (ERROR_INDICATOR: out ERROR_NUMBER;
   EVENTS           : out MORE_EVENTS);

procedure INQ_WS_CONNECTION_AND_TYPE
  (WS                : in  WS_ID;
   ERROR_INDICATOR: out ERROR_NUMBER;
   CONNECTION       : out VARIABLE_CONNECTION_ID;
   TYPE_OF_WS       : out WS_TYPE);

procedure INQ_LIST_OF_VIEW_INDICES
  (WS                  : in  WS_ID;
   ERROR_INDICATOR  : out ERROR_NUMBER;
   DEFINED_VIEW_LIST : out VIEW_INDICES.LIST_OF);

procedure INQ_VIEW_REPRESENTATION
  (WS                                : in  WS_ID;
   VIEW_IND                          : in  VIEW_INDEX;
   ERROR_INDICATOR                   : out ERROR_NUMBER;
   UPDATE                            : out UPDATE_STATE;
   REQUESTED_ORIENTATION_MATRIX : out TRANSFORMATION_MATRIX_3;
   CURRENT_ORIENTATION_MATRIX   : out TRANSFORMATION_MATRIX_3;
   REQUESTED_MAPPING_MATRIX     : out TRANSFORMATION_MATRIX_3;
   CURRENT_MAPPING_MATRIX       : out TRANSFORMATION_MATRIX_3;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
    REQUESTED_CLIPPING_LIMITS      : out  NPC.RECTANGULAR_REGION_3;
    CURRENT_CLIPPING_LIMITS        : out  NPC.RECTANGULAR_REGION_3;
    REQUESTED_XY_CLIPPING          : out  CLIPPING_INDICATOR;
    CURRENT_XY_CLIPPING            : out  CLIPPING_INDICATOR;
    REQUESTED_BACK_CLIPPING        : out  CLIPPING_INDICATOR;
    CURRENT_BACK_CLIPPING          : out  CLIPPING_INDICATOR;
    REQUESTED_FRONT_CLIPPING       : out  CLIPPING_INDICATOR;
    CURRENT_FRONT_CLIPPING         : out  CLIPPING_INDICATOR);

procedure INQ_HLHSR_MODE
  (WS                  : in   WS_ID;
   ERROR_INDICATOR     : out  ERROR_NUMBER;
   UPDATE              : out  UPDATE_STATE;
   CURRENT_MODE        : out  HLHSR_MODE;
   REQUESTED_MODE      : out  HLHSR_MODE);

procedure INQ_POSTED_STRUCTURES
  (WS                     : in   WS_ID;
   ERROR_INDICATOR        : out  ERROR_NUMBER;
   LIST_OF_STRUCTURES     : out  POSTED_STRUCTURES.LIST_OF);

procedure INQ_DISPLAY_UPDATE_STATE
  (WS                  : in   WS_ID;
   ERROR_INDICATOR     : out  ERROR_NUMBER;
   DEFERRAL            : out  DEFERRAL_MODE;
   MODIFICATION        : out  MODIFICATION_MODE;
   EMPTINESS           : out  DISPLAY_SURFACE_EMPTY;
   VISUAL_STATE        : out  VISUAL_REPRESENTATION_STATE);

procedure INQ_LIST_OF_POLYLINE_INDICES
  (WS                  : in   WS_ID;
   ERROR_INDICATOR     : out  ERROR_NUMBER;
   INDICES             : out  POLYLINE_INDICES.LIST_OF);

procedure INQ_POLYLINE_REPRESENTATION
  (WS                     : in   WS_ID;
   POLYLINE_IND           : in   POLYLINE_INDEX;
   RETURNED_VALUES        : in   RETURN_VALUE_TYPE;
   ERROR_INDICATOR        : out  ERROR_NUMBER;
   TYPE_OF_LINE           : out  LINETYPE;
   LINEWIDTH_SF           : out  LINEWIDTH;
   LINE_COLOUR            : out  COLOUR_INDEX);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567461 763 ■


```
procedure INQ_POLYLINE_REPRESENTATION
  (WS                              : in  WS_ID;
   POLYLINE_IND                    : in  POLYLINE_INDEX;
   RETURNED_VALUES                 : in  RETURN_VALUE_TYPE;
   ERROR_INDICATOR                 : out ERROR_NUMBER;
   TYPE_OF_LINE                    : out LINETYPE;
   LINEWIDTH_SF                    : out LINEWIDTH;
   GENERAL_LINE_COLOUR             : out GENERAL_COLOUR;
   LINE_SHADING                    : out POLYLINE_SHADING_METHOD;
   CRITERIA_FOR_CURVE_APPROX       : out CURVE_APPROX_DATA_RECORD);

procedure INQ_LIST_OF_POLYMARKER_INDICES
  (WS                : in  WS_ID;
   ERROR_INDICATOR   : out ERROR_NUMBER;
   INDICES           : out POLYMARKER_INDICES.LIST_OF);

procedure INQ_POLYMARKER_REPRESENTATION
  (WS                : in  WS_ID;
   POLYMARKER_IND    : in  POLYMARKER_INDEX;
   RETURNED_VALUES   : in  RETURN_VALUE_TYPE;
   ERROR_INDICATOR   : out ERROR_NUMBER;
   TYPE_OF_MARKER    : out MARKER_TYPE;
   SIZE              : out MARKER_SIZE;
   MARKER_COLOUR     : out COLOUR_INDEX);

procedure INQ_POLYMARKER_REPRESENTATION
  (WS                              : in  WS_ID;
   POLYMARKER_IND                  : in  POLYMARKER_INDEX;
   RETURNED_VALUES                 : in  RETURN_VALUE_TYPE;
   ERROR_INDICATOR                 : out ERROR_NUMBER;
   TYPE_OF_MARKER                  : out MARKER_TYPE;
   SIZE                            : out MARKER_SIZE;
   GENERAL_MARKER_COLOUR           : out GENERAL_COLOUR);

procedure INQ_LIST_OF_TEXT_INDICES
  (WS                : in  WS_ID;
   ERROR_INDICATOR   : out ERROR_NUMBER;
   INDICES           : out TEXT_INDICES.LIST_OF);

procedure INQ_TEXT_REPRESENTATION
  (WS                : in  WS_ID;
   TEXT_IND          : in  TEXT_INDEX;
   RETURNED_VALUES   : in  RETURN_VALUE_TYPE;
   ERROR_INDICATOR   : out ERROR_NUMBER;
   FONT              : out TEXT_FONT;
   PRECISION         : out TEXT_PRECISION;
   EXPANSION         : out CHAR_EXPANSION;
   SPACING           : out CHAR_SPACING;
   TEXT_COLOUR       : out COLOUR_INDEX);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
procedure INQ_TEXT_REPRESENTATION
  (WS                    : in  WS_ID;
   TEXT_IND              : in  TEXT_INDEX;
   RETURNED_VALUES       : in  RETURN_VALUE_TYPE;
   ERROR_INDICATOR       : out ERROR_NUMBER;
   FONT                  : out TEXT_FONT;
   PRECISION             : out TEXT_PRECISION;
   EXPANSION             : out CHAR_EXPANSION;
   SPACING               : out CHAR_SPACING;
   GENERAL_TEXT_COLOUR   : out GENERAL_COLOUR);

procedure INQ_LIST_OF_INTERIOR_INDICES
  (WS                : in  WS_ID;
   ERROR_INDICATOR   : out ERROR_NUMBER;
   INDICES           : out INTERIOR_INDICES.LIST_OF);

procedure INQ_INTERIOR_REPRESENTATION
  (WS                : in  WS_ID;
   INTERIOR_IND      : in  INTERIOR_INDEX;
   RETURNED_VALUES   : in  RETURN_VALUE_TYPE;
   ERROR_INDICATOR   : out ERROR_NUMBER;
   STYLE_OF_INTERIOR : out INTERIOR_STYLE;
   STYLE_IND         : out STYLE_INDEX;
   INTERIOR_COLOUR   : out COLOUR_INDEX);

procedure INQ_INTERIOR_REPRESENTATION
  (WS                        : in  WS_ID;
   INTERIOR_IND              : in  INTERIOR_INDEX;
   RETURNED_VALUES           : in  RETURN_VALUE_TYPE;
   ERROR_INDICATOR           : out ERROR_NUMBER;
   STYLE_OF_INTERIOR         : out INTERIOR_STYLE;
   STYLE_IND                 : out STYLE_INDEX;
   GENERAL_INTERIOR_COLOUR   : out GENERAL_COLOUR;
   INTERIOR_SHADING          : out INTERIOR_SHADING_METHOD);

procedure INQ_LIST_OF_EDGE_INDICES
  (WS                : in  WS_ID;
   ERROR_INDICATOR   : out ERROR_NUMBER;
   INDICES           : out EDGE_INDICES.LIST_OF);

procedure INQ_EDGE_REPRESENTATION
  (WS                : in  WS_ID;
   EDGE_IND          : in  EDGE_INDEX;
   RETURNED_VALUES   : in  RETURN_VALUE_TYPE;
   ERROR_INDICATOR   : out ERROR_NUMBER;
   FLAG              : out EDGE_FLAG;
   TYPE_OF_EDGE      : out EDGETYPE;
   EDGEWIDTH_SF      : out EDGEWIDTH;
   EDGE_COLOUR       : out COLOUR_INDEX);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567463 536 ■


```
procedure INQ_EDGE_REPRESENTATION
  (WS                        : in  WS_ID;
   EDGE_IND                  : in  EDGE_INDEX;
   RETURNED_VALUES           : in  RETURN_VALUE_TYPE;
   ERROR_INDICATOR           : out ERROR_NUMBER;
   FLAG                      : out EDGE_FLAG;
   TYPE_OF_EDGE              : out EDGETYPE;
   EDGEWIDTH_SF              : out EDGEWIDTH;
   GENERAL_EDGE_COLOUR       : out GENERAL_COLOUR);

procedure INQ_LIST_OF_DATA_MAPPING_INDICES
  (WS                  : in  WS_ID;
   ERROR_INDICATOR     : out ERROR_NUMBER;
   INDICES             : out DATA_MAPPING_INDICES.LIST_OF);

procedure INQ_DATA_MAPPING_REPRESENTATION
  (WS                    : in  WS_ID;
   DATA_MAPPING_IND      : in  DATA_MAPPING_INDEX;
   RETURNED_VALUES       : in  RETURN_VALUE_TYPE;
   ERROR_INDICATOR       : out ERROR_NUMBER;
   DATA_MAPPING          : out DATA_MAPPING_DATA_RECORD);

procedure INQ_LIST_OF_REFLECTANCE_INDICES
  (WS                  : in  WS_ID;
   ERROR_INDICATOR     : out ERROR_NUMBER;
   INDICES             : out REFLECTANCE_INDICES.LIST_OF);

procedure INQ_REFLECTANCE_REPRESENTATION
  (WS                          : in  WS_ID;
   REFLECTANCE_IND             : in  REFLECTANCE_INDEX;
   RETURNED_VALUES             : in  RETURN_VALUE_TYPE;
   ERROR_INDICATOR             : out ERROR_NUMBER;
   MODEL                       : out REFLECTANCE_MODEL;
   REFLECTANCE_PROPERTIES      : out REFLECTANCE_PROPERTIES_DATA_RECORD);

procedure INQ_LIST_OF_PARAMETRIC_SURFACE_INDICES
  (WS                  : in  WS_ID;
   ERROR_INDICATOR     : out ERROR_NUMBER;
   INDICES             : out PARAMETRIC_SURFACE_INDICES.LIST_OF);

procedure INQ_PARAMETRIC_SURFACE_REPRESENTATION
  (WS                                : in  WS_ID;
   PARAMETRIC_SURFACE_IND            : in  PARAMETRIC_SURFACE_INDEX;
   RETURNED_VALUES                   : in  RETURN_VALUE_TYPE;
   ERROR_INDICATOR                   : out ERROR_NUMBER;
   CRITERIA_FOR_SURFACE_APPROX       : out SURFACE_APPROX_DATA_RECORD;
   PARAMETRIC_SURFACE_CHARACTERISTICS
                                     : out PARAMETRIC_SURFACE_DATA_RECORD);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567464 472 ■


```
procedure INQ_LIST_OF_PATTERN_INDICES
  (WS                 : in   WS_ID;
   ERROR_INDICATOR    : out  ERROR_NUMBER;
   INDICES            : out  PATTERN_INDICES.LIST_OF);

procedure INQ_PATTERN_REPRESENTATION
  (WS                 : in   WS_ID;
   PATTERN_IND        : in   PATTERN_INDEX;
   RETURNED_VALUES    : in   RETURN_VALUE_TYPE;
   ERROR_INDICATOR    : out  ERROR_NUMBER;
   PATTERN            : out  ACCESS_COLOUR_MATRIX);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567465 309 ■


```
procedure INQ_PATTERN_REPRESENTATION
  (WS                 : in  WS_ID;
   PATTERN_IND        : in  PATTERN_INDEX;
   RETURNED_VALUES    : in  RETURN_VALUE_TYPE;
   ERROR_INDICATOR    : out ERROR_NUMBER;
   GENERAL_PATTERN    : out ACCESS_COLOUR_VALUE_ARRAY);

procedure INQ_LIST_OF_LIGHT_SOURCE_INDICES
  (WS                 : in  WS_ID;
   ERROR_INDICATOR    : out ERROR_NUMBER;
   INDICES            : out LIGHT_SOURCE_INDICES.LIST_OF);

procedure INQ_LIGHT_SOURCE_REPRESENTATION
  (WS                 : in  WS_ID;
   LIGHT_SOURCE_IND   : in  LIGHT_SOURCE_INDEX;
   RETURNED_VALUES    : in  RETURN_VALUE_TYPE;
   ERROR_INDICATOR    : out ERROR_NUMBER;
   LIGHT_SOURCE       : out LIGHT_SOURCE_DATA_RECORD);

procedure INQ_LIST_OF_DEPTH_CUE_INDICES
  (WS                 : in  WS_ID;
   ERROR_INDICATOR    : out ERROR_NUMBER;
   INDICES            : out DEPTH_CUE_INDICES.LIST_OF);

procedure INQ_DEPTH_CUE_REPRESENTATION
  (WS                   : in  WS_ID;
   DEPTH_CUE_IND        : in  DEPTH_CUE_INDEX;
   RETURNED_VALUES      : in  RETURN_VALUE_TYPE;
   ERROR_INDICATOR      : out ERROR_NUMBER;
   MODE                 : out DEPTH_CUE_MODE;
   REFERENCE_PLANES     : out NPC.RANGE_OF_MAGNITUDES;
   DEPTH_CUE_SF         : out DEPTH_CUE_SCALE_FACTORS;
   DEPTH_CUE_COLOUR     : out GENERAL_COLOUR);

procedure INQ_COLOUR_MAPPING_STATE
  (WS                 : in  WS_ID;
   METHOD             : in  COLOUR_MAPPING_METHOD;
   ERROR_INDICATOR    : out ERROR_NUMBER;
   STATE              : out COLOUR_MAPPING_STATE);

procedure INQ_LIST_OF_COLOUR_MAPPING_INDICES
  (WS                 : in  WS_ID;
   ERROR_INDICATOR    : out ERROR_NUMBER;
   INDICES            : out COLOUR_MAPPING_INDICES.LIST_OF);

procedure INQ_COLOUR_MAPPING_REPRESENTATION
  (WS                   : in  WS_ID;
   COLOUR_MAPPING_IND   : in  COLOUR_MAPPING_INDEX;
   RETURNED_VALUES      : in  RETURN_VALUE_TYPE;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567466 245 ■


```
    ERROR_INDICATOR      : out ERROR_NUMBER;
    COLOUR_MAPPING       : out ACCESS_COLOUR_MAPPING_DATA_RECORD);

procedure INQ_COLOUR_MODEL
  (WS                  : in  WS_ID;
   ERROR_INDICATOR: out  ERROR_NUMBER;
   MODEL               : out  COLOUR_MODEL);

procedure INQ_LIST_OF_COLOUR_INDICES
  (WS                  : in  WS_ID;
   ERROR_INDICATOR: out  ERROR_NUMBER;
   INDICES             : out  COLOUR_INDICES.LIST_OF);

procedure INQ_COLOUR_REPRESENTATION
  (WS                   : in  WS_ID;
   COLOUR_IND           : in  COLOUR_INDEX;
   RETURNED_VALUES  : in  RETURN_VALUE_TYPE;
   ERROR_INDICATOR  : out  ERROR_NUMBER;
   COLOUR               : out  COLOUR_REPRESENTATION);

procedure INQ_HIGHLIGHTING_FILTER
  (WS                  : in  WS_ID;
   ERROR_INDICATOR: out  ERROR_NUMBER;
   HIGHLIGHTING     : out  NAME_SET_FILTER);

procedure INQ_INVISIBILITY_FILTER
  (WS                  : in  WS_ID;
   ERROR_INDICATOR: out  ERROR_NUMBER;
   INVISIBILITY     : out  NAME_SET_FILTER);

procedure INQ_WS_TRANSFORMATION
  (WS                              : in  WS_ID;
   ERROR_INDICATOR                 : out  ERROR_NUMBER;
   UPDATE                          : out  UPDATE_STATE;
   REQUESTED_WINDOW_LIMITS  : out  NPC.RECTANGULAR_REGION_3;
   CURRENT_WINDOW_LIMITS    : out  NPC.RECTANGULAR_REGION_3;
   REQUESTED_VIEWPORT_LIMITS: out  DC.RECTANGULAR_REGION_3;
   CURRENT_VIEWPORT_LIMITS  : out  DC.RECTANGULAR_REGION_3);

procedure INQ_WS_TRANSFORMATION
  (WS                              : in  WS_ID;
   ERROR_INDICATOR                 : out  ERROR_NUMBER;
   UPDATE                          : out  UPDATE_STATE;
   REQUESTED_WINDOW_LIMITS  : out  NPC.RECTANGULAR_REGION_2;
   CURRENT_WINDOW_LIMITS    : out  NPC.RECTANGULAR_REGION_2;
   REQUESTED_VIEWPORT_LIMITS: out  DC.RECTANGULAR_REGION_2;
   CURRENT_VIEWPORT_LIMITS  : out  DC.RECTANGULAR_REGION_2);

procedure INQ_LOCATOR_DEVICE_STATE
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567467 181 ■


```
  (WS                  : in   WS_ID;
   DEVICE              : in   LOCATOR_DEVICE_NUMBER;
   RETURNED_VALUES     : in   RETURN_VALUE_TYPE;
   ERROR_INDICATOR     : out  ERROR_NUMBER;
   MODE                : out  OPERATING_MODE;
   SWITCH              : out  ECHO_SWITCH;
   INITIAL_VIEW_IND    : out  VIEW_INDEX;
   INITIAL_POSITION    : out  WC.POINT_3;
   ECHO_VOLUME         : out  DC.RECTANGULAR_REGION_3;
   DATA_RECORD         : out  LOCATOR_DATA_RECORD);

procedure INQ_LOCATOR_DEVICE_STATE
  (WS                  : in   WS_ID;
   DEVICE              : in   LOCATOR_DEVICE_NUMBER;
   RETURNED_VALUES     : in   RETURN_VALUE_TYPE;
   ERROR_INDICATOR     : out  ERROR_NUMBER;
   MODE                : out  OPERATING_MODE;
   SWITCH              : out  ECHO_SWITCH;
   INITIAL_VIEW_IND    : out  VIEW_INDEX;
   INITIAL_POSITION    : out  WC.POINT_2;
   ECHO_AREA           : out  DC.RECTANGULAR_REGION_2;
   DATA_RECORD         : out  LOCATOR_DATA_RECORD);

procedure INQ_STROKE_DEVICE_STATE
  (WS                       : in   WS_ID;
   DEVICE                   : in   STROKE_DEVICE_NUMBER;
   RETURNED_VALUES          : in   RETURN_VALUE_TYPE;
   ERROR_INDICATOR          : out  ERROR_NUMBER;
   MODE                     : out  OPERATING_MODE;
   SWITCH                   : out  ECHO_SWITCH;
   INITIAL_VIEW_IND         : out  VIEW_INDEX;
   INITIAL_STROKE_POINTS    : out  WC.ACCESS_POINT_LIST_3;
   ECHO_VOLUME              : out  DC.RECTANGULAR_REGION_3;
   DATA_RECORD              : out  STROKE_DATA_RECORD);

procedure INQ_STROKE_DEVICE_STATE
  (WS                       : in   WS_ID;
   DEVICE                   : in   STROKE_DEVICE_NUMBER;
   RETURNED_VALUES          : in   RETURN_VALUE_TYPE;
   ERROR_INDICATOR          : out  ERROR_NUMBER;
   MODE                     : out  OPERATING_MODE;
   SWITCH                   : out  ECHO_SWITCH;
   INITIAL_VIEW_IND         : out  VIEW_INDEX;
   INITIAL_STROKE_POINTS    : out  WC.ACCESS_POINT_LIST_2;
   ECHO_AREA                : out  DC.RECTANGULAR_REGION_2;
   DATA_RECORD              : out  STROKE_DATA_RECORD);

procedure INQ_VALUATOR_DEVICE_STATE
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
    (WS                 : in  WS_ID;
     DEVICE             : in  VALUATOR_DEVICE_NUMBER;
     ERROR_INDICATOR    : out ERROR_NUMBER;
     MODE               : out OPERATING_MODE;
     SWITCH             : out ECHO_SWITCH;
     INITIAL_VALUE      : out VALUATOR_VALUE;
     ECHO_VOLUME        : out DC.RECTANGULAR_REGION_3;
     DATA_RECORD        : out VALUATOR_DATA_RECORD);

  procedure INQ_VALUATOR_DEVICE_STATE
    (WS                 : in  WS_ID;
     DEVICE             : in  VALUATOR_DEVICE_NUMBER;
     ERROR_INDICATOR    : out ERROR_NUMBER;
     MODE               : out OPERATING_MODE;
     SWITCH             : out ECHO_SWITCH;
     INITIAL_VALUE      : out VALUATOR_VALUE;
     ECHO_AREA          : out DC.RECTANGULAR_REGION_2;
     DATA_RECORD        : out VALUATOR_DATA_RECORD);

  procedure INQ_CHOICE_DEVICE_STATE
    (WS                 : in  WS_ID;
     DEVICE             : in  CHOICE_DEVICE_NUMBER;
     ERROR_INDICATOR    : out ERROR_NUMBER;
     MODE               : out OPERATING_MODE;
     SWITCH             : out ECHO_SWITCH;
     INITIAL_STATUS     : out CHOICE_STATUS;
     INITIAL_CHOICE     : out CHOICE_NUMBER;
     ECHO_VOLUME        : out DC.RECTANGULAR_REGION_3;
     DATA_RECORD        : out CHOICE_DATA_RECORD);

  procedure INQ_CHOICE_DEVICE_STATE
    (WS                 : in  WS_ID;
     DEVICE             : in  CHOICE_DEVICE_NUMBER;
     ERROR_INDICATOR    : out ERROR_NUMBER;
     MODE               : out OPERATING_MODE;
     SWITCH             : out ECHO_SWITCH;
     INITIAL_STATUS     : out CHOICE_STATUS;
     INITIAL_CHOICE     : out CHOICE_NUMBER;
     ECHO_AREA          : out DC.RECTANGULAR_REGION_2;
     DATA_RECORD        : out CHOICE_DATA_RECORD);

  procedure INQ_PICK_DEVICE_STATE
    (WS                 : in  WS_ID;
     DEVICE             : in  PICK_DEVICE_NUMBER;
     RETURNED_VALUES    : in  RETURN_VALUE_TYPE;
     ERROR_INDICATOR    : out ERROR_NUMBER;
     MODE               : out OPERATING_MODE;
     SWITCH             : out ECHO_SWITCH;
     PICKABILITY        : out NAME_SET_FILTER;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567469 T54 ■


```
    INITIAL_STATUS      : out  PICK_STATUS;
    INITIAL_PATH        : out  PICK_PATH;
    ECHO_VOLUME         : out  DC.RECTANGULAR_REGION_3;
    DATA_RECORD         : out  PICK_DATA_RECORD;
    ORDER               : out  PATH_ORDER);

  procedure INQ_PICK_DEVICE_STATE
    (WS                 : in   WS_ID;
    DEVICE              : in   PICK_DEVICE_NUMBER;
    RETURNED_VALUES     : in   RETURN_VALUE_TYPE;
    ERROR_INDICATOR     : out  ERROR_NUMBER;
    MODE                : out  OPERATING_MODE;
    SWITCH              : out  ECHO_SWITCH;
    PICKABILITY         : out  NAME_SET_FILTER;
    INITIAL_STATUS      : out  PICK_STATUS;
    INITIAL_PATH        : out  PICK_PATH;
    ECHO_AREA           : out  DC.RECTANGULAR_REGION_2;
    DATA_RECORD         : out  PICK_DATA_RECORD;
    ORDER               : out  PATH_ORDER);

  procedure INQ_STRING_DEVICE_STATE
    (WS                      : in   WS_ID;
    DEVICE                   : in   STRING_DEVICE_NUMBER;
    ERROR_INDICATOR          : out  ERROR_NUMBER;
    MODE                     : out  OPERATING_MODE;
    SWITCH                   : out  ECHO_SWITCH;
    INITIAL_CHAR_STRING      : out  INPUT_STRING;
    ECHO_VOLUME              : out  DC.RECTANGULAR_REGION_3;
    DATA_RECORD              : out  STRING_DATA_RECORD);

  procedure INQ_STRING_DEVICE_STATE
    (WS                      : in   WS_ID;
    DEVICE                   : in   STRING_DEVICE_NUMBER;
    ERROR_INDICATOR          : out  ERROR_NUMBER;
    MODE                     : out  OPERATING_MODE;
    SWITCH                   : out  ECHO_SWITCH;
    INITIAL_CHAR_STRING      : out  INPUT_STRING;
    ECHO_AREA                : out  DC.RECTANGULAR_REGION_2;
    DATA_RECORD              : out  STRING_DATA_RECORD);

  procedure INQ_WS_CATEGORY
    (TYPE_OF_WS         : in   WS_TYPE;
    ERROR_INDICATOR     : out  ERROR_NUMBER;
    CATEGORY            : out  WS_CATEGORY);

  procedure INQ_DISPLAY_SPACE_SIZE
    (TYPE_OF_WS              : in   WS_TYPE;
    ERROR_INDICATOR          : out  ERROR_NUMBER;
    UNITS                    : out  DC_UNITS;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
    MAX_DC_UNIT_SIZE       : out  DC.SIZE_3;
    MAX_RASTER_UNIT_SIZE   : out  RASTER_UNIT_SIZE_3);

procedure INQ_DISPLAY_SPACE_SIZE
  (TYPE_OF_WS              : in   WS_TYPE;
   ERROR_INDICATOR         : out  ERROR_NUMBER;
   UNITS                   : out  DC_UNITS;
   MAX_DC_UNIT_SIZE        : out  DC.SIZE_2;
   MAX_RASTER_UNIT_SIZE    : out  RASTER_UNIT_SIZE_2);

procedure INQ_HLHSR_FACILITIES
  (TYPE_OF_WS              : in   WS_TYPE;
   ERROR_INDICATOR         : out  ERROR_NUMBER;
   LIST_OF_IDENTIFIERS     : out  HLHSR_IDS.LIST_OF;
   LIST_OF_MODES           : out  HLHSR_MODES.LIST_OF);

procedure INQ_VIEW_FACILITIES
  (TYPE_OF_WS              : in   WS_TYPE;
   ERROR_INDICATOR         : out  ERROR_NUMBER;
   NUMBER_OF_INDICES       : out  PHIGS_POSITIVE);

procedure INQ_PREDEFINED_VIEW_REPRESENTATION
  (TYPE_OF_WS              : in   WS_TYPE;
   VIEW_IND                : in   VIEW_INDEX;
   ERROR_INDICATOR         : out  ERROR_NUMBER;
   ORIENTATION_MATRIX      : out  TRANSFORMATION_MATRIX_3;
   MAPPING_MATRIX          : out  TRANSFORMATION_MATRIX_3;
   CLIPPING_LIMITS         : out  NPC.RECTANGULAR_REGION_3;
   XY_CLIPPING             : out  CLIPPING_INDICATOR;
   BACK_CLIPPING           : out  CLIPPING_INDICATOR;
   FRONT_CLIPPING          : out  CLIPPING_INDICATOR);

procedure INQ_WS_CLASSIFICATION
  (TYPE_OF_WS              : in   WS_TYPE;
   ERROR_INDICATOR         : out  ERROR_NUMBER;
   CLASS                   : out  DISPLAY_CLASS);

procedure INQ_DYNAMICS_OF_WS_ATTRIBUTES
  (TYPE_OF_WS                      : in   WS_TYPE;
   ERROR_INDICATOR                 : out  ERROR_NUMBER;
   POLYLINE_REPRESENTATION         : out  DYNAMIC_MODIFICATION;
   POLYMARKER_REPRESENTATION       : out  DYNAMIC_MODIFICATION;
   TEXT_REPRESENTATION             : out  DYNAMIC_MODIFICATION;
   INTERIOR_REPRESENTATION         : out  DYNAMIC_MODIFICATION;
   EDGE_REPRESENTATION             : out  DYNAMIC_MODIFICATION;
   PATTERN_REPRESENTATION          : out  DYNAMIC_MODIFICATION;
   COLOUR_REPRESENTATION           : out  DYNAMIC_MODIFICATION;
   VIEW_REPRESENTATION             : out  DYNAMIC_MODIFICATION;
   WS_TRANSFORMATION               : out  DYNAMIC_MODIFICATION;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567471 602 ■


```
    HIGHLIGHTING_FILTER              : out  DYNAMIC_MODIFICATION;
    INVISIBILITY_FILTER              : out  DYNAMIC_MODIFICATION;
    HLHSR_MODE_CHANGE                : out  DYNAMIC_MODIFICATION);

procedure INQ_DYNAMICS_OF_WS_ATTRIBUTES
  (TYPE_OF_WS                                 : in  WS_TYPE;
    ERROR_INDICATOR                           : out ERROR_NUMBER;
    DATA_MAPPING_REPRESENTATION               : out DYNAMIC_MODIFICATION;
    REFLECTANCE_REPRESENTATION                : out DYNAMIC_MODIFICATION;
    PARAMETRIC_SURFACE_REPRESENTATION         : out DYNAMIC_MODIFICATION;
    WS_LIGHT_SOURCE_REPRESENTATION            : out DYNAMIC_MODIFICATION;
    DEPTH_CUE_REPRESENTATION                  : out DYNAMIC_MODIFICATION;
    COLOUR_MAPPING_REPRESENTATION             : out DYNAMIC_MODIFICATION);

procedure INQ_DEFAULT_DISPLAY_UPDATE_STATE
  (TYPE_OF_WS        : in   WS_TYPE;
    ERROR_INDICATOR: out  ERROR_NUMBER;
    DEFERRAL         : out  DEFERRAL_MODE;
    MODIFICATION     : out  MODIFICATION_MODE);

procedure INQ_POLYLINE_FACILITIES
  (TYPE_OF_WS          : in   WS_TYPE;
    ERROR_INDICATOR    : out  ERROR_NUMBER;
    NUMBER_OF_TYPES    : out  PHIGS_INTEGER;
    LIST_OF_TYPES      : out  LINETYPES.LIST_OF;
    NUMBER_OF_WIDTHS : out  PHIGS_NATURAL;
    NOMINAL_WIDTH      : out  DC.MAGNITUDE;
    RANGE_OF_WIDTHS    : out  DC.RANGE_OF_MAGNITUDES;
    NUMBER_OF_INDICES: out  PHIGS_NATURAL);

procedure INQ_POLYLINE_FACILITIES
  (TYPE_OF_WS                    : in  WS_TYPE;
    ERROR_INDICATOR              : out ERROR_NUMBER;
    NUMBER_OF_TYPES              : out PHIGS_INTEGER;
    LIST_OF_TYPES                : out LINETYPES.LIST_OF;
    NUMBER_OF_WIDTHS             : out PHIGS_NATURAL;
    NOMINAL_WIDTH                : out DC.MAGNITUDE;
    RANGE_OF_WIDTHS              : out DC.RANGE_OF_MAGNITUDES;
    LIST_OF_SHADING_METHODS      : out POLYLINE_SHADING_METHODS.LIST_OF;
    NUMBER_OF_INDICES            : out PHIGS_NATURAL);

procedure INQ_PREDEFINED_POLYLINE_REPRESENTATION
  (TYPE_OF_WS        : in   WS_TYPE;
    POLYLINE_IND     : in   POLYLINE_INDEX;
    ERROR_INDICATOR: out  ERROR_NUMBER;
    TYPE_OF_LINE     : out  LINETYPE;
    LINEWIDTH_SF     : out  LINEWIDTH;
    LINE_COLOUR      : out  COLOUR_INDEX);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567472 549 ■


```
procedure INQ_PREDEFINED_POLYLINE_REPRESENTATION
  (TYPE_OF_WS                    : in  WS_TYPE;
   POLYLINE_IND                  : in  POLYLINE_INDEX;
   ERROR_INDICATOR               : out ERROR_NUMBER;
   TYPE_OF_LINE                  : out LINETYPE;
   LINEWIDTH_SF                  : out LINEWIDTH;
   GENERAL_LINE_COLOUR           : out GENERAL_COLOUR;
   LINE_SHADING                  : out POLYLINE_SHADING_METHOD;
   CRITERIA_FOR_CURVE_APPROX     : out CURVE_APPROX_DATA_RECORD);

procedure INQ_POLYMARKER_FACILITIES
  (TYPE_OF_WS          : in  WS_TYPE;
   ERROR_INDICATOR     : out ERROR_NUMBER;
   NUMBER_OF_TYPES     : out PHIGS_INTEGER;
   LIST_OF_TYPES       : out MARKER_TYPES.LIST_OF;
   NUMBER_OF_SIZES     : out PHIGS_NATURAL;
   NOMINAL_SIZE        : out DC.MAGNITUDE;
   RANGE_OF_SIZES      : out DC.RANGE_OF_MAGNITUDES;
   NUMBER_OF_INDICES   : out PHIGS_NATURAL);

procedure INQ_PREDEFINED_POLYMARKER_REPRESENTATION
  (TYPE_OF_WS       : in  WS_TYPE;
   POLYMARKER_IND   : in  POLYMARKER_INDEX;
   ERROR_INDICATOR  : out ERROR_NUMBER;
   TYPE_OF_MARKER   : out MARKER_TYPE;
   SIZE             : out MARKER_SIZE;
   MARKER_COLOUR    : out COLOUR_INDEX);

procedure INQ_PREDEFINED_POLYMARKER_REPRESENTATION
  (TYPE_OF_WS                    : in  WS_TYPE;
   POLYMARKER_IND                : in  POLYMARKER_INDEX;
   ERROR_INDICATOR               : out ERROR_NUMBER;
   TYPE_OF_MARKER                : out MARKER_TYPE;
   SIZE                          : out MARKER_SIZE;
   GENERAL_MARKER_COLOUR         : out GENERAL_COLOUR);

procedure INQ_TEXT_FACILITIES
  (TYPE_OF_WS                    : in  WS_TYPE;
   ERROR_INDICATOR               : out ERROR_NUMBER;
   LIST_OF_FONT_PRECISION_PAIRS  : out TEXT_FONT_PRECISIONS.LIST_OF;
   NUMBER_OF_HEIGHTS             : out PHIGS_NATURAL;
   RANGE_OF_HEIGHTS              : out DC.RANGE_OF_MAGNITUDES;
   NUMBER_OF_EXPANSIONS          : out PHIGS_NATURAL;
   RANGE_OF_EXPANSIONS           : out DC.RANGE_OF_MAGNITUDES;
   NUMBER_OF_INDICES             : out PHIGS_NATURAL);

procedure INQ_PREDEFINED_TEXT_REPRESENTATION
  (TYPE_OF_WS       : in  WS_TYPE;
   TEXT_IND         : in  TEXT_INDEX;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567473 485 ■


```
    ERROR_INDICATOR: out  ERROR_NUMBER;
    FONT             : out  TEXT_FONT;
    PRECISION        : out  TEXT_PRECISION;
    EXPANSION        : out  CHAR_EXPANSION;
    SPACING          : out  CHAR_SPACING;
    TEXT_COLOUR      : out  COLOUR_INDEX);

procedure INQ_PREDEFINED_TEXT_REPRESENTATION
  (TYPE_OF_WS              : in  WS_TYPE;
   TEXT_IND                : in  TEXT_INDEX;
   ERROR_INDICATOR         : out ERROR_NUMBER;
   FONT                    : out TEXT_FONT;
   PRECISION               : out TEXT_PRECISION;
   EXPANSION               : out CHAR_EXPANSION;
   SPACING                 : out CHAR_SPACING;
   GENERAL_TEXT_COLOUR     : out GENERAL_COLOUR);

procedure INQ_ANNOTATION_FACILITIES
  (TYPE_OF_WS                   : in  WS_TYPE;
   ERROR_INDICATOR              : out ERROR_NUMBER;
   LIST_OF_ANNOTATION_STYLES    : out ANNOTATION_STYLES.LIST_OF;
   NUMBER_OF_CHAR_HEIGHTS       : out PHIGS_NATURAL;
   RANGE_OF_CHAR_HEIGHTS        : out DC.RANGE_OF_MAGNITUDES);

procedure INQ_TEXT_EXTENT
  (TYPE_OF_WS              : in  WS_ID;
   FONT                    : in  TEXT_FONT;
   EXPANSION               : in  CHAR_EXPANSION;
   SPACING                 : in  CHAR_SPACING;
   HEIGHT                  : in  MC.MAGNITUDE;
   PATH                    : in  TEXT_PATH;
   ALIGNMENT               : in  TEXT_ALIGNMENT;
   CHAR_STRING             : in  PHIGS_STRING;
   ERROR_INDICATOR         : out ERROR_NUMBER;
   EXTENT_RECTANGLE        : out MC.RECTANGULAR_REGION_2;
   CONCATENATION_OFFSET    : out MC.VECTOR_2);

-- This procedure is for use with text primitives.

procedure INQ_TEXT_EXTENT
  (TYPE_OF_WS              : in  WS_ID;
   FONT                    : in  TEXT_FONT;
   EXPANSION               : in  CHAR_EXPANSION;
   SPACING                 : in  CHAR_SPACING;
   HEIGHT                  : in  NPC.MAGNITUDE;
   PATH                    : in  TEXT_PATH;
   ALIGNMENT               : in  TEXT_ALIGNMENT;
   CHAR_STRING             : in  PHIGS_STRING;
   ERROR_INDICATOR         : out ERROR_NUMBER;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
    EXTENT_RECTANGLE       : out  NPC.RECTANGULAR_REGION_2;
    CONCATENATION_OFFSET   : out  NPC.VECTOR_2);

-- This procedure is for use with annotation text primitives.

procedure INQ_INTERIOR_FACILITIES
  (TYPE_OF_WS                  : in   WS_TYPE;
   ERROR_INDICATOR             : out  ERROR_NUMBER;
   LIST_OF_INTERIOR_STYLES     : out  INTERIOR_STYLES.LIST_OF;
   NUMBER_OF_HATCH_STYLES      : out  PHIGS_NATURAL;
   LIST_OF_HATCH_STYLES        : out  HATCH_STYLES.LIST_OF;
   NUMBER_OF_INDICES           : out  PHIGS_NATURAL);

procedure INQ_INTERIOR_FACILITIES
  (TYPE_OF_WS                  : in  WS_TYPE;
   ERROR_INDICATOR             : out ERROR_NUMBER;
   LIST_OF_INTERIOR_STYLES     : out INTERIOR_STYLES.LIST_OF;
   NUMBER_OF_HATCH_STYLES      : out PHIGS_NATURAL;
   LIST_OF_HATCH_STYLES        : out HATCH_STYLES.LIST_OF;
   LIST_OF_SHADING_METHODS     : out INTERIOR_SHADING_METHODS.LIST_OF;
   NUMBER_OF_INDICES           : out PHIGS_NATURAL);

procedure INQ_PREDEFINED_INTERIOR_REPRESENTATION
  (TYPE_OF_WS          : in   WS_TYPE;
   INTERIOR_IND        : in   INTERIOR_INDEX;
   ERROR_INDICATOR     : out  ERROR_NUMBER;
   STYLE_OF_INTERIOR   : out  INTERIOR_STYLE;
   STYLE_IND           : out  STYLE_INDEX;
   INTERIOR_COLOUR     : out  COLOUR_INDEX);

procedure INQ_PREDEFINED_INTERIOR_REPRESENTATION
  (TYPE_OF_WS                  : in  WS_TYPE;
   INTERIOR_IND                : in  INTERIOR_INDEX;
   ERROR_INDICATOR             : out ERROR_NUMBER;
   STYLE_OF_INTERIOR           : out INTERIOR_STYLE;
   STYLE_IND                   : out STYLE_INDEX;
   GENERAL_INTERIOR_COLOUR     : out GENERAL_COLOUR;
   INTERIOR_SHADING            : out INTERIOR_SHADING_METHOD);

procedure INQ_EDGE_FACILITIES
  (TYPE_OF_WS          : in   WS_TYPE;
   ERROR_INDICATOR     : out  ERROR_NUMBER;
   NUMBER_OF_TYPES     : out  PHIGS_INTEGER;
   LIST_OF_TYPES       : out  EDGETYPES.LIST_OF;
   NUMBER_OF_WIDTHS    : out  PHIGS_NATURAL;
   NOMINAL_WIDTH       : out  DC.MAGNITUDE;
   RANGE_OF_WIDTHS     : out  DC.RANGE_OF_MAGNITUDES;
   NUMBER_OF_INDICES   : out  PHIGS_NATURAL);
```


--`,,`-`-`,,`,,`,`,,`---

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567475 258 ■


```
procedure INQ_PREDEFINED_EDGE_REPRESENTATION
  (TYPE_OF_WS          : in   WS_TYPE;
   EDGE_IND            : in   EDGE_INDEX;
   ERROR_INDICATOR: out  ERROR_NUMBER;
   FLAG                : out  EDGE_FLAG;
   TYPE_OF_EDGE        : out  EDGETYPE;
   EDGEWIDTH_SF        : out  EDGEWIDTH;
   EDGE_COLOUR         : out  COLOUR_INDEX);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567476 194 ■


```
procedure INQ_PREDEFINED_EDGE_REPRESENTATION
  (TYPE_OF_WS                : in  WS_TYPE;
   EDGE_IND                  : in  EDGE_INDEX;
   ERROR_INDICATOR           : out ERROR_NUMBER;
   FLAG                      : out EDGE_FLAG;
   TYPE_OF_EDGE              : out EDGETYPE;
   EDGEWIDTH_SF              : out EDGEWIDTH;
   GENERAL_EDGE_COLOUR       : out GENERAL_COLOUR);

procedure INQ_DATA_MAPPING_FACILITIES
  (TYPE_OF_WS                : in  WS_TYPE;
   ERROR_INDICATOR           : out ERROR_NUMBER;
   LIST_OF_METHODS           : out DATA_MAPPING_METHODS.LIST_OF;
   NUMBER_OF_INDICES         : out PHIGS_POSITIVE);

procedure INQ_PREDEFINED_DATA_MAPPING_REPRESENTATION
  (TYPE_OF_WS                : in  WS_TYPE;
   DATA_MAPPING_IND          : in  DATA_MAPPING_INDEX;
   ERROR_INDICATOR           : out ERROR_NUMBER;
   DATA_MAPPING              : out DATA_MAPPING_DATA_RECORD);

procedure INQ_REFLECTANCE_FACILITIES
  (TYPE_OF_WS                : in  WS_TYPE;
   ERROR_INDICATOR           : out ERROR_NUMBER;
   MODELS                    : out REFLECTANCE_MODELS.LIST_OF;
   PROPERTIES                : out REFLECTANCE_PROPERTIES_TYPES.LIST_OF;
   NUMBER_OF_INDICES         : out PHIGS_POSITIVE);

procedure INQ_PREDEFINED_REFLECTANCE_REPRESENTATION
  (TYPE_OF_WS                : in  WS_TYPE;
   REFLECTANCE_IND           : in  REFLECTANCE_INDEX;
   ERROR_INDICATOR           : out ERROR_NUMBER;
   MODEL                     : out REFLECTANCE_MODEL;
   REFLECTANCE_PROPERTIES    : out REFLECTANCE_PROPERTIES_DATA_RECORD);

procedure INQ_CURVE_AND_SURFACE_FACILITIES
  (TYPE_OF_WS                : in  WS_TYPE;
   ERROR_INDICATOR           : out ERROR_NUMBER;
   MAX_CURVE_ORDER           : out RETURN_SPLINE_ORDER;
   MAX_SURFACE_ORDER         : out RETURN_SPLINE_ORDER;
   MAX_TRIM_ORDER            : out RETURN_TRIMCURVE_ORDER;
   LIST_OF_CURVE_APPROX_CRITERIA_TYPES
                             : out CURVE_APPROX_CRITERIA_TYPES.LIST_OF;
   LIST_OF_SURFACE_APPROX_CRITERIA_TYPES
                             : out SURFACE_APPROX_CRITERIA_TYPES.LIST_OF;
   LIST_OF_TRIM_APPROX_CRITERIA_TYPES
                             : out CURVE_APPROX_CRITERIA_TYPES.LIST_OF;
   LIST_OF_CHARACTERISTICS
                          : out PARAMETRIC_SURFACE_CHARACTERISTICS.LIST_OF;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567477 020 ■


```
    NUMBER_OF_INDICES          : out PHIGS_POSITIVE);

procedure INQ_PREDEFINED_PARAMETRIC_SURFACE_REPRESENTATION
  (TYPE_OF_WS                        : in  WS_TYPE;
   PARAMETRIC_SURFACE_IND            : in  PARAMETRIC_SURFACE_INDEX;
   ERROR_INDICATOR                   : out ERROR_NUMBER;
   CRITERIA_FOR_SURFACE_APPROX       : out SURFACE_APPROX_DATA_RECORD;
   PARAMETRIC_SURFACE_CHARACTERISTICS
                                     : out PARAMETRIC_SURFACE_DATA_RECORD);

procedure INQ_PATTERN_FACILITIES
  (TYPE_OF_WS           : in  WS_TYPE;
   ERROR_INDICATOR      : out ERROR_NUMBER;
   NUMBER_OF_INDICES    : out PHIGS_NATURAL);

procedure INQ_PREDEFINED_PATTERN_REPRESENTATION
  (TYPE_OF_WS           : in  WS_TYPE;
   PATTERN_IND          : in  PATTERN_INDEX;
   ERROR_INDICATOR      : out ERROR_NUMBER;
   PATTERN              : out ACCESS_COLOUR_MATRIX);

procedure INQ_PREDEFINED_PATTERN_REPRESENTATION
  (TYPE_OF_WS           : in  WS_TYPE;
   PATTERN_IND          : in  PATTERN_INDEX;
   ERROR_INDICATOR      : out ERROR_NUMBER;
   PATTERN              : out ACCESS_COLOUR_VALUE_ARRAY);

procedure INQ_COLOUR_MODEL_FACILITIES
  (TYPE_OF_WS           : in  WS_TYPE;
   ERROR_INDICATOR      : out ERROR_NUMBER;
   LIST_OF_MODELS       : out COLOUR_MODELS.LIST_OF;
   DEFAULT_MODEL        : out COLOUR_MODEL);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567478 T67 ■


```
procedure INQ_LIGHT_SOURCE_FACILITIES
  (TYPE_OF_WS                  : in  WS_TYPE;
   ERROR_INDICATOR             : out ERROR_NUMBER;
   LIST_OF_TYPES               : out LIGHT_SOURCE_TYPES.LIST_OF;
   MAX_SIMULTANEOUS_LIGHTS     : out PHIGS_POSITIVE;
   NUMBER_OF_INDICES           : out RETURN_LIGHT_SOURCE_COUNT);

procedure INQ_PREDEFINED_LIGHT_SOURCE_REPRESENTATION
  (TYPE_OF_WS          : in  WS_TYPE;
   LIGHT_SOURCE_IND    : in  LIGHT_SOURCE_INDEX;
   ERROR_INDICATOR     : out ERROR_NUMBER;
   LIGHT_SOURCE        : out LIGHT_SOURCE_DATA_RECORD);

procedure INQ_DEPTH_CUE_FACILITIES
  (TYPE_OF_WS          : in  WS_TYPE;
   ERROR_INDICATOR     : out ERROR_NUMBER;
   NUMBER_OF_INDICES   : out RETURN_DEPTH_CUE_INDEX_COUNT;
   AVAILABLE_MODES     : out DEPTH_CUE_MODES.LIST_OF);

procedure INQ_PREDEFINED_DEPTH_CUE_REPRESENTATION
  (TYPE_OF_WS          : in  WS_TYPE;
   DEPTH_CUE_IND       : in  DEPTH_CUE_INDEX;
   ERROR_INDICATOR     : out ERROR_NUMBER;
   MODE                : out DEPTH_CUE_MODE;
   REFERENCE_PLANES    : out NPC.RANGE_OF_MAGNITUDES;
   DEPTH_CUE_SF        : out DEPTH_CUE_SCALE_FACTORS;
   DEPTH_CUE_COLOUR    : out GENERAL_COLOUR);

procedure INQ_COLOUR_MAPPING_FACILITIES
  (TYPE_OF_WS          : in  WS_TYPE;
   ERROR_INDICATOR     : out ERROR_NUMBER;
   LIST_OF_METHODS     : out COLOUR_MAPPING_METHODS.LIST_OF;
   NUMBER_OF_INDICES   : out PHIGS_POSITIVE);

procedure INQ_COLOUR_MAPPING_METHOD_FACILITIES
  (TYPE_OF_WS          : in  WS_TYPE;
   METHOD              : in  COLOUR_MAPPING_METHOD;
   ERROR_INDICATOR     : out ERROR_NUMBER;
   FACILITIES          : out COLOUR_MAPPING_METHOD_FACILITIES);

procedure INQ_PREDEFINED_COLOUR_MAPPING_REPRESENTATION
  (TYPE_OF_WS          : in  WS_TYPE;
   COLOUR_MAPPING_IND  : in  COLOUR_MAPPING_INDEX;
   ERROR_INDICATOR     : out ERROR_NUMBER;
   COLOUR_MAPPING      : out ACCESS_COLOUR_MAPPING_DATA_RECORD);

procedure INQ_DIRECT_COLOUR_MODEL_FACILITIES
  (TYPE_OF_WS          : in  WS_TYPE;
   ERROR_INDICATOR     : out ERROR_NUMBER;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567479 9T3 ■


```
    LIST_OF_MODELS    : out COLOUR_MODELS.LIST_OF);

procedure INQ_RENDERING_COLOUR_MODEL_FACILITIES
  (TYPE_OF_WS         : in  WS_TYPE;
   ERROR_INDICATOR    : out ERROR_NUMBER;
   LIST_OF_MODELS     : out COLOUR_MODELS.LIST_OF);

procedure INQ_COLOUR_FACILITIES
  (TYPE_OF_WS                 : in  WS_TYPE;
   ERROR_INDICATOR            : out ERROR_NUMBER;
   NUMBER_OF_COLOURS          : out PHIGS_NATURAL;
   AVAILABILITY_OF_COLOUR     : out COLOUR_AVAILABLE;
   NUMBER_OF_INDICES          : out PHIGS_NATURAL;
   PRIMARY_COLOURS            : out CHROMATICITY_COEFFICIENT_SET);

procedure INQ_PREDEFINED_COLOUR_REPRESENTATION
  (TYPE_OF_WS       : in  WS_TYPE;
   INDEX            : in  COLOUR_INDEX;
   ERROR_INDICATOR  : out ERROR_NUMBER;
   COLOUR           : out COLOUR_REPRESENTATION);

procedure INQ_LIST_OF_AVAILABLE_GDP
  (TYPE_OF_WS       : in  WS_TYPE;
   ERROR_INDICATOR  : out ERROR_NUMBER;
   LIST_OF_GDP      : out GDP_3_IDS.LIST_OF);

procedure INQ_LIST_OF_AVAILABLE_GDP
  (TYPE_OF_WS       : in  WS_TYPE;
   ERROR_INDICATOR  : out ERROR_NUMBER;
   LIST_OF_GDP      : out GDP_IDS.LIST_OF);

procedure INQ_GDP
  (TYPE_OF_WS                   : in  WS_TYPE;
   GDP                          : in  GDP_3_ID;
   ERROR_INDICATOR              : out ERROR_NUMBER;
   LIST_OF_ATTRIBUTE_SETS_USED  : out ATTRIBUTES_USED.LIST_OF);

procedure INQ_GDP
  (TYPE_OF_WS                   : in  WS_TYPE;
   GDP                          : in  GDP_ID;
   ERROR_INDICATOR              : out ERROR_NUMBER;
   LIST_OF_ATTRIBUTE_SETS_USED  : out ATTRIBUTES_USED.LIST_OF);

procedure INQ_LIST_OF_AVAILABLE_GSE
  (TYPE_OF_WS       : in  WS_TYPE;
   ERROR_INDICATOR  : out ERROR_NUMBER;
   LIST_OF_GSE      : out GSE_IDS.LIST_OF);

procedure INQ_NUMBER_OF_DISPLAY_PRIORITIES_SUPPORTED
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567480 615 ■


```
  (TYPE_OF_WS              : in   WS_TYPE;
   ERROR_INDICATOR         : out  ERROR_NUMBER;
   NUMBER_OF_PRIORITIES    : out  PHIGS_NATURAL);

procedure INQ_WS_STATE_TABLE_LENGTHS
  (TYPE_OF_WS              : in   WS_TYPE;
   ERROR_INDICATOR         : out  ERROR_NUMBER;
   MAX_POLYLINE_ENTRIES    : out  PHIGS_NATURAL;
   MAX_POLYMARKER_ENTRIES  : out  PHIGS_NATURAL;
   MAX_TEXT_ENTRIES        : out  PHIGS_NATURAL;
   MAX_INTERIOR_ENTRIES    : out  PHIGS_NATURAL;
   MAX_EDGE_ENTRIES        : out  PHIGS_NATURAL;
   MAX_PATTERN_INDICES     : out  PHIGS_NATURAL;
   MAX_COLOUR_INDICES      : out  PHIGS_NATURAL;
   MAX_VIEW_INDICES        : out  PHIGS_NATURAL);

procedure INQ_WS_STATE_TABLE_LENGTHS
  (TYPE_OF_WS                      : in  WS_TYPE;
   ERROR_INDICATOR                 : out ERROR_NUMBER;
   MAX_DATA_MAPPING_ENTRIES        : out PHIGS_NATURAL;
   MAX_REFLECTANCE_ENTRIES         : out PHIGS_NATURAL;
   MAX_PARAMETRIC_SURFACE_ENTRIES  : out PHIGS_NATURAL;
   MAX_LIGHT_SOURCE_ENTRIES        : out PHIGS_NATURAL;
   MAX_DEPTH_CUE_ENTRIES           : out PHIGS_NATURAL;
   MAX_COLOUR_MAPPING_ENTRIES      : out PHIGS_NATURAL);

procedure INQ_DYNAMICS_OF_STRUCTURES
  (TYPE_OF_WS              : in   WS_TYPE;
   ERROR_INDICATOR         : out  ERROR_NUMBER;
   CONTENT_MODIFICATION    : out  DYNAMIC_MODIFICATION;
   POST                    : out  DYNAMIC_MODIFICATION;
   UNPOST                  : out  DYNAMIC_MODIFICATION;
   DELETE                  : out  DYNAMIC_MODIFICATION;
   REFERENCE_MODIFICATION  : out  DYNAMIC_MODIFICATION);

procedure INQ_NUMBER_OF_AVAILABLE_LOGICAL_INPUT_DEVICES
  (TYPE_OF_WS        : in   WS_TYPE;
   ERROR_INDICATOR   : out  ERROR_NUMBER;
   LOCATOR           : out  PHIGS_NATURAL;
   STROKE            : out  PHIGS_NATURAL;
   VALUATOR          : out  PHIGS_NATURAL;
   CHOICE            : out  PHIGS_NATURAL;
   PICK              : out  PHIGS_NATURAL;
   STRING            : out  PHIGS_NATURAL);

procedure INQ_DEFAULT_LOCATOR_DEVICE_DATA
  (TYPE_OF_WS        : in   WS_TYPE;
   DEVICE            : in   LOCATOR_DEVICE_NUMBER;
   ERROR_INDICATOR   : out  ERROR_NUMBER;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
    INITIAL_POSITION  : out  WC.POINT_3;
    LIST_OF_PROMPT_ECHO_TYPES
                      : out  LOCATOR_PROMPT_ECHO_TYPES.LIST_OF;
    ECHO_VOLUME       : out  DC.RECTANGULAR_REGION_3;
    DATA_RECORD       : out  LOCATOR_DATA_RECORD);

procedure INQ_DEFAULT_LOCATOR_DEVICE_DATA
  (TYPE_OF_WS         : in   WS_TYPE;
    DEVICE            : in   LOCATOR_DEVICE_NUMBER;
    ERROR_INDICATOR   : out  ERROR_NUMBER;
    INITIAL_POSITION  : out  WC.POINT_2;
    LIST_OF_PROMPT_ECHO_TYPES
                      : out  LOCATOR_PROMPT_ECHO_TYPES.LIST_OF;
    ECHO_AREA         : out  DC.RECTANGULAR_REGION_2;
    DATA_RECORD       : out  LOCATOR_DATA_RECORD);

procedure INQ_DEFAULT_STROKE_DEVICE_DATA
  (TYPE_OF_WS         : in   WS_TYPE;
    DEVICE            : in   STROKE_DEVICE_NUMBER;
    ERROR_INDICATOR   : out  ERROR_NUMBER;
    MAX_BUFFER_SIZE   : out  PHIGS_NATURAL;
    LIST_OF_PROMPT_ECHO_TYPES
                      : out  STROKE_PROMPT_ECHO_TYPES.LIST_OF;
    ECHO_VOLUME       : out  DC.RECTANGULAR_REGION_3;
    DATA_RECORD       : out  STROKE_DATA_RECORD);

procedure INQ_DEFAULT_STROKE_DEVICE_DATA
  (TYPE_OF_WS         : in   WS_TYPE;
    DEVICE            : in   STROKE_DEVICE_NUMBER;
    ERROR_INDICATOR   : out  ERROR_NUMBER;
    MAX_BUFFER_SIZE   : out  PHIGS_NATURAL;
    LIST_OF_PROMPT_ECHO_TYPES
                      : out  STROKE_PROMPT_ECHO_TYPES.LIST_OF;
    ECHO_AREA         : out  DC.RECTANGULAR_REGION_2;
    DATA_RECORD       : out  STROKE_DATA_RECORD);

procedure INQ_DEFAULT_VALUATOR_DEVICE_DATA
  (TYPE_OF_WS         : in   WS_TYPE;
    DEVICE            : in   VALUATOR_DEVICE_NUMBER;
    ERROR_INDICATOR   : out  ERROR_NUMBER;
    INITIAL_VALUE     : out  VALUATOR_VALUE;
    LIST_OF_PROMPT_ECHO_TYPES
                      : out  VALUATOR_PROMPT_ECHO_TYPES.LIST_OF;
    ECHO_VOLUME       : out  DC.RECTANGULAR_REGION_3;
    DATA_RECORD       : out  VALUATOR_DATA_RECORD);

procedure INQ_DEFAULT_VALUATOR_DEVICE_DATA
  (TYPE_OF_WS         : in   WS_TYPE;
    DEVICE            : in   VALUATOR_DEVICE_NUMBER;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
    ERROR_INDICATOR: out  ERROR_NUMBER;
    INITIAL_VALUE    : out  VALUATOR_VALUE;
    LIST_OF_PROMPT_ECHO_TYPES
                     : out  VALUATOR_PROMPT_ECHO_TYPES.LIST_OF;
    ECHO_AREA        : out  DC.RECTANGULAR_REGION_2;
    DATA_RECORD      : out  VALUATOR_DATA_RECORD);

  procedure INQ_DEFAULT_CHOICE_DEVICE_DATA
    (TYPE_OF_WS      : in   WS_TYPE;
    DEVICE           : in   CHOICE_DEVICE_NUMBER;
    ERROR_INDICATOR: out  ERROR_NUMBER;
    MAX_CHOICES      : out  CHOICE_NUMBER;
    LIST_OF_PROMPT_ECHO_TYPES
                     : out  CHOICE_PROMPT_ECHO_TYPES.LIST_OF;
    ECHO_VOLUME      : out  DC.RECTANGULAR_REGION_3;
    DATA_RECORD      : out  CHOICE_DATA_RECORD);

  procedure INQ_DEFAULT_CHOICE_DEVICE_DATA
    (TYPE_OF_WS      : in   WS_TYPE;
    DEVICE           : in   CHOICE_DEVICE_NUMBER;
    ERROR_INDICATOR: out  ERROR_NUMBER;
    MAX_CHOICES      : out  CHOICE_NUMBER;
    LIST_OF_PROMPT_ECHO_TYPES
                     : out  CHOICE_PROMPT_ECHO_TYPES.LIST_OF;
    ECHO_AREA        : out  DC.RECTANGULAR_REGION_2;
    DATA_RECORD      : out  CHOICE_DATA_RECORD);

  procedure INQ_DEFAULT_PICK_DEVICE_DATA
    (TYPE_OF_WS                  : in   WS_TYPE;
    DEVICE                       : in   PICK_DEVICE_NUMBER;
    ERROR_INDICATOR              : out  ERROR_NUMBER;
    LIST_OF_PROMPT_ECHO_TYPES : out  PICK_PROMPT_ECHO_TYPES.LIST_OF;
    ECHO_VOLUME                  : out  DC.RECTANGULAR_REGION_3;
    DATA_RECORD                  : out  PICK_DATA_RECORD);

  procedure INQ_DEFAULT_PICK_DEVICE_DATA
    (TYPE_OF_WS                  : in   WS_TYPE;
    DEVICE                       : in   PICK_DEVICE_NUMBER;
    ERROR_INDICATOR              : out  ERROR_NUMBER;
    LIST_OF_PROMPT_ECHO_TYPES : out  PICK_PROMPT_ECHO_TYPES.LIST_OF;
    ECHO_AREA                    : out  DC.RECTANGULAR_REGION_2;
    DATA_RECORD                  : out  PICK_DATA_RECORD);

  procedure INQ_DEFAULT_STRING_DEVICE_DATA
    (TYPE_OF_WS                  : in   WS_TYPE;
    DEVICE                       : in   STRING_DEVICE_NUMBER;
    ERROR_INDICATOR              : out  ERROR_NUMBER;
    MAX_STRING_BUFFER_SIZE       : out  PHIGS_NATURAL;
    LIST_OF_PROMPT_ECHO_TYPES : out  STRING_PROMPT_ECHO_TYPES.LIST_OF;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567483 324 ■


```
    ECHO_VOLUME                    : out  DC.RECTANGULAR_REGION_3;
    DATA_RECORD                    : out  STRING_DATA_RECORD);

procedure INQ_DEFAULT_STRING_DEVICE_DATA
  (TYPE_OF_WS                      : in   WS_TYPE;
    DEVICE                         : in   STRING_DEVICE_NUMBER;
    ERROR_INDICATOR                : out  ERROR_NUMBER;
    MAX_STRING_BUFFER_SIZE         : out  PHIGS_NATURAL;
    LIST_OF_PROMPT_ECHO_TYPES      : out  STRING_PROMPT_ECHO_TYPES.LIST_OF;
    ECHO_AREA                      : out  DC.RECTANGULAR_REGION_2;
    DATA_RECORD                    : out  STRING_DATA_RECORD);

procedure INQ_SET_OF_WS_TO_WHICH_POSTED
  (STRUCTURE_IDENTIFIER : in   STRUCTURE_ID;
    ERROR_INDICATOR     : out  ERROR_NUMBER;
    LIST_OF_WS          : out  WS_IDS.LIST_OF);

procedure INQ_OPEN_STRUCTURE
  (ERROR_INDICATOR      : out  ERROR_NUMBER;
    STATUS              : out  OPEN_STRUCTURE_STATUS;
    STRUCTURE_IDENTIFIER : out STRUCTURE_ID);

procedure INQ_ELEMENT_POINTER
  (ERROR_INDICATOR: out  ERROR_NUMBER;
    POSITION      : out  RETURNED_ELEMENT_POSITION);

procedure INQ_CURRENT_ELEMENT_TYPE_AND_SIZE
  (ERROR_INDICATOR: out  ERROR_NUMBER;
    ELEMENT_TYPE  : out  STRUCTURE_ELEMENT_TYPE;
    SIZE          : out  PHIGS_NATURAL);

procedure INQ_CURRENT_ELEMENT_CONTENT
  (ERROR_INDICATOR:    out ERROR_NUMBER;
    ELEMENT_RECORD : in out STRUCTURE_ELEMENT_RECORD);

procedure INQ_ELEMENT_TYPE_AND_SIZE
  (STRUCTURE_IDENTIFIER : in   STRUCTURE_ID;
    POSITION            : in   ELEMENT_POSITION;
    ERROR_INDICATOR     : out  ERROR_NUMBER;
    ELEMENT_TYPE        : out  STRUCTURE_ELEMENT_TYPE;
    SIZE                : out  PHIGS_NATURAL);

procedure INQ_ELEMENT_CONTENT
  (STRUCTURE_IDENTIFIER : in     STRUCTURE_ID;
    POSITION            : in     ELEMENT_POSITION;
    ERROR_INDICATOR     :    out ERROR_NUMBER;
    ELEMENT_RECORD      : in out STRUCTURE_ELEMENT_RECORD);

procedure INQ_STRUCTURE_STATUS
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567484 260 ■


```
  (STRUCTURE_IDENTIFIER : in   STRUCTURE_ID;
   ERROR_INDICATOR      : out  ERROR_NUMBER;
   STATUS               : out  STRUCTURE_STATUS);

procedure INQ_PATHS_TO_ANCESTORS
  (STRUCTURE_IDENTIFIER : in   STRUCTURE_ID;
   TRUNCATION           : in   TRUNCATION_METHOD;
   DEPTH                : in   PATH_DEPTH;
   ERROR_INDICATOR      : out  ERROR_NUMBER;
   LIST_OF_PATHS        : out  REFERENCE_PATHS.LIST_OF);

procedure INQ_PATHS_TO_DESCENDANTS
  (STRUCTURE_IDENTIFIER : in   STRUCTURE_ID;
   TRUNCATION           : in   TRUNCATION_METHOD;
   DEPTH                : in   PATH_DEPTH;
   ERROR_INDICATOR      : out  ERROR_NUMBER;
   LIST_OF_PATHS        : out  REFERENCE_PATHS.LIST_OF);

procedure ELEMENT_SEARCH
  (STRUCTURE_IDENTIFIER : in   STRUCTURE_ID;
   START_POSITION       : in   ELEMENT_POSITION;
   DIRECTION            : in   SEARCH_DIRECTION;
   INCLUSION_SET        : in   ELEMENT_TYPES.LIST_OF;
   EXCLUSION_SET        : in   ELEMENT_TYPES.LIST_OF;
   ERROR_INDICATOR      : out  ERROR_NUMBER;
   SEARCH_STATUS        : out  SEARCH_STATUS_INDICATOR;
   FOUND_POSITION       : out  RETURNED_ELEMENT_POSITION);

procedure INCREMENTAL_SPATIAL_SEARCH
  (SEARCH_REFERENCE_POINT  : in   WC.POINT_3;
   SEARCH_DISTANCE         : in   WC.MAGNITUDE;
   STARTING_PATH           : in   REFERENCE_PATH;
   MODELLING_CLIPPING      : in   CLIPPING_INDICATOR;
   SEARCH_CEILING_INDEX    : in   ELEMENT_POSITION;
   LIST_OF_NORMAL_FILTERS  : in   NAME_SET_FILTERS.LIST_OF;
   LIST_OF_INVERTED_FILTERS : in  NAME_SET_FILTERS.LIST_OF;
   ERROR_INDICATOR         : out  ERROR_NUMBER;
   FOUND_PATH              : out  REFERENCE_PATH);

procedure INCREMENTAL_SPATIAL_SEARCH
  (SEARCH_REFERENCE_POINT  : in   WC.POINT_2;
   SEARCH_DISTANCE         : in   WC.MAGNITUDE;
   STARTING_PATH           : in   REFERENCE_PATH;
   MODELLING_CLIPPING      : in   CLIPPING_INDICATOR;
   SEARCH_CEILING_INDEX    : in   ELEMENT_POSITION;
   LIST_OF_NORMAL_FILTERS  : in   NAME_SET_FILTERS.LIST_OF;
   LIST_OF_INVERTED_FILTERS : in  NAME_SET_FILTERS.LIST_OF;
   ERROR_INDICATOR         : out  ERROR_NUMBER;
   FOUND_PATH              : out  REFERENCE_PATH);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567485 1T7 ■


```
    procedure INQ_INPUT_QUEUE_OVERFLOW
      (ERROR_INDICATOR: out ERROR_NUMBER;
       WS               : out WS_ID;
       DEVICE           : out EVENT_QUEUE_DEVICE_NUMBER);

    procedure INQ_ERROR_HANDLING_MODE
      (ERROR_INDICATOR: out ERROR_NUMBER;
       MODE             : out ERROR_HANDLING_MODE);

  -- ERROR FUNCTIONS

    procedure EMERGENCY_CLOSE_PHIGS;

    procedure ERROR_HANDLING
      (ERROR_INDICATOR: in ERROR_NUMBER;
       SUBPROGRAM     : in SUBPROGRAM_NAME;
       ERROR_FILE     : in FILE_ID := PHIGS_STRING(DEFAULT_ERROR_FILE));

    procedure ERROR_LOGGING
      (ERROR_INDICATOR: in ERROR_NUMBER;
       SUBPROGRAM     : in SUBPROGRAM_NAME;
       ERROR_FILE     : in FILE_ID := PHIGS_STRING(DEFAULT_ERROR_FILE));

    procedure SET_ERROR_HANDLING_MODE
      (MODE : in ERROR_HANDLING_MODE);

  -- Subprograms for Manipulating Input Data Records

  -- Locator Data Record Operations

    procedure BUILD_LOCATOR_DATA_RECORD
      (PROMPT_ECHO_TYPE : in  LOCATOR_PROMPT_ECHO_TYPE;
       DATA_RECORD      : out LOCATOR_DATA_RECORD);

    procedure BUILD_LOCATOR_DATA_RECORD
      (PROMPT_ECHO_TYPE : in  LOCATOR_PROMPT_ECHO_TYPE;
       CONTENTS         : in  LINE_DATA;
       DATA_RECORD      : out LOCATOR_DATA_RECORD);

    procedure BUILD_LOCATOR_DATA_RECORD
      (PROMPT_ECHO_TYPE : in  LOCATOR_PROMPT_ECHO_TYPE;
       CONTENTS         : in  INTERIOR_DATA;
       DATA_RECORD      : out LOCATOR_DATA_RECORD);

    procedure BUILD_LOCATOR_DATA_RECORD
      (PROMPT_ECHO_TYPE : in  LOCATOR_PROMPT_ECHO_TYPE;
       INTERIOR_CONTENTS: in  INTERIOR_DATA;
       EDGE_CONTENTS    : in  EDGE_DATA;
       DATA_RECORD      : out LOCATOR_DATA_RECORD);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567486 033 ■


```
function LOCATOR_ATTRIBUTES_USED
  (DATA_RECORD : in LOCATOR_DATA_RECORD)
 return POLYLINE_FILL_AREA_CONTROL_FLAG;

function LINE_ATTRIBUTES (DATA_RECORD : in LOCATOR_DATA_RECORD)
 return LINE_DATA;

function INTERIOR_ATTRIBUTES (DATA_RECORD : in  LOCATOR_DATA_RECORD)
 return INTERIOR_DATA;

function EDGE_ATTRIBUTES (DATA_RECORD : in  LOCATOR_DATA_RECORD)
 return EDGE_DATA;

-- Stroke Data Record Operations

procedure BUILD_STROKE_DATA_RECORD
  (PROMPT_ECHO_TYPE : in   STROKE_PROMPT_ECHO_TYPE;
   BUFFER_SIZE      : in   PHIGS_POSITIVE;
   POSITION         : in   PHIGS_POSITIVE;
   INTERVAL         : in   WC.SIZE_2;
   TIME             : in   DURATION;
   DATA_RECORD      : out  STROKE_DATA_RECORD);

procedure BUILD_STROKE_DATA_RECORD
  (PROMPT_ECHO_TYPE : in   STROKE_PROMPT_ECHO_TYPE;
   BUFFER_SIZE      : in   PHIGS_POSITIVE;
   POSITION         : in   PHIGS_POSITIVE;
   INTERVAL         : in   WC.SIZE_2;
   TIME             : in   DURATION;
   CONTENTS         : in   MARKER_DATA;
   DATA_RECORD      : out  STROKE_DATA_RECORD);

procedure BUILD_STROKE_DATA_RECORD
  (PROMPT_ECHO_TYPE : in   STROKE_PROMPT_ECHO_TYPE;
   BUFFER_SIZE      : in   PHIGS_POSITIVE;
   POSITION         : in   PHIGS_POSITIVE;
   INTERVAL         : in   WC.SIZE_2;
   TIME             : in   DURATION;
   CONTENTS         : in   LINE_DATA;
   DATA_RECORD      : out  STROKE_DATA_RECORD);

function BUFFER_SIZE (DATA_RECORD : in STROKE_DATA_RECORD)
 return PHIGS_POSITIVE;

function POSITION (DATA_RECORD : in STROKE_DATA_RECORD)
 return PHIGS_POSITIVE;

function INTERVAL (DATA_RECORD : in STROKE_DATA_RECORD)
 return WC.SIZE_2;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567487 T7T ■


```
function TIME (DATA_RECORD : in STROKE_DATA_RECORD)
 return DURATION;

function MARKER_ATTRIBUTES (DATA_RECORD : in STROKE_DATA_RECORD)
 return MARKER_DATA;

function LINE_ATTRIBUTES (DATA_RECORD : in STROKE_DATA_RECORD)
 return LINE_DATA;

-- Valuator Data Record Operations

procedure BUILD_VALUATOR_DATA_RECORD
  (PROMPT_ECHO_TYPE : in   VALUATOR_PROMPT_ECHO_TYPE;
   LOW_VALUE        : in   VALUATOR_VALUE;
   HIGH_VALUE       : in   VALUATOR_VALUE;
   DATA_RECORD      : out  VALUATOR_DATA_RECORD);

function HIGH_VALUE (DATA_RECORD : in VALUATOR_DATA_RECORD)
 return VALUATOR_VALUE;

function LOW_VALUE (DATA_RECORD : in VALUATOR_DATA_RECORD)
 return VALUATOR_VALUE;

-- Choice Data Record Operations

procedure BUILD_CHOICE_DATA_RECORD
  (PROMPT_ECHO_TYPE : in   CHOICE_PROMPT_ECHO_TYPE;
   DATA_RECORD      : out  CHOICE_DATA_RECORD);

procedure BUILD_CHOICE_DATA_RECORD
  (PROMPT_ECHO_TYPE : in   CHOICE_PROMPT_ECHO_TYPE;
   ARRAY_OF_PROMPTS : in   CHOICE_PROMPTS.LIST_OF;
   DATA_RECORD      : out  CHOICE_DATA_RECORD);

procedure BUILD_CHOICE_DATA_RECORD
  (PROMPT_ECHO_TYPE : in   CHOICE_PROMPT_ECHO_TYPE;
   ARRAY_OF_STRINGS : in   CHOICE_PROMPT_STRING_LIST;
   DATA_RECORD      : out  CHOICE_DATA_RECORD);

procedure BUILD_CHOICE_DATA_RECORD
  (PROMPT_ECHO_TYPE      : in   CHOICE_PROMPT_ECHO_TYPE;
   STRUCTURE_IDENTIFIER : in   STRUCTURE_ID;
   LIST_OF_PICK_IDS      : in   PICK_IDS.LIST_OF;
   DATA_RECORD           : out  CHOICE_DATA_RECORD);

function ARRAY_OF_PROMPTS (DATA_RECORD : in CHOICE_DATA_RECORD)
 return CHOICE_PROMPTS.LIST_OF;

function ARRAY_OF_STRINGS (DATA_RECORD : in CHOICE_DATA_RECORD)
 return CHOICE_PROMPT_STRING_LIST;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
    function STRUCTURE_IDENTIFIER (DATA_RECORD : in CHOICE_DATA_RECORD)
     return STRUCTURE_ID;

    function LIST_OF_PICK_IDS (DATA_RECORD : in CHOICE_DATA_RECORD)
     return PICK_IDS.LIST_OF;

-- Pick Data Record Operations.

    procedure BUILD_PICK_DATA_RECORD
      (PROMPT_ECHO_TYPE : in  PICK_PROMPT_ECHO_TYPE;
       DATA_RECORD      : out PICK_DATA_RECORD);

-- String Data Record Operations

    procedure BUILD_STRING_DATA_RECORD
      (PROMPT_ECHO_TYPE       : in  STRING_PROMPT_ECHO_TYPE;
       INPUT_BUFFER_SIZE      : in  PHIGS_NATURAL;
       INITIAL_CURSOR_POSITION : in  PHIGS_NATURAL;
       DATA_RECORD            : out STRING_DATA_RECORD);

    function INPUT_BUFFER_SIZE (DATA_RECORD : in STRING_DATA_RECORD)
     return PHIGS_NATURAL;

    function INITIAL_CURSOR_POSITION (DATA_RECORD : in STRING_DATA_RECORD)
     return PHIGS_NATURAL;

-- Procedures for deallocating access types

    procedure DEALLOCATE
      (OBJECT : in out CHOICE_DATA_RECORD);

    procedure DEALLOCATE
      (OBJECT : in out ESCAPE_DATA_RECORD);

    procedure DEALLOCATE
      (OBJECT : in out GDP_3_RECORD);

    procedure DEALLOCATE
      (OBJECT : in out GDP_RECORD);

    procedure DEALLOCATE
      (OBJECT : in out GSE_RECORD);

    procedure DEALLOCATE
      (OBJECT : in out LOCATOR_DATA_RECORD);

    procedure DEALLOCATE
      (OBJECT : in out METAFILE_DATA_RECORD);

    procedure DEALLOCATE
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567489 842 ■


```
      (OBJECT : in out NAME_SET)
    renames PHIGS_NAME_SET_FACILITY.DEALLOCATE;

    procedure DEALLOCATE
      (OBJECT : in out PICK_DATA_RECORD);

    procedure DEALLOCATE
      (OBJECT : in out STRING_DATA_RECORD);

    procedure DEALLOCATE
      (OBJECT : in out STROKE_DATA_RECORD);

    procedure DEALLOCATE
      (OBJECT : in out STRUCTURE_ELEMENT_RECORD);

    procedure DEALLOCATE
      (OBJECT : in out VALUATOR_DATA_RECORD);

  -- Procedures for deallocating PHIGS PLUS acess types

    procedure DEALLOCATE (OBJECT: in out COLOUR_VALUE_LIST);

    procedure DEALLOCATE (OBJECT: in out CURVE_COLOURSPLINE);

    procedure DEALLOCATE (OBJECT: in out DATA_MAPPING_DATA_RECORD);

    procedure DEALLOCATE (OBJECT: in out DATASPLINE_LIST);

    procedure DEALLOCATE (OBJECT: in out DATA_VALUE_SET);

    procedure DEALLOCATE (OBJECT: in out LIST_OF_COLOUR_VALUE_LIST);

    procedure DEALLOCATE (OBJECT: in out LIST_OF_EDGE_FLAG_LIST);

    procedure DEALLOCATE (OBJECT: in out LIST_OF_LIST_OF_EDGE_FLAG_LIST);

    procedure DEALLOCATE (OBJECT: in out LIST_OF_LIST_OF_VERTEX_INDEX_LIST);

    procedure DEALLOCATE (OBJECT: in out LIST_OF_TRIMCURVE_LIST);

    procedure DEALLOCATE (OBJECT: in out LIST_OF_VERTEX_INDEX_LIST);

    procedure DEALLOCATE (OBJECT: in out SURFACE_COLOURSPLINE);

    procedure DEALLOCATE (OBJECT: in out TRIMCURVE_LIST);

  -- Subprograms for Manipulating Metafile Data Records

    procedure BUILD_NEW_METAFILE_DATA_RECORD
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
    (TYPE_OF_ITEM : in   METAFILE_ITEM_TYPE;
     ITEM_DATA    : in   PHIGS_STRING;
     ITEM         : out  METAFILE_DATA_RECORD);

  function ITEM_DATA_RECORD_STRING
    (ITEM : in METAFILE_DATA_RECORD) return ACCESS_STRING;

-- Subprograms for Manipulating Name Sets

  procedure BUILD_NAME_SET
    (MEMBERS : in   NAMES.LIST_OF;
     SET     : out  NAME_SET)
  renames PHIGS_NAME_SET_FACILITY.BUILD_NAME_SET;

  -- Constructs a name set from a list of names.

  procedure BUILD_NAME_SET_UNION
    (LEFT_SET  : in   NAME_SET;
     RIGHT_SET : in   NAME_SET;
     UNION     : out  NAME_SET)
  renames PHIGS_NAME_SET_FACILITY.BUILD_NAME_SET_UNION;

  -- Constructs the logical union of two name sets.

  procedure BUILD_NAME_SET_INTERSECTION
    (LEFT_SET      : in   NAME_SET;
     RIGHT_SET     : in   NAME_SET;
     INTERSECTION  : out  NAME_SET)
  renames PHIGS_NAME_SET_FACILITY.BUILD_NAME_SET_INTERSECTION;

  -- Constructs the logical intersection of two name sets.

  procedure BUILD_NAME_SET_DIFFERENCE
    (LEFT_SET    : in   NAME_SET;
     RIGHT_SET   : in   NAME_SET;
     DIFFERENCE  : out  NAME_SET)
  renames PHIGS_NAME_SET_FACILITY.BUILD_NAME_SET_DIFFERENCE;

  -- Constructs the logical difference of two name sets.

  procedure COPY_NAME_SET
    (ORIGINAL : in   NAME_SET;
     COPY     : out  NAME_SET)
  renames PHIGS_NAME_SET_FACILITY.COPY_NAME_SET;

  -- Duplicates the provided name set.

  function IS_EQUAL
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
    (LEFT_SET   : in NAME_SET;
     RIGHT_SET : in NAME_SET)
   return BOOLEAN
  renames PHIGS_NAME_SET_FACILITY.IS_EQUAL;

  -- Compares two name sets.

  procedure EXTRACT_NAMES
    (SET          : in   NAME_SET;
     MEMBERS : out  NAMES.LIST_OF)
  renames PHIGS_NAME_SET_FACILITY.EXTRACT_NAMES;

  -- Extracts the members from the provided name set.

  procedure ADD_NAMES
    (MEMBERS : in      NAMES.LIST_OF;
     SET          : in out  NAME_SET)
  renames PHIGS_NAME_SET_FACILITY.ADD_NAMES;

  -- Adds members to the name set if not already there.

  procedure REMOVE_NAMES
    (MEMBERS : in      NAMES.LIST_OF;
     SET          : in out  NAME_SET)
  renames PHIGS_NAME_SET_FACILITY.REMOVE_NAMES;

  -- Removes members from the name set if they are there.

  function APPLY_FILTER
    (SET      : in NAME_SET;
     FILTER : in NAME_SET_FILTER)
   return NAME_SET_ACCEPTANCE
  renames PHIGS_NAME_SET_FACILITY.APPLY_FILTER;

  -- Applies a filter to a name set and indicates whether the name
  -- set passes through the filter.

  function INQUIRE_MEMBERSHIP
    (MEMBER : in NAME;
     SET         : in NAME_SET)
   return NAME_SET_MEMBERSHIP
  renames PHIGS_NAME_SET_FACILITY.INQUIRE_MEMBERSHIP;

  -- Returns an indication of whether the indicated member
  -- is contained in the provided name set.

-- PRIVATE TYPE DEFINITIONS

private
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
-- The following types define the specifications for the private
-- types. Null records are used soley for the purposes of
-- demonstrating compilability in this annex. An implementation
-- would use some more appropriate declaration.

    type METAFILE_DATA_RECORD (TYPE_OF_ITEM : METAFILE_ITEM_TYPE := 0) is
      record
        null;
      end record;

    type CHOICE_DATA_RECORD
      (PROMPT_ECHO_TYPE : CHOICE_PROMPT_ECHO_TYPE := 1) is
        record
          null;
        end record;

    type LOCATOR_DATA_RECORD
      (PROMPT_ECHO_TYPE : LOCATOR_PROMPT_ECHO_TYPE := 1) is
        record
          null;
        end record;

    type PICK_DATA_RECORD
      (PROMPT_ECHO_TYPE : PICK_PROMPT_ECHO_TYPE := 1) is
        record
          null;
        end record;

    type STRING_DATA_RECORD
      (PROMPT_ECHO_TYPE : STRING_PROMPT_ECHO_TYPE := 1) is
        record
          null;
        end record;

    type STROKE_DATA_RECORD
      (PROMPT_ECHO_TYPE : STROKE_PROMPT_ECHO_TYPE := 1) is
        record
          null;
        end record;

    type VALUATOR_DATA_RECORD
      (PROMPT_ECHO_TYPE : VALUATOR_PROMPT_ECHO_TYPE := 1) is
        record
          null;
        end record;

end PHIGS;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567493 273 ■


```
-- Minimal PHIGS_GDP Package

with PHIGS_TYPES;
use PHIGS_TYPES;
package PHIGS_GDP is


    type GDP_INTEGER_ARRAY  is array (SMALL_NATURAL range <>) of
      PHIGS_INTEGER;

    type GDP_FLOAT_ARRAY   is array (SMALL_NATURAL range <>) of
      GENERAL_FLOAT_PARAMETER;

    type GDP_STRING_ARRAY  is array (SMALL_NATURAL range <>) of
      PHIGS_STRING(1..80);

    type GDP_DATA_RECORD ( NUM_OF_INTEGERS :  SMALL_NATURAL := 0;
                           NUM_OF_REALS    : SMALL_NATURAL := 0;
                           NUM_OF_STRINGS  : SMALL_NATURAL := 0) is
      record
        INTEGER_ARRAY : GDP_INTEGER_ARRAY (1..NUM_OF_INTEGERS);
        REAL_ARRAY    : GDP_FLOAT_ARRAY (1..NUM_OF_REALS);
        GDP_STRINGS   : GDP_STRING_ARRAY (1..NUM_OF_STRINGS);
      end record;

    procedure GENERALIZED_GDP
      (GDP          : in GDP_3_ID;
       POINTS       : in MC.POINT_LIST_3;
       GDP_DATA     : in GDP_DATA_RECORD);

    procedure GENERALIZED_GDP
      (GDP          : in GDP_ID;
       POINTS       : in MC.POINT_LIST_2;
       GDP_DATA     : in GDP_DATA_RECORD);

end PHIGS_GDP;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567494 10T ■


```
-- Minimal PHIGS_GSE Package


with PHIGS_TYPES;
use PHIGS_TYPES;
package PHIGS_GSE is


    type GSE_INTEGER_ARRAY is array (SMALL_NATURAL range <>) of
      PHIGS_INTEGER;

    type GSE_FLOAT_ARRAY   is array (SMALL_NATURAL range <>) of
      GENERAL_FLOAT_PARAMETER;

    type GSE_STRING_ARRAY  is array (SMALL_NATURAL range <>) of
      PHIGS_STRING(1..80);

    type GSE_DATA_RECORD ( NUM_OF_INTEGERS : SMALL_NATURAL := 0;
                           NUM_OF_REALS    : SMALL_NATURAL := 0;
                           NUM_OF_STRINGS  : SMALL_NATURAL := 0) is
      record
        INTEGER_ARRAY : GSE_INTEGER_ARRAY (1..NUM_OF_INTEGERS);
        REAL_ARRAY    : GSE_FLOAT_ARRAY  (1..NUM_OF_REALS);
        GSE_STRINGS   : GSE_STRING_ARRAY (1..NUM_OF_STRINGS);
      end record;

    procedure GENERALIZED_GSE
      (GSE       : in GSE_ID;
       GSE_DATA  : in GSE_DATA_RECORD);

end PHIGS_GSE;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
-- Minimal PHIGS_ESCAPE Package

with PHIGS_TYPES;
use PHIGS_TYPES;
package PHIGS_ESCAPE is


    type ESC_INTEGER_ARRAY  is array (SMALL_NATURAL range <>) of
      PHIGS_INTEGER;

    type ESC_FLOAT_ARRAY   is array (SMALL_NATURAL range <>) of
      GENERAL_FLOAT_PARAMETER;

    type ESC_STRING_ARRAY  is array (SMALL_NATURAL range <>) of
      PHIGS_STRING(1..80);

    type ESC_DATA_RECORD ( NUM_OF_INTEGERS : SMALL_NATURAL := 0;
                           NUM_OF_REALS    : SMALL_NATURAL := 0;
                           NUM_OF_STRINGS  : SMALL_NATURAL := 0) is
      record
        INTEGER_ARRAY : ESC_INTEGER_ARRAY (1..NUM_OF_INTEGERS);
        REAL_ARRAY    : ESC_FLOAT_ARRAY  (1..NUM_OF_REALS);
        ESC_STRINGS   : ESC_STRING_ARRAY (1..NUM_OF_STRINGS);
      end record;

    procedure GENERALIZED_ESC
      (ESC_NAME      : in  ESCAPE_ID;
       ESC_DATA_IN   : in  ESC_DATA_RECORD;
       ESC_DATA_OUT  : out ESC_DATA_RECORD);

end PHIGS_ESCAPE;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567496 T82 ■

Annex B

(informative)

Cross Reference Listing of Implementation Defined Items

The following should be appended:

MAX_COLOUR_MAPPING_INDICES_SUPPORTED	4.2.5
MAX_COLOUR_MAPPING_METHODS_SUPPORTED	4.2.5
MAX_COLOUR_VALUES_SUPPORTED	4.2.5
MAX_CURVE_APPROX_CRITERIA_TYPES_SUPPORTED	4.2.5
MAX_DATA_MAPPING_INDICES_SUPPORTED	4.2.5
MAX_DATA_MAPPING_METHODS_SUPPORTED	4.2.5
MAX_DATA_VALUES_PER_FACET	4.2.5
MAX_DATA_VALUES_PER_VERTEX	4.2.5
MAX_DEPTH_CUE_INDICES_SUPPORTED	4.2.5
MAX_FACETS_SUPPORTED	4.2.5
MAX_INTERIOR_SHADING_METHODS_SUPPORTED	4.2.5
MAX_LIGHT_SOURCE_INDICES_SUPPORTED	4.2.5
MAX_LIGHT_SOURCE_TYPES_SUPPORTED	4.2.5
MAX_PARAMETRIC_SURFACE_CHARACTERISTICS_SUPPORTED	4.2.5
MAX_PARAMETRIC_SURFACE_INDICES_SUPPORTED	4.2.5
MAX_POLYLINE_SHADING_METHODS_SUPPORTED	4.2.5
MAX_REFLECTANCE_INDICES_SUPPORTED	4.2.5
MAX_REFLECTANCE_MODELS_SUPPORTED	4.2.5
MAX_REFLECTANCE_PROPERTIES_TYPES_SUPPORTED	4.2.5
MAX_SURFACE_APPROX_CRITERIA_TYPES_SUPPORTED	4.2.5
MAX_VERTICES_SUPPORTED	4.2.5
PHIGS_PI	4.2.5

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567497 919 ■

Annex C

(informative)

Example Programs

C.1 Example Program 1: STAR

No changes.

C.2 Example Program 2: IRON

No changes.

C.3 Example Program 3: DYNASTAR

No changes.

C.4 Example Program 4: TRANSFORM_POLYLINE

No changes.

C.5 Example Program 5: SHOW_LINETYPES

No changes.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

page 273: The following example programs should be inserted following the SHOW_LINETYPES program.

C.6 Example Program 6: DODECAHEDRON

```
-- PROGRAM DODECAHEDRON

-- A modified version for the PHIGS PLUS/Ada binding

with PHIGS, PHIGS_TYPES;
use PHIGS, PHIGS_TYPES;

-- Mathematics library with functions SIN (sine) and COS (cosine) assumed.

with MATH;
use MATH;
procedure DODECAHEDRON is

   -- This program draws a dodecahedron using just a single PHIGS PLUS primitive.

   -- Define workstation variables and error logging file.

   MY_WS_ID              : constant WS_ID := 1;
   SOME_CONNECTION       : constant CONNECTION_ID := "UNIT_1";
   SOME_OUTPUT_TYPE      : constant WS_TYPE := 1;
   ERROR_FILE            : constant FILE_ID := "MY_ERROR_FILE";
   EI                    : ERROR_NUMBER;

   -- Define type of HLHSR to be z-buffered
   --  (implementation-dependent value allowed but assumed to be 1
   --   for this example program)

   Z_BUFFER              : constant HLHSR_MODE := 1;

   -- Define variables for view table entry 1.

   VIEW_ORIENTATION_MATRIX   : TRANSFORMATION_MATRIX_3;
   VIEW_MAPPING_MATRIX       : TRANSFORMATION_MATRIX_3;
   VIEW_FOR_DODECAHEDRON     : VIEW_INDEX := 1;

   -- Define various colour representations

   BACKGROUND_BLUE   : COLOUR_REPRESENTATION := (RGB, 0.10, 0.10, 0.5);
   WHITE             : COLOUR_REPRESENTATION := (RGB, 1.0, 1.0, 1.0);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
RED                 : COLOUR_REPRESENTATION := (RGB, 1.0, 0.0, 0.0);
GREEN               : COLOUR_REPRESENTATION := (RGB, 0.0, 1.0, 0.0);
BLUE                : COLOUR_REPRESENTATION := (RGB, 0.0, 0.0, 1.0);
YELLOW              : COLOUR_REPRESENTATION := (RGB, 1.0, 1.0, 0.0);
CYAN                : COLOUR_REPRESENTATION := (RGB, 0.0, 1.0, 1.0);
MAGENTA             : COLOUR_REPRESENTATION := (RGB, 1.0, 0.0, 1.0);
DARK_RED            : COLOUR_REPRESENTATION := (RGB, 0.7, 0.0, 0.0);
DARK_GREEN          : COLOUR_REPRESENTATION := (RGB, 0.0, 0.7, 0.0);
DARK_BLUE           : COLOUR_REPRESENTATION := (RGB, 0.0, 0.0, 0.7);
DARK_YELLOW         : COLOUR_REPRESENTATION := (RGB, 0.7, 0.7, 0.0);
DARK_CYAN           : COLOUR_REPRESENTATION := (RGB, 0.0, 0.7, 0.7);
DARK_MAGENTA        : COLOUR_REPRESENTATION := (RGB, 0.7, 0.0, 0.7);

-- Define structure Identifiers

PENTAGON            : constant STRUCTURE_ID := 1;
BOTTOMSIDES         : constant STRUCTURE_ID := 2;
TOPSIDES            : constant STRUCTURE_ID := 3;
PICTURE             : constant STRUCTURE_ID := 4;

-- Define transformation variables.

TRANSFORMATION_1    : TRANSFORMATION_MATRIX_3;
TRANSFORMATION_2    : TRANSFORMATION_MATRIX_3;
TRANSFORMATION_3    : TRANSFORMATION_MATRIX_3;
RESULT_MATRIX       : TRANSFORMATION_MATRIX_3;
TRANSFORMATION_21   : TRANSFORMATION_MATRIX_3;
TRANSFORMATION_22   : TRANSFORMATION_MATRIX_3;
TRANSFORMATION_23   : TRANSFORMATION_MATRIX_3;
TRANSFORMATION_11   : TRANSFORMATION_MATRIX_3;
TRANSFORMATION_12   : TRANSFORMATION_MATRIX_3;
TRANSFORMATION_13   : TRANSFORMATION_MATRIX_3;
TRANSFORMATION_14   : TRANSFORMATION_MATRIX_3;
TRANSFORMATION_16   : TRANSFORMATION_MATRIX_3;
TRANSFORMATION_17   : TRANSFORMATION_MATRIX_3;

-- Define light activation and deactivation lists

LIGHT_LIST          : LIGHT_SOURCE_INDICES.LIST_VALUES(1..2) := (1,2);
ACTIVE_LIGHTS       : LIGHT_SOURCE_INDICES.LIST_OF;
INACTIVE_LIGHTS     : LIGHT_SOURCE_INDICES.LIST_OF;

-- Define types and variables for building the pentagons with circle cutouts.

subtype PENTAGON_RANGE is POSITIVE range 1..5;
subtype CIRCLE_RANGE   is POSITIVE range 1..32;
PENTAGON_POINTS     : MC.POINT_LIST_3(PENTAGON_RANGE);
CIRCLE_POINTS       : MC.POINT_LIST_3(CIRCLE_RANGE);
DODECAHEDRON_SIDE   : MC.LIST_OF_POINT_LIST_3(1..2);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
   PENTAGON_ANGLE          : ANGLE;
   COSINE_5_SIDES          : MC_TYPE := COS(2.0*PI/5.0);
   SINE_5_SIDES            : MC_TYPE := SIN(2.0*PI/5.0);
   COSINE_32_SIDES         : MC_TYPE := COS(2.0*PI/32.0);
   SINE_32_SIDES           : MC_TYPE := SIN(2.0*PI/32.0);
   RADIUS                  : MC_TYPE := 0.9;

begin

   -- Initialize first point in pentagon

   PENTAGON_POINTS(1) := (0.0, 0.6089172*RADIUS, -RADIUS*0.7932338);

   -- Set remainder of points in pentagon

   for I in 2..PENTAGON_RANGE'last loop
      PENTAGON_POINTS(I) := (    PENTAGON_POINTS(I-1).X*COSINE_5_SIDES
                                   - PENTAGON-POINTS(I-1).Y*SINE_5_SIDES,
                                 PENTAGON_POINTS(I-1).X*SINE_5_SIDES
                                   + PENTAGON_POINTS(I-1).Y*COSINE_5_SIDES,
                                 PENTAGON_POINTS(1).Z );
   end loop;

   -- Initialize first point in circle

   CIRCLE_POINTS(1) := (0.0, 0.2*RADIUS, -RADIUS*0.7932338);

   -- Set remainder of points in circle

   for I in 2..CIRCLE_RANGE'last loop
      CIRCLE_POINT(I) := (  CIRCLE_POINTS(I-1).X*COSINE_32_SIDES
                              - CIRCLE_POINTS(I-1).Y*SINE_32_SIDES,
                            CIRCLE_POINTS(I-1).X*SINE_32_SIDES,
                              + CIRCLE_POINTS(I-1).Y*COSINE_32_SIDES,
                            CIRCLE_POINTS(1).Z );
   end loop;

   -- Define a single side of the dodecahedron.

   DODECAHEDRON_SIDE(1) := PENTAGON_POINTS;
   DODECAHEDRON_SIDE(2) := CIRCLE_POINTS;

   -- Open PHIGS and a workstation

   OPEN_PHIGS(ERROR_FILE);
   OPEN_WS (MY_WS_ID, SOME_CONNECTION, SOME_OUTPUT_TYPE);

   -- Establish type of HLHSR

   SET_HLHSR_MODE(MY_WS_ID, Z_BUFFER);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567501 17T ■


```
-- Build PHIGS colour table

SET_COLOUR_REPRESENTATION(MY_WS_ID, 0, BACKGROUND_BLUE);
SET_COLOUR_REPRESENTATION(MY_WS_ID, 1, WHITE);
SET_COLOUR_REPRESENTATION(MY_WS_ID, 2, RED);
SET_COLOUR_REPRESENTATION(MY_WS_ID, 3, GREEN);
SET_COLOUR_REPRESENTATION(MY_WS_ID, 4, BLUE);
SET_COLOUR_REPRESENTATION(MY_WS_ID, 5, YELLOW);
SET_COLOUR_REPRESENTATION(MY_WS_ID, 6, CYAN);
SET_COLOUR_REPRESENTATION(MY_WS_ID, 7, MAGENTA);
SET_COLOUR_REPRESENTATION(MY_WS_ID, 8, DARK_RED);
SET_COLOUR_REPRESENTATION(MY_WS_ID, 9, DARK_GREEN);
SET_COLOUR_REPRESENTATION(MY_WS_ID, 10, DARK_BLUE);
SET_COLOUR_REPRESENTATION(MY_WS_ID, 11, DARK_YELLOW);
SET_COLOUR_REPRESENTATION(MY_WS_ID, 12, DARK_CYAN);
SET_COLOUR_REPRESENTATION(MY_WS_ID, 13, DARK_MAGENTA);

-- Establish the view.

-- Calculate the view orientation matrix.

EVALUATE_VIEW_ORIENTATION_MATRIX
    (REFERENCE_POINT          => (0.0, 0.0, 0.0),
     NORMAL_VECTOR            => (0.0, 0.0, 1.0),
     UP_VECTOR                => (0.0, 1.0, 0.0),
     ERROR_INDICATOR          => EI,
     ORIENTATION_MATRIX       => VIEW_ORIENTATION_MATRIX);

-- Calculate the view mapping matrix.

EVALUATE_VIEW_MAPPING_MATRIX
    (WINDOW_LIMITS        =>  (XMIN => -1.0, XMAX => 1.0,
                               YMIN => -1.0, YMAX => 1.0),
     VIEWPORT_LIMITS      =>  (XMIN => 0.0, XMAX => 1.0,
                               YMIN => 0.0, YMAX => 1.0,
                               ZMIN => 0.0, ZMAX => 1.0),
     TYPE_OF_PROJECTION   =>  PARALLEL,
     REFERENCE_POINT      =>  (0.0, 0.0, 10.0),
     VIEW_DISTANCE        =>  0.0,
     BACK_DISTANCE        =>  -1.0,
     FRONT_DISTANCE       =>  1.0,
     ERROR_INDICATOR      =>  EI,
     MAPPING_MATRIX       =>  VIEW_MAPPING_MATRIX);

-- Store the view in the view table as view 1.

SET_VIEW_REPRESENTATION
    (MY_WS_ID,
     VIEW_IND             =>  VIEW_FOR_DODECAHEDRON,
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567502 006 ■


```
        ORIENTATION_MATRIX  =>   VIEW_ORIENTATION_MATRIX,
        MAPPING_MATRIX      =>   VIEW_MAPPING_MATRIX,
        CLIPPING_LIMITS     =>   (XMIN => 0.0, XMAX => 1.0,
                                  YMIN => 0.0, YMAX => 1.0,
                                  ZMIN => 0.0, ZMAX => 1.0),
        XY_CLIPPING         =>   CLIP,
        BACK_CLIPPING       =>   CLIP,
        FRONT_CLIPPING      =>   CLIP);

-- Define structure for a single side

OPEN_STRUCTURE (PENTAGON);
SET_INTERIOR_STYLE (SOLID);
FILL_AREA_SET (DODECAHEDRON_SIDE);
CLOSE_STRUCTURE;

-- Define structure for topsides of dodecahedron

OPEN_STRUCTURE (TOPSIDES);
TOPSIDE_PENTAGONS:
  declare
     HUE_IND : COLOUR_INDEX := 2;
  begin

  -- First define sample facets

     SET_INTERIOR_COLOUR_INDEX (HUE_IND);
     EXECUTE_STRUCTURE (PENTAGON);
     TRANSLATE ((0.0, 0.0, 0.0), EI, TRANSFORMATION_11);
     ROTATE_X (-2.03444393579, EI, TRANSFORMATION_12);
     COMPOSE_MATRIX ( TRANSFORMATION_12,
                      TRANSFORMATION_11,
                      EI,
                      TRANSFORMATION_13 );
     ROTATE_Z (PHIGS_PI, EI, TRANSFORMATION_14);
     COMPOSE_MATRIX ( TRANSFORMATION_14,
                      TRANSFORMATION_13,
                      EI,
                      RESULT_MATRIX);
     SET_LOCAL_TRANSFORMATION (RESULT_MATRIX, REPLACE);
     HUE_IND := HUE_IND+1;
     SET_INTERIOR_COLOUR_INDEX (HUE_IND);
     EXECUTE_STRUCTURE (PENTAGON);

     -- Define remaining top side facets

     for I in 2..PENTAGON_RANGE'last loop
        PENTAGON_ANGLE := ANGLE(-(I-1))*1.256637;
        ROTATE_Z (PENTAGON_ANGLE, EI, TRANSFORMATION_16);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567503 T42 ■


```
        COMPOSE_MATRIX ( TRANSFORMATION_16,
                         RESULT_MATRIX,
                         EI,
                         TRANSFORMATION_17);
        SET_LOCAL_TRANSFORMATION (TRANSFORMATION_17, REPLACE);
        SET_INTERIOR_COLOUR_INDEX (HUE_IND);
        EXECUTE_STRUCTURE (PENTAGON);
      end loop;
  end TOPSIDE_PENTAGONS;
CLOSE_STRUCTURE;

-- Define structure for topsides of dodecahedron

OPEN_STRUCTURE (BOTTOMSIDES);
BOTTOMSIDE_PENTAGONS:
  declare
    HUE_IND : COLOUR_INDEX := 8;
  begin

    -- First define sample facets

    SET_INTERIOR_COLOUR_INDEX (HUE_IND);
    EXECUTE_STRUCTURE (PENTAGON);
    TRANSLATE ((0.0, 0.0, 0.0), EI, TRANSFORMATION_11);
    ROTATE_X (-2.03444393579, EI, TRANSFORMATION_12);
    COMPOSE_MATRIX ( TRANSFORMATION_12,
                     TRANSFORMATION_11,
                     EI,
                     TRANSFORMATION_13);
    ROTATE_Z (PHIGS_PI, EI, TRANSFORMATION_14);
    COMPOSE_MATRIX ( TRANSFORMATION_14,
                     TRANSFORMATION_13,
                     EI,
                     RESULT_MATRIX);
    SET_LOCAL_TRANSFORMATION (RESULT_MATRIX, REPLACE);
    HUE_IND := HUE_IND+1;
    SET_INTERIOR_COLOUR_INDEX (HUE_IND);
    EXECUTE_STRUCTURE (PENTAGON);

    -- Define remaining bottomside facets

    for I in 1..PENTAGON_RANGE'last loop
      PENTAGON_ANGLE := ANGLE(-(I-1))*1.256637;
      ROTATE_Z (PENTAGON_ANGLE, EI, TRANSFORMATION_16);
      COMPOSE_MATRIX ( TRANSFORMATION_16,
                       RESULT_MATRIX,
                       EI,
                       TRANSFORMATION_17);
      SET_LOCAL_TRANSFORMATION (TRANSFORMATION_17, REPLACE);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
        SET_INTERIOR_COLOUR_INDEX (HUE_IND);
        EXECUTE_STRUCTURE (PENTAGON);
      end loop;
    end BOTTOMSIDE_PENTAGONS;
  CLOSE_STRUCTURE;

  -- Define the light sources

  SET_LIGHT_SOURCE_REPRESENTATION
      (MY_WS_ID,
       LIGHT_SOURCE_IND    => 1,
       LIGHT_SOURCE        => (  TYPE_OF_LIGHT => AMBIENT_LIGHT,
                                 AMBIENT_COLOUR => (RGB, (0.5, 0.5, 0.5))));
  SET_LIGHT_SOURCE_REPRESENTATION
      (MY_WS_ID,
       LIGHT_SOURCE_IND    => 2,
       LIGHT_SOURCE        => (  TYPE_OF_LIGHT => DIRECTIONAL_LIGHT,
                                 DIRECTIONAL_COLOUR => (RGB, (0.5, 0.5, 0.5)),
                                 DIRECTIONAL_DIRECTION => (0.75, 0.75, -1.0)));

  -- Build the light source activation and deactivation lists

  ACTIVE_LIGHTS := LIGHT_SOURCE_INDICES.LIST(LIGHT_LIST);
  INACTIVE_LIGHTS := LIGHT_SOURCE_INDICES.NULL_LIST;

  -- Define the complete dodecahedron picture

  OPEN_STRUCTURE (PICTURE);
  SET_VIEW_INDEX (VIEW_FOR_DODECAHEDRON);
  SET_REFLECTANCE_MODEL (AMBIENT_DIFFUSE_REFLECTANCE);
  SET_INTERIOR_SHADING_METHOD (INTERIOR_COLOUR_SHADING);
  SET_LIGHT_SOURCE_STATE (ACTIVE_LIGHTS, INACTIVE_LIGHTS);
  EXECUTE_STRUCTURE (TOPSIDES);
  ROTATE_X (PHIGS_PI, EI, TRANSFORMATION_21);
  ROTATE_Z (3.76991118, EI, TRANSFORMATION_22);
  COMPOSE_MATRIX ( TRANSFORMATION_22,
                   TRANSFORMATION_21,
                   EI,
                   TRANSFORMATION_23);
  SET_LOCAL_TRANSFORMATION (TRANSFORMATION_23, REPLACE);
  EXECUTE_STRUCTURE (BOTTOMSIDES);
  CLOSE_STRUCTURE;

  -- Post to workstation

  POST_STRUCTURE (MY_WS_ID, PICTURE, 1.0);

  -- Insure up-to-date picture

  UPDATE_WS (MY_WS_ID, POSTPONE);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567505 815 ■


```
    -- End PHIGS session

    CLOSE_WS (MY_WS_ID);
    CLOSE_PHIGS;

end DODECAHEDRON;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567506 751 ■

C.7 Example Program 7: TRIMMED_SURFACE

```
-- PROGRAM TRIMMED_SURFACE

-- converted from:
-- /* surface_trimmed.c 1.3 */
-- /* Copyright 1992 O'Reilly and Associates, Inc.  Permission to use, copy,
--  * and modify this program is hereby granted, as long as this copyright
--  * notice appears in each copy of the program source code.
-- */

with PHIGS, PHIGS_TYPES;
use  PHIGS, PHIGS_TYPES;
procedure TRIMMED_SURFACE is

-- This program draws a trimmed non-uniform B-Spline surface.

   -- Define workstation variables and error logging file.

   MY_WS_ID              : constant WS_ID := 1;
   SOME_CONNECTION       : constant CONNECTION_ID := "UNIT_1";
   SOME_OUTPUT_TYPE      : constant WS_TYPE := 1;
   ERROR_FILE            : constant FILE_ID := "MY_ERROR_FILE";
   EI                    : ERROR_NUMBER := 0;

   -- Define structure identifier for trimmed surface model.

   MODEL                 : constant STRUCTURE_ID := 1;

   -- Specify the surface knots and control points.

   SURFACE_U_ORDER  : SPLINE_ORDER := 3;
   SURFACE_V_ORDER  : SPLINE_ORDER := 4;

   SURFACE_GEOMETRY : MC.SURFACE_GEOMETRY_SPLINE
         ( RATIONALITY       => NON_RATIONAL,
           KNOT_COUNT_U      => SMALL_NATURAL(2*SURFACE_U_ORDER),
           KNOT_COUNT_V      => SMALL_NATURAL(2*SURFACE_V_ORDER),
           LENGTH_U          => VERTEX_SET_DIMENSION(SPLINE_ORDER),
           LENGTH_V          => VERTEX_SET_DIMENSION(SPLINE_ORDER) )
     := ( U_ORDER            => SURFACE_U_ORDER,
          V_ORDER            => SURFACE_V_ORDER,
          U_KNOTS            => (0.0, 0.0, 0.0, 1.0, 1.0, 1.0),
          V_KNOTS            => (0.0, 0.0, 0.0, 0.0, 1.0, 1.0, 1.0, 1.0),
          CONTROL_POINTS     => (   (0.0, 0.5, 0.0), (0.0, 0.6, 0.5), (0.0, 0.5, 1.0),
                                    (0.3, 0.5, 0.0), (0.3, 1.0, 0.5), (0.3, 0.5, 1.0),
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567507 698 ■


```
                                (0.6, 0.4, 0.0), (0.6, 0.5, 0.5), (0.6, 0.4, 1.0),
                                (0.9, 0.5, 0.0), (0.9, 0.3, 0.5), (0.9, 0.2, 1.0)) );

-- Specify the trimming Curves, from inner-most to outer-most.

TRIM_CURVE_0    : TRIMCURVE
        (CRITERIA_TYPE      => CONSTANT_SUBDIVISION_CURVE,
         RATIONALITY        => NON_RATIONAL,
         KNOT_COUNT         => 12,
         LENGTH             => 9 )
    :=  (CRITERIA_FOR_CURVE_APPROX=> (   CRITERIA_TYPE, 10),
         FLAG                       => ON,
         ORDER                      => 3,
         KNOTS                      => (   0.0, 0.0, 0.0, 1.0, 1.0, 2.0, 2.0, 3.0, 3.0,
                                           4.0, 4.0, 4.0 ),
         CONTROL_POINTS             => (   RATIONALITY,
                                           ((0.5,0.32),
                                           (0.68,0.32),
                                           (0.68,0.5),
                                           (0.68,0.68),
                                           (0.5,0.68),
                                           (0.32,0.68),
                                           (0.32,0.5),
                                           (0.32,0.32),
                                           (0.5,0.32)) ),
         LIMITS                     => (MIN => 0.0, MAX => 4.0)  );

TRIM_CURVE_1    : TRIMCURVE
        (CRITERIA_TYPE      => CONSTANT_SUBDIVISION_CURVE,
         RATIONALITY        => NON_RATIONAL,
         KNOT_COUNT         => 12,
         LENGTH             => 9 )
    := ( CRITERIA_FOR_CURVE_APPROX  => (CRITERIA_TYPE, 10),
         FLAG                       => ON,
         ORDER                      => 3,
         KNOTS                      => (   0.0, 0.0, 0.0, 1.0, 1.0, 2.0, 2.0, 3.0, 3.0,
                                           4.0, 4.0, 4.0 ),
         CONTROL_POINTS             => (   RATIONALITY,
                                           ((0.5,0.25),
                                           (0.75,0.25),
                                           (0.75,0.5),
                                           (0.75,0.75),
                                           (0.5,0.75),
                                           (0.25,0.75),
                                           (0.25,0.5),
                                           (0.25,0.25),
                                           (0.5,0.25)) ),
         LIMITS                     => (MIN => 0.0, MAX => 4.0)  );
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
TRIM_CURVE_2   : TRIMCURVE
        (CRITERIA_TYPE      => CONSTANT_SUBDIVISION_CURVE,
         RATIONALITY        => NON_RATIONAL,
         KNOT_COUNT         => 7,
         LENGTH             => 5)
     := (CRITERIA_FOR_CURVE_APPROX => (CRITERIA_TYPE, 10),
         FLAG                      => ON,
         ORDER                     => 2,
         KNOTS                     => (0.0, 0.0, 1.0, 2.0, 3.0, 4.0, 4.0),
         CONTROL_POINTS            => (   RATIONALITY,
                                          ((0.0, 0.0),
                                          (1.0, 0.0),
                                          (1.0, 1.0),
                                          (0.0, 1.0),
                                          (0.0, 0.0)) ),
         LIMITS                    => (MIN => 0.0, MAX => 4.0)  );

TRIMCURVES      : TRIMCURVE_LIST(1..3)
                    := (TRIM_CURVE_0, TRIM_CURVE_1, TRIM_CURVE_2);
ALL_TRIM_CURVES      : LIST_OF_TRIMCURVE_LIST(1..1) := TRIMCURVES;

-- Define procedure to establish the view.

procedure SET_VIEW is

   -- Define local variables.

   VIEW_ORIENTATION_MATRIX  : TRANSFORMATION_MATRIX_3;
   VIEW_MAPPING_MATRIX      : TRANSFORMATION_MATRIX_3;

begin

   -- Calculate the view orientation matrix.

   EVALUATE_VIEW_ORIENTATION_MATRIX
       (REFERENCE_POINT      => (0.5, 0.5, 0.5),
        NORMAL_VECTOR        => (0.5643, 0.4775, 0.6735),
        UP_VECTOR            => (0.0, 1.0, 0.0),
        ERROR_INDICATOR      => EI,
        ORIENTATION_MATRIX   => VIEW_ORIENTATION_MATRIX );

   -- Calculate the view mapping matrix.

   EVALUATE_VIEW_MAPPING_MATRIX
       (WINDOW_LIMITS        => (  XMIN => -0.78, XMAX => 0.7,
                                   YMIN => -0.78, YMAX => 0.7),
        VIEW_PORT_LIMITS     => (  XMIN => 0.0, XMAX => 1.0,
                                   YMIN => 0.0, YMAX => 1.0,
                                   ZMIN => 0.0, ZMAX => 1.0),
        TYPE_OF_PROJECTION   => PARALLEL,
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567509 460 ■


```
            REFERENCE_POINT      => (0.0, 0.0, 10.0),
            VIEW_DISTANCE        => 0.0,
            BACK_DISTANCE        => -1.0,
            FRONT_DISTANCE       => 1.0,
            ERROR_INDICATOR      => EI,
            MAPPING_MATRIX       => VIEW_MAPPING_MATRIX );

        -- Store the view in the view table as view 1.

        SET_VIEW_REPRESENTATION
            (MY_WS_ID,
             VIEW_IND            => 1,
             ORIENTATION_MATRIX  => VIEW_ORIENTATION_MATRIX,
             MAPPING_MATRIX      => VIEW_MAPPING_MATRIX,
             CLIPPING_LIMITS     => (  XMIN => 0.0, XMAX => 1.0,
                                       YMIN => 0.0, YMAX => 1.0,
                                       ZMIN => 0.0, ZMAX => 1.0),
             XY_CLIPPING         => CLIP,
             BACK_CLIPPING       => CLIP,
             FRONT_CLIPPING      => CLIP);

    end SET_VIEW;

begin

    -- Open PHIGS and a workstation

    OPEN_PHIGS(ERROR_FILE);
    OPEN_WS (MY_WS_ID, SOME_CONNECTION, SOME_OUTPUT_TYPE);

    -- Create the structure

    OPEN_STRUCTURE (MODEL);
    SET_VIEW_INDEX (1);
    SET_INTERIOR_STYLE (EMPTY);

    -- Set the parametric surface characteristics

    SET_PARAMETRIC_SURFACE_CHARACTERISTICS
        (CHARACTERISTIC      => ISOPARAMETRIC_CURVES,
         CURVE_PLACEMENT     => UNIFORM,
         U_COUNT             => 20,
         V_COUNT             => 20 );

    SET_SURFACE_APPROXIMATION_CRITERIA
        (CRITERIA     => CONSTANT_SUBDIVISION_SURFACE_BETWEEN_KNOTS,
         CONSTANT_SUBDIVISION_UCOUNT => 10.0,
         CONSTANT_SUBDIVISION_VCOUNT => 10.0 );
```


--`,,`-`-`,,`,,`,`,,`---
Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
    -- Create the surface

    SET_EDGE_FLAG (ON);
    NON_UNIFORM_B_SPLINE_SURFACE ( SURFACE_SPLINE  => SURFACE_GEOMETRY,
                                   TRIM_CURVES     => ALL_TRIM_CURVES );

    -- End of structure definition

    CLOSE_STRUCTURE;

    -- Set the view, and post the top-level structure

    SET_VIEW;
    POST_STRUCTURE (MY_WS_ID, MODEL, 1.0);
    UPDATE_WS (MY_WS_ID, PERFORM);

    -- End the session

    CLOSE_WS (MY_WS_ID);
    CLOSE_PHIGS;

end TRIMMED_SURFACE;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567511 019 ■

Annex D

(informative)

PHIGS Multi-Tasking

No changes.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Annex E

(informative)

Index

The following index applies only to this amendment.

--`,,,`-`-`,,`,,`,`,,`---

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567513 991 ■

A

ABANDON, **174**
Abbreviations
  APPROX, 14
  INQ, 5
  WS, 6
ACCEPTED, 117
ACCESS_ARRAY_OF_EDGE_FLAG_PAIR, 27, **30**, **194**, **228**
ACCESS_ARRAY_OF_MAGNITUDE_LISTS, **95**, **139**
ACCESS_COLOUR_MAPPING_DATA_RECORD, **30**, 86, 92, **172**, 268, 280
ACCESS_COLOUR_MATRIX, 21, 222
ACCESS_COLOUR_VALUE_ARRAY, 25, **30**, 43, 85, 91, **106**, **123**, **158**, 161, 227, 267, 279
ACCESS_COLOUR_VALUE_LIST, **30**, 36, 43, **106**, **123**, **158**, 160, 172
ACCESS_CONTROL_POINT_ARRAY, **107**, **124**
ACCESS_CONTROL_POINT_ARRAY_3, **94**, **137**
ACCESS_CONTROL_POINT_LIST, **107**, **124**
ACCESS_CONTROL_POINT_LIST_2, **94**, **137**
ACCESS_CONTROL_POINT_LIST_3, **94**, **136**
ACCESS_CURVE_COLOURSPLINE, 27, **30**, **163**, 228
ACCESS_CURVE_GEOMETRYSPLINE_2, **96**, **140**
ACCESS_CURVE_GEOMETRYSPLINE_3, 27, **95**, **139**, 228
ACCESS_DATA_VALUE_SET, **31**, 43, **126**, 160, 161
ACCESS_DATASPLINE_LIST, 27, **31**, **164**, 229
ACCESS_EDGE_FLAG_LIST, 26, **31**, 54, **193**, 228
ACCESS_FACET_DATA_ARRAY_SET, 27, **99**, **143**, 228
ACCESS_FACET_DATA_LIST_SET, 26, **100**, **144**, 227, 228
ACCESS_FACET_DATA_SET, 25, **97**, **141**, 227
ACCESS_HALF_SPACE_LIST_2, 24, **138**, 226
ACCESS_HALF_SPACE_LIST_3, 24, **138**, 226
ACCESS_HOMOGENEOUS_COLOUR_VALUE_ARRAY, **107**, **123**
ACCESS_HOMOGENEOUS_COLOUR_VALUE_LIST, **106**, **123**
ACCESS_LIST_OF_COLOUR_VALUE_LIST, **31**, 36, 43, **106**, **123**, **158**, 160, 172
ACCESS_LIST_OF_EDGE_FLAG_LIST, 25, **31**, 55, **193**, 227
ACCESS_LIST_OF_EDGE_FLAG_TRIPLET, 26, **31**, **194**, 227
ACCESS_LIST_OF_LIST_OF_EDGE_FLAG_LIST, 26, **31**, **193**, 227
ACCESS_LIST_OF_LIST_OF_VERTEX_INDEX_LIST, 26, **32**, **219**, 227
ACCESS_LIST_OF_POINT_LIST_2, 21, **136**, 222
ACCESS_LIST_OF_POINT_LIST_3, 21, **136**, 222
ACCESS_LIST_OF_POINT_LIST_4, **93**, **136**
ACCESS_LIST_OF_TRIMCURVE_LIST, 27, **32**, **215**, 228
ACCESS_LIST_OF_VERTEX_COLOUR_LIST_SET_3, 25, **105**, **149**, 226
ACCESS_LIST_OF_VERTEX_DATA_LIST_SET_2, 25, **105**, **149**, 227
ACCESS_LIST_OF_VERTEX_DATA_LIST_SET_3, 25, **104**, **148**, 227
ACCESS_LIST_OF_VERTEX_INDEX_LIST, **32**, 55, **219**
ACCESS_LIST_OF_VERTEX_INDEX_TRIPLET, 26, **32**, **219**, 227
ACCESS_MAGNITUDE_LIST, **95**, **139**
ACCESS_POINT_ARRAY_2, **93**, **136**
ACCESS_POINT_ARRAY_3, **93**, **136**
ACCESS_POINT_ARRAY_4, **93**, **136**
ACCESS_POINT_LIST_2, 20, 21, **135**, 136, 221, 222, 223
ACCESS_POINT_LIST_3, 20, 21, **135**, 136, 221, 222
ACCESS_POINT_LIST_4, **93**, **135**, 136
ACCESS_STRING, 20, 222
ACCESS_SURFACE_COLOURSPLINE, 27, **32**, **164**, 229
ACCESS_SURFACE_GEOMETRYSPLINE, 27, **96**, **140**, 228
ACCESS_TRIMCURVE_LIST, **32**, 55, **215**
ACCESS_VECTOR_ARRAY_3, **95**, **137**
ACCESS_VECTOR_LIST_3, **95**, **137**
ACCESS_VERTEX_COLOUR_LIST_SET_3, **105**, **149**
ACCESS_VERTEX_DATA_ARRAY_SET_2, 27, **103**, **147**, 228
ACCESS_VERTEX_DATA_ARRAY_SET_3, 27, **102**, **146**, 228
ACCESS_VERTEX_DATA_LIST_SET_2, 26, **105**, **149**, 227, 228
ACCESS_VERTEX_DATA_LIST_SET_3, 26, **104**, **148**, 227, 228
ACCESS_VERTEX_INDEX_LIST, **33**, 55, **218**
ACCESS_WEIGHT_LIST, **33**, 36, **171**, 172
ADD_NAMES, **119**
ADD_NAMES_TO_SET, **18**, 23, **180**, 225
ADD_TO_LIST, **114**, **115**
ALL_ELEMENT_TYPES, **16**, **179**
ALLOWED, **46**, **177**
ALWAYS, **174**
AMBIENT_DIFFUSE_REFLECTANCE, **67**, **203**, 306
AMBIENT_DIFFUSE_SPECULAR_REFLECTANCE, **67**, **203**
AMBIENT_LIGHT, 52, **68**, **190**, 306
AMBIENT_REFLECTANCE, **67**, **203**
ANGLE, 62, 69, **165**, 190, 301, 304, 305
ANNOTATION_STYLE, 23, **165**, 224
ANNOTATION_STYLES, **165**
ANNOTATION_TEXT_RELATIVE, **16**, 20, **179**, 222
ANNOTATION_TEXT_RELATIVE_3, **16**, 20, **179**, 222
APPLICATION_DATA, **18**, 25, **181**, 226
APPLICATION_DATA_RANGE, **165**
APPLICATION_DATA_RECORD, 25, **165**, 226
APPLY_FILTER, **119**
ARCHIVE_ID, **165**, 166
ARCHIVE_IDS, **166**
ARCHIVE_STATE, **166**
ARCL, **166**
AROP, **166**

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567514 828 ■

ARRAY_OF_DATA_VALUE_SET, **33**, 98, 102, **126**, 142, 146, **151**
ARRAY_OF_EDGE_FLAG_PAIR, 30, **33**, 73, 74, **194**, 238
ARRAY_OF_MAGNITUDE_LISTS, **95**, **139**
ASAP, **176**
ASF, 24, **166**, 225
ASPECT, **14**, 24, 225
ASTI, **176**
ATTENUATION_COEFFICIENTS, **33**, 52, **168**, 190, 191
ATTRIBUTES_USED, **168**
ATTRIBUTES_USED_TYPE, **16**, 168

B

BACK_INTERIOR_COLOUR_DIR, **15**, **167**
BACK_METHOD_OF_DATA_MAPPING, **15**, **167**
BACK_METHOD_OF_INTERIOR_SHADING, **15**, **167**
BACK_REFLECTANCE_MODEL, **15**, **167**
BACK_REFLECTANCE_PROPERTIES, **15**, **167**
BACK_STYLE_IND, **15**, **167**
BACK_STYLE_OF_INTERIOR, **15**, **167**
BACKFACING, **47**, **183**
BASE_TYPES, **121**, 125, 151
BI_VALUE_NON_UNIFORM, 43, **67**, **155**, 161
BI_VALUE_UNIFORM, 43, **67**, **155**, 160
Binding Specific Error 2002, 9, 73, 74
BNIG, **176**
BNIL, **176**
BOOLEAN, 113, 115, 118
BUILD_NAME_SET, **118**
BUILD_NAME_SET_DIFFERENCE, **118**
BUILD_NAME_SET_INTERSECTION, **118**
BUILD_NAME_SET_UNION, **118**
BUNDLED, **166**

C

CBS, **178**
cell array 3 plus, 6, 71
CELL_ARRAY, 6, **17**, 21, **71**, **179**, 222, **236**
CELL_ARRAY_3, **17**, 21, **179**, 222
CELL_ARRAY_3_PLUS, **18**, 25, **181**, 227
CHAR_EXPANSION, 22, 80, 83, 88, **168**, 224, 244, 263, 275
CHAR_SET, **168**
CHAR_SETS, **168**
CHAR_SPACING, 22, 80, 83, 88, **168**, 224, 244, 263, 275
CHARACTER, 121
CHARACTERISTICS_OF_PARAMETRIC_SURFACE, **15**, **167**
CHOICE_DEVICE_NUMBER, **169**
CHOICE_NUMBER, **169**
CHOICE_PROMPT, **169**
CHOICE_PROMPT_ECHO_TYPE, **169**
CHOICE_PROMPT_ECHO_TYPES, **169**
CHOICE_PROMPT_STRING, **170**
CHOICE_PROMPT_STRING_ARRAY, **170**
CHOICE_PROMPT_STRING_LIST, 153, **170**
CHOICE_PROMPTS, **169**
CHOICE_REQUEST_STATUS, **170**
CHOICE_SMALL_NATURAL, **153**, 170
CHOICE_STATUS, **170**
CHROMATICITY_COEFFICIENT, **157**, 158
CHROMATICITY_COEFFICIENT_SET, **158**
CIELUV, 35, 39, 40, 48, 54, **127**, 131, 132, 133, **152**, 159
CIELUV_COLOUR_VALUE, **34**, 48, **127**, 128, 131, **152**
CIELUV_HOMOGENEOUS_COLOUR_VALUE, **34**, **128**, **152**
CIELUV_TYPES, **34**, 35, 39, 40, 54, **128**, 132, 133, **152**, 159
CLIP, **170**
CLIPPING_INDICATOR, 24, **170**, 226
CLOSE_PHIGS, 307, 312
CLOSE_STRUCTURE, 304, 305, 306, 312
CLOSE_WS, 306, 312
COLOUR, **156**
COLOUR_ASPECT, **61**, **155**
COLOUR_AVAILABLE, **156**
COLOUR_COEFFICIENT, 33, 34, 49, 50, 51, 61, 66, **127**, 128, 129, 130, **152**, 156, 157, 168, 171
COLOUR_COEFFICIENT_ARRAY, 49, **127**, 128, 131, **152**, 157
COLOUR_COEFFICIENTS, **156**
COLOUR_COMPONENTS, 66, **126**, 127, **151**, 171
COLOUR_CONTROL_POINT_ARRAY, **34**, 64, **158**, 163
COLOUR_CONTROL_POINT_LIST, **35**, 42, **159**, 163
COLOUR_INDEX, 22, 23, 48, 51, **126**, 130, 131, **151**, 156, 223, 224, 225, 304, 305
COLOUR_INDICES, **156**
COLOUR_MAPPING_DATA_RECORD, 30, **36**, 82, 108, **171**, 172, 246
COLOUR_MAPPING_INDEX, 29, **37**, 79, 82, 86, 92, **172**, 231, 243, 246, 267, 280
COLOUR_MAPPING_INDICES, **37**, 86, **172**, 267
COLOUR_MAPPING_METHOD, 36, **37**, 38, 68, 86, 92, **171**, 173, 267, 280
COLOUR_MAPPING_METHOD_FACILITIES, **37**, 92, **173**, 280
COLOUR_MAPPING_METHODS, **38**, 92, **171**, 280
COLOUR_MAPPING_STATE, **38**, 86, **173**, 267
COLOUR_MATRIX, **156**
COLOUR_MATRIX_SMALL_NATURAL, **153**, 156
COLOUR_MODEL, 29, 34, 35, 36, 38, 39, 42, 44, 48, 53, 64, 67, 79, 96, 97, 99, 100, 102, 103, 104, 105, **126**, 127, 131, 132, 133, 140, 141, 143, 144, 146, 147, 148, 149, **152**, 156, 157, 158, 159, 163, 164, 172, 230, 243
COLOUR_MODELS, 87, **156**, 280, 281
COLOUR_REPRESENTATION, **157**, 300, 301
COLOUR_VALUE, **106**, **123**
COLOUR_VALUE_ARRAY, 30, **38**, 39, 71, 81, 98, 101, **106**, 107, **123**, 124, **131**, 132, 142, 145, **153**, 158, 237, 246
COLOUR_VALUE_LIST, 30, **39**, 40, 99, 103, **106**, 107, 108, **123**, 124, **132**, 133, 143, 147, **153**, 158, 291

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

9593 PT*3 90 ■ 4851903 0567515 764 ■

COLOUR_VALUE_SET_DIMENSION, 34, 35, 39, **40**, 42, 44, 53, 64, 106, 107, **122**, 123, 124, 131, 132, 133, **151**, 158, 159, 163, 164
COMPOSE_MATRIX, 304, 305, 306
COMPOSITION_TYPE, 24, **174**, 225
CONDITIONALLY, **174**
CONFLICT_RESOLUTION, **174**
Conformance, 3
CONNECTION_ID, **174**, 300, 308
CONSTANT_SUBDIVISION_CURVE, 41, 64, **67**, **175**, 214, 309
CONSTANT_SUBDIVISION_SURFACE, 63, **68**, **210**
CONSTANT_SUBDIVISION_SURFACE_BETWEEN_KNOTS, 311
CONTROL_FLAG, **174**
CONTROL_POINT_ARRAY, 34, 35, **107**, **124**, 158, 159
CONTROL_POINT_ARRAY_3, 44, **94**, 96, **137**, 140, 164
CONTROL_POINT_LIST, 35, 36, **107**, **123**, 124, 159, 160
CONTROL_POINT_LIST_2, 64, **94**, 95, **136**, 137, 140, 214
CONTROL_POINT_LIST_3, **93**, 94, 95, **136**, 139
COORDINATE_COMPONENT_TYPE, 93, **135**, 138
COORDINATES, **65**, **134**
COORDINATES_COLOUR, **65**, 101, 103, **134**, 145, 147
COORDINATES_COLOUR_DATA, **65**, 101, 102, 103, 104, **134**, 145, 146, 147, 148
COORDINATES_COLOUR_NORMAL, **65**, 101, 103, **134**, 145, 147
COORDINATES_COLOUR_NORMAL_DATA, **65**, 101, 102, 103, 104, **134**, 145, 146, 147, 148
COORDINATES_DATA, **65**, 102, 104, **134**, 146, 148
COORDINATES_NORMAL, **65**, 101, 103, **134**, 145, 147
COORDINATES_NORMAL_DATA, **65**, 101, 102, 103, 104, **134**, 145, 146, 147, 148
COPY_NAME_SET, **118**
CRITERIA_FOR_CURVE_APPROX, **15**, **167**
CRITERIA_FOR_SURFACE_APPROX, **15**, **167**
CURVE_APPROX_CRITERIA_TYPE, **40**, 41, 64, 67, 68, **174**, 175, 214
CURVE_APPROX_CRITERIA_TYPES, **40**, 90, **174**, 278
CURVE_APPROX_DATA_RECORD, 29, **41**, 64, 78, 79, 82, **88**, **175**, 214, 230, 243, 244, 262, 274
CURVE_COLOURSPLINE, 30, **42**, 74, 108, **163**, 238, 291
CURVE_GEOMETRYSPLINE_2, **95**, 96, **139**, 140
CURVE_GEOMETRYSPLINE_3, 74, **95**, **139**, 238
CURVE_PLACEMENT_TYPE, **42**, 56, **176**, 200
CURVE_VISIBILITY_FLAG, **42**, 64, **193**, 214

D

DATA_MAPPING_COLOUR, 43, **67**, **155**, 160
DATA_MAPPING_DATA_RECORD, 28, **43**, 77, 78, 81, 84, 89, 108, **160**, 230, 242, 245, 265, 278, 291
DATA_MAPPING_INDEX, 27, 28, **44**, 75, 81, 84, 89, **126**, **151**, 155, 229, 239, 245, 265, 278
DATA_MAPPING_INDICES, **44**, 84, **155**, 265
DATA_MAPPING_METHOD, 29, 43, **44**, 67, **155**, 160, 230
DATA_MAPPING_METHODS, **44**, 89, **155**, 278
DATA_VALUE, 33, **45**, 46, 54, **125**, 126, **151**, 154
DATA_VALUE_INDEX, 33, 43, 45, 46, 54, 97, 98, 99, 100, 101, 102, 103, 104, **125**, 126, 141, 142, 143, 144, 145, 146, 147, 148, **151**, 154, 160, 164
DATA_VALUE_INDEX_PAIR, 43, **45**, **154**, 160, 161
DATA_VALUE_RANGE, 43, **45**, **154**, 160
DATA_VALUE_SET, 31, **46**, 97, 108, **125**, 126, 141, **151**, 291
DATASPLINE, **44**, 45, **164**
DATASPLINE_LIST, 31, **45**, 75, 108, **164**, 238, 291
DC, 41, 63, 87, **162**, 175, 176, 210, 211, 273
DC_CHORDAL_DEVIATION_CURVE, 41, **68**, **175**
DC_CHORDAL_SIZE_CURVE, 41, **67**, **175**
DC_CHORDAL_SIZE_SURFACE, 63, **68**, **210**
DC_PLANAR_DEVIATION_SURFACE, 63, **68**, **210**, 211
DC_RELATIVE_CURVE, 41, **68**, **175**, 176
DC_RELATIVE_SURFACE, 63, **68**, **210**, 211
DC_TYPE, **162**
DC_UNITS, **176**
DEALLOCATE, **149**, **150**
- COLOUR_MAPPING_DATA_RECORD, **108**
- COLOUR_VALUE_LIST, **108**, **291**
- CURVE_COLOURSPLINE, **108**, **291**
- DATA_MAPPING_DATA_RECORD, **108**, **291**
- DATA_VALUE_SET, **108**, **291**
- DATASPLINE_LIST, **108**, **291**
- LIST_OF_COLOUR_VALUE_LIST, **108**, **291**
- LIST_OF_EDGE_FLAG_LIST, **108**, **291**
- LIST_OF_LIST_OF_EDGE_FLAG_LIST, **108**, **291**
- LIST_OF_LIST_OF_VERTEX_INDEX_LIST, **109**, **291**
- LIST_OF_TRIMCURVE_LIST, **109**, **291**
- LIST_OF_VERTEX_COLOUR_LIST_SET_3, **105**, **149**
- LIST_OF_VERTEX_DATA_LIST_SET_2, **105**, **149**
- LIST_OF_VERTEX_DATA_LIST_SET_3, **105**, **149**
- LIST_OF_VERTEX_INDEX_LIST, **109**, **291**
- NAME_SET, **119**
- SURFACE_COLOURSPLINE, **109**, **291**
- TRIMCURVE_LIST, **109**, **291**

DEFAULT_ERROR_FILE, **110**
DEFAULT_LIST_SIZE, **110**, 113
DEFAULT_MEMORY_UNITS, **110**
DEFERRAL_MODE, **176**
DELETE_FROM_LIST, **114**, **115**
DEPTH_CUE_INDEX, 29, **46**, 79, 82, 86, 91, **176**, 177, 231, 243, 246, 267, 280
DEPTH_CUE_INDICES, **46**, 86, **177**, 267
DEPTH_CUE_MODE, **46**, 82, 86, 91, **177**, 246, 267, 280
DEPTH_CUE_MODES, **46**, 91, **177**, 280
DEPTH_CUE_SCALE_FACTOR, **46**, 47, **177**
DEPTH_CUE_SCALE_FACTORS, **47**, 82, 86, 91, **177**, 246, 267, 280
DEVICE_NUMBER, **169**
DIRECTIONAL_LIGHT, 52, **68**, **190**, 306

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567516 6T0 ■

DISPLAY_CLASS, **178**
DISPLAY_SURFACE_EMPTY, **178**
DODECAHEDRON, **300**
DYNAMIC_MODIFICATION, 87, **178**, 273

E

ECHO, **178**
ECHO_SWITCH, **178**
EDGE_ATTRIBUTES, **16**, **167**
EDGE_COLOUR, **15**, **167**
EDGE_FLAG, 23, 42, 47, 80, 84, 89, 193, 194, 224, 245, 265, **278**
EDGE_FLAG_LIST, 31, **47**, 73, **193**, 238
EDGE_FLAG_PAIR, 33, **47**, **194**
EDGE_FLAG_TRIPLET, **47**, 55, **194**
EDGE_INDEX, 21, 80, 84, 89, 223, 245, 265, 278
EDGETYPE, 23, 80, 84, 89, 224, 245, 265, 278
EDGEWIDTH, 23, 80, 84, 89, 225, 245, 265, 278
EDGEWIDTH_SF, **15**, **167**
EDIT_MODE, **178**
ELEMENT_POSITION, **178**, 179
ELEMENT_TYPE, **16**, **113**, 114, 115, 182
ELEMENT_TYPES, **182**
EMPTY, **178**, 311
ERROR_HANDLING_MODE, **182**
ERROR_NUMBER, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, **182**, 262, 263, 264, 265, 267, 273, 274, 275, 276, 278, 279, 280, 281, 282, 300, 308
ESCAPE, 164
ESCAPE_DATA_RECORD, **183**
ESCAPE_ID, **182**, 183
ESCAPE_IDS, **183**
EVALUATE_VIEW_MAPPING_MATRIX, 303, 310
EVALUATE_VIEW_ORIENTATION_MATRIX, 303, 310
EXECUTE_STRUCTURE, **18**, 25, **180**, 226, 304, 305, 306
EXPANSION, **15**, **166**
EXTRACT_NAMES, **118**

F

FACET_COLOUR, **48**, **61**, 96, 98, 99, **134**, 140, 142, 143, **155**
FACET_COLOUR_ARRAY, **97**, 99, **141**, 143
FACET_COLOUR_DATA, **48**, 96, 97, 98, 99, 100, **134**, 140, 141, 142, 143, 144
FACET_COLOUR_ITEM, **96**, 97, **140**, 141
FACET_COLOUR_LIST, **99**, 100, **143**, 144
FACET_COLOUR_NORMAL, **48**, 96, 97, 98, 99, 100, **134**, 140, 141, 142, 143, 144
FACET_COLOUR_NORMAL_DATA, **48**, 96, 97, 98, 99, 100, **134**, 140, 141, 142, 143, 144
FACET_CULLING_MODE, 28, **47**, 76, **183**, 229, 241
FACET_DATA, **48**, **61**, 97, 98, 100, **134**, 141, 142, 144, **155**
FACET_DATA_ARRAY, **98**, 99, **142**, 143
FACET_DATA_ARRAY_SET, 73, 74, **98**, 99, **142**, 143, 238
FACET_DATA_FLAG, **48**, 96, 97, 98, 99, 100, **134**, 140, 141, 142, 143, 144, **153**
FACET_DATA_ITEM, **97**, **141**
FACET_DATA_LIST, **100**, **144**
FACET_DATA_LIST_SET, 71, 72, 73, **100**, **144**, 237, 238
FACET_DATA_SET, 70, 71, **97**, **141**, 236
FACET_DISTINGUISHING_MODE, 28, **48**, 76, **183**, 229, 241
FACET_NONE, **48**, **134**
FACET_NORMAL, **48**, 97, 98, 100, **134**, 141, 142, 144
FACET_NORMAL_ARRAY, **98**, 99, **142**, 143
FACET_NORMAL_DATA, **48**, 97, 98, 100, **134**, 141, 142, 144
FACET_NORMAL_ITEM, **96**, 97, **140**, 141
FACET_NORMAL_LIST, **99**, 100, **143**, 144
FACET_SET_DIMENSION, **48**, 97, 98, 99, 100, **122**, 141, 142, 143, 144, **151**
FILE_ID, 300, 308
FILE_SMALL_NATURAL, **154**
fill area set 3 with data, 6
FILL AREA SET 3 WITH DATA (with Edge Flags), 70
FILL AREA SET 3 WITH DATA (without Edge Flags), 70
fill area set with data, 6
FILL AREA SET WITH DATA (with Edge Flags), 71
FILL AREA SET WITH DATA (without Edge Flags), 70
FILL_AREA, **17**, 21, **179**, 222
FILL_AREA_3, **17**, 21, **179**, 222
FILL_AREA_SET, 6, **17**, 21, **179**, 222, 304
FILL_AREA_SET_3, **17**, 21, **179**, 222
FILL_AREA_SET_3_WITH_DATA, **18**, 25, **181**, 227
FILL_AREA_SET_WITH_DATA, **18**, 25, **70**, **71**, **181**, 227, **236**
FLAG, **15**, **167**
FONT, **15**, **166**
FRONTFACING, **47**, **183**

G

GDP, **17**, 21, 164, **179**, 223
GDP_3, **17**, 21, **179**, 222
GDP_3_RECORD, 21, 222
GDP_RECORD, 21, 223
GENERAL_COLOUR, 28, 29, **48**, 52, 59, 76, 77, 78, 79, 80, 82, 83, 84, 86, 88, 89, 91, 96, **131**, 140, **153**, 190, 191, 204, 229, 230, 240, 242, 243, 244, 245, 246, 262, 263, 264, 265, 267, 274, 275, 276, 278, 280
GENERAL_EDGE_COLOUR, **15**, **167**
GENERAL_FLOAT_PARAMETER, **164**
GENERAL_INTERIOR_COLOUR, **15**, **167**
GENERAL_POLYLINE_COLOUR, **15**, **167**
GENERAL_POLYMARKER_COLOUR, **15**, **167**
GENERAL_TEXT_COLOUR, **15**, **167**
GENERIC_COLOUR_TYPES, 35, 36, 39, 40, **49**, 54, **128**, 132, 133, **152**, 159, 160
GSE, **18**, 25, 164, **181**, 226
GSE_RECORD, 25, 226

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

9593 PT*3 90 ■ 4851903 0567517 537 ■

H

HALF_SPACE_2, **138**
HALF_SPACE_3, **138**
HALF_SPACE_LIST_2, **138**
HALF_SPACE_LIST_3, **138**
HATCH_STYLES, 88, 276
HLHSR_ID, 24, 225
HLHSR_MODE, 300
HLS, 35, 36, 39, 40, 49, 54, **127**, 131, 132, 133, **152**, 159, 160
HLS_COLOUR_VALUE, 49, 50, **128**, 129, 131, **152**
HLS_HOMOGENEOUS_COLOUR_VALUE, **49**, 50, **129**, **152**
HLS_TYPES, 35, 36, 39, 40, **50**, 54, **129**, 132, 133, **152**, 159, 160
HOMOGENEOUS_COLOUR_VALUE, **106**, 107, **123**
HOMOGENEOUS_COLOUR_VALUE_ARRAY, **107**, **123**, 124
HOMOGENEOUS_COLOUR_VALUE_LIST, **106**, 107, **123**, 124
HSV, 35, 36, 39, 40, 48, 54, **127**, 131, 132, 133, **152**, 159, 160
HSV_COLOUR_VALUE, 48, **50**, **129**, 131, **152**
HSV_HOMOGENEOUS_COLOUR_VALUE, **50**, **129**, **152**
HSV_TYPES, 35, 36, 39, 40, **50**, 54, **129**, 132, 133, **152**, 159, 160

I

IMM, **178**
INDIRECT, 34, 35, 39, 48, 53, **67**, **127**, 131, 132, 133, **152**, 158, 159
INDIRECT_HOMOGENEOUS_COLOUR_VALUE, **51**, **130**, **152**
INDIRECT_TYPES, 34, 35, 39, **51**, 53, **130**, 132, 133, **152**, 158, 159
INDIVIDUAL, **166**
INPUT_STRING, 154
INPUT_STRING_SMALL_NATURAL, **154**
INQ_COLOUR_MAPPING_FACILITIES, 5, 6, **92**, **280**
INQ_COLOUR_MAPPING_METHOD_FACILITIES, 5, 6, **92**, **280**
INQ_COLOUR_MAPPING_REPRESENTATION, 5, 6, **86**, **267**
INQ_COLOUR_MAPPING_STATE, 5, 6, **86**, **267**
INQ_CURVE_AND_SURFACE_FACILITIES, 5, 6, **90**, **278**
INQ_DATA_MAPPING_FACILITIES, 5, 6, **89**, **278**
INQ_DATA_MAPPING_REPRESENTATION, 5, 6, **84**, **265**
INQ_DEPTH_CUE_FACILITIES, 5, 6, **91**, **280**
INQ_DEPTH_CUE_REPRESENTATION, 5, 6, **86**, **267**
INQ_DIRECT_COLOUR_MODEL_FACILITIES, 5, 6, **87**, **280**
INQ_DYNAMICS_OF_WS_ATTRIBUTES, 6, **87**, **273**
INQ_DYNAMICS_OF_WS_ATTRIBUTES (PLUS), 5, 6
INQ_EDGE_REPRESENTATION, 6, **84**, **265**
INQ_EDGE_REPRESENTATION (PLUS), 5
INQ_GDP, 168
INQ_INTERIOR_FACILITIES, 6, **88**, **276**
INQ_INTERIOR_FACILITIES (PLUS), 5
INQ_INTERIOR_REPRESENTATION, 6, **83**, **264**
INQ_INTERIOR_REPRESENTATION (PLUS), 5
INQ_LIGHT_SOURCE_FACILITIES, 5, 6, **91**, **280**
INQ_LIGHT_SOURCE_REPRESENTATION, 5, 6, **85**, **267**
INQ_LIST_OF_COLOUR_MAPPING_INDICES, 5, 6, **86**, **267**
INQ_LIST_OF_DATA_MAPPING_INDICES, 5, 6, **84**, **265**
INQ_LIST_OF_DEPTH_CUE_INDICES, 5, 6, **86**, **267**
INQ_LIST_OF_LIGHT_SOURCE_INDICES, 5, 6, **85**, **267**
INQ_LIST_OF_PARAMETRIC_SURFACE_INDICES, 5, 6, **85**, **265**
INQ_LIST_OF_REFLECTANCE_INDICES, 5, 6, **84**, **265**
INQ_PARAMETRIC_SURFACE_REPRESENTATION, 5, 6, **85**, **265**
INQ_PATTERN_REPRESENTATION, 6, **85**, **267**
INQ_PATTERN_REPRESENTATION (PLUS), 5
INQ_POLYLINE_FACILITIES, 6, **87**, **273**
INQ_POLYLINE_FACILITIES (PLUS), 5
INQ_POLYLINE_REPRESENTATION, 6, **82**, **262**
INQ_POLYLINE_REPRESENTATION (PLUS), 5
INQ_POLYMARKER_REPRESENTATION, 6, **83**, **263**
INQ_POLYMARKER_REPRESENTATION (PLUS), 5
INQ_PREDEFINED_COLOUR_MAPPING_REPRESENTATION, 5, 6, **92**, **280**
INQ_PREDEFINED_DATA_MAPPING_REPRESENTATION, 5, 6, **89**, **278**
INQ_PREDEFINED_DEPTH_CUE_REPRESENTATION, 5, 6, **91**, **280**
INQ_PREDEFINED_EDGE_REPRESENTATION, 6, **89**, **278**
INQ_PREDEFINED_EDGE_REPRESENTATION (PLUS), 5
INQ_PREDEFINED_INTERIOR_REPRESENTATION, 6, **89**, **276**
INQ_PREDEFINED_INTERIOR_REPRESENTATION (PLUS), 5
INQ_PREDEFINED_LIGHT_SOURCE_REPRESENTATION, 5, 6, **91**, **280**
INQ_PREDEFINED_PARAMETRIC_SURFACE_REPRESENTATION, 6, **90**, **279**
INQ_PREDEFINED_PATTERN_REPRESENTATION, 7, **91**, **279**
INQ_PREDEFINED_PATTERN_REPRESENTATION (PLUS), 6
INQ_PREDEFINED_POLYLINE_REPRESENTATION, 7, **88**, **274**
INQ_PREDEFINED_POLYLINE_REPRESENTATION (PLUS), 6
INQ_PREDEFINED_POLYMARKER_REPRESENTATION, 7, **88**, **274**
INQ_PREDEFINED_POLYMARKER_REPRESENTATION (PLUS), 6

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567518 473 ■

INQ_PREDEFINED_REFLECTANCE_REPRESENTATION, 6, 7, **90**, **278**
INQ_PREDEFINED_TEXT_REPRESENTATION, 7, **88**, **275**
INQ_PREDEFINED_TEXT_REPRESENTATION (PLUS), 6
INQ_REFLECTANCE_FACILITIES, 6, 7, **90**, **278**
INQ_REFLECTANCE_REPRESENTATION, 6, 7, **84**, **265**
INQ_RENDERING_COLOUR_MODEL_FACILITIES, 6, 7, **87**, **281**
INQ_TEXT_REPRESENTATION, 7, **83**, **263**
INQ_TEXT_REPRESENTATION (PLUS), 6
INQ_WS_STATE_TABLE_LENGTHS, 7, **92**, **282**
INQ_WS_STATE_TABLE_LENGTHS (PLUS), 6
inquire colour mapping facilities, 6
inquire colour mapping method facilities, 6
inquire colour mapping representation, 6
inquire colour mapping state, 6
inquire curve and surface facilities, 6
inquire data mapping facilities, 6
inquire depth cue facilities, 6
inquire depth cue representation, 6
inquire direct colour model facilities, 6
inquire dynamics of workstation attributes plus, 6
inquire edge representation plus, 6
inquire interior facilities plus, 6
inquire interior representation plus, 6
inquire light source facilities, 6
inquire light source representation, 6
inquire list of colour mapping indices, 6
inquire list of data mapping indices, 6
inquire list of depth cue indices, 6
inquire list of light source indices, 6
inquire list of parametric surface indices, 6
inquire list of reflectance indices, 6
inquire parametric surface representation, 6
inquire pattern representation plus, 6
inquire polyline facilities plus, 6
inquire polyline representation plus, 6
inquire polymarker representation plus, 6
inquire predefined colour mapping representation, 6
inquire predefined data mapping representation, 6
inquire predefined depth cue representation, 6
inquire predefined edge representation plus, 6
inquire predefined interior representation plus, 6
inquire predefined light source representation, 6
inquire predefined parametric surface representation, 6
inquire predefined pattern representation plus, 7
inquire predefined polyline representation plus, 7
inquire predefined polymarker representation plus, 7
inquire predefined reflectance representation, 7
inquire predefined text representation plus, 7
inquire reflectance facilities, 7
inquire reflectance representation, 7
inquire rendering colour model facilities, 7
inquire text representation plus, 7
inquire workstation state table lengths plus, 7
INQUIRE_MEMBERSHIP, **119**
INSERT, **178**
INTENSITY, 49, 50, 60, **127**, 128, 129, 130, **152**, 157
INTERIOR_ATTRIBUTES, **16**, **167**
INTERIOR_COLOUR, **15**, **166**
INTERIOR_COLOUR_SHADING, **67**, **188**, 306
INTERIOR_DATA_AND_DOT_SHADING, **67**, **188**
INTERIOR_DATA_AND_NORMAL_SHADING, **67**, **188**
INTERIOR_DATA_SHADING, **67**, **188**
INTERIOR_INDEX, 21, 28, 75, 80, 83, 89, 223, 229, 239, 245, 264, 276
INTERIOR_SHADING_METHOD, 28, 29, **51**, 67, 77, 80, 83, 89, **188**, 229, 230, 242, 245, 264, 276
INTERIOR_SHADING_METHODS, **51**, 88, **188**, 276
INTERIOR_STYLE, 23, 28, 77, 80, 83, 89, 224, 230, 242, 245, 264, 276
INTERIOR_STYLES, 88, 276
IRG, **178**
IS_EQUAL, **118**
IS_IN_LIST, **113**, **115**
ISOPARAMETRIC_CURVES, 56, **68**, **200**, 311

K

KNOT, **51**, **189**
KNOT_VECTOR, 42, 44, **52**, 64, **163**, 164, 214

L

LABEL, **18**, 25, **181**, 226
LABEL_ID, 25, 226
LIGHT_SOURCE_DATA_RECORD, **52**, 81, 85, 91, **190**, 246, 267, 280
LIGHT_SOURCE_INDEX, **53**, 81, 85, 91, **191**, 246, 267, 280
LIGHT_SOURCE_INDICES, 29, **53**, 78, 85, **191**, 230, 242, 267, 301, 306
LIGHT_SOURCE_TYPE, 52, **53**, 68, **190**
LIGHT_SOURCE_TYPES, **53**, 91, **190**, 280
LIGHTING_EXPONENT, **52**, 59, **189**, 191, 204
LINE_COLOUR, **14**, **166**
LINETYPE, 22, 79, 82, 88, 223, 244, 262, 274
LINETYPES, 87, 273
LINEWIDTH, 22, 79, 82, 88, 223, 244, 262, 274
LINEWIDTH_SF, **14**, **166**
LIST, **113**, **115**, 306
LIST_ELEMENT, **113**, **115**
LIST_OF, 29, 78, 84, 85, 86, 87, 88, 89, 90, 91, 92, **113**, 114, 115, 118, 119, 230, 242, 265, 267, 273, 276, 278, 280, 281, 301
LIST_OF_COLOUR_VALUE_LIST, 31, **53**, 54, **106**, 108, **123**, **133**, **153**, 158, 291
LIST_OF_DATA_VALUE_SET, **54**, 100, 104, **126**, 144, 148, **151**
LIST_OF_EDGE_FLAG_LIST, 31, **54**, 70, 71, 108, **193**, 236, 291
LIST_OF_EDGE_FLAG_TRIPLET, 31, **55**, 72, **194**, 237
LIST_OF_LIST_OF_EDGE_FLAG_LIST, 31, **55**, 71, 72, 108, **193**, 237, 291

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

9593 PT*3 90 ■ 4851903 0567519 30T ■

LIST_OF_LIST_OF_VERTEX_INDEX_LIST, 32, **55**, 71, 72, 109, **219**, 237, 291
LIST_OF_POINT_LIST_2, **136**
LIST_OF_POINT_LIST_3, **136**, 301
LIST_OF_POINT_LIST_4, **93**, **136**
LIST_OF_TRIMCURVE_LIST, 32, **55**, 74, 75, 109, **215**, 238, 291, 310
LIST_OF_VERTEX_COLOUR_LIST_SET_3, 70, **105**, **149**, 150, 236
LIST_OF_VERTEX_DATA_LIST_SET_2, 70, 71, **105**, **149**, 150, 236
LIST_OF_VERTEX_DATA_LIST_SET_3, 70, **104**, 105, **148**, 149, 236
LIST_OF_VERTEX_INDEX_LIST, 32, **55**, 109, **218**, 219, 291
LIST_OF_VERTEX_INDEX_TRIPLET, 32, **55**, 72, **219**, 237
LIST_SIZE, **113**, 114
LIST_VALUES, **113**, 114, 115, 301
LOCATOR_ATTRIBUTES_USED, 168

M

MAGNITUDE, 22, 41, 51, 63, 87, 95, 138, 139, 175, 189, 210, 211, 224, 273
MAGNITUDE_BASE_TYPE, **138**
MAGNITUDE_LIST, 52, 56, **95**, 96, **139**, 140, 163, 200, 201
MAGNITUDE_LIST_ARRAY, **95**, **139**
MAINTAIN, **174**
MARKER_COLOUR, **15**, **166**
MARKER_SIZE, 22, 79, 83, 88, 223, 244, 263, 274
MARKER_TYPE, 22, 79, 83, 88, 223, 244, 263, 274
MAX_ANNOTATION_STYLES_SUPPORTED, **110**, 165
MAX_APPLICATION_DATA, **110**, 165
MAX_ARCHIVE_IDS_SUPPORTED, **110**, 166
MAX_CHAR_SETS_SUPPORTED, **110**, 168
MAX_CHOICE_PROMPT_ECHO_TYPES_SUPPORTED, **110**, 169
MAX_CHOICE_PROMPTS_SUPPORTED, **110**, 169
MAX_CHOICE_SMALL_NATURAL, **110**
MAX_COLOUR_COEFFICIENTS, 66, **110**, **126**, 156, 171
MAX_COLOUR_INDICES_SUPPORTED, **110**, 156
MAX_COLOUR_MAPPING_INDICES_SUPPORTED, 37, **69**, **112**, 172
MAX_COLOUR_MAPPING_METHODS_SUPPORTED, 38, **69**, 171
MAX_COLOUR_MATRIX_SMALL_NATURAL, **111**, 153
MAX_COLOUR_MODELS_SUPPORTED, **111**, 156
MAX_COLOUR_VALUES_SUPPORTED, 40, **69**, **112**, 122
MAX_CURVE_APPROX_CRITERIA_TYPES_SUPPORTED, 40, **69**, **112**, 174
MAX_DATA_MAPPING_INDICES_SUPPORTED, 44, **69**, **112**, 155
MAX_DATA_MAPPING_METHODS_SUPPORTED, 44, **69**, **112**, 155
MAX_DATA_VALUES_PER_FACET, **69**, **112**
MAX_DATA_VALUES_PER_VERTEX, **69**, **112**
MAX_DEPTH_CUE_INDICES_SUPPORTED, 46, 59, **69**, **112**, 177
MAX_EDGE_INDICES_SUPPORTED, **111**
MAX_EDGETYPES_SUPPORTED, **111**
MAX_ESCAPE_IDS_SUPPORTED, **111**, 183
MAX_FACETS_SUPPORTED, 48, **69**, **112**, 122
MAX_FILE_IDS_SUPPORTED, **111**
MAX_FILE_SMALL_NATURAL, **111**, 154
MAX_GDP_3_IDS_SUPPORTED, **111**
MAX_GDP_IDS_SUPPORTED, **111**
MAX_GSE_IDS_SUPPORTED, **111**
MAX_HATCH_STYLES_SUPPORTED, **111**
MAX_HLHSR_IDS_SUPPORTED, **111**
MAX_HLHSR_MODES_SUPPORTED, **111**
MAX_INPUT_STRING_SMALL_NATURAL, **111**, 154
MAX_INTEGER_VALUE, **111**, 121
MAX_INTERIOR_INDICES_SUPPORTED, **111**
MAX_INTERIOR_SHADING_METHODS_SUPPORTED, 51, **69**, **112**, 188
MAX_LIGHT_SOURCE_INDICES_SUPPORTED, 53, **69**, **112**, 191
MAX_LIGHT_SOURCE_TYPES_SUPPORTED, 53, **69**, **112**, 190
MAX_LINETYPES_SUPPORTE, **111**
MAX_LIST_SIZE, **113**
MAX_LOCATOR_PROMPT_ECHO_TYPES_SUPPORTED, **111**
MAX_MARKER_TYPES_SUPPORTED, **111**
MAX_MEMORY_UNITS, **111**
MAX_METAFILE_ITEM_LENGTH, **111**
MAX_METAFILE_ITEM_TYPE, **111**
MAX_MODELLING_CLIP_DISTINCT_PLANES_SUPPORTED, **111**
MAX_MODELLING_CLIP_OPERATION_TYPES_SUPPORTED, **111**
MAX_NAME_SET_FILTER_LIST_LENGTH_SUPPORTED, **111**
MAX_NAME_SUPPORTED, **111**, 117
MAX_OPEN_WS_SUPPORTED, **111**
MAX_PARAMETRIC_SURFACE_CHARACTERISTICS_SUPPORTED, 56, **69**, **112**, 200
MAX_PARAMETRIC_SURFACE_INDICES_SUPPORTED, 57, **69**, **112**, 201
MAX_PATH_DEPTH_SUPPORTED, **111**
MAX_PATTERN_INDICES_SUPPORTED, **111**
MAX_PICK_IDS_SUPPORTED, **111**
MAX_PICK_PROMPT_ECHO_TYPES_SUPPORTED, **111**
MAX_POLYLINE_INDICES_SUPPORTED, **111**
MAX_POLYLINE_SHADING_METHODS_SUPPORTED, 57, **69**, **112**, 206
MAX_POLYMARKER_INDICES_SUPPORTED, **111**
MAX_POSTED_STRUCTURES_SUPPORTED, **111**
MAX_REFERENCE_PATHS_SUPPORTED, **111**

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567520 021 ■

MAX_REFLECTANCE_INDICES_SUPPORTED, 58, **69**, **112**, 204
MAX_REFLECTANCE_MODELS_SUPPORTED, 58, **69**, **112**, 203
MAX_REFLECTANCE_PROPERTIES_TYPES_SUPPORTED, 59, **69**, **112**, 204
MAX_SMALL_NATURAL, **111**, 122
MAX_STRING_PROMPT_ECHO_TYPES_SUPPORTED, **111**
MAX_STRING_SMALL_NATURAL, **111**, 154
MAX_STROKE_PROMPT_ECHO_TYPES_SUPPORTED, **111**
MAX_STRUCTURE_IDS_SUPPORTED, **111**
MAX_SURFACE_APPROX_CRITERIA_TYPES_SUPPORTED, 62, **69**, **112**, 209
MAX_TEXT_FONT_PRECISION_PAIRS_SUPPORTED, **111**
MAX_TEXT_INDICES_SUPPORTED, **111**
MAX_VALUATOR_PROMPT_ECHO_TYPES_SUPPORTED, **111**
MAX_VERTEX_SETS_SUPPORTED, 66, **69**, 122
MAX_VERTICES_SUPPORTED, 65, 66, **69**, **112**, 122, 218
MAX_VIEW_INDICES_SUPPORTED, **111**
MAX_WS_IDS_SUPPORTED, **111**
MAX_WS_TYPES_SUPPORTED, **111**
MC, 20, 21, 22, 23, 24, 25, 26, 27, 56, 70, 71, 72, 73, 74, 75, **161**, 200, 221, 222, 223, 224, 225, 226, 227, 228, 236, 237, 238, 301, 308
MC_LEVEL_CURVES, 56, **68**, **200**
MC_TYPE, **161**, 301, 302
MEMBER, 117
METHOD_OF_DATA_MAPPING, **15**, **167**
METHOD_OF_INTERIOR_SHADING, **15**, **167**
METHOD_OF_POLYLINE_SHADING, **15**, **167**
METRES, 176
MIN_INTEGER_VALUE, **111**, 121
MIN_METAFILE_ITEM_TYPE, **111**
MODEL_OF_REFLECTANCE, **15**, **167**
MODELLING_CLIP_OPERATION_TYPE, 24, 226
MONOCHROME, **156**

N

NAME, **117**, 119
NAME_SET, 23, 24, **117**, 118, 119, 225
NAME_SET_ACCEPTANCE, **117**, 119
NAME_SET_FILTER, 117, 119
NAME_SET_MEMBERSHIP, **117**, 119
NAMES, **117**, 118, 119
NATURAL, 66, 113, 115, 122
NIL, **16**, 20, **179**, 221
NO_FACET_CULLING, **47**, **183**
NO_INTERIOR_SHADING, **67**, **188**
NO_PARAMETRIC_SURFACE, 56, **68**, **200**
NO_POLYLINE_SHADING, **67**, **206**
NO_REFLECTANCE, **67**, **203**
NOCHOICE, **170**
NOCLIP, **170**
NOECHO, **178**
NON_MEMBER, 117
NON_RATIONAL, **62**, 94, 107, **122**, 124, 136, 137, 308, 309, 310
NON_UNIFORM, **42**, **176**
non_uniform B-spline surface with data, 7
NON_UNIFORM_B_SPLINE_CURVE, 7, **18**, 27, **74**, **181**, 228, **238**
NON_UNIFORM_B_SPLINE_CURVE_WITH_COLOUR, **18**, 27, **181**, 228
NON_UNIFORM_B_SPLINE_SURFACE, 7, **18**, 27, **74**, **181**, 228, **238**, 311
NON_UNIFORM_B_SPLINE_SURFACE_WITH_DATA, **18**, 27, **75**, **181**, 228, **238**
NONE, **170**
non-uniform B-spline curve, 7
non-uniform B-spline curve with colour, 7
non-uniform B-spline surface, 7
NOTEMPTY, **178**
NPC, 20, 22, 23, 41, 63, 82, 86, 91, **162**, 175, 210, 211, 222, 224, 246, 267, 280
NPC_CHORDAL_DEVIATION_CURVE, 41, **67**, **175**
NPC_CHORDAL_SIZE_CURVE, 41, **67**, **175**
NPC_CHORDAL_SIZE_SURFACE, 63, **68**, **210**
NPC_PLANAR_DEVIATION_SURFACE, 63, **68**, **210**
NPC_RELATIVE_CURVE, 41, **68**, **175**
NPC_RELATIVE_SURFACE, 63, **68**, **210**, 211
NPC_TYPE, **162**
NULL_LIST, **113**, **115**, 306

O

OFF, **165**
OFF_ON, 48, **165**, 169, 182, 183
OK, **170**
ON, **165**, 309, 310, 311
OPEN_PHIGS, 302, 311
OPEN_STRUCTURE, 304, 305, 306, 311
OPEN_WS, 302, 311
OTHER, 176
OTHER_DISPLAY, **178**

P

PARALLEL, 303, 310
PARAMETRIC_SURFACE_ATTRIBUTES, **16**, **167**
PARAMETRIC_SURFACE_CHARACTERISTIC, **56**, 68, **200**
PARAMETRIC_SURFACE_CHARACTERISTICS, **56**, 90, **200**, 278
PARAMETRIC_SURFACE_DATA_RECORD, 29, **56**, 78, 81, 85, 90, **200**, 230, 243, 245, 265, 279
PARAMETRIC_SURFACE_INDEX, 28, **57**, 76, 81, 85, 90, **201**, 229, 239, 245, 265, 279
PARAMETRIC_SURFACE_INDICES, **57**, 85, **201**, 265
PATTERN_INDEX, 81, 85, 91, 246, 267, 279
PERFORM, 306
PHIGS, 300, 308
PHIGS generic coordinate system package, 93

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

9593 PT*3 90 ■ 4851903 0567521 T68 ■

PHIGS PLUS, 2
PHIGS_BASE_TYPES, 151
PHIGS_COLOUR_TYPES, 34, 49, 50, 51, 61, 105, **106**, **123**, 128, 129, 130, 131, 151
PHIGS_CONFIGURATION, **110**, 113, 117, 125, 151
PHIGS_COORDINATE_SYSTEM, 61, **135**, **149**, 151, 161, 162
PHIGS_INTEGER, 37, 40, 41, 44, 51, 53, 56, 58, 59, 62, 63, 87, **121**, **125**, 126, **151**, 155, **165**, 169, 171, 173, 174, 175, 178, 182, 188, 190, 200, 203, 204, 209, 210, **273**
PHIGS_LIST_UTILITIES, 37, 38, 40, 44, 46, 51, 53, 56, 57, 58, 59, 62, **113**, **115**, 117, 151, 155, 156, 165, 166, 168, 169, 171, 172, 174, 177, 182, 183, 188, 190, 191, 200, 201, 203, 204, 206, 209
PHIGS_NAME_SET_FACILITY, **117**, 125, 151
PHIGS_NATURAL, 37, 44, 46, 56, 57, 58, 65, 87, 88, 92, **121**, 122, **125**, 126, **151**, 153, 154, 165, 168, 172, 176, 200, 201, 203, 206, 218, 273, 276, 282
PHIGS_PI, 62, **69**, **111**, 190, 304, 305, 306
PHIGS_POSITIVE, 45, 53, 59, 60, 89, 90, 91, 92, **121**, **125**, **151**, 169, 170, 177, 191, 278, 280
PHIGS_PRECISION, 45, 52, 57, 62, **111**, 121, 125, 127, 138, 161, 162, 164, 165, 189, 203
PHIGS_STANDARD_TYPES, 151, 152, 153
PHIGS_STRING, **121**, **125**, **151**, 165, 170, 174
PHIGS_TYPES, **125**, **151**, 300, 308
PICK_ID, 25, 226
POINT_2, 20, 21, 23, 93, **135**, 136, 137, 138, 222, 225
POINT_3, 20, 21, 23, 25, 52, 56, 71, 93, **135**, 136, 137, 138, 190, 191, 200, 201, 222, 225, 227, 236
POINT_4, **93**, **135**, 136
POINT_ARRAY_2, **93**, 102, **136**, 146
POINT_ARRAY_3, **93**, 94, 102, **136**, 137, 146
POINT_ARRAY_4, **93**, 94, **136**, 137
POINT_LIST_2, 94, 104, **135**, 137, 148
POINT_LIST_3, 94, 104, 105, **135**, 136, 137, 148, 149, 301
POINT_LIST_4, **93**, 94, **135**, 136
POLYLINE, **16**, 20, **179**, 221
polyline set 3 with colour, 7, 70
POLYLINE_3, **16**, 20, **179**, 221
POLYLINE_ATTRIBUTES, **16**, **167**
POLYLINE_COLOUR_SHADING, **67**, **206**
POLYLINE_INDEX, 21, 79, 82, 88, 223, 244, 262, 274
POLYLINE_SET, 7, **70**, **236**
POLYLINE_SET_3_WITH_COLOUR, **18**, 25, **181**, 226
POLYLINE_SHADING_METHOD, 28, **57**, 67, 76, 79, 82, 88, **206**, 229, 240, 244, 262, 274
POLYLINE_SHADING_METHODS, **57**, 87, **206**, 273
POLYMARKER, **16**, 20, **179**, 222
POLYMARKER_3, **16**, 20, **179**, 221
POLYMARKER_ATTRIBUTES, **16**, **167**
POLYMARKER_INDEX, 21, 79, 83, 88, 223, 244, 263, 274
POSITIONAL_LIGHT, 52, **68**, **190**
POSITIVE, 33, 40, 47, 48, 54, 55, 62, 64, 65, 93, 95, 113, 115, 122, 135, 136, 137, 138, 139, 193, 194, 215, 218, 219, 301
POST_STRUCTURE, 306, 312
POSTCONCATENATE, **174**
PRECISION, **15**, **166**
PRECONCATENATE, **174**
PSEUDO_COLOUR_MAPPING, 36, 37, 38, **68**, **171**, 172, 173
PSEUDO_N_COLOUR_MAPPING, 36, 37, 38, 68, 171, 172, 173

Q

quadrilateral mesh 3 with data, 7
QUADRILATERAL MESH 3 WITH DATA (with Edge Flags), 73
QUADRILATERAL MESH 3 WITH DATA (without Edge Flags), 73
quadrilateral mesh with data, 7
QUADRILATERAL MESH WITH DATA (with Edge Flags), 74
QUADRILATERAL MESH WITH DATA (without Edge Flags), 74
QUADRILATERAL_MESH, 7
QUADRILATERAL_MESH_3_WITH_DATA, **18**, 27, **181**, 228
QUADRILATERAL_MESH_WITH_DATA, **18**, 27, **73**, **74**, **181**, 228, **238**

R

RANGE_OF_MAGNITUDES, 27, 64, 74, **82**, 86, 87, 91, **139**, 214, 228, 238, 246, 267, 273, 280
RASTER_DISPLAY, **178**
RATIONAL, 34, 35, 44, **62**, 64, 94, 107, **122**, 124, 136, 137, 158, 159, 164, 214
RECTANGULAR_REGION_2, **138**
RECTANGULAR_REGION_3, **137**
REFLECTANCE_ATTRIBUTES, **16**, **167**
REFLECTANCE_COEFFICIENT, **57**, 59, **203**, 204
REFLECTANCE_INDEX, 27, 28, **58**, 75, 81, 84, 90, **203**, 204, 229, 239, 245, 265, 278
REFLECTANCE_INDICES, **58**, 84, **204**, 265
REFLECTANCE_MODEL, 28, 29, **58**, 67, 77, 78, 81, 84, 90, **203**, 230, 242, 245, 265, 278
REFLECTANCE_MODELS, **58**, 90, **203**, 278
REFLECTANCE_PROPERTIES, **15**, **167**
REFLECTANCE_PROPERTIES_DATA_RECORD, 28, 29, **58**, 77, 78, 81, 84, 90, **204**, 230, 242, 245, 265, 278
REFLECTANCE_PROPERTIES_TYPE, 58, **59**, 67, **204**
REFLECTANCE_PROPERTIES_TYPES, **59**, 90, **204**
REJECTED, 117
RELATIVE_MAGNITUDE, 41, 63, **95**, **138**, 175, 176, 211
REMOVE_NAMES, **119**
REMOVE_NAMES_FROM_SET, **18**, 24, **180**, 225
REPLACE, **174**, **178**, 304, 305, 306
RESTORE_MODELLING_CLIPPING_VOLUME, **18**, 24, **180**, 226
RETURN_DEPTH_CUE_INDEX_COUNT, **59**, 91, **177**, 280

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

RETURN_LIGHT_SOURCE_COUNT, **60**, 91, **191**, 280
RETURN_SPLINE_ORDER, **60**, 90, **156**, 278
RETURN_TRIMCURVE_ORDER, **60**, 90, **215**, 278
RETURN_VALUE_TYPE, 82, 83, 84, 85, 86, 262, 263, 264, 265, 267
RETURNED_ELEMENT_POSITION, **178**
RGB, 34, 35, 38, 39, 44, 48, 53, 54, **127**, 131, 132, 133, **152**, 157, 158, 159, 164, 300, 301, 306
RGB_COLOUR_VALUE, 48, **60**, 61, **130**, 131, **153**
RGB_HOMOGENEOUS_COLOUR_VALUE, **61**, **130**, 131, **153**
RGB_TYPES, 35, 39, 54, **61**, **131**, 132, 133, **153**, 159
ROTATE_X, 304, 305, 306
ROTATE_Z, 304, 305, 306

S

SCALE_FACTOR, 46, **121**, **125**, **151**, 168, 177
Scope, 1
set back data mapping index, 7
set back data mapping method, 7
set back interior colour, 7
set back interior index, 7
set back interior shading method, 7
set back interior style, 7
set back interior style index, 7
set back reflectance index, 7
set back reflectance model, 7
set back reflectance properties, 7
set colour mapping index, 7
set colour mapping representation, 7
set curve approximation criteria, 7
set data mapping index, 7
set data mapping method, 7
set data mapping representation, 7
set depth cue index, 7
set depth cue representation, 7
set edge colour, 7
set edge representation plus, 7
set facet culling mode, 7
set facet distinguishing mode, 7
set interior colour, 7
set interior representation plus, 7
set interior shading method, 7
set light source representation, 7
set light source state, 7
set of fill area sets 3 with data, 7
SET OF FILL AREA SETS 3 WITH DATA (with Edge Flags), 71
SET OF FILL AREA SETS 3 WITH DATA (without Edge Flags), 71
set of fill area sets with data, 7
SET OF FILL AREA SETS WITH DATA (with Edge Flags), 72
SET OF FILL AREA SETS WITH DATA (without Edge Flags), 71
set parametric surface characteristics, 7
set parametric surface index, 7
set parametric surface representation, 7
set pattern representation plus, 7
set polyline colour, 7
set polyline representation plus, 7
set polyline shading method, 7
set polymarker colour, 7
set polymarker representation plus, 7
set reflectance index, 7
set reflectance model, 7
set reflectance properties, 7
set reflectance representation, 7
set rendering colour model, 7
set surface approximation criteria, 7
set text colour, 7
set text representation plus, 7
SET_ANNOTATION_STYLE, **17**, 23, **180**, 224
SET_ANNOTATION_TEXT_ALIGNMENT, **17**, 23, **180**, 224
SET_ANNOTATION_TEXT_CHAR_HEIGHT, **17**, 22, **180**, 224
SET_ANNOTATION_TEXT_CHAR_UP_VECTOR, **17**, 23, **180**, 224
SET_ANNOTATION_TEXT_PATH, **17**, 23, **180**, 224
SET_BACK_DATA_MAPPING_INDEX, 7, **19**, 28, **75**, **181**, 229, **239**
SET_BACK_DATA_MAPPING_METHOD, 7, **19**, 29, **78**, **182**, 230, **242**
SET_BACK_INTERIOR_COLOUR, 7, **19**, 29, **77**, **182**, 230, **242**
SET_BACK_INTERIOR_INDEX, 7, **19**, 28, **75**, **181**, 229, **239**
SET_BACK_INTERIOR_SHADING_METHOD, 7, **19**, 29, **77**, **182**, 230, **242**
SET_BACK_INTERIOR_STYLE, 7, **19**, 28, **77**, **182**, 230, **242**
SET_BACK_INTERIOR_STYLE_INDEX, 7, **19**, 29, **77**, **182**, 230, **242**
SET_BACK_REFLECTANCE_INDEX, 7, **19**, 28, **75**, **181**, 229, **239**
SET_BACK_REFLECTANCE_MODEL, 7, **19**, 29, **78**, **182**, 230, **242**
SET_BACK_REFLECTANCE_PROPERTIES, 7, **19**, 29, **78**, **182**, 230, **242**
SET_CHAR_EXPANSION_FACTOR, **17**, 22, **180**, 224
SET_CHAR_HEIGHT, **17**, 22, **180**, 224
SET_CHAR_SPACING, **17**, 22, **180**, 224
SET_CHAR_UP_VECTOR, **17**, 22, **180**, 224
SET_COLOUR_MAPPING_INDEX, 7, **19**, 29, **79**, **182**, 231, **243**
SET_COLOUR_MAPPING_REPRESENTATION, 7, **82**, **246**
SET_COLOUR_REPRESENTATION, 303
SET_CURVE_APPROX_CRITERIA, **19**, 29, **182**, 230
SET_CURVE_APPROXIMATION_CRITERIA, 7, **78**, **243**
SET_DATA_MAPPING_INDEX, 7, **18**, 27, **75**, **181**, 229, **239**
SET_DATA_MAPPING_METHOD, 7, **19**, 28, **77**, **181**, 230, **242**
SET_DATA_MAPPING_REPRESENTATION, 7, **81**, **245**
SET_DEPTH_CUE_INDEX, 7, **19**, 29, **79**, **182**, 231, **243**

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

9593 PT*3 90 ■ 4851903 0567523 830 ■

SET_DEPTH_CUE_REPRESENTATION, 7, **82**, **246**
SET_EDGE_COLOUR, 7, **19**, 29, **78**, **182**, 230, **243**
SET_EDGE_COLOUR_INDEX, **17**, 23, **180**, 225
SET_EDGE_FLAG, **17**, 23, **180**, 224, 311
SET_EDGE_INDEX, **17**, 21, **179**, 223
SET_EDGE_REPRESENTATION, 7, **80**, **245**
SET_EDGETYPE, **17**, 23, **180**, 224
SET_EDGEWIDTH_SCALE_FACTOR, **17**, 23, **180**, 225
SET_FACET_CULLING_MODE, 7, **19**, 28, **76**, **181**, 229, **241**
SET_FACET_DISTINGUISHING_MODE, 7, **19**, 28, **76**, **181**, 229, **241**
SET_GLOBAL_TRANSFORMATION, **18**, 24, **180**, 226
SET_GLOBAL_TRANSFORMATION_3, **18**, 24, **180**, 225
SET_HLHSR_IDENTIFIER, **18**, 24, **180**, 225
SET_HLHSR_MODE, 302
SET_INDIVIDUAL_ASF, **18**, 24, **180**, 225
SET_INTERIOR_COLOUR, 7, **19**, 28, **76**, **181**, 229, **242**
SET_INTERIOR_COLOUR_INDEX, **17**, 23, **180**, 224, 304, 305
SET_INTERIOR_INDEX, **17**, 21, **179**, 223
SET_INTERIOR_REPRESENTATION, 7, **80**, **245**
SET_INTERIOR_SHADING_METHOD, 7, **19**, 28, **77**, **181**, 229, **242**, 306
SET_INTERIOR_STYLE, **17**, 23, **180**, 224, 304, 311
SET_INTERIOR_STYLE_INDEX, **17**, 23, **180**, 224
SET_LIGHT_SOURCE_REPRESENTATION, 7, **81**, **246**, 306
SET_LIGHT_SOURCE_STATE, 7, **19**, 29, **78**, **182**, 230, **242**, 306
SET_LINETYPE, **17**, 22, **179**, 223
SET_LINEWIDTH_SCALE_FACTOR, **17**, 22, **179**, 223
SET_LOCAL_TRANSFORMATION, **18**, 24, **180**, 225, 304, 305, 306
SET_LOCAL_TRANSFORMATION_3, **18**, 24, **180**, 225
SET_MARKER_SIZE_SCALE_FACTOR, **17**, 22, **179**, 223
SET_MARKER_TYPE, **17**, 22, **179**, 223
SET_MODELLING_CLIPPING_INDICATOR, **18**, 24, **180**, 226
SET_MODELLING_CLIPPING_VOLUME, **18**, 24, **180**, 226
SET_MODELLING_CLIPPING_VOLUME_3, **18**, 24, **180**, 226
SET_OF_FILL_AREA_SET_WITH_DATA, **71**, **72**, **237**
SET_OF_FILL_AREA_SETS, 7
SET_OF_FILL_AREA_SETS_3_WITH_DATA, **18**, 26, **181**, 227
SET_OF_FILL_AREA_SETS_WITH_DATA, **18**, 26, **181**, 227
SET_PARAMETRIC_SURFACE_CHARACTERISTICS, 7, **19**, 29, **78**, **182**, 230, **243**, 311
SET_PARAMETRIC_SURFACE_INDEX, 7, **19**, 28, **76**, **181**, 229, **239**
SET_PARAMETRIC_SURFACE_REPRESENTATION, 7, **81**, **245**
SET_PATTERN_REFERENCE_POINT, **18**, 23, **180**, 225
SET_PATTERN_REFERENCE_POINT_AND_VECTORS, **18**, 23, **180**, 225
SET_PATTERN_REPRESENTATION, 7, **81**, **246**
SET_PATTERN_SIZE, **18**, 23, **180**, 225
SET_PICK_IDENTIFIER, **18**, 25, **181**, 226
SET_POLYLINE_COLOUR, 7, **19**, 28, **76**, **181**, 229, **240**
SET_POLYLINE_COLOUR_INDEX, **17**, 22, **179**, 223
SET_POLYLINE_INDEX, **17**, 21, **179**, 223
SET_POLYLINE_REPRESENTATION, 7, **79**, **244**
SET_POLYLINE_SHADING_METHOD, 7, **19**, 28, **76**, **181**, 229, **240**
SET_POLYMARKER_COLOUR, 7, **19**, 28, **76**, **181**, 229, **240**
SET_POLYMARKER_COLOUR_INDEX, **17**, 22, **179**, 223
SET_POLYMARKER_INDEX, **17**, 21, **179**, 223
SET_POLYMARKER_REPRESENTATION, 7, **79**, **244**
SET_REFLECTANCE_INDEX, 7, **18**, 27, **75**, **181**, 229, **239**
SET_REFLECTANCE_MODEL, 7, **19**, 28, **77**, **181**, 230, **242**, 306
SET_REFLECTANCE_PROPERTIES, 7, **19**, 28, **77**, **181**, 230, **242**
SET_REFLECTANCE_REPRESENTATION, 7, **81**, **245**
SET_RENDERING_COLOUR_MODEL, 7, **19**, 29, **79**, **182**, 230, **243**
SET_SURFACE_APPROX_CRITERIA, **19**, 29, **182**, 230
SET_SURFACE_APPROXIMATION_CRITERIA, 7, **78**, **243**, 311
SET_TEXT_ALIGNMENT, **17**, 22, **180**, 224
SET_TEXT_COLOUR, 7, **19**, 28, **76**, **181**, 229, **240**
SET_TEXT_COLOUR_INDEX, **17**, 22, **180**, 224
SET_TEXT_FONT, **17**, 22, **180**, 223
SET_TEXT_INDEX, **17**, 21, **179**, 223
SET_TEXT_PATH, **17**, 22, **180**, 224
SET_TEXT_PRECISION, **17**, 22, **180**, 223
SET_TEXT_REPRESENTATION, 7, **80**, **244**
SET_VIEW_INDEX, **18**, 24, **180**, 226, 306, 311
SET_VIEW_REPRESENTATION, 303, 311
SIMPLE_REFLECTANCE, 58, 59, **67**, **204**
SINGLE_VALUE_NON_UNIFORM, 43, **67**, **155**, 160
SINGLE_VALUE_UNIFORM, 43, **67**, **155**, 160
SIZE, **15**, **166**
SIZE_2, 23, **139**, 225
SIZE_3, **138**
SIZE_OF_LIST, **113**, **115**
SMALL_NATURAL, 42, 44, 61, 64, 95, 96, **122**, 139, 140, **151**, 155, 163, 164, 214, 308
SOURCE_SELECTOR, **61**, **155**
SOURCE_SELECTOR_LIST, 43, **61**, **155**, 160, 161
SPACING, **15**, **166**
SPC, 27, 44, 51, 52, **61**, 64, 74, **162**, 163, 164, 189, 214, 228, 238
SPC_TYPE, 61, **62**, **162**
SPLINE_ORDER, 42, 44, 60, **62**, 64, 65, 95, 96, **122**, 139, 140, **151**, 156, 163, 164, 214, 308

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0567524 777 ■

SPLINE_RATIONALITY, 34, 35, 42, 44, **62**, 64, 93, 94, 95, 96, 107, **122**, 123, 124, 136, 137, 139, 140, **151**, 158, 159, 163, 164, 214
SPOT_LIGHT, 52, **68**, **190**, 191
SPREAD_ANGLE, 52, **62**, **190**, 191
STORAGE_ERROR, 122, 153, 154
STRING, 110
STRING_SMALL_NATURAL, **154**, 170
STRUCTURE_ELEMENT_RECORD, **20**
STRUCTURE_ELEMENT_TYPE, 20, 221
STRUCTURE_ID, 25, 226, 301, 308
STYLE_IND, **15**, **166**
STYLE_INDEX, 23, 29, 77, 80, 83, 89, 224, 230, 242, 245, 264, 276
STYLE_OF_INTERIOR, **15**, **166**
SUPPRESSED, **46**, **177**
SURFACE_APPROX_CRITERA_TYPE, 62, 210
SURFACE_APPROX_CRITERIA_TYPE, **62**, 68, **209**, 210
SURFACE_APPROX_CRITERIA_TYPES, **62**, 90, **209**, 278
SURFACE_APPROX_DATA_RECORD, 29, **62**, 78, 81, 85, 90, **210**, 230, 243, 245, 265, 279
SURFACE_COLOURSPLINE, 32, **64**, 75, 109, **163**, 164, 238, 291
SURFACE_GEOMETRY_SPLINE, 308
SURFACE_GEOMETRYSPLINE, 74, 75, **96**, **140**, 238

T

TEXT, **16**, 20, **179**, 222
TEXT_3, **16**, 20, **179**, 222
TEXT_ALIGNMENT, 22, 23, 224
TEXT_ATTRIBUTES, **16**, **167**
TEXT_COLOUR, **15**, **166**
TEXT_FONT, 22, 80, 83, 88, 223, 244, 263, 275
TEXT_INDEX, 21, 80, 83, 88, 223, 244, 263, 275
TEXT_PATH, 22, 23, 224
TEXT_PRECISION, 22, 80, 83, 88, 223, 244, 263, 275
TRANSFORMATION_MATRIX_2, 24, 225, 226
TRANSFORMATION_MATRIX_3, 24, 225, 300, 301, 310
TRANSLATE, 304, 305
triangle set 3 with data, 7
TRIANGLE SET 3 WITH DATA (with Edge Flags), 72
TRIANGLE SET 3 WITH DATA (without Edge Flags), 72
triangle set with data, 7
TRIANGLE SET WITH DATA (with Edge Flags), 72
TRIANGLE SET WITH DATA (without Edge Flags), 72
triangle strip 3 with data, 7
TRIANGLE STRIP 3 WITH DATA (with Edge Flags), 73
TRIANGLE STRIP 3 WITH DATA (without Edge Flags), 72
triangle strip with data, 7
TRIANGLE STRIP WITH DATA (with Edge Flags), 73
TRIANGLE STRIP WITH DATA (without Edge Flags), 73
TRIANGLE_SET, 7
TRIANGLE_SET_3_WITH_DATA, **18**, 26, **181**, 227
TRIANGLE_SET_WITH_DATA, **18**, 26, **72**, **181**, 227, **237**
TRIANGLE_STRIP, 7
TRIANGLE_STRIP_3_WITH_DATA, **18**, 26, **181**, 228
TRIANGLE_STRIP_WITH_DATA, **18**, 26, **72**, **73**, **181**, 228, **237**, **238**
TRIMCURVE, **64**, **214**, 215, 309
TRIMCURVE_LIST, 32, **64**, 109, **215**, 291, 310
TRIMCURVE_ORDER, 60, 64, **65**, **214**, 215
TRUE_COLOUR_MAPPING, 36, 37, 38, **68**, **171**, 172, 173
TRUNCATION_METHOD, **215**, 253, 285, 286
TYPE_OF_EDGE, **15**, **167**
TYPE_OF_LINE, **14**, **166**
TYPE_OF_MARKER, **15**, **166**

U

UNIFORM, **42**, **176**, 311
UPDATE, **174**
UPDATE_WS, 306, 312

V

VARIABLE_COLOUR_MATRIX, 153
VARIABLE_FILE_ID, 154
VECTOR_2, 22, 23, **137**, 138, 224
VECTOR_3, 52, 56, 95, **137**, 138, 190, 191, 200, 201
VECTOR_ARRAY_3, **95**, 98, 101, **137**, 142, 145
VECTOR_DISPLAY, **178**
VECTOR_LIST_3, **95**, 100, 103, **137**, 144, 147
VECTOR_PAIR_2, **137**
VECTOR_PAIR_3, 20, 23, **137**, 222, 225
VERTEX_COLOUR, **61**, **155**
VERTEX_COLOUR_ARRAY, **100**, 102, **144**, 146
VERTEX_COLOUR_FLAG, **65**, 105, **134**, 149, **153**
VERTEX_COLOUR_LIST, **103**, 104, 105, **147**, 148, 149
VERTEX_COLOUR_LIST_SET_3, **105**, **149**
VERTEX_DATA, **61**, **155**
VERTEX_DATA_ARRAY, **101**, 102, 103, **145**, 146, 147
VERTEX_DATA_ARRAY_SET_2, 74, **102**, 103, **146**, 147, 238
VERTEX_DATA_ARRAY_SET_3, 73, **102**, **146**, 238
VERTEX_DATA_FLAG, 65, 100, 101, 102, 103, 104, **134**, 144, 145, 146, 147, 148, **153**
VERTEX_DATA_LIST, **103**, 104, **147**, 148
VERTEX_DATA_LIST_SET_2, 71, 72, 73, **104**, 105, **148**, 149, 237, 238
VERTEX_DATA_LIST_SET_3, 71, 72, 73, **104**, **148**, 237, 238
VERTEX_INDEX, **65**, **218**
VERTEX_INDEX_LIST, 33, **65**, 66, **218**, 219
VERTEX_INDEX_TRIPLET, 55, **66**, **219**
VERTEX_NORMAL_ARRAY, **101**, 102, 103, **145**, 146, 147
VERTEX_NORMAL_LIST, **103**, 104, **147**, 148
VERTEX_SET_COUNT, **66**, 104, 105, **122**, 148, 149, **151**

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

VERTEX_SET_DIMENSION, 33, 54, 64, **66**, 94, 95, 96, 100, 101, 102, 103, 104, 105, **122**, 126, 136, 137, 139, 140, 144, 145, 146, 147, 148, 149, **151**, 214, 308
VIEW_INDEX, 24, 226, 300
VRC, **162**
VRC_TYPE, **161**, 162

W

WAIT, **176**
WC, 41, 52, 56, 63, **161**, 175, 190, 191, 201, 210, 211
WC_CHORDAL_DEVIATION_CURVE, 41, **67**, **175**
WC_CHORDAL_SIZE_CURVE, 41, **67**, **175**
WC_CHORDAL_SIZE_SURFACE, 63, **68**, **210**
WC_LEVEL_CURVES, 56, **68**, **200**, 201
WC_PLANAR_DEVIATION_SURFACE, 63, **68**, **210**
WC_RELATIVE_CURVE, 41, **68**, **175**
WC_RELATIVE_SURFACE, 63, **68**, **210**, 211
WC_TYPE, **161**
WEIGHT_LIST, 33, **66**, **171**
WEIGHT_LIST_ARRAY, 66, **171**
WS_DEPENDENT_COLOUR_MODEL, **67**, **156**
WS_DEPENDENT_CURVE, 41, **67**, **175**
WS_DEPENDENT_PARAMETRIC_SURFACE, 56, **68**, **200**
WS_DEPENDENT_SURFACE, 63, **68**, **210**
WS_ID, 79, 80, 81, 82, 83, 84, 85, 86, 244, 245, 246, 262, 263, 264, 265, 267, 300, 308
WS_TYPE, 87, 88, 89, 90, 91, 92, 273, 274, 275, 276, 278, 279, 280, 281, 282, 300, 308

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

ICS 35.140.00; 35.060.00

Descriptors: data processing, information interchange, graphic data processing, programming (computers), programming languages, C (programming language).

Price based on 333 pages

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0539835 385 ■

INTERNATIONAL STANDARD ISO/IEC 9593-3:1990
TECHNICAL CORRIGENDUM 1

Published 1993-07-15

INTERNATIONAL ORGANIZATION FOR STANDARDIZATION • МЕЖДУНАРОДНАЯ ОРГАНИЗАЦИЯ ПО СТАНДАРТИЗАЦИИ • ORGANISATION INTERNATIONALE DE NORMALISATION
INTERNATIONAL ELECTROTECHNICAL COMMISSION • МЕЖДУНАРОДНАЯ ЭЛЕКТРОТЕХНИЧЕСКАЯ КОМИССИЯ • COMMISSION ÉLECTROTECHNIQUE INTERNATIONALE

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

Part 3:
Ada

TECHNICAL CORRIGENDUM 1

Technologies de l'information — Infographie — Interfaces langage avec PHIGS —

Partie 3: Ada

RECTIFICATIF TECHNIQUE 1

Technical corrigendum 1 to International Standard ISO/IEC 9593-3:1990 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology.*

Page 22

Change the CHROMATICITY_COEFFICIENT data type to read:

```
type CHROMATICITY_COEFFICIENT is
        record
          UPRIME   : COLOUR_COEFFICIENT;
          VPRIME   : COLOUR_COEFFICIENT;
          Y        : COLOUR_COEFFICIENT;
        end record;
```

UDC 681.3.01:681.3.072.2(084):800.92 **Ref. No. ISO/IEC 9593-3:1990/Cor.1:1993(E)**

Descriptors : data processing, information interchange, graphic data processing, programming (computers), computer programs, programming languages, Ada.

© ISO/IEC 1993

Printed in Switzerland

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS

Not for Resale

9593 PT*3 90 ■ 4851903 0575373 618 ■

INTERNATIONAL STANDARD

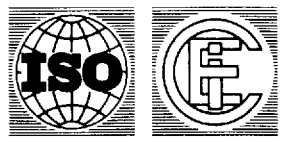
ISO/IEC 9593-3

First edition
1990-04-15

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

Part 3 :
Ada

Technologies de l'information — Infographie — Interfaces langage avec PHIGS — Partie 3 : Ada



Reference number
ISO/IEC 9593-3 : 1990 (E)

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS

Not for Resale

Contents

Foreword iv

Introduction v

1 Scope 1

2 Normative references 2

3 Principles 3

3.1 Conformance 3
3.2 Implications of the Language 3

3.2.1 Functional Mapping 3
3.2.2 Implementation and Host Dependencies 4
3.2.3 Error Handling 4
3.2.4 Data mapping 4
3.2.5 Multi-tasking 6
3.2.6 Packaging 6
3.2.7 Application Program Environment 7
3.2.8 Registration 7

4 Tables 8

4.1 Abbreviations used in procedure names 8

4.1.1 List of procedures using the abbreviations 8
4.1.2 Alphabetical by bound name 11
4.1.3 Alphabetical PHIGS functions 15

4.2 Data type definitions 15

4.2.1 Abbreviations used in the data type definitions 16
4.2.2 Alphabetical list of type definitions 16
4.2.3 Alphabetical List of Private Type Definitions 66
4.2.4 List of Constant Declarations 68
4.2.5 PHIGS Configuration Values 69

4.3 Error Codes 71

4.3.1 Precluded Error Codes 72

© ISO/IEC 1990
All rights reserved. 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

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

5 Functions in the Ada Binding of PHIGS 73

5.1 Control functions 73
5.2 Output primitive functions 74
5.3 Attribute specification functions 77
5.4 Transformation and clipping functions 84
5.5 Structure content functions 91
5.6 Structure manipulation functions 94
5.7 Structure display functions 95
5.8 Structure archive functions 95
5.9 Input functions 98
5.10 Metafile functions 106
5.11 Inquiry functions 107
5.12 Error control functions 132
5.13 Special interface functions 133
5.14 Additional Functions 134

5.14.1 Subprograms for Manipulating Input Data Records 134
5.14.2 PHIGS Generic Coordinate System Package 138
5.14.3 PHIGS Generic List Utility Package 141
5.14.4 PHIGS Name Set Facility Package 144
5.14.5 Deallocation of structure element records 147
5.14.6 Metafile Function Utilities 149

5.15 Conformal Variants 149

Annexes

A Compilable PHIGS Specification 151

B Cross Reference Listing of Implementation Defined Items 243

C Example Programs 245

C.1 Example Program 1: STAR 245
C.2 Example Program 2: IRON 248
C.3 Example Program 3: DYNASTAR 255
C.4 Example Program 4: TRANSFORM_POLYLINE 261
C.5 Example Program 5: SHOW_LINETYPES 269

D PHIGS Multi-Tasking 274

E Index 279

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) together form a system for worldwide standardization as a whole. 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 approval before their acceptance as International Standards. They are approved in accordance with procedures requiring at least 75 % approval by the national bodies voting.

International Standard ISO/IEC 9593-3 was prepared by Technical Committee ISO/IEC JTC 1, *Information technology*.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Introduction

ISO/IEC 9592 is specified in a language independent manner and needs to be embedded in language dependent layers (language bindings) for use with particular programming languages.

The purpose of this part of ISO/IEC 9593 is to define a standard binding of PHIGS to the Ada computer programming language.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

INTERNATIONAL STANDARD ISO/IEC 9593-3 : 1990 (E)

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

Part 3 : Ada

1 Scope

ISO/IEC 9592 specifies a language independent nucleus of a graphics system. For integration into a programming language, PHIGS is 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 Ada computer programming language.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

9593 PT*3 90 ■ 4851903 0575379 036 ■

2 Normative references

The following standards contain provisions which, through reference in this text, constitute provisions of this part of ISO/IEC 9593. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this part of ISO/IEC 9593 are encouraged to investigate the possibility of applying the most recent editions of the standards listed below. Members of IEC and ISO maintain registers of currently valid International Standards.

ISO 8652 : 1987, *Programming languages - Ada (Endorsement of ANSI Standard 1815A-1983).*

ISO/IEC 9592-1 : 1989, *Information processing systems - Computer graphics - Programmer's Hierarchical Interactive Graphics System (PHIGS) - Part 1: Functional description.*

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS

Not for Resale

9593 PT*3 90 ■ 4851903 0575380 858 ■

3 Principles

The PHIGS Binding to Ada is intended to be implementation independent except as it can be adapted by the PHIGS_CONFIGURATION package. The PHIGS Binding to Ada makes no assumptions concerning the availability of implementation dependent facilities defined by the Ada language. It does, however, limit the use of multi-tasking as described in 3.2.5. The Ada compiler shall be able to support the number of declarations contained in this PHIGS Binding to Ada.

This binding does not make any assumptions regarding the machine representation of the predefined Ada numeric types.

This binding assumes that the application programmer will supply an error file name, archive file names, and connection identifiers that are in an acceptable format for the Ada implementation.

This binding makes no assumptions regarding the format of a string specifying an error file name, archive file names, or connection identifiers for devices or metafiles.

3.1 Conformance

This binding incorporates the rules of conformance defined in the PHIGS Standard (ISO/IEC 9592) for PHIGS implementations, with these additional requirements specifically defined for Ada implementations of PHIGS.

The following criteria are established for determining conformance or non-conformance of an implementation to this binding:

- The semantics of an implementation shall be those stated in ISO/IEC 9592 as modified or extended for Ada as stated in this part of ISO/IEC 9593.
- The package corresponding to PHIGS shall be an available Ada library unit, with all names as specified by this part of ISO/IEC 9593.

3.2 Implications of the Language

3.2.1 Functional Mapping

The functions of PHIGS are all mapped to Ada procedures. The mapping utilizes a one-to-one correspondence between the PHIGS functions and Ada procedures except for the PHIGS function INQUIRE TEXT EXTENT. This function is mapped to two overloaded Ada functions each named INQ_TEXT_EXTENT one which

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

supports modelling text and one which supports annotation text. Certain functions required by the binding but not defined by PHIGS are mapped to Ada functions and procedures.

3.2.2 Implementation and Host Dependencies

There are a number of implementation and host dependencies associated with the Ada compiler and runtime system used. These will affect the portability of application programs and their use of PHIGS. The application programmer should follow accepted practices for ensuring portability of Ada programs to avoid introducing problems when rehosting the application on another system. Implementation dependencies include runtime storage management and processor management.

3.2.3 Error Handling

The inquiry functions utilize error indicator parameters for the error returns, and do not raise Ada exceptions. The application program shall ensure that these error indicators are checked before attempting to access other parameters, since Ada implementations are not required to raise an exception if an undefined value is accessed.

The error handling requirements of PHIGS can be summarized as follows:

1. By default, a procedure named ERROR_HANDLING will be provided that simply reports the error by calling ERROR_LOGGING. This is called from the PHIGS function that detects the error.

2. The ERROR_HANDLING procedure may be replaced by one defined by the user.

The procedure ERROR_HANDLING is defined as a library subprogram:

```
with PHIGS_TYPES;
use PHIGS_TYPES;
procedure ERROR_HANDLING
  (ERROR_INDICATOR  : in ERROR_NUMBER;
   PHIGS_FUNCTION   : in STRING;
   ERROR_FILE       : in FILE_ID := DEFAULT_ERROR_FILE);

-- The procedure ERROR_HANDLING is defined as a library subprogram,
-- and is not declared within package PHIGS.
```

This binding defines two different bodies for this subprogram; each shall be supplied by the implementation. The default body is the one required by PHIGS semantics. It simply calls ERROR_LOGGING and returns. The second body calls ERROR_LOGGING and then raises the exception PHIGS_ERROR. The PHIGS function shall be written so as not to handle PHIGS_ERROR (this is a requirement of the implementation). Thus, by Ada rules, the exception will be propagated back to the application program that called the PHIGS function in which the error was detected.

The means by which the user replaces the default body by either the exception-raising version or another one of his or her choosing is dependent upon the Ada library manager. Some implementations support multiple versions of a body with a single specification or otherwise allow hierarchical libraries with the sharing of common units. In other implementations, it may be necessary to duplicate the PHIGS library for each version of ERROR_HANDLING.

3.2.4 Data mapping

The simple and compound data types of PHIGS are bound to a variety of Ada scalar and compound types. Constraints on permitted values are reflected where possible in the type definitions. The general correspondence between the PHIGS data types and Ada binding data types is summarized below:

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

- PHIGS integer types (I) are mapped to Ada integer types.
- PHIGS real types (R) are mapped to Ada floating-point types.
- PHIGS string types (S) are mapped to the predefined Ada type STRING, or to a type providing for variable length strings.
- PHIGS point types (P2, P3) are mapped to Ada record types.
- PHIGS vector types (V2, V3) are mapped to Ada record types.
- PHIGS enumeration types (E) are mapped to Ada enumeration types.
- PHIGS name types (NM) are mapped to an Ada integer type. The PHIGS name set composite type SET(NM) is mapped to an Ada private type. A set of subprograms for operating on objects of this private type is explicitly defined by this binding.
- PHIGS filter types (FR) are mapped to an Ada record type.
- PHIGS pick path item type (PP) is mapped to an Ada record type. PHIGS pick paths are mapped to an Ada array type.
- PHIGS element reference type (ER) is mapped to an Ada record type.
- PHIGS half-space types (HS2, HS3) are mapped to Ada record types.
- PHIGS font/precision pair type (FP) is mapped to an Ada record type.
- PHIGS structure element type (SE) is mapped to an Ada record type.
- PHIGS posted structure type (PS) is mapped to an Ada record type.
- PHIGS bounding range types (B) are mapped to Ada record types.
- PHIGS colour specification type (CLR) is mapped to an Ada record type.
- PHIGS chromaticity coefficient type (CC) is mapped to an Ada record type.
- PHIGS connection identifier type (C) is mapped to the Ada STRING type.
- PHIGS file types (F) are mapped to Ada STRING types.
- PHIGS workstation type (W) is mapped to an Ada integer type.
- PHIGS modelling clipping volume type (MCV) is not used by the binding. Implementations can map this type to an implementation specific private type.
- PHIGS generalized drawing primitive identifier types (G2, G3) are mapped to Ada integer types.
- PHIGS generalized structure element identifier type (GS) is mapped to an Ada integer type.
- PHIGS archive file identifier type (AI) is mapped to an Ada integer type.
- PHIGS pick identifier type (PI) is mapped to an Ada integer type.
- PHIGS escape identifier type (EI) is mapped to an Ada integer type.
- PHIGS function name type (FN) is mapped to an Ada type providing for variable length strings.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

- PHIG workstation identifier type (WI) is mapped to an Ada integer type.
- PHIGS tuples are mapped to Ada record types.
- PHIGS matrices are mapped to Ada array types.
- PHIGS arrays are mapped to either an unconstrained Ada array type, or to a record type providing for variable length arrays.
- PHIGS lists are mapped to an Ada private type declared in the generic PHIGS_LIST_UTILITIES package.
- PHIGS set types are mapped to Ada private types with generic accessing functions provided as necessary.
- PHIGS data records are mapped to Ada private types. In some cases, a set of subprograms for operating on the data records is explicitly defined by this binding. This is because the content and structure of the data record is implementation dependent. An implementation of PHIGS may provide other subprograms for manipulating implementation dependent data records.

Additional types used by the binding are declared as needed in a manner compatible with the PHIGS types.

3.2.5 Multi-tasking

The Ada language definition provides explicit support for concurrency. The Ada tasking model includes facilities for declaring and allocating tasks, and operations allowing intertask communication and synchronization.

The PHIGS standard, and hence this binding standard, neither requires nor prohibits an implementation from protecting against problems which could arise from asynchronous access to the PHIGS data structures from concurrent tasks. Implementors of PHIGS should provide information in the user's documentation regarding whether protection against such problems is implemented.

Annex D contains guidelines for implementors who want to support multi-tasking application programs. This annex does not form an integral part of the binding standard, but provides additional information.

3.2.6 Packaging

The PHIGS standard defines all of its graphic functionality as a cohesive whole. An implementation of PHIGS shall implement the entire functionality of PHIGS. To support this concept, this binding defines a single Ada package which corresponds to all of the PHIGS functionality. This package is named

```
package PHIGS is ... end PHIGS;
```

Associated with this package is a data type package which provides the type declarations as defined in 4.2 and the exception defined in 4.3. This package is

```
package PHIGS_TYPES is ... end PHIGS_TYPES;
```

A minimal program referencing PHIGS is shown below.

```
with PHIGS,
     PHIGS_TYPES;

procedure APPLICATION is
begin
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
    null;
end APPLICATION;
```

Several additional Ada packages are defined in this binding. These packages are

- generic package PHIGS_COORDINATE_SYSTEM
- generic package PHIGS_LIST_UTILITIES
- package PHIGS_NAME_SET_FACILITY
- package PHIGS_CONFIGURATION

These packages support the declaration types in the PHIGS_TYPES package described above. PHIGS_COORDINATE_SYSTEM is a generic package which defines an assortment of types supporting each of the PHIGS coordinate systems. PHIGS_LIST_UTILITIES is also a generic package which provides type declarations and operations for list types which correspond to the PHIGS list types. The PHIGS_NAME_SET_FACILITY package defines private types for name sets and provides functions for building and manipulating name sets. The PHIGS_CONFIGURATION package sets implementation dependent limit values for various types defined in this binding. The PHIGS_CONFIGURATION package also provides for possible application modification of these limits.

3.2.7 Application Program Environment

An application program utilizing an Ada implementation of PHIGS will need to be aware of the environment in which both PHIGS and the application program(s) reside.

One aspect of the environment is the Ada program library. The Ada language requires that the application program have access to the program library in which the PHIGS software resides. The Ada standard ISO 8652 does not specify whether there is a single library or multiple libraries, or how access to the libraries is granted or managed. The user's documentation for the PHIGS implementation should specify where the PHIGS library exists in the system, and how access to the library is acquired.

Input/Output interfaces are also implementation-dependent, and are required to be described in the user's documentation. Besides the obvious graphics device interface information, interfaces to the file system shall be included in the documentation. Specifically, this includes the interface to the PHIGS error file, archive files, and metafile storage.

Subclause 3.2.2 discusses implications about the application program environment which may also apply to application programs.

3.2.8 Registration[1]

PHIGS reserves certain value ranges for registration as graphical items. The registered graphical items will be bound to Ada (and other programming languages). The registered item bindings will be consistent with the binding presented in this document.

---

1) For the purpose of this International Standard and according to the rules for the designation and operation of registration authorities in the ISO/IEC JTC 1 procedures, the ISO and IEC Councils have designated the National Institute of Standards and Technology (National Computer Systems Laboratory) A-266 Technology Building, Gaithersburg, MD 20899, USA, to act as registration authority.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0575385 33T ■

4 Tables

4.1 Abbreviations used in procedure names

ASF	aspect source flag
CHAR	character
ESC	escape
GDP	generalized drawing primitive
GSE	generalized structure element
HLHSR	hidden line/hidden surface removal
INQ	inquire
PHIGS	Programmer's Hierarchical Interactive Graphics System
U	unregistered
WS	workstation

4.1.1 List of procedures using the abbreviations

ASF	SET_INDIVIDUAL_ASF
CHAR	SET_ANNOTATION_TEXT_CHAR_HEIGHT SET_ANNOTATION_TEXT_CHAR_UP_VECTOR SET_CHAR_EXPANSION_FACTOR SET_CHAR_HEIGHT SET_CHAR_SPACING SET_CHAR_UP_VECTOR
ESC	PHIGS_ESCAPE.GENERALIZED_ESC PHIGS_ESC_<name of the escape procedure>.ESC PHIGS_UESC_<name of the escape procedure>.ESC
GDP	INQ_GDP (2D) INQ_GDP (3D) INQ_LIST_OF_AVAILABLE_GDP (2D) INQ_LIST_OF_AVAILABLE_GDP (3D) PHIGS_GDP.GENERALIZED_GDP PHIGS_GDP_<name of the GDP procedure>.GDP PHIGS_UGDP_<name of the GDP procedure>.GDP
GSE	INQ_GSE_FACILITIES INQ_LIST_OF_AVAILABLE_GSE PHIGS_GSE.GENERALIZED_GSE PHIGS_GSE_<name of the GSE procedure>.GSE PHIGS_UGSE_<name of the GSE procedure>.GSE
HLHSR	INQ_HLHSR_FACILITIES INQ_HLHSR_MODE SET_HLHSR_IDENTIFIER SET_HLHSR_MODE
INQ	INQ_ALL_CONFLICTING_STRUCTURES

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

9593 PT*3 90 ■ 4851903 0575386 276 ■

INQ_ANNOTATION_FACILITIES
INQ_ARCHIVE_FILES
INQ_ARCHIVE_STATE_VALUE
INQ_CHOICE_DEVICE_STATE (2D)
INQ_CHOICE_DEVICE_STATE (3D)
INQ_COLOUR_FACILITIES
INQ_COLOUR_MODEL
INQ_COLOUR_MODEL_FACILITIES
INQ_COLOUR_REPRESENTATION
INQ_CONFLICT_RESOLUTION
INQ_CONFLICTING_STRUCTURES_IN_NETWORK
INQ_CURRENT_ELEMENT_CONTENT
INQ_CURRENT_ELEMENT_TYPE_AND_SIZE
INQ_DEFAULT_CHOICE_DEVICE_DATA (2D)
INQ_DEFAULT_CHOICE_DEVICE_DATA (3D)
INQ_DEFAULT_DISPLAY_UPDATE_STATE
INQ_DEFAULT_LOCATOR_DEVICE_DATA (2D)
INQ_DEFAULT_LOCATOR_DEVICE_DATA (3D)
INQ_DEFAULT_PICK_DEVICE_DATA (2D)
INQ_DEFAULT_PICK_DEVICE_DATA (3D)
INQ_DEFAULT_STRING_DEVICE_DATA (2D)
INQ_DEFAULT_STRING_DEVICE_DATA (3D)
INQ_DEFAULT_STROKE_DEVICE_DATA (2D)
INQ_DEFAULT_STROKE_DEVICE_DATA (3D)
INQ_DEFAULT_VALUATOR_DEVICE_DATA (2D)
INQ_DEFAULT_VALUATOR_DEVICE_DATA (3D)
INQ_DISPLAY_SPACE_SIZE (2D)
INQ_DISPLAY_SPACE_SIZE (3D)
INQ_DISPLAY_UPDATE_STATE
INQ_DYNAMICS_OF_STRUCTURES
INQ_DYNAMICS_OF_WS_ATTRIBUTES
INQ_EDGE_FACILITIES
INQ_EDGE_REPRESENTATION
INQ_EDIT_MODE
INQ_ELEMENT_CONTENT
INQ_ELEMENT_POINTER
INQ_ELEMENT_TYPE_AND_SIZE
INQ_ERROR_HANDLING_MODE
INQ_GDP (2D)
INQ_GDP (3D)
INQ_GSE_FACILITIES
INQ_HIGHLIGHTING_FILTER
INQ_HLHSR_FACILITIES
INQ_HLHSR_MODE
INQ_INPUT_QUEUE_OVERFLOW
INQ_INTERIOR_FACILITIES
INQ_INTERIOR_REPRESENTATION
INQ_INVISIBILITY_FILTER
INQ_LIST_OF_AVAILABLE_GDP (2D)
INQ_LIST_OF_AVAILABLE_GDP (3D)
INQ_LIST_OF_AVAILABLE_GSE
INQ_LIST_OF_AVAILABLE_WS_TYPES
INQ_LIST_OF_COLOUR_INDICES
INQ_LIST_OF_EDGE_INDICES
INQ_LIST_OF_INTERIOR_INDICES
INQ_LIST_OF_PATTERN_INDICES
INQ_LIST_OF_POLYLINE_INDICES
INQ_LIST_OF_POLYMARKER_INDICES
INQ_LIST_OF_TEXT_INDICES
INQ_LIST_OF_VIEW_INDICES
INQ_LOCATOR_DEVICE_STATE (2D)
INQ_LOCATOR_DEVICE_STATE (3D)
INQ_MODELLING_CLIPPING_FACILITIES
INQ_MORE_SIMULTANEOUS_EVENTS
INQ_NUMBER_OF_AVAILABLE_LOGICAL_INPUT_DEVICES
INQ_NUMBER_OF_DISPLAY_PRIORITIES_SUPPORTED
INQ_OPEN_STRUCTURE
INQ_PATHS_TO_ANCESTORS
INQ_PATHS_TO_DESCENDANTS
INQ_PATTERN_FACILITIES
INQ_PATTERN_REPRESENTATION

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0575387 102 ■

	INQ_PHIGS_FACILITIES
	INQ_PICK_DEVICE_STATE (2D)
	INQ_PICK_DEVICE_STATE (3D)
	INQ_POLYLINE_FACILITIES
	INQ_POLYLINE_REPRESENTATION
	INQ_POLYMARKER_FACILITIES
	INQ_POLYMARKER_REPRESENTATION
	INQ_POSTED_STRUCTURES
	INQ_PREDEFINED_COLOUR_REPRESENTATION
	INQ_PREDEFINED_EDGE_REPRESENTATION
	INQ_PREDEFINED_INTERIOR_REPRESENTATION
	INQ_PREDEFINED_PATTERN_REPRESENTATION
	INQ_PREDEFINED_POLYLINE_REPRESENTATION
	INQ_PREDEFINED_POLYMARKER_REPRESENTATION
	INQ_PREDEFINED_TEXT_REPRESENTATION
	INQ_PREDEFINED_VIEW_REPRESENTATION
	INQ_SET_OF_OPEN_WS
	INQ_SET_OF_WS_TO_WHICH_POSTED
	INQ_STRING_DEVICE_STATE (2D)
	INQ_STRING_DEVICE_STATE (3D)
	INQ_STROKE_DEVICE_STATE (2D)
	INQ_STROKE_DEVICE_STATE (3D)
	INQ_STRUCTURE_IDENTIFIERS
	INQ_STRUCTURE_STATE_VALUE
	INQ_STRUCTURE_STATUS
	INQ_SYSTEM_STATE_VALUE
	INQ_TEXT_EXTENT (text)
	INQ_TEXT_EXTENT (annotation text)
	INQ_TEXT_FACILITIES
	INQ_TEXT_REPRESENTATION
	INQ_VALUATOR_DEVICE_STATE (2D)
	INQ_VALUATOR_DEVICE_STATE (3D)
	INQ_VIEW_FACILITIES
	INQ_VIEW_REPRESENTATION
	INQ_WS_CATEGORY
	INQ_WS_CLASSIFICATION
	INQ_WS_CONNECTION_AND_TYPE
	INQ_WS_STATE_TABLE_LENGTHS
	INQ_WS_STATE_VALUE
	INQ_WS_TRANSFORMATION (2D)
	INQ_WS_TRANSFORMATION (3D)
PHIGS	CLOSE_PHIGS
	EMERGENCY_CLOSE_PHIGS
	INQ_PHIGS_FACILITIES
	OPEN_PHIGS
U	PHIGS_UESC_<name of the escape procedure>.ESC
	PHIGS_UGDP_<name of the GDP procedure>.GDP
	PHIGS_UGSE_<name of the GSE procedure>.GSE
WS	CLOSE_WS
	INQ_DYNAMICS_OF_WS_ATTRIBUTES
	INQ_LIST_OF_AVAILABLE_WS_TYPES
	INQ_SET_OF_OPEN_WS
	INQ_SET_OF_WS_TO_WHICH_POSTED
	INQ_WS_CATEGORY
	INQ_WS_CLASSIFICATION
	INQ_WS_CONNECTION_AND_TYPE
	INQ_WS_STATE_TABLE_LENGTHS
	INQ_WS_STATE_VALUE
	INQ_WS_TRANSFORMATION (2D)
	INQ_WS_TRANSFORMATION (3D)
	OPEN_WS
	SET_WS_VIEWPORT (2D)
	SET_WS_VIEWPORT (3D)
	SET_WS_WINDOW (2D)
	SET_WS_WINDOW (3D)
	UPDATE_WS

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0575388 049 ■

4.1.2 Alphabetical by bound name

This subclause alphabetically lists the Ada procedures corresponding to each PHIGS function. Instances in which the binding of a name corresponds to more than one PHIGS function indicates the use of Ada overloading.

ADD_NAMES_TO_SET	add names to set
ANNOTATION_TEXT_RELATIVE	annotation text relative
ANNOTATION_TEXT_RELATIVE	annotation text relative 3
APPLICATION_DATA	application data
ARCHIVE_ALL_STRUCTURES	archive all structures
ARCHIVE_STRUCTURE_NETWORKS	archive structure networks
ARCHIVE_STRUCTURES	archive structures
AWAIT_EVENT	await event
BUILD_TRANSFORMATION_MATRIX	build transformation matrix
BUILD_TRANSFORMATION_MATRIX	build transformation matrix 3
CELL_ARRAY	cell array
CELL_ARRAY	cell array 3
CHANGE_STRUCTURE_IDENTIFIER	change structure identifier
CHANGE_STRUCTURE_IDENTIFIER_AND_REFERENCES	change structure identifier and references
CHANGE_STRUCTURE_REFERENCES	change structure references
CLOSE_ARCHIVE_FILE	close archive file
CLOSE_PHIGS	close PHIGS
CLOSE_STRUCTURE	close structure
CLOSE_WS	close workstation
COMPOSE_MATRIX	compose matrix
COMPOSE_MATRIX	compose matrix 3
COMPOSE_TRANSFORMATION_MATRIX	compose transformation matrix
COMPOSE_TRANSFORMATION_MATRIX	compose transformation matrix 3
COPY_ALL_ELEMENTS_FROM_STRUCTURE	copy all elements from structure
DELETE_ALL_STRUCTURES	delete all structures
DELETE_ALL_STRUCTURES_FROM_ARCHIVE	delete all structures from archive
DELETE_ELEMENT	delete element
DELETE_ELEMENT_RANGE	delete element range
DELETE_ELEMENTS_BETWEEN_LABELS	delete elements between labels
DELETE_STRUCTURE	delete structure
DELETE_STRUCTURE_NETWORK	delete structure network
DELETE_STRUCTURE_NETWORKS_FROM_ARCHIVE	delete structure networks from archive
DELETE_STRUCTURES_FROM_ARCHIVE	delete structures from archive
ELEMENT_SEARCH	element search
EMERGENCY_CLOSE_PHIGS	emergency close PHIGS
EMPTY_STRUCTURE	empty structure
ERROR_HANDLING	error handling
ERROR_LOGGING	error logging
ESCAPE	escape
EVALUATE_VIEW_MAPPING_MATRIX	evaluate view mapping matrix
EVALUATE_VIEW_MAPPING_MATRIX	evaluate view mapping matrix 3
EVALUATE_VIEW_ORIENTATION_MATRIX	evaluate view orientation matrix
EVALUATE_VIEW_ORIENTATION_MATRIX	evaluate view orientation matrix 3
EXECUTE_STRUCTURE	execute structure
FILL_AREA	fill area
FILL_AREA	fill area 3
FILL_AREA_SET	fill area set
FILL_AREA_SET	fill area set 3
FLUSH_DEVICE_EVENTS	flush device events
GENERALIZED_GDP	generalized drawing primitive
GENERALIZED_GDP	generalized drawing primitive 3
GENERALIZED_GSE	generalized structure element
GET_CHOICE	get choice
GET_ITEM_TYPE_FROM_METAFILE	get item type from metafile
GET_LOCATOR	get locator
GET_LOCATOR	get locator 3
GET_PICK	get pick
GET_STRING	get string
GET_STROKE	get stroke

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

GET_STROKE	get stroke 3
GET_VALUATOR	get valuator
INCREMENTAL_SPATIAL_SEARCH	incremental spatial search
INCREMENTAL_SPATIAL_SEARCH	incremental spatial search 3
INITIALIZE_CHOICE	initialize choice
INITIALIZE_CHOICE	initialize choice 3
INITIALIZE_LOCATOR	initialize locator
INITIALIZE_LOCATOR	initialize locator 3
INITIALIZE_PICK	initialize pick
INITIALIZE_PICK	initialize pick 3
INITIALIZE_STRING	initialize string
INITIALIZE_STRING	initialize string 3
INITIALIZE_STROKE	initialize stroke
INITIALIZE_STROKE	initialize stroke 3
INITIALIZE_VALUATOR	initialize valuator
INITIALIZE_VALUATOR	initialize valuator 3
INQ_ALL_CONFLICTING_STRUCTURES	inquire all conflicting structures
INQ_ANNOTATION_FACILITIES	inquire annotation facilities
INQ_ARCHIVE_FILES	inquire archive files
INQ_ARCHIVE_STATE_VALUE	inquire archive state value
INQ_CHOICE_DEVICE_STATE	inquire choice device state
INQ_CHOICE_DEVICE_STATE	inquire choice device state 3
INQ_COLOUR_FACILITIES	inquire colour facilities
INQ_COLOUR_MODEL	inquire colour model
INQ_COLOUR_MODEL_FACILITIES	inquire colour model facilities
INQ_COLOUR_REPRESENTATION	inquire colour representation
INQ_CONFLICT_RESOLUTION	inquire conflict resolution
INQ_CONFLICTING_STRUCTURES_IN_NETWORK	inquire conflicting structures in network
INQ_CURRENT_ELEMENT_CONTENT	inquire current element content
INQ_CURRENT_ELEMENT_TYPE_AND_SIZE	inquire current element type and size
INQ_DEFAULT_CHOICE_DEVICE_DATA	inquire default choice device data
INQ_DEFAULT_CHOICE_DEVICE_DATA	inquire default choice device data 3
INQ_DEFAULT_DISPLAY_UPDATE_STATE	inquire default display update state
INQ_DEFAULT_LOCATOR_DEVICE_DATA	inquire default locator device data
INQ_DEFAULT_LOCATOR_DEVICE_DATA	inquire default locator device data 3
INQ_DEFAULT_PICK_DEVICE_DATA	inquire default pick device data
INQ_DEFAULT_PICK_DEVICE_DATA	inquire default pick device data 3
INQ_DEFAULT_STRING_DEVICE_DATA	inquire default string device data
INQ_DEFAULT_STRING_DEVICE_DATA	inquire default string device data 3
INQ_DEFAULT_STROKE_DEVICE_DATA	inquire default stroke device data
INQ_DEFAULT_STROKE_DEVICE_DATA	inquire default stroke device data 3
INQ_DEFAULT_VALUATOR_DEVICE_DATA	inquire default valuator device data
INQ_DEFAULT_VALUATOR_DEVICE_DATA	inquire default valuator device data 3
INQ_DISPLAY_SPACE_SIZE	inquire display space size
INQ_DISPLAY_SPACE_SIZE	inquire display space size 3
INQ_DISPLAY_UPDATE_STATE	inquire display update state
INQ_DYNAMICS_OF_STRUCTURES	inquire dynamics of structures
INQ_DYNAMICS_OF_WS_ATTRIBUTES	inquire dynamics of workstation attributes
INQ_EDGE_FACILITIES	inquire edge facilities
INQ_EDGE_REPRESENTATION	inquire edge representation
INQ_EDIT_MODE	inquire edit mode
INQ_ELEMENT_CONTENT	inquire element content
INQ_ELEMENT_POINTER	inquire element pointer
INQ_ELEMENT_TYPE_AND_SIZE	inquire element type and size
INQ_ERROR_HANDLING_MODE	inquire error handling mode
INQ_GDP	inquire generalized drawing primitives
INQ_GDP	inquire generalized drawing primitives 3
INQ_GSE_FACILITIES	inquire generalized structure element facilities
INQ_HIGHLIGHTING_FILTER	inquire highlighting filter
INQ_HLHSR_FACILITIES	inquire hlhsr facilities
INQ_HLHSR_MODE	inquire hlhsr mode
INQ_INPUT_QUEUE_OVERFLOW	inquire input queue overflow
INQ_INTERIOR_FACILITIES	inquire interior facilities
INQ_INTERIOR_REPRESENTATION	inquire interior representation
INQ_INVISIBILITY_FILTER	inquire invisibility filter
INQ_LIST_OF_AVAILABLE_GDP	inquire list of available generalized drawing primitives
INQ_LIST_OF_AVAILABLE_GDP	inquire list of available generalized drawing primitives 3
INQ_LIST_OF_AVAILABLE_GSE	inquire list of available generalized structure elements
INQ_LIST_OF_AVAILABLE_WS_TYPES	inquire list of available workstation types
INQ_LIST_OF_COLOUR_INDICES	inquire list of colour indices
INQ_LIST_OF_EDGE_INDICES	inquire list of edge indices

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

9593 PT*3 90 ■ 4851903 0575390 7T7 ■

INQ_LIST_OF_INTERIOR_INDICES	inquire list of interior indices
INQ_LIST_OF_PATTERN_INDICES	inquire list of pattern indices
INQ_LIST_OF_POLYLINE_INDICES	inquire list of polyline indices
INQ_LIST_OF_POLYMARKER_INDICES	inquire list of polymarker indices
INQ_LIST_OF_TEXT_INDICES	inquire list of text indices
INQ_LIST_OF_VIEW_INDICES	inquire list of view indices
INQ_LOCATOR_DEVICE_STATE	inquire locator device state
INQ_LOCATOR_DEVICE_STATE	inquire locator device state 3
INQ_MODELLING_CLIPPING_FACILITIES	inquire modelling clipping facilities
INQ_MORE_SIMULTANEOUS_EVENTS	inquire more simultaneous events
INQ_NUMBER_OF_AVAILABLE_LOGICAL_INPUT_DEVICES	inquire number of available logical input devices
INQ_NUMBER_OF_DISPLAY_PRIORITIES_SUPPORTED	inquire number of structure priorities supported
INQ_OPEN_STRUCTURE	inquire open structure
INQ_PATHS_TO_ANCESTORS	inquire paths to ancestors
INQ_PATHS_TO_DESCENDANTS	inquire paths to descendants
INQ_PATTERN_FACILITIES	inquire pattern facilities
INQ_PATTERN_REPRESENTATION	inquire pattern representation
INQ_PHIGS_FACILITIES	inquire phigs facilities
INQ_PICK_DEVICE_STATE	inquire pick device state
INQ_PICK_DEVICE_STATE	inquire pick device state 3
INQ_POLYLINE_FACILITIES	inquire polyline facilities
INQ_POLYLINE_REPRESENTATION	inquire polyline representation
INQ_POLYMARKER_FACILITIES	inquire polymarker facilities
INQ_POLYMARKER_REPRESENTATION	inquire polymarker representation
INQ_POSTED_STRUCTURES	inquire posted structures
INQ_PREDEFINED_COLOUR_REPRESENTATION	inquire predefined colour representation
INQ_PREDEFINED_EDGE_REPRESENTATION	inquire predefined edge representation
INQ_PREDEFINED_INTERIOR_REPRESENTATION	inquire predefined interior representation
INQ_PREDEFINED_PATTERN_REPRESENTATION	inquire predefined pattern representation
INQ_PREDEFINED_POLYLINE_REPRESENTATION	inquire predefined polyline representation
INQ_PREDEFINED_POLYMARKER_REPRESENTATION	inquire predefined polymarker representation
INQ_PREDEFINED_TEXT_REPRESENTATION	inquire predefined text representation
INQ_PREDEFINED_VIEW_REPRESENTATION	inquire predefined view representation
INQ_SET_OF_OPEN_WS	inquire set of open workstations
INQ_SET_OF_WS_TO_WHICH_POSTED	inquire set of workstations to which posted
INQ_STRING_DEVICE_STATE	inquire string device state
INQ_STRING_DEVICE_STATE	inquire string device state 3
INQ_STROKE_DEVICE_STATE	inquire stroke device state
INQ_STROKE_DEVICE_STATE	inquire stroke device state 3
INQ_STRUCTURE_IDENTIFIERS	inquire structure identifiers
INQ_STRUCTURE_STATE_VALUE	inquire structure state value
INQ_STRUCTURE_STATUS	inquire structure status
INQ_SYSTEM_STATE_VALUE	inquire system state value
INQ_TEXT_EXTENT (text)	inquire text extent
INQ_TEXT_EXTENT (annotation text)	inquire text extent
INQ_TEXT_FACILITIES	inquire text facilities
INQ_TEXT_REPRESENTATION	inquire text representation
INQ_VALUATOR_DEVICE_STATE	inquire valuator device state
INQ_VALUATOR_DEVICE_STATE	inquire valuator device state 3
INQ_VIEW_FACILITIES	inquire view facilities
INQ_VIEW_REPRESENTATION	inquire view representation
INQ_WS_CATEGORY	inquire workstation category
INQ_WS_CLASSIFICATION	inquire workstation classification
INQ_WS_CONNECTION_AND_TYPE	inquire workstation connection and type
INQ_WS_STATE_TABLE_LENGTHS	inquire workstation state table lengths
INQ_WS_STATE_VALUE	inquire workstation state value
INQ_WS_TRANSFORMATION	inquire workstation transformation
INQ_WS_TRANSFORMATION	inquire workstation transformation 3
INTERPRET_ITEM	interpret item
LABEL	label
MESSAGE	message
OFFSET_ELEMENT_POINTER	offset element pointer
OPEN_ARCHIVE_FILE	open archive fill
OPEN_PHIGS	open PHIGS
OPEN_STRUCTURE	open structure
OPEN_WS	open workstation
POLYLINE	polyline
POLYLINE	polyline 3
POLYMARKER	polymarker
POLYMARKER	polymarker 3
POST_STRUCTURE	post structure

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

READ_ITEM_FROM_METAFILE	read item from metafile
REDRAW_ALL_STRUCTURES	redraw all structures
REMOVE_NAMES_FROM_SET	remove names from set
REQUEST_CHOICE	request choice
REQUEST_LOCATOR	request locator
REQUEST_LOCATOR	request locator 3
REQUEST_PICK	request pick
REQUEST_STRING	request string
REQUEST_STROKE	request stroke
REQUEST_STROKE	request stroke 3
REQUEST_VALUATOR	request valuator
RESTORE_MODELLING_CLIPPING_VOLUME	restore modelling clipping volume
RETRIEVE_ALL_STRUCTURES	retrieve all structures
RETRIEVE_PATHS_TO_ANCESTORS	retrieve paths to ancestors
RETRIEVE_PATHS_TO_DESCENDANTS	retrieve paths to descendants
RETRIEVE_STRUCTURE_IDENTIFIERS	retrieve structure identifiers
RETRIEVE_STRUCTURE_NETWORKS	retrieve structure networks
RETRIEVE_STRUCTURES	retrieve structures
ROTATE	rotate
ROTATE_X	rotate x
ROTATE_Y	rotate y
ROTATE_Z	rotate z
SAMPLE_CHOICE	sample choice
SAMPLE_LOCATOR	sample locator
SAMPLE_LOCATOR	sample locator 3
SAMPLE_PICK	sample pick
SAMPLE_STRING	sample string
SAMPLE_STROKE	sample stroke
SAMPLE_STROKE	sample stroke 3
SAMPLE_VALUATOR	sample valuator
SCALE	scale
SCALE	scale 3
SET_ANNOTATION_STYLE	set annotation style
SET_ANNOTATION_TEXT_ALIGNMENT	set annotation text alignment
SET_ANNOTATION_TEXT_CHAR_HEIGHT	set annotation text character height
SET_ANNOTATION_TEXT_CHAR_UP_VECTOR	set annotation text character up vector
SET_ANNOTATION_TEXT_PATH	set annotation text path
SET_CHAR_EXPANSION_FACTOR	set character expansion factor
SET_CHAR_HEIGHT	set character height
SET_CHAR_SPACING	set character spacing
SET_CHAR_UP_VECTOR	set character up vector
SET_CHOICE_MODE	set choice mode
SET_COLOUR_MODEL	set colour model
SET_COLOUR_REPRESENTATION	set colour representation
SET_CONFLICT_RESOLUTION	set conflict resolution
SET_DISPLAY_UPDATE_STATE	set display update state
SET_EDGE_COLOUR_INDEX	set edge colour index
SET_EDGE_FLAG	set edge flag
SET_EDGE_INDEX	set edge index
SET_EDGE_REPRESENTATION	set edge representation
SET_EDGETYPE	set edgetype
SET_EDGEWIDTH_SCALE_FACTOR	set edgewidth scale factor
SET_EDIT_MODE	set edit mode
SET_ELEMENT_POINTER	set element pointer
SET_ELEMENT_POINTER_AT_LABEL	set element pointer at label
SET_ERROR_HANDLING_MODE	set error handling mode
SET_GLOBAL_TRANSFORMATION	set global transformation
SET_GLOBAL_TRANSFORMATION	set global transformation 3
SET_HIGHLIGHTING_FILTER	set highlighting filter
SET_HLHSR_IDENTIFIER	set hlhsr identifier
SET_HLHSR_MODE	set hlhsr mode
SET_INDIVIDUAL_ASF	set individual asf
SET_INTERIOR_COLOUR_INDEX	set interior colour index
SET_INTERIOR_INDEX	set interior index
SET_INTERIOR_REPRESENTATION	set interior representation
SET_INTERIOR_STYLE	set interior style
SET_INTERIOR_STYLE_INDEX	set interior style index
SET_INVISIBILITY_FILTER	set invisibility filter
SET_LINETYPE	set linetype
SET_LINEWIDTH_SCALE_FACTOR	set linewidth scale factor
SET_LOCAL_TRANSFORMATION	set local transformation

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

SET_LOCAL_TRANSFORMATION	set local transformation 3
SET_LOCATOR_MODE	set locator mode
SET_MARKER_SIZE_SCALE_FACTOR	set marker size scale factor
SET_MARKER_TYPE	set marker type
SET_MODELLING_CLIPPING_INDICATOR	set modelling clipping indicator
SET_MODELLING_CLIPPING_VOLUME	set modelling clipping volume
SET_MODELLING_CLIPPING_VOLUME	set modelling clipping volume 3
SET_PATTERN_REFERENCE_POINT	set pattern reference point
SET_PATTERN_REFERENCE_POINT_AND_VECTORS	set pattern reference point and vectors
SET_PATTERN_REPRESENTATION	set pattern representation
SET_PATTERN_SIZE	set pattern size
SET_PICK_FILTER	set pick filter
SET_PICK_IDENTIFIER	set pick identifier
SET_PICK_MODE	set pick mode
SET_POLYLINE_COLOUR_INDEX	set polyline colour index
SET_POLYLINE_INDEX	set polyline index
SET_POLYLINE_REPRESENTATION	set polyline representation
SET_POLYMARKER_COLOUR_INDEX	set polymarker colour index
SET_POLYMARKER_INDEX	set polymarker index
SET_POLYMARKER_REPRESENTATION	set polymarker representation
SET_STRING_MODE	set string mode
SET_STROKE_MODE	set stroke mode
SET_TEXT_ALIGNMENT	set text alignment
SET_TEXT_COLOUR_INDEX	set text colour index
SET_TEXT_FONT	set text font
SET_TEXT_INDEX	set text index
SET_TEXT_PATH	set text path
SET_TEXT_PRECISION	set text precision
SET_TEXT_REPRESENTATION	set text representation
SET_VALUATOR_MODE	set valuator mode
SET_VIEW_INDEX	set view index
SET_VIEW_REPRESENTATION	set view representation
SET_VIEW_REPRESENTATION	set view representation 3
SET_VIEW_TRANSFORMATION_INPUT_PRIORITY	set view transformation input priority
SET_WS_VIEWPORT	set workstation viewport
SET_WS_VIEWPORT	set workstation viewport 3
SET_WS_WINDOW	set workstation window
SET_WS_WINDOW	set workstation window 3
TEXT	text
TEXT	text 3
TRANSFORM_POINT	transform point
TRANSFORM_POINT	transform point 3
TRANSLATE	translate
TRANSLATE	translate 3
UNPOST_ALL_STRUCTURES	unpost all structures
UNPOST_STRUCTURE	unpost structure
UPDATE_WS	update workstation
WRITE_ITEM_TO_METAFILE	write item to metafile

4.1.3 Alphabetical PHIGS functions

The functions are in the same order when listed alphabetically according to the PHIGS function names as they are when listed alphabetically according to the bound names. Therefore, the table provided in 4.1.2 alphabetically lists the PHIGS functions.

4.2 Data type definitions

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

4.2.1 Abbreviations used in the data type definitions

ARCL	archive closed
AROP	archive open
ASAP	as soon as possible
ASF	aspect source flag
ASTI	at some time
BNIG	before next interaction globally
BNIL	before next interaction locally
CBS	can be simulated
CHAR	character
CSS	central structure store
DC	device coordinate
GDP	generalized drawing primitive
GSE	generalized structure element
HLHSR	hidden line/hidden surface removal
ID	identifier
IDS	identifiers
IMM	immediate
IND	index
IRG	implicit regeneration
MAX	maximum
MC	modelling coordinates
MI	metafile input
MIN	minimum
MISC	miscellaneous
MO	metafile output
NIVE	no immediate visual effect
NPC	normalized projection coordinates
PHCL	PHIGS closed
PHIGS	Programmer's Hierarchical Interactive Graphics System
PHOP	PHIGS open
PT	point
REF	reference
SF	scale factor
STCL	structure closed
STOP	structure open
UQUM	use quick update method
UWOR	update without regeneration
VC	viewing coordinates
WC	world coordinates
WS	workstation
WSCL	workstation closed
WSOP	workstation open

4.2.2 Alphabetical list of type definitions

This subclause contains an alphabetical listing of all of the data type definitions used to define the Ada binding to PHIGS. Each element declared includes a comment about the type and/or use of the type. Some of the declarations employ constant values in the definition of the type. All of these constant declarations are included in the PHIGS_TYPES package.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

ACCESS_COLOUR_MATRIX

```
type ACCESS_COLOUR_MATRIX is access COLOUR_MATRIX;

-- Provides for returning variable sized matrices containing colour
-- indices corresponding to a cell array or pattern array.
```

ACCESS_STRING

```
type ACCESS_STRING is access PHIGS_STRING;

-- Defines a variable length string needed to return strings in
-- structure element records.
```

ANGLE

```
type ANGLE is digits PHIGS_PRECISION;

-- Values used in the modelling transformation utility functions
-- for specifying angles of rotation in radians. Positive
-- indicates a counterclockwise direction.
```

ANNOTATION_STYLE

```
type ANNOTATION_STYLE is new PHIGS_INTEGER;

-- Defines the annotation styles for annotation primitives.
```

ANNOTATION_STYLES

```
package ANNOTATION_STYLES is
        new PHIGS_LIST_UTILITIES (ANNOTATION_STYLE,
                                  MAX_ANNOTATION_STYLES_SUPPORTED);

-- Provides for lists of annotation styles.
```

APPLICATION_DATA_RANGE

```
subtype APPLICATION_DATA_RANGE is
        PHIGS_NATURAL range 0..MAX_APPLICATION_DATA;

-- Defines range of lengths for application data objects.
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

APPLICATION_DATA_RECORD

```
type APPLICATION_DATA_RECORD (LENGTH : APPLICATION_DATA_RANGE := 0) is
        record
          DATA : PHIGS_STRING (1..LENGTH);
        end record;

-- Defines type for defining application data objects.
```

ARCHIVE_ID

```
type ARCHIVE_ID is new PHIGS_NATURAL;

-- Provides for an application specified pointer to archive
-- files separate from the file identifier.
```

ARCHIVE_IDS

```
package ARCHIVE_IDS is new PHIGS_LIST_UTILITIES (ARCHIVE_ID,
                                                 MAX_ARCHIVE_IDS_SUPPORTED);

-- Provides for lists of archive identifiers.
```

ARCHIVE_STATE

```
type ARCHIVE_STATE is (ARCL,
                       AROP);

-- The type used to return the archive state.
```

ASF

```
type ASF is (BUNDLED,
             INDIVIDUAL);

-- This type defines an aspect source flag whose
-- value indicates whether an aspect of a primitive
-- should be set from a bundle table or from an
-- individual attribute.
```

ASPECT

```
type ASPECT is (TYPE_OF_LINE,
                LINEWIDTH_SF,
                LINE_COLOUR,
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
                    TYPE_OF_MARKER,
                    SIZE,
                    MARKER_COLOUR,

                    FONT,
                    PRECISION,
                    EXPANSION,
                    SPACING,
                    TEXT_COLOUR,

                    STYLE_OF_INTERIOR,
                    STYLE_IND,
                    INTERIOR_COLOUR,

                    FLAG,
                    TYPE_OF_EDGE,
                    EDGEWIDTH_SF,
                    EDGE_COLOUR);

    -- This type lists the aspects for which an aspect source
    -- flag exists.
```

ATTRIBUTES_USED

```
    package ATTRIBUTES_USED is
            new PHIGS_LIST_UTILITIES
                (ATTRIBUTES_USED_TYPE,
                 ATTRIBUTES_USED_TYPE'POS(ATTRIBUTES_USED_TYPE'LAST)+1);

    -- Provides for a list of the attributes used which is
    -- returned by the INQ_GDP and LOCATOR_ATTRIBUTES_USED
    -- functions.
```

ATTRIBUTES_USED_TYPE

```
    type ATTRIBUTES_USED_TYPE is (POLYLINE_ATTRIBUTES,
                                  POLYMARKER_ATTRIBUTES,
                                  TEXT_ATTRIBUTES,
                                  INTERIOR_ATTRIBUTES,
                                  EDGE_ATTRIBUTES);

    -- The types of attributes which may be used in generating
    -- output for a GDP and in generating prompt and echo
    -- information for certain prompt and echo types of certain
    -- classes of input devices.
```

CHAR_EXPANSION

```
    type CHAR_EXPANSION is new SCALE_FACTOR range
            SCALE_FACTOR'SAFE_SMALL..SCALE_FACTOR'LAST;

    -- Defines a character expansion factor.
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

CHAR_SET

```
type CHAR_SET is new PHIGS_NATURAL;

-- Provides identifications for the available character
-- sets. ISO 646 character set maps to value zero.
```

CHAR_SETS

```
package CHAR_SETS is new PHIGS_LIST_UTILITIES (CHAR_SET,
                                               MAX_CHAR_SETS_SUPPORTED);

-- Provides for lists of character set identifications.
```

CHAR_SPACING

```
type CHAR_SPACING is new SCALE_FACTOR;

-- Defines a character spacing factor. The factors are
-- unitless. A positive value indicates the amount of extra
-- space between character boxes in a text string, and a
-- negative value indicates the amount of overlap between
-- character boxes in a text string.
```

CHOICE_DEVICE_NUMBER

```
type CHOICE_DEVICE_NUMBER is new DEVICE_NUMBER;

-- Provides for choice device identifiers.
```

CHOICE_NUMBER

```
type CHOICE_NUMBER is new PHIGS_POSITIVE;

-- Defines the choice numbers available on an implementation.
```

CHOICE_PROMPT

```
type CHOICE_PROMPT is new OFF_ON;

-- Indicates whether a specified choice prompt is to be displayed
-- or not.
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

---

CHOICE_PROMPTS

```
package CHOICE_PROMPTS is
        new PHIGS_LIST_UTILITIES
            (CHOICE_PROMPT,
             MAX_CHOICE_PROMPTS_SUPPORTED);

-- Provides for lists of choice prompts.
```

CHOICE_PROMPT_ECHO_TYPE

```
type CHOICE_PROMPT_ECHO_TYPE is new PHIGS_INTEGER;

-- Defines the choice prompt and echo type.
```

CHOICE_PROMPT_ECHO_TYPES

```
package CHOICE_PROMPT_ECHO_TYPES is
        new PHIGS_LIST_UTILITIES
            (CHOICE_PROMPT_ECHO_TYPE,
             MAX_CHOICE_PROMPT_ECHO_TYPES_SUPPORTED);

-- Provides for lists of choice prompt and echo types.
```

CHOICE_PROMPT_STRING

```
type CHOICE_PROMPT_STRING (LENGTH : STRING_SMALL_NATURAL := 0) is
        record
          CONTENTS : PHIGS_STRING (1..LENGTH);
        end record;

-- Provides for a variable length prompt.  Objects of this
-- type should be declared unconstrained to allow for dynamic
-- modification of the length.
```

CHOICE_PROMPT_STRING_ARRAY

```
type CHOICE_PROMPT_STRING_ARRAY is array (PHIGS_POSITIVE range <>) of
        CHOICE_PROMPT_STRING;

-- Provides for an array of prompt strings.
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

---

CHOICE_PROMPT_STRING_LIST

```
type CHOICE_PROMPT_STRING_LIST (LENGTH : CHOICE_SMALL_NATURAL := 0) is
        record
          LIST : CHOICE_PROMPT_STRING_ARRAY (1..LENGTH);
        end record;

-- Provides for lists of prompt strings.
```

CHOICE_REQUEST_STATUS

```
type CHOICE_REQUEST_STATUS is (OK,
                               NOCHOICE,
                               NONE);

-- Defines the status of a choice input operation for the
-- request function.
```

CHOICE_SMALL_NATURAL

```
subtype CHOICE_SMALL_NATURAL is
        PHIGS_NATURAL range 0..MAX_CHOICE_SMALL_NATURAL;

-- This is a subtype declaration which allows for
-- unconstrained record objects for CHOICE_PROMPT_STRING_LIST
-- type without causing the exception STORAGE_ERROR to be raised.
```

CHOICE_STATUS

```
subtype CHOICE_STATUS is
        CHOICE_REQUEST_STATUS range OK..NOCHOICE;

-- Indicates if a choice was made by the operator for the
-- sample, get, and inquiry functions.
```

CHROMATICITY_COEFFICIENT

```
type CHROMATICITY_COEFFICIENT is
        record
          X : COLOUR_COEFFICIENT;
          Y : COLOUR_COEFFICIENT;
        end record;

-- Defines a CIE colour coefficient pair.
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

---

CHROMATICITY_COEFFICIENT_SET

```
type CHROMATICITY_COEFFICIENT_SET is
        record
          R : CHROMATICITY_COEFFICIENT;
          G : CHROMATICITY_COEFFICIENT;
          B : CHROMATICITY_COEFFICIENT;
        end record;

-- Defines a set of CIE primary colour chromaticity coefficients.
```

CLIPPING_INDICATOR

```
type CLIPPING_INDICATOR is (CLIP,
                            NOCLIP);

-- Indicates whether or not clipping is to be performed.
```

COLOUR_AVAILABLE

```
type COLOUR_AVAILABLE is (COLOUR,
                          MONOCHROME);

-- Indicates whether colour output is available on a
-- workstation.
```

COLOUR_COEFFICIENT

```
type COLOUR_COEFFICIENT is digits PHIGS_PRECISION;

-- Defines a general type for colour components.
```

COLOUR_COEFFICIENT_ARRAY

```
type COLOUR_COEFFICIENT_ARRAY is
        array (COLOUR_COMPONENTS) of COLOUR_COEFFICIENT;

-- Defines a general type for colour components.
```

COLOUR_COEFFICIENTS

```
package COLOUR_COEFFICIENTS is
          new PHIGS_LIST_UTILITIES (COLOUR_COEFFICIENT,
                                    MAX_COLOUR_COEFFICIENTS);

-- Provides for a lists of colour coefficients for the generalized
-- colour model.
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

COLOUR_COMPONENTS

```
type COLOUR_COMPONENTS is range 1..MAX_COLOUR_COEFFICIENTS;

-- Defines a type for the size of colour component arrays.
```

COLOUR_INDEX

```
type COLOUR_INDEX is new PHIGS_NATURAL;

-- Indices into colour tables are of this type.
```

COLOUR_INDICES

```
package COLOUR_INDICES is
        new PHIGS_LIST_UTILITIES (COLOUR_INDEX,
                                  MAX_COLOUR_INDICES_SUPPORTED);

-- Provides for a set of colour indices which are available
-- on a particular workstation.
```

COLOUR_MATRIX

```
type COLOUR_MATRIX is array (COLOUR_MATRIX_SMALL_NATURAL range <>,
                             COLOUR_MATRIX_SMALL_NATURAL range <>)
                    of COLOUR_INDEX;

-- Provides for matrices containing colour indices corresponding
-- to a cell array or pattern array.
```

COLOUR_MATRIX_SMALL_NATURAL

```
subtype COLOUR_MATRIX_SMALL_NATURAL is
        PHIGS_NATURAL range 0..MAX_COLOUR_MATRIX_SMALL_NATURAL;

-- This is a subtype declaration which allows for unconstrained
-- record objects for COLOUR_MATRIX type without causing the
-- exception STORAGE_ERROR to be raised.
```

COLOUR_MODEL

```
type COLOUR_MODEL is new PHIGS_INTEGER;

-- Indicates the colour models available in PHIGS.
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

---

COLOUR_MODELS

```
package COLOUR_MODELS is
        new PHIGS_LIST_UTILITIES (COLOUR_MODEL,
                                  MAX_COLOUR_MODELS_SUPPORTED);

-- Provides for lists of colour models.
```

COLOUR_REPRESENTATION

```
type COLOUR_REPRESENTATION (MODEL : COLOUR_MODEL := RGB) is
      record
        case MODEL is

          when RGB = >
            RED_RGB              : INTENSITY;
            GREEN_RGB            : INTENSITY;
            BLUE_RGB             : INTENSITY;

          when CIELUV = >
            L_STAR_CIE           : COLOUR_COEFFICIENT;
            U_STAR_CIE           : COLOUR_COEFFICIENT;
            V_STAR_CIE           : COLOUR_COEFFICIENT;

          when HSV = >
            HUE_HSV              : INTENSITY;
            SATURATION_HSV       : INTENSITY;
            VALUE_HSV            : INTENSITY;

          when HLS = >
            HUE_HLS              : INTENSITY;
            LIGHTNESS_HLS        : INTENSITY;
            SATURATION_HLS       : INTENSITY;

          when others = >
            GENERIC_COMPONENTS   : COLOUR_COEFFICIENT_ARRAY;

        end case;
      end record;

-- Defines the representation of a colour as a combination of components
-- whose meaning depends on the colour model. Additional variants may be
-- included when additional registered or unregistered colour models
-- are supported by an implementation.
```

COMPOSITION_TYPE

```
type COMPOSITION_TYPE is (PRECONCATENATE,
                          POSTCONCATENATE,
                          REPLACE);

-- Defines the type of matrix composition allowed between
-- modelling transformations.
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

CONFLICT_RESOLUTION

```
type CONFLICT_RESOLUTION is (MAINTAIN,
                             ABANDON,
                             UPDATE);

-- Defines a type for specifying the conflict resolution mode.
```

CONNECTION_ID

```
type CONNECTION_ID is new PHIGS_STRING;

-- Provides for specifying connection identifiers.
```

CONTROL_FLAG

```
type CONTROL_FLAG is (CONDITIONALLY,
                      ALWAYS);

-- The control flag is used to indicate the conditions
-- under which the display surface should be cleared.
```

DC

```
package DC is new PHIGS_COORDINATE_SYSTEM (DC_TYPE);

-- Defines the Device Coordinate System.
```

DC_TYPE

```
type DC_TYPE is digits PHIGS_PRECISION;

-- The type of a coordinate component in the Device Coordinate
-- System.
```

DC_UNITS

```
type DC_UNITS is (METRES,
                  OTHER);

-- Device coordinate units for a particular workstation
-- may be in metres, or some other unit (such as inches).
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

---

DEFERRAL_MODE

```
type DEFERRAL_MODE is (ASAP,
                       BNIG,
                       BNIL,
                       ASTI,
                       WAIT);

-- Defines the five PHIGS deferral modes.
```

DEVICE_NUMBER

```
subtype DEVICE_NUMBER is PHIGS_POSITIVE;

-- Defines the base type for input device numbers.
```

DISPLAY_CLASS

```
type DISPLAY_CLASS is (VECTOR_DISPLAY,
                       RASTER_DISPLAY,
                       OTHER_DISPLAY);

-- The classification of a workstation of category OUTPUT
-- or OUTIN.
```

DISPLAY_PRIORITY

```
type DIPLAY_PRIORITY is
      digits PHIGS_PRECISION range 0.0..1.0;

-- Defines type for specifying structure display priority.
```

DISPLAY_SURFACE_EMPTY

```
type DISPLAY_SURFACE_EMPTY is (EMPTY,
                               NOTEMPTY);

-- Indicates whether there is graphical output on the display
-- surface.
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

DYNAMIC_MODIFICATION

```
type DYNAMIC_MODIFICATION is (IRG,
                              IMM,
                              CBS);

-- Indicates whether an update to the state list is performed
-- immediately (IMM), requires implicit regeneration (IRG), or
-- can be simulated (CBS).
```

ECHO_SWITCH

```
type ECHO_SWITCH is (ECHO,
                     NOECHO);

-- Indicates whether or not echoing of the measure is
-- performed.
```

EDGE_DATA

```
type EDGE_DATA is record
                    FLAG_ASF          : ASF;
                    TYPE_OF_EDGE_ASF  : ASF;
                    EDGEWIDTH_SF_ASF  : ASF;
                    EDGE_COLOUR_ASF   : ASF;
                    FLAG              : EDGE_FLAG;
                    TYPE_OF_EDGE      : EDGETYPE;
                    EDGEWIDTH_SF      : EDGEWIDTH;
                    EDGE_COLOUR       : COLOUR_INDEX;
                  end record;

-- A record containing information needed to specify the
-- appearance of a fill area set edge.
```

EDGE_FLAG

```
type EDGE_FLAG is new OFF_ON;

-- Defines a type for indicating whether or not to depict
-- edges on fill area set and related primitives.
```

EDGE_INDEX

```
type EDGE_INDEX is new PHIGS_POSITIVE;

-- Defines the edge bundle table indices.
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

---

EDGE_INDICES

```
package EDGE_INDICES is
        new PHIGS_LIST_UTILITIES (EDGE_INDEX,
                                  MAX_EDGE_INDICES_SUPPORTED);

-- Provides for a list of edge indices.
```

EDGETYPE

```
type EDGETYPE is new PHIGS_INTEGER;

-- Defines a type for describing the form of edges.
```

EDGETYPES

```
package EDGETYPES is new PHIGS_LIST_UTILITIES (EDGETYPE,
                                               MAX_EDGETYPES_SUPPORTED);

-- Provides for lists of edgetypes.
```

EDGEWIDTH

```
type EDGEWIDTH is
      new SCALE_FACTOR range 0.0..SCALE_FACTOR'LAST;

-- Defines a type for describing the width scale factor of
-- edges.
```

EDIT_MODE

```
type EDIT_MODE is (INSERT,
                   REPLACE);

-- Defines a type for specifying the mode in which editing
-- takes place.
```

ELEMENT_POSITION

```
type ELEMENT_POSITION is new PHIGS_INTEGER;

-- Base type for element pointer values and offsets.
```

ELEMENT_TYPE

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
type ELEMENT_TYPE is
        (ALL_ELEMENT_TYPES,
         NIL,

         POLYLINE_3,
         POLYLINE,

         POLYMARKER_3,
         POLYMARKER,

         TEXT_3,
         TEXT,

         ANNOTATION_TEXT_RELATIVE_3,
         ANNOTATION_TEXT_RELATIVE,

         FILL_AREA_3,
         FILL_AREA,
         FILL_AREA_SET_3,
         FILL_AREA_SET,

         CELL_ARRAY_3,
         CELL_ARRAY,

         GDP_3,
         GDP,

         SET_POLYLINE_INDEX,
         SET_POLYMARKER_INDEX,
         SET_TEXT_INDEX,
         SET_INTERIOR_INDEX,
         SET_EDGE_INDEX,

         SET_LINETYPE,
         SET_LINEWIDTH_SCALE_FACTOR,
         SET_POLYLINE_COLOUR_INDEX,

         SET_MARKER_TYPE,
         SET_MARKER_SIZE_SCALE_FACTOR,
         SET_POLYMARKER_COLOUR_INDEX,

         SET_TEXT_FONT,
         SET_TEXT_PRECISION,
         SET_CHAR_EXPANSION_FACTOR,
         SET_CHAR_SPACING,
         SET_TEXT_COLOUR_INDEX,
         SET_CHAR_HEIGHT,
         SET_CHAR_UP_VECTOR,
         SET_TEXT_PATH,
         SET_TEXT_ALIGNMENT,

         SET_ANNOTATION_TEXT_CHAR_HEIGHT,
         SET_ANNOTATION_TEXT_CHAR_UP_VECTOR,
         SET_ANNOTATION_TEXT_PATH,
         SET_ANNOTATION_TEXT_ALIGNMENT,
         SET_ANNOTATION_STYLE,
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
        SET_INTERIOR_STYLE,
        SET_INTERIOR_STYLE_INDEX,
        SET_INTERIOR_COLOUR_INDEX,

        SET_EDGE_FLAG,
        SET_EDGETYPE,
        SET_EDGEWIDTH_SCALE_FACTOR,
        SET_EDGE_COLOUR_INDEX,

        SET_PATTERN_SIZE,
        SET_PATTERN_REFERENCE_POINT_AND_VECTORS,
        SET_PATTERN_REFERENCE_POINT,

        ADD_NAMES_TO_SET,
        REMOVE_NAMES_FROM_SET,

        SET_INDIVIDUAL_ASF,
        SET_HLHSR_IDENTIFIER,

        SET_LOCAL_TRANSFORMATION_3,
        SET_LOCAL_TRANSFORMATION,
        SET_GLOBAL_TRANSFORMATION_3,
        SET_GLOBAL_TRANSFORMATION,
        SET_MODELLING_CLIPPING_VOLUME_3,
        SET_MODELLING_CLIPPING_VOLUME,
        SET_MODELLING_CLIPPING_INDICATOR,
        RESTORE_MODELLING_CLIPPING_VOLUME,
        SET_VIEW_INDEX,

        EXECUTE_STRUCTURE,

        LABEL,
        APPLICATION_DATA,
        GSE,
        SET_PICK_IDENTIFIER);

    -- Provides for specification of element types.

-----------------------------------------------------------------------------------------------------------------------

ELEMENT_TYPES

    package ELEMENT_TYPES is
            new PHIGS_LIST_UTILITIES (ELEMENT_TYPE,
                                      ELEMENT_TYPE'POS(ELEMENT_TYPE'LAST)+1);

    -- Provides for lists of element types.

-----------------------------------------------------------------------------------------------------------------------

ERROR_HANDLING_MODE

    type ERROR_HANDLING_MODE is new OFF_ON;

    -- Type for inquiring the error handling mode.
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

ERROR_NUMBER

```
type ERROR_NUMBER is new PHIGS_INTEGER;

-- Defines the type for error indicator values.
```

ESCAPE_ID

```
type ESCAPE_ID is new PHIGS_INTEGER;

-- Defines a type for identifying different escapes.
```

ESCAPE_IDS

```
package ESCAPE_IDS is new PHIGS_LIST_UTILITIES (ESCAPE_ID,
                                                MAX_ESCAPE_IDS_SUPPORTED);

-- Provides for lists of Escape ID's.
```

ESCAPE_DATA_RECORD

```
type ESCAPE_DATA_RECORD (ESCAPE : ESCAPE_ID := 0) is
        record
          case ESCAPE is

            -- For each ESCAPE which needs a record, an
            -- implementation dependent entry is defined.
            -- For others, a null record is defined.

            when others = >
               null;
          end case;
        end record;

-- Provides for the definition of ESCAPE data records.
```

EVENT_DEVICE_NUMBER

```
type EVENT_DEVICE_NUMBER (CLASS : INPUT_CLASS := NONE) is
        record
          case CLASS is

            when NONE = > null;

            when LOCATOR_INPUT
               = > LOCATOR_EVENT_DEVICE : LOCATOR_DEVICE_NUMBER;

            when STROKE_INPUT
               = > STROKE_EVENT_DEVICE : STROKE_DEVICE_NUMBER;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
                when VALUATOR_INPUT
                   => VALUATOR_EVENT_DEVICE : VALUATOR_DEVICE_NUMBER;

                when CHOICE_INPUT
                   => CHOICE_EVENT_DEVICE : CHOICE_DEVICE_NUMBER;

                when PICK_INPUT
                   => PICK_EVENT_DEVICE : PICK_DEVICE_NUMBER;

                when STRING_INPUT
                   => STRING_EVENT_DEVICE : STRING_DEVICE_NUMBER;

              end case;
            end record;

    -- Provides for returning any class of device number from the
    -- event queue.
```

EVENT_QUEUE_DEVICE_NUMBER

```
    type EVENT_QUEUE_DEVICE_NUMBER
            (CLASS : INPUT_QUEUE_CLASS := LOCATOR_INPUT) is
                record
                  case CLASS is

                    when LOCATOR_INPUT
                       => LOCATOR_EVENT_DEVICE : LOCATOR_DEVICE_NUMBER;

                    when STROKE_INPUT
                       => STROKE_EVENT_DEVICE : STROKE_DEVICE_NUMBER;

                    when VALUATOR_INPUT
                       => VALUATOR_EVENT_DEVICE : VALUATOR_DEVICE_NUMBER;

                    when CHOICE_INPUT
                       => CHOICE_EVENT_DEVICE : CHOICE_DEVICE_NUMBER;

                    when PICK_INPUT
                       => PICK_EVENT_DEVICE : PICK_DEVICE_NUMBER;

                    when STRING_INPUT
                       => STRING_EVENT_DEVICE : STRING_DEVICE_NUMBER;

                  end case;
                end record;

    -- Allows EVENT_DEVICE_NUMBERs which cannot have value NONE.  This is
    -- used when returning only real input classes as causes of event
    -- queue overflow and for flushing the queue.
```

FILE_ID

```
    subtype FILE_ID is PHIGS_STRING;

    -- Provides for file identifiers.
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

FILE_SMALL_NATURAL

```
subtype FILE_SMALL_NATURAL is
        PHIGS_NATURAL range 0..MAX_FILE_SMALL_NATURAL;

-- This is a subtype declaration which allows for unconstrained
-- record objects for VARIABLE_FILE_ID type without
-- causing the exception STORAGE_ERROR to be raised.
```

GDP_3_ID

```
type GDP_3_ID is new PHIGS_INTEGER;

-- Defines a type for selecting a Generalized Drawing
-- Primitive.
```

GDP_3_IDS

```
package GDP_3_IDS is new PHIGS_LIST_UTILITIES (GDP_3_ID,
                                               MAX_GDP_3_IDS_SUPPORTED);

-- Provides for lists of 3D Generalized Drawing Primitive ID's.
```

GDP_3_RECORD

```
type GDP_3_RECORD (GDP : GDP_3_ID := 0) is
        record
          case GDP is

            -- For each supported GDP_3 which needs a record, an
            -- implementation dependent entry is defined.
            -- For others, a generic record is defined.

            when others = >
              GDP_INFO : APPLICATION_DATA_RECORD;
          end case;
        end record;

-- Provides for the definition of GDP_3 data records.
```

GDP_ID

```
type GDP_ID is new PHIGS_INTEGER;

-- Defines a type for selecting a Generalized Drawing
-- Primitive.
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

---

GDP_IDS

```
package GDP_IDS is new PHIGS_LIST_UTILITIES (GDP_ID,
                                              MAX_GDP_IDS_SUPPORTED);

-- Provides for lists of Generalized Drawing Primitive ID's.
```

GDP_RECORD

```
type GDP_RECORD (GDP : GDP_ID := 0) is
        record
          case GDP is

            -- For each supported GDP which needs a record, an
            -- implementation dependent entry is defined.
            -- For others, a generic record is defined.

            when others = >
              GDP_INFO : APPLICATION_DATA_RECORD;
          end case;
        end record;

-- Provides for the definition of GDP data records.
```

GENERAL_FLOAT_PARAMETER

```
type GENERAL_FLOAT_PARAMETER is digits PHIGS_PRECISION;

-- Defines a type for specifying implementation-specific real
-- data in GDP, GSE, and ESCAPE data records.
```

GSE_ID

```
type GSE_ID is new PHIGS_INTEGER;

-- Defines a type for selecting a Generalized Structure
-- Element.
```

GSE_IDS

```
package GSE_IDS is new PHIGS_LIST_UTILITIES (GSE_ID,
                                             MAX_GSE_IDS_SUPPORTED);

-- Provides for lists of Generalized Structure Element ID's.
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

---

GSE_RECORD

```
type GSE_RECORD (GSE : GSE_ID := 0) is
        record
          case GSE is

            -- For each supported GSE which needs a record, an
            -- implementation dependent entry is defined.
            -- For others, a generic record is defined.

            when others = >
               GSE_INFO : APPLICATION_DATA_RECORD;
          end case;
        end record;
```

-- Provides for the definition of GSE data records.

HATCH_STYLE

```
subtype HATCH_STYLE is STYLE_INDEX;
```

-- Defines the interior hatch styles type.

HATCH_STYLES

```
package HATCH_STYLES is
            new PHIGS_LIST_UTILITIES (HATCH_STYLE,
                                      MAX_HATCH_STYLES_SUPPORTED);
```

-- Provides for lists of hatch styles.

HLHSR_ID

```
type HLHSR_ID is new PHIGS_INTEGER;
```

-- A number used as an attribute for workstation independent
-- HLHSR factors.

HLHSR_IDS

```
package HLHSR_IDS is new PHIGS_LIST_UTILITIES (HLHSR_ID,
                                               MAX_HLHSR_IDS_SUPPORTED);
```

-- Provides for lists of HLHSR identifiers.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

---

HLHSR_MODE

```
type HLHSR_MODE is new PHIGS_INTEGER;

-- A number used for controlling workstation dependent
-- HLHSR factors.
```

HLHSR_MODES

```
package HLHSR_MODES is
        new PHIGS_LIST_UTILITIES (HLHSR_MODE,
                                  MAX_HLHSR_MODES_SUPPORTED);

-- Provides for lists of HLHSR modes.
```

HORIZONTAL_ALIGNMENT

```
type HORIZONTAL_ALIGNMENT is (NORMAL,
                              LEFT,
                              CENTRE,
                              RIGHT);

-- The alignment of the text extent rectangle with
-- respect to horizontal positioning of the text.
```

INPUT_CLASS

```
type INPUT_CLASS is (NONE,
                     LOCATOR_INPUT,
                     STROKE_INPUT,
                     VALUATOR_INPUT,
                     CHOICE_INPUT,
                     PICK_INPUT,
                     STRING_INPUT);

-- Defines the input device classifications for
-- workstations of category INPUT or OUTIN.
```

INPUT_QUEUE_CLASS

```
subtype INPUT_QUEUE_CLASS is
        INPUT_CLASS range LOCATOR_INPUT .. STRING_INPUT;

-- Defines the input device classifications for workstations
-- of category INPUT and OUTIN that do not return a NONE
-- classification.
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

INPUT_STATUS

```
type INPUT_STATUS is (OK,
                      NONE);

-- Defines the status of a locator, stroke, valuator, or
-- string operation.
```

INPUT_STRING

```
type INPUT_STRING is (LENGTH : INPUT_STRING_SMALL_NATURAL := 0) is
      record
        CONTENTS : PHIGS_STRING (1..LENGTH);
      end record;

-- Provides a variable length string for use in STRING input functions.
-- Objects of this type should be declared unconstrained to allow for
-- dynamic modification of the length.
```

INPUT_STRING_SMALL_NATURAL

```
subtype INPUT_STRING_SMALL_NATURAL is
          PHIGS_NATURAL range 0..MAX_INPUT_STRING_SMALL_NATURAL;

-- This is a subtype declaration which allows for unconstrained
-- record objects for INPUT_STRING type without
-- causing the exception STORAGE_ERROR to be raised.
```

INTENSITY

```
subtype INTENSITY is COLOUR_COEFFICIENT range 0.0..1.0;

-- Defines the restricted range of colour components required
-- by some colour models.
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

---

INTERIOR_DATA

```
type INTERIOR_DATA is
        record
          STYLE_OF_INTERIOR_ASF : ASF;
          STYLE_IND_ASF         : ASF;
          INTERIOR_COLOUR_ASF   : ASF;
          INTERIOR_IND          : INTERIOR_INDEX;
          STYLE_OF_INTERIOR     : INTERIOR_STYLE;
          STYLE_IND             : STYLE_INDEX;
          INTERIOR_COLOUR       : COLOUR_INDEX;
        end record;

-- A record containing information needed to specify the
-- appearance of a filled area interior.
```

INTERIOR_INDEX

```
type INTERIOR_INDEX is new PHIGS_POSITIVE;

-- Defines interior bundle table indices.
```

INTERIOR_INDICES

```
package INTERIOR_INDICES is
        new PHIGS_LIST_UTILITIES (INTERIOR_INDEX,
                                  MAX_INTERIOR_INDICES_SUPPORTED);

-- Provides for lists of interior bundle table indices.
```

INTERIOR_STYLE

```
type INTERIOR_STYLE is (HOLLOW,
                        SOLID,
                        PATTERN,
                        HATCH,
                        EMPTY);

-- Defines the interior styles for filled areas.
```

INTERIOR_STYLES

```
package INTERIOR_STYLES is
        new PHIGS_LIST_UTILITIES
              (INTERIOR_STYLE,
               INTERIOR_STYLE'POS(INTERIOR_STYLE'LAST)+1);

-- Provides for lists of interior styles.
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

---

INVALID_VALUES_INDICATOR

```
type INVALID_VALUES_INDICATOR is (ABSENT,
                                  PRESENT);

-- Indicates whether invalid values are contained in
-- a colour array or matrix.
```

LABEL_ID

```
type LABEL_ID is new PHIGS_INTEGER;

-- Defines a type for specifying structure element labels.
```

LINE_DATA

```
type LINE_DATA is
    record
      TYPE_OF_LINE_ASF : ASF;
      WIDTH_ASF        : ASF;
      LINE_COLOUR_ASF  : ASF;
      POLYLINE_IND     : POLYLINE_INDEX;
      TYPE_OF_LINE     : LINETYPE;
      WIDTH            : LINEWIDTH;
      LINE_COLOUR      : COLOUR_INDEX;
    end record;

-- A record containing information needed to specify the
-- appearance of a line for input data records.
```

LINETYPE

```
type LINETYPE is new PHIGS_INTEGER;

-- Defines the types of line styles provided by PHIGS.
```

LINETYPES

```
package LINETYPES is new PHIGS_LIST_UTILITIES (LINETYPE,
                                               MAX_LINETYPES_SUPPORTED);

-- Provides for lists of linetypes.
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

9593 PT*3 90 ■ 4851903 0575418 606 ■

LINEWIDTH

```
type LINEWIDTH is
      new SCALE_FACTOR range 0.0..SCALE_FACTOR'LAST;

-- The width of a line is indicated by a scale factor.
```

LOCATOR_DEVICE_NUMBER

```
type LOCATOR_DEVICE_NUMBER is new DEVICE_NUMBER;

-- Provides for locator device identifiers.
```

LOCATOR_PROMPT_ECHO_TYPE

```
type LOCATOR_PROMPT_ECHO_TYPE is new PHIGS_INTEGER;

-- Defines the locator prompt and echo types supported by
-- the implementation.
```

LOCATOR_PROMPT_ECHO_TYPES

```
package LOCATOR_PROMPT_ECHO_TYPES is
      new PHIGS_LIST_UTILITIES
            (LOCATOR_PROMPT_ECHO_TYPE,
             MAX_LOCATOR_PROMPT_ECHO_TYPES_SUPPORTED);

-- Provides for lists of locator prompt and echo types.
```

MARKER_DATA

```
type MARKER_DATA is
      record
        TYPE_OF_MARKER_ASF  : ASF;
        SIZE_ASF            : ASF;
        MARKER_COLOUR_ASF   : ASF;
        POLYMARKER_IND      : POLYMARKER_INDEX;
        TYPE_OF_MARKER      : MARKER_TYPE;
        SIZE                : MARKER_SIZE;
        MARKER_COLOUR       : COLOUR_INDEX;
      end record;

-- A record containing information needed to specify the
-- appearance of a marker for input data records.
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

---

MARKER_SIZE

```
type MARKER_SIZE is new SCALE_FACTOR;

-- The size of a marker is indicated by a scale factor.
```

MARKER_TYPE

```
type MARKER_TYPE is new PHIGS_INTEGER;

-- Defines the type for markers provided by PHIGS.
```

MARKER_TYPES

```
package MARKER_TYPES is
        new PHIGS_LIST_UTILITIES (MARKER_TYPE,
                                  MAX_MARKER_TYPES_SUPPORTED);

-- Provides for lists of marker types.
```

MC

```
package MC is new PHIGS_COORDINATE_SYSTEM (MC_TYPE);

-- Defines the Modelling Coordinate System.
```

MC_TYPE

```
type MC_TYPE is digits PHIGS_PRECISION;

-- Defines the type of coordinate in the Modelling
-- Coordinate System.
```

MEMORY_UNITS

```
type MEMORY_UNITS is range 0..MAX_MEMORY_UNITS;

-- Defines the type for units of memory that may be
-- allocated by PHIGS.
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

---

METAFILE_ITEM_LENGTH

```
type METAFILE_ITEM_LENGTH is
        new PHIGS_NATURAL range 0..MAX_METAFILE_ITEM_LENGTH;

-- The length of an item contained in a metafile.
```

METAFILE_ITEM_TYPE

```
type METAFILE_ITEM_TYPE is
        new PHIGS_NATURAL
                range MIN_METAFILE_ITEM_TYPE..MAX_METAFILE_ITEM_TYPE;

-- The type of an item contained in a metafile.
```

MODELLING_CLIP_OPERATION_TYPE

```
type MODELLING_CLIP_OPERATION_TYPE is new PHIGS_INTEGER;

-- This type provides for specifying modelling clip operation
-- types.
```

MODELLING_CLIP_OPERATION_TYPES

```
package MODELLING_CLIP_OPERATION_TYPES is
        new PHIGS_LIST_UTILITIES
                (MODELLING_CLIP_OPERATION_TYPE,
                 MAX_MODELLING_CLIP_OPERATION_TYPES_SUPPORTED);

-- This type provides for lists of modelling clip operation types.
```

MODIFICATION_MODE

```
type MODIFICATION_MODE is (NIVE,
                           UWOR,
                           UQUM);

-- Defines type for specifying modification modes.
```

MORE_EVENTS

```
type MORE_EVENTS is (NOMORE,
                     MORE);

-- Indicates whether more simultaneous events are contained
-- in the input queue.
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

NAME

```
subtype NAME is
        PHIGS_NAME_SET_FACILITY.NAME;

-- Provides for names for each of the name set classes.
-- See 5.14.4.
```

NAMES

```
package NAMES renames
        PHIGS_NAME_SET_FACILITY.NAMES;

-- Provides for lists of name set classes. See 5.14.4.
```

NAME_SET

```
subtype NAME_SET is PHIGS_NAME_SET_FACILITY.NAME_SET;

-- Used to define name sets. The internal structure of
-- a name set is implementation-dependent. "Build" functions
-- contained in the package PHIGS_NAME_SET_FACILITIES shall be
-- used to create and manipulate name sets (See 5.14.4).
```

NAME_SET_ACCEPTANCE

```
subtype NAME_SET_ACCEPTANCE is
        PHIGS_NAME_SET_FACILITY.NAME_SET_ACCEPTANCE;

-- Used to indicate the results of applying a name set
-- against a filter pair. See 5.14.4.
```

NAME_SET_FILTER

```
subtype NAME_SET_FILTER is
        PHIGS_NAME_SET_FACILITY.NAME_SET_FILTER;

-- Used to define name set pairs used as filters.
-- See 5.14.4.
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

---

NAME_SET_FILTERS

```
package NAME_SET_FILTERS is
        new PHIGS_LIST_UTILITIES
                (NAME_SET_FILTER,
                 MAX_NAME_SET_FILTER_LIST_LENGTH_SUPPORTED);

-- Used to provide lists of name set filters.
-- See 5.14.4.
```

NAME_SET_MEMBERSHIP

```
subtype NAME_SET_MEMBERSHIP is
        PHIGS_NAME_SET_FACILITY.NAME_SET_MEMBERSHIP;

-- Provides for indicating whether a class is contained in a
-- name set. See 5.14.4.
```

NPC

```
package NPC is new PHIGS_COORDINATE_SYSTEM (NPC_TYPE);

-- Defines the Normalized Projection Coordinate System.
```

NPC_TYPE

```
type NPC_TYPE is digits PHIGS_PRECISION;

-- Defines the type of a coordinate in the Normalized
-- Projection Coordinate System.
```

OFF_ON

```
type OFF_ON is (OFF,
                ON);

-- Generic type for off/on switches.
```

OPEN_STRUCTURE_STATUS

```
type OPEN_STRUCTURE_STATUS is (NONE,
                               OPEN);

-- Returns the status of the open structure.
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

OPERATING_MODE

```
type OPERATING_MODE is (REQUEST_MODE,
                        SAMPLE_MODE,
                        EVENT_MODE);

-- Defines the operating modes of an input device.
```

PATH_DEPTH

```
subtype PATH_DEPTH is
        PHIGS_NATURAL range 0..MAX_PATH_DEPTH_SUPPORTED;

-- Defines a type for specifying the depth of a
-- traversal path.
```

PATH_ORDER

```
type PATH_ORDER is (TOPFIRST,
                    BOTTOMFIRST);

-- Provides for specifying the order of the pick and
-- reference paths.
```

PATTERN_INDEX

```
subtype PATTERN_INDEX is
        STYLE_INDEX range 1..STYLE_INDEX'LAST;

-- Defines the range of pattern table indices.
```

PATTERN_INDICES

```
package PATTERN_INDICES is
        new PHIGS_LIST_UTILITIES (PATTERN_INDEX,
                                  MAX_PATTERN_INDICES_SUPPORTED);

-- Provides for lists of pattern table indices.
```

PHIGS_INTEGER

```
type PHIGS_INTEGER is new INTEGER;

-- Base type for PHIGS integer types.
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

---

PHIGS_NATURAL

```
subtype PHIGS_NATURAL is
        PHIGS_INTEGER range 0..PHIGS_INTEGER'last;

-- Base type for PHIGS natural types.
```

PHIGS_POSITIVE

```
subtype PHIGS_POSITIVE is
        PHIGS_INTEGER range 1..PHIGS_INTEGER'last;

-- Base type for PHIGS positive types.
```

PHIGS_STRING

```
type PHIGS_STRING is
     array (PHIGS_INTEGER range < >) of CHARACTER;

-- Base type for PHIGS string types.
```

PICK_DEVICE_NUMBER

```
type PICK_DEVICE_NUMBER is new DEVICE_NUMBER;

-- Provides for pick device identifiers.
```

PICK_ID

```
type PICK_ID is new PHIGS_INTEGER;

-- Defines the range of pick identifiers available on an
-- implementation.
```

PICK_IDS

```
package PICK_IDS is new PHIGS_LIST_UTILITIES (PICK_ID,
                                              MAX_PICK_IDS_SUPPORTED);

-- Provides for a list of pick identifiers.
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

PICK_PATH

```
type PICK_PATH (DEPTH : PATH_DEPTH := 0) is
      record
        PATH : PICK_PATH_ARRAY (0..DEPTH);
      end record;

-- Provides for the specification of a pick path.
```

PICK_PATH_ARRAY

```
type PICK_PATH_ARRAY is array (PATH_DEPTH range <>) of
                                PICK_PATH_ENTRY;

-- Provides for the specification of an unconstrained array of pick
-- path entries.
```

PICK_PATH_ENTRY

```
type PICK_PATH_ENTRY is
      record
        STRUCTURE_IDENTIFIER  : STRUCTURE_ID;
        PICK_IDENTIFIER       : PICK_ID;
        ELEMENT_NUMBER        : ELEMENT_POSITION;
      end record;

-- Defines the entries of a pick path.
```

PICK_PROMPT_ECHO_TYPE

```
type PICK_PROMPT_ECHO_TYPE is new PHIGS_INTEGER;

-- Defines the pick prompt and echo type.
```

PICK_PROMPT_ECHO_TYPES

```
package PICK_PROMPT_ECHO_TYPES is
        new PHIGS_LIST_UTILITIES (PICK_PROMPT_ECHO_TYPE,
                                  MAX_PICK_PROMPT_ECHO_TYPES_SUPPORTED);

-- Provides for lists of pick prompt and echo types.
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0575426 782 ■

PICK_REQUEST_STATUS

```
type PICK_REQUEST_STATUS is (OK,
                             NOPICK,
                             NONE);

-- Defines the status of a pick input operation for the
-- request function.
```

PICK_STATUS

```
subtype PICK_STATUS is PICK_REQUEST_STATUS range OK..NOPICK;

-- Defines the status of pick input operation for the sample
-- get, and inquiry functions.
```

POLYLINE_FILL_AREA_CONTROL_FLAG

```
type POLYLINE_FILL_AREA_CONTROL_FLAG is (POLYLINE,
                                         FILL_AREA,
                                         FILL_AREA_SET);

-- Provides for indicating the type of primitive used
-- to produce locator prompt and echo type 5.
```

POLYLINE_INDEX

```
type POLYLINE_INDEX is new PHIGS_POSITIVE;

-- Defines the range of polyline indices.
```

POLYLINE_INDICES

```
package POLYLINE_INDICES is
        new PHIGS_LIST_UTILITIES (POLYLINE_INDEX,
                                  MAX_POLYLINE_INDICES_SUPPORTED);

-- Provides for lists of polyline indices.
```

POLYMARKER_INDEX

```
type POLYMARKER_INDEX is new PHIGS_POSITIVE;

-- Defines the range of polymarker bundle table indices.
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

---

POLYMARKER_INDICES

```
package POLYMARKER_INDICES is
        new PHIGS_LIST_UTILITIES (POLYMARKER_INDEX,
                                  MAX_POLYMARKER_INDICES_SUPPORTED);
```

-- Provides for lists of polymarker indices.

POSITIVE_VIEW_INDEX

```
subtype POSITIVE_VIEW_INDEX is
        VIEW_INDEX range 1..VIEW_INDEX'LAST;
```

-- Defines a data type for specifying settable view representations.

POSTED_STRUCTURE

```
type POSTED_STRUCTURE is
     record
       STRUCTURE_IDENTIFIER : STRUCTURE_ID;
       PRIORITY             : DISPLAY_PRIORITY;
     end record;
```

-- Defines a data type for describing posted structure networks.

POSTED_STRUCTURES

```
package POSTED_STRUCTURES is
        new PHIGS_LIST_UTILITIES (POSTED_STRUCTURE,
                                  MAX_POSTED_STRUCTURES_SUPPORTED);
```

-- Provides for the specification of lists of posted structure
-- networks.

PROJECTION_TYPE

```
type PROJECTION_TYPE is (PARALLEL,
                         PERSPECTIVE);
```

-- Defines the type of projections supported by PHIGS.

RASTER_UNITS

```
type RASTER_UNITS is new PHIGS_POSITIVE;
```

-- Defines the range of raster units.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0575428 555 ■

RASTER_UNIT_SIZE_3

```
type RASTER_UNIT_SIZE_3 is
        record
          X : RASTER_UNITS;
          Y : RASTER_UNITS;
          Z : RASTER_UNITS;
        end record;

-- Defines the size of an object in raster units on a
-- raster device.
```

RASTER_UNIT_SIZE_2

```
type RASTER_UNIT_SIZE_2 is
        record
          X : RASTER_UNITS;
          Y : RASTER_UNITS;
        end record;

-- Defines the size of an object in raster units on a
-- raster device.
```

REFERENCE_HANDLING_FLAG

```
type REFERENCE_HANDLING_FLAG is (DELETE,
                                 KEEP);

-- Provides for specification of the handling of deleted
-- structure networks.
```

REFERENCE_PATH

```
type REFERENCE_PATH (DEPTH : PATH_DEPTH := 0) is
        record
          PATH : REFERENCE_PATH_ARRAY (0..DEPTH);
        end record;

-- Provides for the specification of a variable size
-- reference path.
```

REFERENCE_PATH_ARRAY

```
type REFERENCE_PATH_ARRAY is array (PATH_DEPTH range <>) of
                             REFERENCE_PATH_ENTRY;

-- Provides for the specification of a basic reference path.
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0575429 491 ■

REFERENCE_PATH_ENTRY

```
type REFERENCE_PATH_ENTRY is
        record
          STRUCTURE_IDENTIFIER  : STRUCTURE_ID;
          ELEMENT_NUMBER        : ELEMENT_POSITION;
        end record;

-- Defines the entries of a reference path.
```

REFERENCE_PATHS

```
package REFERENCE_PATHS is
            new PHIGS_LIST_UTILITIES (REFERENCE_PATH,
                                      MAX_REFERENCE_PATHS_SUPPORTED);

-- Provides for the specification of lists of reference
-- paths.
```

RELATIVE_PRIORITY

```
type RELATIVE_PRIORITY is (HIGHER,
                           LOWER);

-- Indicates the relative input priority between two views.
```

RETURNED_ELEMENT_POSITION

```
subtype RETURNED_ELEMENT_POSITION is
            ELEMENT_POSITION range 0..ELEMENT_POSITION'LAST;

-- Defines a type for returning element pointer values.
```

RETURN_VALUE_TYPE

```
type RETURN_VALUE_TYPE is (SET,
                           REALIZED);

-- Indicates whether the returned values should be as they
-- were set by the program or as they were actually realized
-- on the device.
```

SCALE_FACTOR

```
type SCALE_FACTOR is digits PHIGS_PRECISION;

-- The type used for unitless scaling factors.
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0575430 103 ■

SEARCH_DIRECTION

```
type SEARCH_DIRECTION is (BACKWARD,
                          FORWARD);

-- This type provides for indicating search direction.
```

SEARCH_STATUS_INDICATOR

```
type SEARCH_STATUS_INDICATOR is (FAILURE,
                                 SUCCESS);

-- This type provides for indicating the result of a search.
```

SMALL_NATURAL

```
subtype SMALL_NATURAL is
        PHIGS_NATURAL range 0..MAX_SMALL_NATURAL;

-- This is a subtype declaration which allows for
-- unconstrained record objects for various record types
-- without causing the exception STORAGE_ERROR to be raised.
```

STRING_DEVICE_NUMBER

```
type STRING_DEVICE_NUMBER is new DEVICE_NUMBER;

-- Provides for string device identifiers.
```

STRING_PROMPT_ECHO_TYPE

```
type STRING_PROMPT_ECHO_TYPE is new PHIGS_INTEGER;

-- Defines the string prompt and echo types.
```

STRING_PROMPT_ECHO_TYPES

```
package STRING_PROMPT_ECHO_TYPES is
        new PHIGS_LIST_UTILITIES
            (STRING_PROMPT_ECHO_TYPE,
             MAX_STRING_PROMPT_ECHO_TYPES_SUPPORTED);

-- Provides for lists of string prompt and echo types.
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

---

STRING_SMALL_NATURAL

```
subtype STRING_SMALL_NATURAL is
        PHIGS_NATURAL range 0..MAX_STRING_SMALL_NATURAL;

-- This is a subtype declaration which allows for unconstrained
-- record objects for various string record types without
-- causing the exception STORAGE_ERROR to be raised.
```

STROKE_DEVICE_NUMBER

```
type STROKE_DEVICE_NUMBER is new DEVICE_NUMBER;

-- Provides for stroke device identifiers.
```

STROKE_PROMPT_ECHO_TYPE

```
type STROKE_PROMPT_ECHO_TYPE is new PHIGS_INTEGER;

-- Defines the stroke prompt and echo types.
```

STROKE_PROMPT_ECHO_TYPES

```
package STROKE_PROMPT_ECHO_TYPES is
        new PHIGS_LIST_UTILITIES
                (STROKE_PROMPT_ECHO_TYPE
                 MAX_STROKE_PROMPT_ECHO_TYPES_SUPPORTED);

-- Provides for lists of stroke prompt and echo types
```

STRUCTURE_ELEMENT_RECORD

```
type STRUCTURE_ELEMENT_RECORD
        (ELEMENT_TYPE : STRUCTURE_ELEMENT_TYPE := NIL) is
    record
      case ELEMENT_TYPE is

        -- The empty element

        when NIL = >
          null;

        -- Primitive elements

        when POLYLINE_3 = >
          POLYLINE_3_POINTS : MC.ACCESS_POINT_LIST_3;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
when POLYLINE = >
   POLYLINE_POINTS : MC.ACCESS_POINT_LIST_2;

when POLYMARKER_3 = >
   POLYMARKER_3_POINTS : MC.ACCESS_POINT_LIST_3;

when POLYMARKER = >
   POLYMARKER_POINTS : MC.ACCESS_POINT_LIST_2;

when TEXT_3 = >
   TEXT_3_POINT : MC.POINT_3;
   TEXT_DIRECTION_VECTORS : MC.VECTOR_PAIR_3;
   TEXT_3_CHAR_STRING : ACCESS_STRING;

when TEXT = >
   TEXT_POINT : MC.POINT_2;
   TEXT_CHAR_STRING : ACCESS_STRING;

when ANNOTATION_TEXT_RELATIVE_3 = >
   ANNOTATION_TEXT_RELATIVE_3_REF_POINT : MC.POINT_3;
   ANNOTATION_TEXT_RELATIVE_3_OFFSET : NPC.POINT_3;
   ANNOTATION_TEXT_3_CHAR_STRING : ACCESS_STRING;

when ANNOTATION_TEXT_RELATIVE = >
   ANNOTATION_TEXT_RELATIVE_REF_POINT : MC.POINT_2;
   ANNOTATION_TEXT_RELATIVE_OFFSET : NPC.POINT_2;
   ANNOTATION_TEXT_RELATIVE_CHAR_STRING : ACCESS_STRING;

when FILL_AREA_3 = >
   FILL_AREA_3_POINTS : MC.ACCESS_POINT_LIST_3;

when FILL_AREA = >
   FILL_AREA_POINTS : MC.ACCESS_POINT_LIST_2;

when FILL_AREA_SET_3 = >
   FILL_AREA_SET_3_POINT_LISTS : MC.ACCESS_LIST_OF_POINT_LIST_3;

when FILL_AREA_SET = >
   FILL_AREA_SET_POINT_LISTS : MC.ACCESS_LIST_OF_POINT_LIST_2;

when CELL_ARRAY_3 = >
   CORNER_P_3 : MC.POINT_3;
   CORNER_Q_3 : MC.POINT_3;
   CORNER_R_3 : MC.POINT_3;
   CELL_ARRAY_3_CELLS : ACCESS_COLOUR_MATRIX;

when CELL_ARRAY = >
   CORNER_P : MC.POINT_2;
   CORNER_Q : MC.POINT_2;
   CELL_ARRAY_CELLS : ACCESS_COLOUR_MATRIX;

when GDP_3 = >
   GDP_3_POINTS : MC.ACCESS_POINT_LIST_3;
   GDP_3_DATA : GDP_3_RECORD;

when GDP = >
   GDP_POINTS : MC.ACCESS_POINT_LIST_2;
   GDP_DATA : GDP_RECORD;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
-- Bundle index elements

when SET_POLYLINE_INDEX = >
   POLYLINE_IND : POLYLINE_INDEX;

when SET_POLYMARKER_INDEX = >
   POLYMARKER_IND : POLYMARKER_INDEX;

when SET_TEXT_INDEX = >
   TEXT_IND : TEXT_INDEX;

when SET_INTERIOR_INDEX = >
   INTERIOR_IND : INTERIOR_INDEX;

when SET_EDGE_INDEX = >
   EDGE_IND : EDGE_INDEX;

-- Individual aspect elements

when SET_LINETYPE = >
   TYPE_OF_LINE : LINETYPE;

when SET_LINEWIDTH_SCALE_FACTOR = >
   LINEWIDTH_SF : LINEWIDTH;

when SET_POLYLINE_COLOUR_INDEX = >
   LINE_COLOUR : COLOUR_INDEX;

when SET_MARKER_TYPE = >
   TYPE_OF_MARKER : MARKER_TYPE;

when SET_MARKER_SIZE_SCALE_FACTOR = >
   SIZE : MARKER_SIZE;

when SET_POLYMARKER_COLOUR_INDEX = >
   MARKER_COLOUR : COLOUR_INDEX;

when SET_TEXT_FONT = >
   FONT : TEXT_FONT;

when SET_TEXT_PRECISION = >
   PRECISION : TEXT_PRECISION;

when SET_CHAR_EXPANSION_FACTOR = >
   EXPANSION : CHAR_EXPANSION;

when SET_CHAR_SPACING = >
   SPACING : CHAR_SPACING;

when SET_TEXT_COLOUR_INDEX = >
   TEXT_COLOUR : COLOUR_INDEX;

when SET_CHAR_HEIGHT = >
   HEIGHT : MC.MAGNITUDE;

when SET_CHAR_UP_VECTOR = >
   CHAR_UP_VECTOR : MC.VECTOR_2;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
when SET_TEXT_PATH = >
   PATH : TEXT_PATH;

when SET_TEXT_ALIGNMENT = >
   ALIGNMENT : TEXT_ALIGNMENT;

when SET_ANNOTATION_TEXT_CHAR_HEIGHT = >
   ANNOTATION_HEIGHT : NPC.MAGNITUDE;

when SET_ANNOTATION_TEXT_CHAR_UP_VECTOR = >
   ANNOTATION_CHAR_UP_VECTOR : NPC.VECTOR_2;

when SET_ANNOTATION_TEXT_PATH = >
   ANNOTATION_PATH : TEXT_PATH;

when SET_ANNOTATION_TEXT_ALIGNMENT = >
   ANNOTATION_ALIGNMENT : TEXT_ALIGNMENT;

when SET_ANNOTATION_STYLE = >
   STYLE_OF_ANNOTATION : ANNOTATION_STYLE;

when SET_INTERIOR_STYLE = >
   STYLE_OF_INTERIOR : INTERIOR_STYLE;

when SET_INTERIOR_STYLE_INDEX = >
   STYLE_IND : STYLE_INDEX;

when SET_INTERIOR_COLOUR_INDEX = >
   INTERIOR_COLOUR : COLOUR_INDEX;

when SET_EDGE_FLAG = >
   FLAG : EDGE_FLAG;

when SET_EDGETYPE = >
   TYPE_OF_EDGE : EDGETYPE;

when SET_EDGEWIDTH_SCALE_FACTOR = >
   EDGEWIDTH_SF : EDGEWIDTH;

when SET_EDGE_COLOUR_INDEX = >
   EDGE_COLOUR : COLOUR_INDEX;

-- Pattern attribute elements

when SET_PATTERN_SIZE = >
   PATTERN_SIZE : MC.SIZE_2;

when SET_PATTERN_REFERENCE_POINT_AND_VECTORS = >
   PATTERN_REFERENCE_POINT_3 : MC.POINT_3;
   PATTERN_REFERENCE_VECTORS : MC.VECTOR_PAIR_3;

when SET_PATTERN_REFERENCE_POINT = >
   PATTERN_REFERENCE_POINT : MC.POINT_2;

-- Name set elements

when ADD_NAMES_TO_SET = >
   NAMES_TO_ADD : NAME_SET;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
when REMOVE_NAMES_FROM_SET = >
   NAMES_TO_REMOVE : NAME_SET;

-- ASF elements

when SET_INDIVIDUAL_ASF = >
   ATTRIBUTE_ID : ASPECT;
   SOURCE_FLAG : ASF;

-- HLHSR elements

when SET_HLHSR_IDENTIFIER = >
   HLHSR_IDENTIFIER : HLHSR_ID;

-- Transformation elements

when SET_LOCAL_TRANSFORMATION_3 = >
   LOCAL_MATRIX_3 : TRANSFORMATION_MATRIX_3;
   HOW_APPLIED_3 : COMPOSITION_TYPE;

when SET_LOCAL_TRANSFORMATION = >
   LOCAL_MATRIX : TRANSFORMATION_MATRIX_2;
   HOW_APPLIED : COMPOSITION_TYPE;

when SET_GLOBAL_TRANSFORMATION_3 = >
   GLOBAL_MATRIX_3 : TRANSFORMATION_MATRIX_3;

when SET_GLOBAL_TRANSFORMATION = >
   GLOBAL_MATRIX : TRANSFORMATION_MATRIX_2;

when SET_MODELLING_CLIPPING_VOLUME_3 = >
   MODELLING_CLIPPING_OPERATOR_3 : MODELLING_CLIP_OPERATION_TYPE;
   MODELLING_CLIPPING_LIMITS_3 : MC.ACCESS_HALF_SPACE_LIST_3;

when SET_MODELLING_CLIPPING_VOLUME = >
   MODELLING_CLIPPING_OPERATOR : MODELLING_CLIP_OPERATION_TYPE;
   MODELLING_CLIPPING_LIMITS : MC.ACCESS_HALF_SPACE_LIST_2;

when SET_MODELLING_CLIPPING_INDICATOR = >
   MODELLING_CLIPPING_INDICATOR : CLIPPING_INDICATOR;

when RESTORE_MODELLING_CLIPPING_VOLUME = >
   null;

when SET_VIEW_INDEX = >
   VIEW_IND : VIEW_INDEX;

-- Invocation elements

when EXECUTE_STRUCTURE = >
   STRUCTURE_IDENTIFIER : STRUCTURE_ID;

-- Structure content identification elements

when LABEL = >
   LABEL_IDENTIFIER : LABEL_ID;

when APPLICATION_DATA = >
   DATA : APPLICATION_DATA_RECORD;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
            when GSE = >
              GSE_DATA : GSE_RECORD;

            when SET_PICK_IDENTIFIER = >
              PICK_IDENTIFIER : PICK_ID;

          end case;
        end record;

    -- This type defines the format of structure contents and
    -- is used to return structure elements during inquiry.
```

STRUCTURE_ELEMENT_TYPE

```
    subtype STRUCTURE_ELEMENT_TYPE is
            ELEMENT_TYPE range NIL..ELEMENT_TYPE'LAST;

    -- Provides for identifying the types of individual
    -- structure elements.
```

STRUCTURE_ID

```
    type STRUCTURE_ID is new PHIGS_NATURAL;

    -- Defines a type for structure identifiers.
```

STRUCTURE_IDS

```
    package STRUCTURE_IDS is
            new PHIGS_LIST_UTILITIES (STRUCTURE_ID,
                                      MAX_STRUCTURE_IDS_SUPPORTED);

    -- Provides for lists of structure identifiers.
```

STRUCTURE NETWORK SOURCE

```
    type STRUCTURE_NETWORK_SOURCE is (CSS,
                                      ARCHIVE);

    -- Provides for indicating source for check of conflicting
    -- structure identifiers.
```

STRUCTURE_STATE

```
    type STRUCTURE_STATE is (STCL,
                             STOP);

    -- The type used to return the structure state value.
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

STRUCTURE_STATUS

```
type STRUCTURE_STATUS is (NON_EXISTENT,
                          EMPTY,
                          NOTEMPTY);

-- The type used to return the structure status.
```

STYLE_INDEX

```
type STYLE_INDEX is new PHIGS_INTEGER;

-- Defines an interior style index.
```

SUBPROGRAM_NAME

```
subtype SUBPROGRAM_NAME is PHIGS_STRING;

-- Defines the name of a PHIGS function detecting an error.
```

SYSTEM_STATE

```
type SYSTEM_STATE is (PHCL,
                      PHOP);

-- The type used to return the system state value.
```

TEXT_ALIGNMENT

```
type TEXT_ALIGNMENT is
        record
          HORIZONTAL : HORIZONTAL_ALIGNMENT;
          VERTICAL   : VERTICAL_ALIGNMENT;
        end record;

-- The type of the attribute controlling the positioning of
-- the text extent parallelogram in relation to the text
-- position, having horizontal and vertical components as
-- defined above.
```

TEXT_FONT

```
type TEXT_FONT is new PHIGS_INTEGER;

-- Defines the types of fonts provided by the implementation.
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

---

TEXT_FONT_PRECISION

```
type TEXT_FONT_PRECISION is
        record
          FONT        : TEXT_FONT;
          PRECISION   : TEXT_PRECISION;
        end record;

-- This type defines a record describing the text font and
-- precision supported.
```

TEXT_FONT_PRECISIONS

```
package TEXT_FONT_PRECISIONS is
            new PHIGS_LIST_UTILITIES
                    (TEXT_FONT_PRECISION,
                     MAX_TEXT_FONT_PRECISION_PAIRS_SUPPORTED);

-- Provides for lists of text font and precision pairs.
```

TEXT_INDEX

```
type TEXT_INDEX is new PHIGS_POSITIVE;

-- Defines the range of text bundle table indices.
```

TEXT_INDICES

```
package TEXT_INDICES is
            new PHIGS_LIST_UTILITIES (TEXT_INDEX,
                                      MAX_TEXT_INDICES_SUPPORTED);

-- Provides for lists of text indices.
```

TEXT_PATH

```
type TEXT_PATH is (RIGHT,
                   LEFT,
                   UP,
                   DOWN);

-- The direction taken by a text string.
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

---

TEXT_PRECISION

```
type TEXT_PRECISION is (STRING_PRECISION,
                        CHAR_PRECISION,
                        STROKE_PRECISION);

-- The precision with which text appears.
```

TRANSFORMATION_MATRIX_2

```
type TRANSFORMATION_MATRIX_2 is
      array (1..3, 1..3) of MC_TYPE;

-- For 2D modelling and view transformation matrices.
```

TRANSFORMATION_MATRIX_3

```
type TRANSFORMATION_MATRIX_3 is
      array (1..4, 1..4) of MC_TYPE;

-- For 3D modelling and view transformation matrices.
```

UPDATE_REGENERATION_FLAG

```
type UPDATE_REGENERATION_FLAG is (PERFORM,
                                  POSTPONE);

-- Flag indicating regeneration action on display.
```

UPDATE_STATE

```
type UPDATE_STATE is (NOTPENDING,
                      PENDING);

-- Indicates whether or not a workstation transformation
-- change has been requested and not yet provided.
```

VALUATOR_DEVICE_NUMBER

```
type VALUATOR_DEVICE_NUMBER is new DEVICE_NUMBER;

-- Provides for valuator device identifiers.
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

---

VALUATOR_PROMPT_ECHO_TYPE

```
type VALUATOR_PROMPT_ECHO_TYPE is new PHIGS_INTEGER;

-- Defines the possible range of valuator prompt and
-- echo types.
```

VALUATOR_PROMPT_ECHO_TYPES

```
package VALUATOR_PROMPT_ECHO_TYPES is
        new PHIGS_LIST_UTILITIES
                (VALUATOR_PROMPT_ECHO_TYPE,
                 MAX_VALUATOR_PROMPT_ECHO_TYPES_SUPPORTED);

-- Provides for lists of valuator prompt and echo types.
```

VALUATOR_VALUE

```
type VALUATOR_VALUE is digits PHIGS_PRECISION;

-- Defines the range of accuracy of valuator input values
-- on an implementation.
```

VARIABLE_CONNECTION_ID

```
type VARIABLE_CONNECTION_ID (LENGTH : STRING_SMALL_NATURAL := 0) is
        record
          CONNECT : CONNECTION_ID (1..LENGTH);
        end record;

-- Defines a variable length connection identifier for
-- INQ_WS_CONNECTION_AND_TYPE.
```

VARIABLE_FILE_ID

```
type VARIABLE_FILE_ID (LENGTH : FILE_SMALL_NATURAL := 0) is
        record
          NAME : FILE_ID (1..LENGTH);
        end record;

-- Provides for variable length file identifiers.
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

VARIABLE_FILE_IDS

```
package VARIABLE_FILE_IDS is
        new PHIGS_LIST_UTILITIES(VARIABLE_FILE_ID,
                                 MAX_FILE_IDS_SUPPORTED);

-- This type is used to define lists of file identifiers.
```

VERTICAL_ALIGNMENT

```
type VERTICAL_ALIGNMENT is (NORMAL,
                            TOP,
                            CAP,
                            HALF,
                            BASE,
                            BOTTOM);

-- The alignment of the text extent parallelogram with
-- respect to the vertical positioning of the text.
```

VIEW_INDEX

```
type VIEW_INDEX is new PHIGS_NATURAL;

-- Defines a type for specifying view indices.
```

VIEW_INDICES

```
package VIEW_INDICES is
        new PHIGS_LIST_UTILITIES (VIEW_INDEX,
                                  MAX_VIEW_INDICES_SUPPORTED);

-- Provides for lists of view indices.
```

VISUAL_REPRESENTATION_STATE

```
type VISUAL_REPRESENTATION_STATE is (CORRECT,
                                     DEFERRED,
                                     SIMULATED);

-- Specifies state of image on display surface.
```

VRC

```
package VRC is new PHIGS_COORDINATE_SYSTEM (VRC_TYPE);

-- Defines the view reference coordinate system.
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

---

VRC_TYPE

```
type VRC_TYPE is digits PHIGS_PRECISION;

-- Defines the type of view reference coordinate components.
```

WC

```
package WC is new PHIGS_COORDINATE_SYSTEM (WC_TYPE);

-- Defines the world coordinate system.
```

WC_TYPE

```
type WC_TYPE is digits PHIGS_PRECISION;

-- Defines the type of coordinate in the World Coordinate
-- System.
```

WS_CATEGORY

```
type WS_CATEGORY is (OUTPUT,
                     INPUT,
                     OUTIN,
                     MO,
                     MI);

-- Type for PHIGS workstation categories.
```

WS_DEPENDENCY

```
type WS_DEPENDENCY is (WS_INDEPENDENT,
                       WS_DEPENDENT);

-- Type for PHIGS workstation categories.
```

WS_DEPENDENCIES

```
package WS_DEPENDENCIES is
        new PHIGS_LIST_UTILITIES (WS_DEPENDENCY,
                                  MAX_GSE_IDS_SUPPORTED);

-- Provides for lists of workstation dependencies.
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

---

WS_ID

```
type WS_ID is new PHIGS_POSITIVE;

-- Defines the range of workstation identifiers.
```

WS_IDS

```
package WS_IDS is new PHIGS_LIST_UTILITIES (WS_ID,
                                            MAX_OPEN_WS_SUPPORTED);

-- Provides for lists of workstation identifiers.
```

WS_STATE

```
type WS_STATE is (WSCL,
                  WSOP);

-- The type used to return the workstation state value.
```

WS_TYPE

```
type WS_TYPE is new PHIGS_POSITIVE;

-- Range of values corresponding to valid workstation types.
-- Constants specifying names for the various types of
-- workstations should be provided by an implementation.
```

WS_TYPES

```
package WS_TYPES is new PHIGS_LIST_UTILITIES (WS_TYPE,
                                              MAX_WS_TYPES_SUPPORTED);

-- Provides for lists of workstation types.
```

4.2.3 Alphabetical List of Private Type Definitions

This section contains an alphabetical listing of all of the private type definitions used to define the PHIGS binding to Ada. All of these elements are Ada PRIVATE type declarations. These declarations except for NAME_SET are included in the PHIGS package to facilitate the manipulations of the private types. The NAME_SET private type is declared as part of the PHIGS_NAME_SET_FACILITY package.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

---

CHOICE_DATA_RECORD

```
type CHOICE_DATA_RECORD
      (PROMPT_ECHO_TYPE : CHOICE_PROMPT_ECHO_TYPE := 1) is private;

-- Defines a record for initialising choice input. The
-- structure of the record is implementation-defined. Since
-- it is a private type, the components of the record may be
-- retrieved only through the use of the subprograms for
-- manipulating the input data records (5.14.1).
```

LOCATOR_DATA_RECORD

```
type LOCATOR_DATA_RECORD
      (PROMPT_ECHO_TYPE : LOCATOR_PROMPT_ECHO_TYPE := 1) is private;

-- Defines a locator input data record.
```

METAFILE_DATA_RECORD

```
type METAFILE_DATA_RECORD
      (TYPE_OF_ITEM : METAFILE_ITEM_TYPE := 0
       LENGTH       : NATURAL := 0) is private;

-- A data record for metafiles.
```

NAME_SET

```
type NAME_SET is private;

-- Defines a name set data element.
```

PICK_DATA_RECORD

```
type PICK_DATA_RECORD
      (PROMPT_ECHO_TYPE : PICK_PROMPT_ECHO_TYPE := 1) is private;

-- Defines a record for initialising pick input. The
-- structure of the record is implementation-defined. Since
-- it is a private type, the components of the record may be
-- retrieved only through the use of the subprograms for
-- manipulating the input data records ( 5.14.1).
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

---

STRING_DATA_RECORD

```
type STRING_DATA_RECORD
      (PROMPT_ECHO_TYPE : STRING_PROMPT_ECHO_TYPE := 1) is private;

-- Defines a string input data record.
```

STROKE_DATA_RECORD

```
type STROKE_DATA_RECORD
      (PROMPT_ECHO_TYPE : STROKE_PROMPT_ECHO_TYPE := 1) is private;

-- Defines a stroke input data record.
```

VALUATOR_DATA_RECORD

```
type VALUATOR_DATA_RECORD
      (PROMPT_ECHO_TYPE : VALUATOR_PROMPT_ECHO_TYPE := 1) is private;

-- Defines a valuator input data record.
```

4.2.4 List of Constant Declarations

This section contains the declarations of implementation dependent constants for defining PHIGS/Ada types. Some of the constants are used for defining default parameter values for PHIGS procedures defined in 5.0. This section also contains the constants that provide the PHIGS standard values defined for some PHIGS/Ada types. The following constants define the PHIGS standard linetypes:

```
SOLID_LINE              : constant LINETYPE := 1;
DASHED_LINE             : constant LINETYPE := 2;
DOTTED_LINE             : constant LINETYPE := 3;
DASHED_DOTTED_LINE      : constant LINETYPE := 4;
```

The following constants define the PHIGS standard marker types:

```
DOT_MARKER              : constant MARKER_TYPE := 1;
PLUS_MARKER             : constant MARKER_TYPE := 2;
STAR_MARKER             : constant MARKER_TYPE := 3;
ZERO_MARKER             : constant MARKER_TYPE := 4;
X_MARKER                : constant MARKER_TYPE := 5;
```

The following constants define the PHIGS standard edgetypes:

```
SOLID_EDGE              : constant EDGETYPE := 1;
DASHED_EDGE             : constant EDGETYPE := 2;
DOTTED_EDGE             : constant EDGETYPE := 3;
DASHED_DOTTED_EDGE      : constant EDGETYPE := 4;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

The following constants define the PHIGS standard annotation styles:

```
UNCONNECTED_ANNOTATION      : constant ANNOTATION_STYLE := 1;
LEAD_LINE_ANNOTATION        : constant ANNOTATION_STYLE := 2;
```

The following constants define the colour models specified by PHIGS:

```
RGB                         : constant COLOUR_MODEL := 1;
CIELUV                      : constant COLOUR_MODEL := 2;
HSV                         : constant COLOUR_MODEL := 3;
HLS                         : constant COLOUR_MODEL := 4;
```

The following constants define the prompt and echo types specified by PHIGS:

```
DEFAULT_LOCATOR             : constant LOCATOR_PROMPT_ECHO_TYPE := 1;
CROSS_HAIR_LOCATOR          : constant LOCATOR_PROMPT_ECHO_TYPE := 2;
TRACKING_CROSS_LOCATOR      : constant LOCATOR_PROMPT_ECHO_TYPE := 3;
RUBBER_BAND_LINE_LOCATOR    : constant LOCATOR_PROMPT_ECHO_TYPE := 4;
RECTANGLE_LOCATOR           : constant LOCATOR_PROMPT_ECHO_TYPE := 5;
DIGITAL_LOCATOR             : constant LOCATOR_PROMPT_ECHO_TYPE := 6;

DEFAULT_STROKE              : constant STROKE_PROMPT_ECHO_TYPE := 1;
DIGITAL_STROKE              : constant STROKE_PROMPT_ECHO_TYPE := 2;
MARKER_STROKE               : constant STROKE_PROMPT_ECHO_TYPE := 3;
LINE_STROKE                 : constant STROKE_PROMPT_ECHO_TYPE := 4;

DEFAULT_VALUATOR            : constant VALUATOR_PROMPT_ECHO_TYPE := 1;
GRAPHICAL_VALUATOR          : constant VALUATOR_PROMPT_ECHO_TYPE := 2;
DIGITAL_VALUATOR            : constant VALUATOR_PROMPT_ECHO_TYPE := 3;

DEFAULT_CHOICE              : constant CHOICE_PROMPT_ECHO_TYPE := 1;
PROMPT_ECHO_CHOICE          : constant CHOICE_PROMPT_ECHO_TYPE := 2;
STRING_PROMPT_CHOICE        : constant CHOICE_PROMPT_ECHO_TYPE := 3;
STRING_INPUT_CHOICE         : constant CHOICE_PROMPT_ECHO_TYPE := 4;
STRUCTURE_CHOICE            : constant CHOICE_PROMPT_ECHO_TYPE := 5;

DEFAULT_PICK                : constant PICK_PROMPT_ECHO_TYPE := 1;
GROUP_HIGHLIGHT_PICK        : constant PICK_PROMPT_ECHO_TYPE := 2;
STRUCTURE_HIGHLIGHT_PICK    : constant PICK_PROMPT_ECHO_TYPE := 3;
```

The following constants are used for defining the modelling clipping operators specified by PHIGS:

```
REPLACE_VOLUME              : constant MODELLING_CLIP_OPERATION_TYPE := 1;
INTERSECT_VOLUME            : constant MODELLING_CLIP_OPERATION_TYPE := 2;
```

4.2.5 PHIGS Configuration Values

To facilitate the configuration of PHIGS implementations, all implementation-dependent values are specified in package PHIGS_CONFIGURATION. When this package is **with**ed, these values become accessible. The names by which the values are accessed shall be contained within the visible part of package PHIGS_CONFIGURATION. There are three kinds of values which may be represented in this package:

- PHIGS base type characteristics
- The name of the default error file
- Range limits for various PHIGS types.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

The precision of the PHIGS floating point base types is a named number defined in the visible portion of the PHIGS_CONFIGURATION package specification:

PHIGS_PRECISION

The range limits of PHIGS integer base types are obtained from names which specify the largest negative and largest positive integers to be represented. These names are defined in the visible portion of the PHIGS_CONFIGURATION package specification:

MIN_INTEGER_VALUE
MAX_INTEGER_VALUE

The name DEFAULT_ERROR_FILE specifies the default name of the PHIGS error file. This name is a string constant defined in the visible portion of the PHIGS_CONFIGURATION package specification.

The remaining names may be any Ada construct which can represent a NATURAL value. Such constructs include constant declarations, variables, and parameterless functions which return integers. The body of the PHIGS_CONFIGURATION package may be used to compute these values and/or provide initial values. An implementation may permit the application programmer to replace the body of the PHIGS_CONFIGURATION package. Instructions for doing this shall be provided in the implementation documentation, if allowed by an implementation.

The following name is used for defining default parameter values for PHIGS procedures defined in 5.0:

DEFAULT_MEMORY_UNITS

The following name is used for defining the default list size for the MAX_LIST_SIZE parameter of the generic PHIGS_LIST_UTILITIES package:

DEFAULT_LIST_SIZE

The following names are used to define the maximum values for various "..._SMALL_NATURAL" data types defined in the binding:

MAX_CHOICE_SMALL_NATURAL
MAX_COLOUR_MATRIX_SMALL_NATURAL
MAX_FILE_SMALL_NATURAL
MAX_INPUT_STRING_SMALL_NATURAL
MAX_SMALL_NATURAL
MAX_STRING_SMALL_NATURAL

The following name is used to define the maximum length of an application data record:

MAX_APPLICATION_DATA

The following names are used to specify the maximum number of entities of the indicated type which are supported by the implementation:

MAX_ANNOTATION_STYLES_SUPPORTED
MAX_ARCHIVE_IDS_SUPPORTED
MAX_CHAR_SETS_SUPPORTED
MAX_CHOICE_PROMPTS_SUPPORTED
MAX_CHOICE_PROMPT_ECHO_TYPES_SUPPORTED
MAX_COLOUR_INDICES_SUPPORTED
MAX_COLOUR_MODELS_SUPPORTED
MAX_EDGE_INDICES_SUPPORTED
MAX_EDGETYPES_SUPPORTED
MAX_ESCAPE_IDS_SUPPORTED
MAX_FILE_IDS_SUPPORTED

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

MAX_GDP_3_IDS_SUPPORTED
MAX_GDP_IDS_SUPPORTED
MAX_GSE_IDS_SUPPORTED
MAX_HATCH_STYLES_SUPPORTED
MAX_HLHSR_IDS_SUPPORTED
MAX_HLHSR_MODES_SUPPORTED
MAX_INTERIOR_INDICES_SUPPORTED
MAX_LINETYPES_SUPPORTED
MAX_LOCATOR_PROMPT_ECHO_TYPES_SUPPORTED
MAX_MARKER_TYPES_SUPPORTED
MAX_MODELLING_CLIP_OPERATION_TYPES_SUPPORTED
MAX_MODELLING_CLIP_DISTINCT_PLANES_SUPPORTED
MAX_NAME_SUPPORTED
MAX_NAME_SET_FILTER_LIST_LENGTH_SUPPORTED
MAX_OPEN_WS_SUPPORTED
MAX_PATH_DEPTH_SUPPORTED
MAX_PATTERN_INDICES_SUPPORTED
MAX_PICK_IDS_SUPPORTED
MAX_PICK_PROMPT_ECHO_TYPES_SUPPORTED
MAX_POLYLINE_INDICES_SUPPORTED
MAX_POLYMARKER_INDICES_SUPPORTED
MAX_POSTED_STRUCTURES_SUPPORTED
MAX_REFERENCE_PATHS_SUPPORTED
MAX_STRING_PROMPT_ECHO_TYPES_SUPPORTED
MAX_STROKE_PROMPT_ECHO_TYPES_SUPPORTED
MAX_STRUCTURE_IDS_SUPPORTED
MAX_TEXT_FONT_PRECISION_PAIRS_SUPPORTED
MAX_TEXT_INDICES_SUPPORTED
MAX_VALUATOR_PROMPT_ECHO_TYPES_SUPPORTED
MAX_VIEW_INDICES_SUPPORTED
MAX_WS_IDS_SUPPORTED
MAX_WS_TYPES_SUPPORTED

The following is used to define the maximum memory units which may be allocated:

MAX_MEMORY_UNITS

The following name is used to specify the maximum number of components which can be represented in the generalized colour representation:

MAX_COLOUR_COEFFICIENTS

The following name is used to specify the length of the maximum metafile item which can be handled:

MAX_METAFILE_ITEM_LENGTH

The following names are used to define the range of values acceptable as metafile item types:

MIN_METAFILE_ITEM_TYPE
MAX_METAFILE_ITEM_TYPE

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

4.3 Error Codes

This binding requires the use of the PHIGS procedure ERROR_HANDLING to process any errors that occur in PHIGS procedures, except the inquiry procedures. A complete description of the error handling requirements of this binding is available in 3.2.3 of this binding.

The PHIGS inquiry functions do not generate errors. Instead, they return an error indicator parameter containing the number of the "error" which was detected. This is consistent with the PHIGS philosophy that no errors occur during inquiries. The error numbers correspond to the error numbers from Appendix C of the PHIGS specification, plus additional errors defined in this binding. Note that certain known error conditions may be detected outside the control of PHIGS due to the nature of the Ada language, and may result in an exception being raised on an inquiry.

4.3.1 Precluded Error Codes

The following PHIGS errors are listed separately because due to some feature of the Ada language or its use by this binding, they could never be detected by the PHIGS implementation. The errors might be detected by the Ada compiler, or at run-time outside the scope of PHIGS.

100 Ignoring function, the bundle index value is less than one.
112 Ignoring function, the pattern index value is less than one.
113 Ignoring function, the colour index value is less than zero.
114 Ignoring function, the view index value is less than zero.
115 Ignoring function, the view index value is less than one.
116 Ignoring function, one of the dimensions of pattern colour array is less than one.
117 Ignoring function, one of the dimensions of the colour index array is less than zero.
118 Ignoring function, one of the components of the colour specification is out of range. The valid range is dependent upon the current colour model.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0575450 TT1 ■

5 Functions in the Ada Binding of PHIGS

5.1 Control functions

OPEN PHIGS

```
procedure OPEN_PHIGS
  (ERROR_FILE          : in FILE_ID := PHIGS_STRING(DEFAULT_ERROR_FILE);
   AMOUNT_OF_MEMORY    : in PHIGS_NATURAL := DEFAULT_MEMORY_UNITS);
```

CLOSE PHIGS

```
procedure CLOSE_PHIGS;
```

OPEN WORKSTATION

```
procedure OPEN_WS
  (WS              : in WS_ID;
   CONNECTION      : in CONNECTION_ID;
   TYPE_OF_WS      : in WS_TYPE);
```

CLOSE WORKSTATION

```
procedure CLOSE_WS
  (WS : in WS_ID);
```

REDRAW ALL STRUCTURES

```
procedure REDRAW_ALL_STRUCTURES
  (WS     : in WS_ID;
   FLAG   : in CONTROL_FLAG);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

UPDATE WORKSTATION

```
procedure UPDATE_WS
  (WS                 : in WS_ID;
   REGENERATION       : in UPDATE_REGENERATION_FLAG);
```

SET DISPLAY UPDATE STATE

```
procedure SET_DISPLAY_UPDATE_STATE
  (WS                 : in WS_ID;
   DEFERRAL           : in DEFERRAL_MODE;
   MODIFICATION       : in MODIFICATION_MODE);
```

MESSAGE

```
procedure MESSAGE
  (WS                 : in WS_ID;
   CONTENTS           : in PHIGS_STRING);
```

5.2 Output primitive functions

POLYLINE 3

```
procedure POLYLINE
  (POINTS : in MC.POINT_LIST_3);
```

POLYLINE

```
procedure POLYLINE
  (POINTS : in MC.POINT_LIST_2);
```

POLYMARKER 3

```
procedure POLYMARKER
  (POINTS : in MC.POINT_LIST_3);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0575452 874 ■

POLYMARKER

```
procedure POLYMARKER
  (POINTS : in MC.POINT_LIST_2);
```

TEXT 3

```
procedure TEXT
  (POSITION          : in MC.POINT_3;
   DIRECTION_VECTORS : in MC.VECTOR_PAIR_3;
   CHAR_STRING       : in PHIGS_STRING);
```

TEXT

```
procedure TEXT
  (POSITION    : in MC.POINT_2;
   CHAR_STRING : in PHIGS_STRING);
```

ANNOTATION TEXT RELATIVE 3

```
procedure ANNOTATION_TEXT_RELATIVE
  (REFERENCE_POINT   : in MC.POINT_3;
   ANNOTATION_OFFSET : in NPC.POINT_3;
   CHAR_STRING       : in PHIGS_STRING);
```

ANNOTATION TEXT RELATIVE

```
procedure ANNOTATION_TEXT_RELATIVE
  (REFERENCE_POINT   : in MC.POINT_2;
   ANNOTATION_OFFSET : in NPC.POINT_2;
   CHAR_STRING       : in PHIGS_STRING);
```

FILL AREA 3

```
procedure FILL_AREA
  (POINTS : in MC.POINT_LIST_3);
```

FILL AREA

```
procedure FILL_AREA
  (POINTS : in MC.POINT_LIST_2);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

FILL AREA SET 3

```
procedure FILL_AREA_SET
  (POINT_LISTS : in MC.LIST_OF_POINT_LIST_3);
```

FILL AREA SET

```
procedure FILL_AREA_SET
  (POINT_LISTS : in MC.LIST_OF_POINT_LIST_2);
```

CELL ARRAY 3

```
procedure CELL_ARRAY
  (CORNER_P : in MC.POINT_3;
   CORNER_Q : in MC.POINT_3;
   CORNER_R : in MC.POINT_3;
   CELLS    : in COLOUR_MATRIX);
```

CELL ARRAY

```
procedure CELL_ARRAY
  (CORNER_P : in MC.POINT_2;
   CORNER_Q : in MC.POINT_2;
   CELLS    : in COLOUR_MATRIX);
```

GENERALIZED DRAWING PRIMITIVE 3
GENERALIZED DRAWING PRIMITIVE

The Generalized Drawing Primitive (GDP) is bound in a one-to-many fashion, with a separate procedure implemented for each GDP, each with its own parameter interface. GDP names and parameters are registered in the ISO International Register of Graphical Items which is maintained by the Registration Authority. All supported registered GDP's are in a library package named PHIGS_GDP.

The minimal support for GDP's required by this standard is that defined in the following specification for the PHIGS_GDP package. The minimal support provides an interface to registered and unregistered GDP's not supported by the implementation. Unsupported GDP's may be inserted in structures using either form of the procedure GENERALIZED_GDP whose external specifications are shown.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
with PHIGS_TYPES;
use PHIGS_TYPES;
package PHIGS_GDP is
  type GDP_INTEGER_ARRAY is array (SMALL_NATURAL range <>) of
                                  INTEGER;
  type GDP_FLOAT_ARRAY   is array (SMALL_NATURAL range <>) of
                                  GENERAL_FLOAT_PARAMETER;
  type GDP_STRING_ARRAY  is array (SMALL_NATURAL range <>) of
                                  PHIGS_STRING(1..80);

  type GDP_DATA_RECORD (NUM_OF_INTEGERS : SMALL_NATURAL := 0;
                        NUM_OF_REALS    : SMALL_NATURAL := 0;
                        NUM_OF_STRINGS  : SMALL_NATURAL := 0) is

    record
      INTEGER_ARRAY : GDP_INTEGER_ARRAY (1..NUM_OF_INTEGERS);
      REAL_ARRAY    : GDP_FLOAT_ARRAY   (1..NUM_OF_REALS);
      GDP_STRINGS   : GDP_STRING_ARRAY  (1..NUM_OF_STRINGS);
    end record;

procedure GENERALIZED_GDP
  (GDP         : in GDP_3_ID;
   POINTS      : in MC.POINT_LIST_3;
   GDP_DATA    : in GDP_DATA_RECORD);

procedure GENERALIZED_GDP
  (GDP         : in GDP_ID;
   POINTS      : in MC.POINT_LIST_2;
   GDP_DATA    : in GDP_DATA_RECORD);

end PHIGS_GDP;
```

Each registered GDP procedure supported by the implementation will be included in package PHIGS_GDP as a separate procedure using the following naming convention which results in references to the GDP by the expression PHIGS_GDP.<name of the GDP procedure>. Specific names will be assigned when the GDP is registered.

```
procedure <name of the GDP procedure> (<parameters as required>);
```

Each unregistered GDP procedure supported by the implementation will be a library package using the following naming convention:

```
package PHIGS_UGDP_<name of the GDP procedure> is
  procedure GDP;
    -- Ada code for UGDP procedure

  -- The only procedure name used in the package
  -- will be GDP.

end PHIGS_UGDP_<name of the GDP procedure>;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

5.3 Attribute specification functions

SET POLYLINE INDEX

```
procedure SET_POLYLINE_INDEX
  (POLYLINE_IND : in POLYLINE_INDEX);
```

SET POLYMARKER INDEX

```
procedure SET_POLYMARKER_INDEX
  (POLYMARKER_IND : in POLYMARKER_INDEX);
```

SET TEXT INDEX

```
procedure SET_TEXT_INDEX
  (TEXT_IND : in TEXT_INDEX);
```

SET INTERIOR INDEX

```
procedure SET_INTERIOR_INDEX
  (INTERIOR_IND : in INTERIOR_INDEX);
```

SET EDGE INDEX

```
procedure SET_EDGE_INDEX
  (EDGE_IND : in EDGE_INDEX);
```

SET LINETYPE

```
procedure SET_LINETYPE
  (TYPE_OF_LINE : in LINETYPE);
```

SET LINEWIDTH SCALE FACTOR

```
procedure SET_LINEWIDTH_SCALE_FACTOR
  (LINEWIDTH_SF : in LINEWIDTH);
```

SET POLYLINE COLOUR INDEX

```
procedure SET_POLYLINE_COLOUR_INDEX
  (LINE_COLOUR : in COLOUR_INDEX);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

9593 PT*3 90 ■ 4851903 0575456 41T ■

SET MARKER TYPE

```
procedure SET_MARKER_TYPE
  (TYPE_OF_MARKER : in MARKER_TYPE);
```

SET MARKER SIZE SCALE FACTOR

```
procedure SET_MARKER_SIZE_SCALE_FACTOR
  (SIZE : in MARKER_SIZE);
```

SET POLYMARKER COLOUR INDEX

```
procedure SET_POLYMARKER_COLOUR_INDEX
  (MARKER_COLOUR : in COLOUR_INDEX);
```

SET TEXT FONT

```
procedure SET_TEXT_FONT
  (FONT : in TEXT_FONT);
```

SET TEXT PRECISION

```
procedure SET_TEXT_PRECISION
  (PRECISION : in TEXT_PRECISION);
```

SET CHARACTER EXPANSION FACTOR

```
procedure SET_CHAR_EXPANSION_FACTOR
  (EXPANSION : in CHAR_EXPANSION);
```

SET CHARACTER SPACING

```
procedure SET_CHAR_SPACING
  (SPACING : in CHAR_SPACING);
```

SET TEXT COLOUR INDEX

```
procedure SET_TEXT_COLOUR_INDEX
  (TEXT_COLOUR : in COLOUR_INDEX);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

SET CHARACTER HEIGHT

```
procedure SET_CHAR_HEIGHT
  (HEIGHT : in MC.MAGNITUDE);
```

SET CHARACTER UP VECTOR

```
procedure SET_CHAR_UP_VECTOR
  (CHAR_UP_VECTOR : in MC.VECTOR_2);
```

SET TEXT PATH

```
procedure SET_TEXT_PATH
  (PATH : in TEXT_PATH);
```

SET TEXT ALIGNMENT

```
procedure SET_TEXT_ALIGNMENT
  (ALIGNMENT : in TEXT_ALIGNMENT);
```

SET ANNOTATION TEXT CHARACTER HEIGHT

```
procedure SET_ANNOTATION_TEXT_CHAR_HEIGHT
  (ANNOTATION_HEIGHT : in NPC.MAGNITUDE);
```

SET ANNOTATION TEXT CHARACTER UP VECTOR

```
procedure SET_ANNOTATION_TEXT_CHAR_UP_VECTOR
  (ANNOTATION_CHAR_UP_VECTOR : in NPC.VECTOR_2);
```

SET ANNOTATION TEXT PATH

```
procedure SET_ANNOTATION_TEXT_PATH
  (ANNOTATION_PATH : in TEXT_PATH);
```

SET ANNOTATION TEXT ALIGNMENT

```
procedure SET_ANNOTATION_TEXT_ALIGNMENT
  (ANNOTATION_ALIGNMENT : in TEXT_ALIGNMENT);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0575458 292 ■

SET ANNOTATION STYLE

```
procedure SET_ANNOTATION_STYLE
  (STYLE_OF_ANNOTATION : in ANNOTATION_STYLE);
```

SET INTERIOR STYLE

```
procedure SET_INTERIOR_STYLE
  (STYLE_OF_INTERIOR : in INTERIOR_STYLE);
```

SET INTERIOR STYLE INDEX

```
procedure SET_INTERIOR_STYLE_INDEX
  (STYLE_IND : in STYLE_INDEX);
```

SET INTERIOR COLOUR INDEX

```
procedure SET_INTERIOR_COLOUR_INDEX
  (INTERIOR_COLOUR : in COLOUR_INDEX);
```

SET EDGE FLAG

```
procedure SET_EDGE_FLAG
  (FLAG : in EDGE_FLAG);
```

SET EDGETYPE

```
procedure SET_EDGETYPE
  (TYPE_OF_EDGE : in EDGETYPE);
```

SET EDGEWIDTH SCALE FACTOR

```
procedure SET_EDGEWIDTH_SCALE_FACTOR
  (EDGEWIDTH_SF : in EDGEWIDTH);
```

SET EDGE COLOUR INDEX

```
procedure SET_EDGE_COLOUR_INDEX
  (EDGE_COLOUR : in COLOUR_INDEX);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

SET PATTERN SIZE

```
procedure SET_PATTERN_SIZE
  (SIZE : in MC.SIZE_2);
```

SET PATTERN REFERENCE POINT AND VECTORS

```
procedure SET_PATTERN_REFERENCE_POINT_AND_VECTORS
  (REFERENCE_POINT    : in MC.POINT_3;
   REFERENCE_VECTORS  : in MC.VECTOR_PAIR_3);
```

SET PATTERN REFERENCE POINT

```
procedure SET_PATTERN_REFERENCE_POINT
  (REFERENCE_POINT : MC.POINT_2);
```

ADD NAMES TO SET

```
procedure ADD_NAMES_TO_SET
  (NAMES_TO_ADD : in NAME_SET);
```

REMOVE NAMES FROM SET

```
procedure REMOVE_NAMES_FROM_SET
  (NAMES_TO_REMOVE : in NAME_SET);
```

SET INDIVIDUAL ASF

```
procedure SET_INDIVIDUAL_ASF
  (ASPECT_ID      : in ASPECT;
   SOURCE_FLAG    : in ASF);
```

SET POLYLINE REPRESENTATION

```
procedure SET_POLYLINE_REPRESENTATION
  (WS               : in WS_ID;
   POLYLINE_IND     : in POLYLINE_INDEX;
   TYPE_OF_LINE     : in LINETYPE;
   LINEWIDTH_SF     : in LINEWIDTH;
   LINE_COLOUR      : in COLOUR_INDEX);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0575460 940 ■

SET POLYMARKER REPRESENTATION

```
procedure SET_POLYMARKER_REPRESENTATION
  (WS                : in WS_ID;
   POLYMARKER_IND    : in POLYMARKER_INDEX;
   TYPE_OF_MARKER    : in MARKER_TYPE;
   SIZE              : in MARKER_SIZE;
   MARKER_COLOUR     : in COLOUR_INDEX);
```

SET TEXT REPRESENTATION

```
procedure SET_TEXT_REPRESENTATION
  (WS                : in WS_ID;
   TEXT_IND          : in TEXT_INDEX;
   FONT              : in TEXT_FONT;
   PRECISION         : in TEXT_PRECISION;
   EXPANSION         : in CHAR_EXPANSION;
   SPACING           : in CHAR_SPACING;
   TEXT_COLOUR       : in COLOUR_INDEX);
```

SET INTERIOR REPRESENTATION

```
procedure SET_INTERIOR_REPRESENTATION
  (WS                : in WS_ID;
   INTERIOR_IND      : in INTERIOR_INDEX;
   STYLE_OF_INTERIOR : in INTERIOR_STYLE;
   STYLE_IND         : in STYLE_INDEX;
   INTERIOR_COLOUR   : in COLOUR_INDEX);
```

SET EDGE REPRESENTATION

```
procedure SET_EDGE_REPRESENTATION
  (WS                : in WS_ID;
   EDGE_IND          : in EDGE_INDEX;
   FLAG              : in EDGE_FLAG;
   TYPE_OF_EDGE      : in EDGETYPE;
   EDGEWIDTH_SF      : in EDGEWIDTH;
   EDGE_COLOUR       : in COLOUR_INDEX);
```

SET PATTERN REPRESENTATION

```
procedure SET_PATTERN_REPRESENTATION
  (WS                : in WS_ID;
   PATTERN_IND       : in PATTERN_INDEX;
   PATTERN           : in COLOUR_MATRIX);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

SET COLOUR REPRESENTATION

```
procedure SET_COLOUR_REPRESENTATION
  (WS              : in WS_ID;
   COLOUR_IND      : in COLOUR_INDEX;
   COLOUR          : in COLOUR_REPRESENTATION);
```

SET HIGHLIGHTING FILTER

```
procedure SET_HIGHLIGHTING_FILTER
  (WS                : in WS_ID;
   HIGHLIGHTING      : in NAME_SET_FILTER);
```

SET INVISIBILITY FILTER

```
procedure SET_INVISIBILITY_FILTER
  (WS                : in WS_ID;
   INVISIBILITY      : in NAME_SET_FILTER);
```

SET COLOUR MODEL

```
procedure SET_COLOUR_MODEL
  (WS        : in WS_ID;
   MODEL     : in COLOUR_MODEL);
```

SET HLHSR IDENTIFIER

```
procedure SET_HLHSR_IDENTIFIER
  (HLHSR_IDENTIFIER : in HLHSR_ID);
```

SET HLHSR MODE

```
procedure SET_HLHSR_MODE
  (WS        : WS_ID;
   MODE      : in HLHSR_MODE);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

9593 PT*3 90 ■ 4851903 0575462 713 ■

5.4 Transformation and clipping functions

SET LOCAL TRANSFORMATION 3

```
procedure SET_LOCAL_TRANSFORMATION
  (MATRIX          : in TRANSFORMATION_MATRIX_3;
   HOW_APPLIED     : in COMPOSITION_TYPE);
```

SET LOCAL TRANSFORMATION

```
procedure SET_LOCAL_TRANSFORMATION
  (MATRIX          : in TRANSFORMATION_MATRIX_2;
   HOW_APPLIED     : in COMPOSITION_TYPE);
```

SET GLOBAL TRANSFORMATION 3

```
procedure SET_GLOBAL_TRANSFORMATION
  (MATRIX : in TRANSFORMATION_MATRIX_3);
```

SET GLOBAL TRANFORMATION

```
procedure SET_GLOBAL_TRANSFORMATION
  (MATRIX : in TRANSFORMATION_MATRIX_2);
```

SET MODELLING CLIPPING VOLUME

```
procedure SET_MODELLING_CLIPPING_VOLUME
  (OPERATOR        : in MODELLING_CLIP_OPERATION_TYPE;
   HALF_SPACES     : in MC.HALF_SPACE_LIST_3);
```

SET MODELLING CLIPPING VOLUME

```
procedure SET_MODELLING_CLIPPING_VOLUME
  (OPERATOR        : in MODELLING_CLIP_OPERATION_TYPE;
   HALF_SPACES     : in MC.HALF_SPACE_LIST_2);
```

SET MODELLING CLIPPING INDICATOR

```
procedure SET_MODELLING_CLIPPING_INDICATOR
  (MODELLING_CLIPPING : in CLIPPING_INDICATOR);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

9593 PT*3 90 ■ 4851903 0575463 65T ■

RESTORE MODELLING CLIPPING VOLUME

```
procedure RESTORE_MODELLING_CLIPPING_VOLUME;
```

SET VIEW INDEX

```
procedure SET_VIEW_INDEX
  (VIEW_IND : in VIEW_INDEX);
```

SET VIEW REPRESENTATION 3

```
procedure SET_VIEW_REPRESENTATION
  (WS                  : in WS_ID;
   VIEW_IND            : in POSITIVE_VIEW_INDEX;
   ORIENTATION_MATRIX  : in TRANSFORMATION_MATRIX_3;
   MAPPING_MATRIX      : in TRANSFORMATION_MATRIX_3;
   CLIPPING_LIMITS     : in NPC.RECTANGULAR_REGION_3;
   XY_CLIPPING         : in CLIPPING_INDICATOR;
   BACK_CLIPPING       : in CLIPPING_INDICATOR;
   FRONT_CLIPPING      : in CLIPPING_INDICATOR);
```

SET VIEW REPRESENTATION

```
procedure SET_VIEW_REPRESENTATION
  (WS                  : in WS_ID;
   VIEW_IND            : in POSITIVE_VIEW_INDEX;
   ORIENTATION_MATRIX  : in TRANSFORMATION_MATRIX_2;
   MAPPING_MATRIX      : in TRANSFORMATION_MATRIX_2;
   CLIPPING_LIMITS     : in NPC.RECTANGULAR_REGION_2;
   XY_CLIPPING         : in CLIPPING_INDICATOR);
```

SET VIEW TRANSFORMATION INPUT PRIORITY

```
procedure SET_VIEW_TRANSFORMATION_INPUT_PRIORITY
  (WS                  : in WS_ID;
   VIEW_IND            : in VIEW_INDEX;
   REFERENCE_IND       : in VIEW_INDEX;
   PRIORITY            : in RELATIVE_PRIORITY);
```

SET WORKSTATION WINDOW 3

```
procedure SET_WS_WINDOW
  (WS                  : in WS_ID;
   WINDOW_LIMITS       : in NPC.RECTANGULAR_REGION_3);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

---

SET WORKSTATION WINDOW

```
procedure SET_WS_WINDOW
  (WS                : in WS_ID;
   WINDOW_LIMITS     : in NPC.RECTANGULAR_REGION_2);
```

SET WORKSTATION VIEWPORT 3

```
procedure SET_WS_VIEWPORT
  (WS                : in WS_ID;
   VIEWPORT_LIMITS   : in DC.RECTANGULAR_REGION_3);
```

SET WORKSTATION VIEWPORT

```
procedure SET_WS_VIEWPORT
  (WS                : in WS_ID;
   VIEWPORT_LIMITS   : in DC.RECTANGULAR_REGION_2);
```

TRANSLATE 3

```
procedure TRANSLATE
  (VECTOR            : in  MC.VECTOR_3;
   ERROR_INDICATOR   : out ERROR_NUMBER;
   MATRIX            : out TRANSFORMATION_MATRIX_3);
```

TRANSLATE

```
procedure TRANSLATE
  (VECTOR            : in  MC.VECTOR_2;
   ERROR_INDICATOR   : out ERROR_NUMBER;
   MATRIX            : out TRANSFORMATION_MATRIX_2);
```

SCALE 3

```
procedure SCALE
  (FACTOR            : in  MC.VECTOR_3;
   ERROR_INDICATOR   : out ERROR_NUMBER;
   MATRIX            : out TRANSFORMATION_MATRIX_3);
```

SCALE

```
procedure SCALE
  (FACTOR            : in  MC.VECTOR_2;
   ERROR_INDICATOR   : out ERROR_NUMBER;
   MATRIX            : out TRANSFORMATION_MATRIX_2);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

ROTATE X

```
procedure ROTATE_X
  (ANGLE_X           : in  ANGLE;
   ERROR_INDICATOR   : out ERROR_NUMBER;
   MATRIX            : out TRANSFORMATION_MATRIX_3);
```

ROTATE Y

```
procedure ROTATE_Y
  (ANGLE_Y           : in ANGLE;
   ERROR_INDICATOR   : out ERROR_NUMBER;
   MATRIX            : out TRANSFORMATION_MATRIX_3);
```

ROTATE Z

```
procedure ROTATE_Z
  (ANGLE_Z           : in  ANGLE;
   ERROR_INDICATOR   : out ERROR_NUMBER;
   MATRIX            : out TRANSFORMATION_MATRIX_3);
```

ROTATE

```
procedure ROTATE
  (ANGLE_Z           : in  ANGLE;
   ERROR_INDICATOR   : out ERROR_NUMBER;
   MATRIX            : out TRANSFORMATION_MATRIX_2);
```

COMPOSE MATRIX 3

```
procedure COMPOSE_MATRIX
  (MATRIX_A          : in  TRANSFORMATION_MATRIX_3;
   MATRIX_B          : in  TRANSFORMATION_MATRIX_3;
   ERROR_INDICATOR   : out ERROR_NUMBER;
   COMPOSED_MATRIX   : out TRANSFORMATION_MATRIX_3);
```

COMPOSE MATRIX

```
procedure COMPOSE_MATRIX
  (MATRIX_A          : in  TRANSFORMATION_MATRIX_2;
   MATRIX_B          : in  TRANSFORMATION_MATRIX_2;
   ERROR_INDICATOR   : out ERROR_NUMBER;
   COMPOSED_MATRIX   : out TRANSFORMATION_MATRIX_2);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0575466 369 ■

TRANSFORM POINT 3

```
procedure TRANSFORM_POINT
  (POINT                : in  MC.POINT_3;
   MATRIX               : in  TRANSFORMATION_MATRIX_3;
   ERROR_INDICATOR      : out ERROR_NUMBER;
   TRANSFORMED_POINT    : out MC.POINT_3);
```

TRANSFORM POINT

```
procedure TRANSFORM_POINT
  (POINT                : in  MC.POINT_2;
   MATRIX               : in  TRANSFORMATION_MATRIX_2;
   ERROR_INDICATOR      : out ERROR_NUMBER;
   TRANSFORMED_POINT    : out MC.POINT_2);
```

BUILD TRANSFORMATION MATRIX 3

```
procedure BUILD_TRANSFORMATION_MATRIX
  (FIXED_POINT          : in  MC.POINT_3;
   SHIFT_VECTOR         : in  MC.VECTOR_3;
   ANGLE_X              : in  ANGLE;
   ANGLE_Y              : in  ANGLE;
   ANGLE_Z              : in  ANGLE;
   SCALE_FACTORS        : in  MC.VECTOR_3;
   ERROR_INDICATOR      : out ERROR_NUMBER;
   MATRIX               : out TRANSFORMATION_MATRIX_3);
```

BUILD TRANSFORMATION MATRIX

```
procedure BUILD_TRANSFORMATION_MATRIX
  (FIXED_POINT          : in  MC.POINT_2;
   SHIFT_VECTOR         : in  MC.VECTOR_2;
   ANGLE_Z              : in  ANGLE;
   SCALE_FACTORS        : in  MC.VECTOR_2;
   ERROR_INDICATOR      : out ERROR_NUMBER;
   MATRIX               : out TRANSFORMATION_MATRIX_2);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

9593 PT*3 90 ■ 4851903 0575467 2T5 ■

COMPOSE TRANSFORMATION MATRIX 3

```
procedure COMPOSE_TRANSFORMATION_MATRIX
  (MATRIX              : in  TRANSFORMATION_MATRIX_3;
   FIXED_POINT         : in  MC.POINT_3;
   SHIFT_VECTOR        : in  MC.VECTOR_3;
   ANGLE_X             : in  ANGLE;
   ANGLE_Y             : in  ANGLE;
   ANGLE_Z             : in  ANGLE;
   SCALE_FACTORS       : in  MC.VECTOR_3;
   ERROR_INDICATOR     : out ERROR_NUMBER;
   COMPOSED_MATRIX     : out TRANSFORMATION_MATRIX_3);
```

COMPOSE TRANSFORMATION MATRIX

```
procedure COMPOSE_TRANSFORMATION_MATRIX
  (MATRIX              : in  TRANSFORMATION_MATRIX_2;
   FIXED_POINT         : in  MC.POINT_2;
   SHIFT_VECTOR        : in  MC.VECTOR_2;
   ANGLE_Z             : in  ANGLE;
   SCALE_FACTORS       : in  MC.VECTOR_2;
   ERROR_INDICATOR     : out ERROR_NUMBER;
   COMPOSED_MATRIX     : out TRANSFORMATION_MATRIX_2);
```

EVALUATE VIEW ORIENTATION MATRIX 3

```
procedure EVALUATE_VIEW_ORIENTATION_MATRIX
  (REFERENCE_POINT     : in  WC.POINT_3;
   NORMAL_VECTOR       : in  WC.VECTOR_3;
   UP_VECTOR           : in  WC.VECTOR_3;
   ERROR_INDICATOR     : out ERROR_NUMBER;
   ORIENTATION_MATRIX  : out TRANSFORMATION_MATRIX_3);
```

EVALUATE VIEW ORIENTATION MATRIX

```
procedure EVALUATE_VIEW_ORIENTATION_MATRIX
  (REFERENCE_POINT     : in  WC.POINT_2;
   UP_VECTOR           : in  WC.VECTOR_2;
   ERROR_INDICATOR     : out ERROR_NUMBER;
   ORIENTATION_MATRIX  : out TRANSFORMATION_MATRIX_2);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

---

EVALUATE VIEW MAPPING MATRIX 3

```
procedure EVALUATE_VIEW_MAPPING_MATRIX
  (WINDOW_LIMITS       : in  VRC.RECTANGULAR_REGION_2;
   VIEWPORT_LIMITS     : in  NPC.RECTANGULAR_REGION_3;
   TYPE_OF_PROJECTION  : in  PROJECTION_TYPE;
   REFERENCE_POINT     : in  VRC.POINT_3;
   VIEW_DISTANCE       : in  VRC_TYPE;
   BACK_DISTANCE       : in  VRC_TYPE;
   FRONT_DISTANCE      : in  VRC_TYPE;
   ERROR_INDICATOR     : out ERROR_NUMBER;
   MAPPING_MATRIX      : out TRANSFORMATION_MATRIX_3);
```

EVALUATE VIEW MAPPING MATRIX

```
procedure EVALUATE_VIEW_MAPPING_MATRIX
  (WINDOW_LIMITS       : in  VRC.RECTANGULAR_REGION_2;
   VIEWPORT_LIMITS     : in  NPC.RECTANGULAR_REGION_2;
   ERROR_INDICATOR     : out ERROR_NUMBER;
   MAPPING_MATRIX      : out TRANSFORMATION_MATRIX_2);
```

5.5 Structure content functions

OPEN STRUCTURE

```
procedure OPEN_STRUCTURE
  (STRUCTURE_IDENTIFIER : in STRUCTURE_ID);
```

CLOSE STRUCTURE

```
procedure CLOSE_STRUCTURE;
```

EXECUTE STRUCTURE

```
procedure EXECUTE_STRUCTURE
  (STRUCTURE_IDENTIFIER : in STRUCTURE_ID);
```

LABEL

```
procedure LABEL
  (LABEL_IDENTIFIER : in LABEL_ID);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

---

APPLICATION DATA

```
procedure APPLICATION_DATA
  (DATA : in APPLICATION_DATA_RECORD);
```

GENERALIZED STRUCTURE ELEMENT

The Generalized Structure Element (GSE) is bound in a one-to-many fashion, with a separate procedure implemented for each GSE, each with its own parameter interface. GSE names and parameters are registered in the ISO International Register of Graphical Items which is maintained by the Registration Authority. All supported registered GSE's are in a library package PHIGS_GSE.

The minimal support for GSE's required by this standard is that defined in the following specification for the PHIGS_GSE package. The minimal support provides an interface to registered and unregistered GSE's not supported by the implementation. Unsupported GSE's may be inserted in structures using the procedure GENERALIZED_GSE whose external specification is shown.

```
with PHIGS_TYPES;
use PHIGS_TYPES;
package PHIGS_GSE is
  type GSE_INTEGER_ARRAY is array (SMALL_NATURAL range <>) of
                                  INTEGER;
  type GSE_FLOAT_ARRAY   is array (SMALL_NATURAL range <>) of
                                  GENERAL_FLOAT_PARAMETER;
  type GSE_STRING_ARRAY  is array (SMALL_NATURAL range <>) of
                                  PHIGS_STRING(1..80);

  type GSE_DATA_RECORD (NUM_OF_INTEGERS : SMALL_NATURAL := 0;
                        NUM_OF_REALS    : SMALL_NATURAL := 0;
                        NUM_OF_STRINGS  : SMALL_NATURAL := 0) is

    record
      INTEGER_ARRAY : GSE_INTEGER_ARRAY (1..NUM_OF_INTEGERS);
      REAL_ARRAY    : GSE_FLOAT_ARRAY   (1..NUM_OF_REALS);
      GSE_STRINGS   : GSE_STRING_ARRAY  (1..NUM_OF_STRINGS);
    end record;

procedure GENERALIZED_GSE
  (GSE       : GSE_ID;
   GSE_DATA  : in GSE_DATA_RECORD);

end PHIGS_GSE;
```

Each registered GSE procedure supported by the implementation will be included in package PHIGS_GSE as a separate procedure using the following naming convention which results in references to the GSE by the expression PHIGS_GSE.<name of the GSE procedure>. Specific names will be assigned when the GSE is registered.

```
procedure <name of the GSE procedure> (<parameters as required>);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0575470 89T ■

Each unregistered GSE procedure will be a library package using the following naming convention:

```
package PHIGS_UGSE_<name of GSE procedure> is
  procedure GSE;
    -- Ada code for UGSE procedure

  -- The only procedure name used in the package
  -- will be GSE.

end PHIGS_UGSE_<name of the GSE procedure>;
```

SET EDIT MODE

```
procedure SET_EDIT_MODE
  (MODE : in EDIT_MODE);
```

COPY ALL ELEMENTS FROM STRUCTURE

```
procedure COPY_ALL_ELEMENTS_FROM_STRUCTURE
  (STRUCTURE_IDENTIFIER : in STRUCTURE_ID);
```

SET ELEMENT POINTER

```
procedure SET_ELEMENT_POINTER
  (POSITION : in ELEMENT_POSITION);
```

OFFSET ELEMENT POINTER

```
procedure OFFSET_ELEMENT_POINTER
  (OFFSET : in ELEMENT_POSITION);
```

SET ELEMENT POINTER AT LABEL

```
procedure SET_ELEMENT_POINTER_AT_LABEL
  (LABEL_IDENTIFIER : in LABEL_ID);
```

DELETE ELEMENT

```
procedure DELETE_ELEMENT;
```

DELETE ELEMENT RANGE

```
procedure DELETE_ELEMENT_RANGE
  (POSITION_1 : in ELEMENT_POSITION;
   POSITION_2 : in ELEMENT_POSITION);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

DELETE ELEMENTS BETWEEN LABELS

```
procedure DELETE_ELEMENTS_BETWEEN_LABELS
  (LABEL_IDENTIFIER_1 : in LABEL_ID;
   LABEL_IDENTIFIER_2 : in LABEL_ID);
```

EMPTY STRUCTURE

```
procedure EMPTY_STRUCTURE
  (STRUCTURE_IDENTIFIER : in STRUCTURE_ID);
```

5.6 Structure manipulation functions

DELETE STRUCTURE

```
procedure DELETE_STRUCTURE
  (STRUCTURE_IDENTIFIER : in STRUCTURE_ID);
```

DELETE STRUCTURE NETWORK

```
procedure DELETE_STRUCTURE_NETWORK
  (STRUCTURE_IDENTIFIER  : in STRUCTURE_ID;
   HANDLING_FLAG         : in REFERENCE_HANDLING_FLAG);
```

DELETE ALL STRUCTURES

```
procedure DELETE_ALL_STRUCTURES;
```

CHANGE STRUCTURE IDENTIFIER

```
procedure CHANGE_STRUCTURE_IDENTIFIER
  (ORIGINAL_IDENTIFIER   : in STRUCTURE_ID;
   RESULTING_IDENTIFIER  : in STRUCTURE_ID);
```

CHANGE STRUCTURE REFERENCES

```
procedure CHANGE_STRUCTURE_REFERENCES
  (ORIGINAL_IDENTIFIER   : in STRUCTURE_ID;
   RESULTING_IDENTIFIER  : in STRUCTURE_ID);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

9593 PT*3 90 ■ 4851903 0575472 662 ■

CHANGE STRUCTURE IDENTIFIER AND REFERENCES

```
procedure CHANGE_STRUCTURE_IDENTIFIER_AND_REFERENCES
  (ORIGINAL_IDENTIFIER   : in STRUCTURE_ID;
   RESULTING_IDENTIFIER  : in STRUCTURE_ID);
```

5.7 Structure display functions

POST STRUCTURE

```
procedure POST_STRUCTURE
  (WS                    : in WS_ID;
   STRUCTURE_IDENTIFIER  : in STRUCTURE_ID;
   PRIORITY              : in DISPLAY_PRIORITY);
```

UNPOST STRUCTURE

```
procedure UNPOST_STRUCTURE
  (WS                    : in WS_ID;
   STRUCTURE_IDENTIFIER  : in STRUCTURE_ID);
```

UNPOST ALL STRUCTURES

```
procedure UNPOST_ALL_STRUCTURES
  (WS : in WS_ID);
```

5.8 Structure archive functions

OPEN ARCHIVE FILE

```
procedure OPEN_ARCHIVE_FILE
  (ARCHIVE_IDENTIFIER : in ARCHIVE_ID;
   ARCHIVE_FILE       : in FILE_ID);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

CLOSE ARCHIVE FILE

```
procedure CLOSE_ARCHIVE_FILE
  (ARCHIVE_IDENTIFIER : in ARCHIVE_ID);
```

ARCHIVE STRUCTURES

```
procedure ARCHIVE_STRUCTURES
  (ARCHIVE_IDENTIFIER      : in ARCHIVE_ID;
   STRUCTURE_IDENTIFIERS : in STRUCTURE_IDS.LIST_OF);
```

ARCHIVE STRUCTURE NETWORKS

```
procedure ARCHIVE_STRUCTURE_NETWORKS
  (ARCHIVE_IDENTIFIER      : in ARCHIVE_ID;
   STRUCTURE_IDENTIFIERS : in STRUCTURE_IDS.LIST_OF);
```

ARCHIVE ALL STRUCTURES

```
procedure ARCHIVE_ALL_STRUCTURES
  (ARCHIVE_IDENTIFIER : in ARCHIVE_ID);
```

SET CONFLICT RESOLUTION

```
procedure SET_CONFLICT_RESOLUTION
  (ARCHIVAL_CONFLICT   : in CONFLICT_RESOLUTION;
   RETRIEVAL_CONFLICT : in CONFLICT_RESOLUTION);
```

RETRIEVE STRUCTURE IDENTIFIERS

```
procedure RETRIEVE_STRUCTURE_IDENTIFIERS
  (ARCHIVE_IDENTIFIER      : in ARCHIVE_ID;
   STRUCTURE_IDENTIFIERS : out STRUCTURE_IDS.LIST_OF);
```

RETRIEVE STRUCTURES

```
procedure RETRIEVE_STRUCTURES
  (ARCHIVE_IDENTIFIER      : in ARCHIVE_ID;
   STRUCTURE_IDENTIFIERS : in STRUCTURE_IDS.LIST_OF);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

---

RETRIEVE STRUCTURE NETWORKS

```
procedure RETRIEVE_STRUCTURE_NETWORKS
  (ARCHIVE_IDENTIFIER      : in ARCHIVE_ID;
   STRUCTURE_IDENTIFIERS   : in STRUCTURE_IDS.LIST_OF);
```

RETRIEVE ALL STRUCTURES

```
procedure RETRIEVE_ALL_STRUCTURES
  (ARCHIVE_IDENTIFIER : in ARCHIVE_ID);
```

RETRIEVE PATHS TO ANCESTORS

```
procedure RETRIEVE_PATHS_TO_ANCESTORS
  (ARCHIVE_IDENTIFIER      : in ARCHIVE_ID;
   STRUCTURE_IDENTIFIER    : in STRUCTURE_ID;
   ORDER                   : in PATH_ORDER;
   DEPTH                   : in PATH_DEPTH;
   LIST_OF_PATHS           : out REFERENCE_PATHS.LIST_OF);
```

RETRIEVE PATHS TO DESCENDANTS

```
procedure RETRIEVE_PATHS_TO_DESCENDANTS
  (ARCHIVE_IDENTIFIER      : in ARCHIVE_ID;
   STRUCTURE_IDENTIFIER    : in STRUCTURE_ID;
   ORDER                   : in PATH_ORDER;
   DEPTH                   : in PATH_DEPTH;
   LIST_OF_PATHS           : out REFERENCE_PATHS.LIST_OF);
```

DELETE STRUCTURES FROM ARCHIVE

```
procedure DELETE_STRUCTURES_FROM_ARCHIVE
  (ARCHIVE_IDENTIFIER      : in ARCHIVE_ID;
   STRUCTURE_IDENTIFIERS   : in STRUCTURE_IDS.LIST_OF);
```

DELETE STRUCTURE NETWORKS FROM ARCHIVE

```
procedure DELETE_STRUCTURE_NETWORKS_FROM_ARCHIVE
  (ARCHIVE_IDENTIFIER      : in ARCHIVE_ID;
   STRUCTURE_IDENTIFIERS   : in STRUCTURE_IDS.LIST_OF);
```

DELETE ALL STRUCTURES FROM ARCHIVE

```
procedure DELETE_ALL_STRUCTURES_FROM_ARCHIVE
  (ARCHIVE_IDENTIFIER : in ARCHIVE_ID);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

5.9 Input functions

SET PICK IDENTIFIER

```
procedure SET_PICK_IDENTIFIER
  (PICK_IDENTIFIER : in PICK_ID);
```

SET PICK FILTER

```
procedure SET_PICK_FILTER
  (WS              : in WS_ID;
   PICK_DEVICE     : in PICK_DEVICE_NUMBER;
   PICKABILITY     : in NAME_SET_FILTER);
```

INITIALIZE LOCATOR 3

```
procedure INITIALIZE_LOCATOR
  (WS                  : in WS_ID;
   DEVICE              : in LOCATOR_DEVICE_NUMBER;
   INITIAL_VIEW_IND    : in VIEW_INDEX;
   INITIAL_POSITION    : in WC.POINT_3;
   ECHO_VOLUME         : in DC.RECTANGULAR_REGION_3;
   DATA_RECORD         : in LOCATOR_DATA_RECORD);
```

INITIALIZE LOCATOR

```
procedure INITIALIZE_LOCATOR
  (WS                  : in WS_ID;
   DEVICE              : in LOCATOR_DEVICE_NUMBER;
   INITIAL_VIEW_IND    : in VIEW_INDEX;
   INITIAL_POSITION    : in WC.POINT_2;
   ECHO_AREA           : in DC.RECTANGULAR_REGION_2;
   DATA_RECORD         : in LOCATOR_DATA_RECORD);
```

INITIALIZE STROKE 3

```
procedure INITIALIZE_STROKE
  (WS                  : in WS_ID;
   DEVICE              : in STROKE_DEVICE_NUMBER;
   INITIAL_VIEW_IND    : in VIEW_INDEX;
   INITIAL_STROKE      : in WC.POINT_LIST_3;
   ECHO_VOLUME         : in DC.RECTANGULAR_REGION_3;
   DATA_RECORD         : in STROKE_DATA_RECORD);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

INITIALIZE STROKE

```
procedure INITIALIZE_STROKE
  (WS                : in WS_ID;
   DEVICE            : in STROKE_DEVICE_NUMBER;
   INITIAL_VIEW_IND  : in VIEW_INDEX;
   INITIAL_STROKE    : in WC.POINT_LIST_2;
   ECHO_AREA         : in DC.RECTANGULAR_REGION_2;
   DATA_RECORD       : in STROKE_DATA_RECORD);
```

INITIALIZE VALUATOR 3

```
procedure INITIALIZE_VALUATOR
  (WS                : in WS_ID;
   DEVICE            : in VALUATOR_DEVICE_NUMBER;
   INITIAL_VALUE     : in VALUATOR_VALUE;
   ECHO_VOLUME       : in DC.RECTANGULAR_REGION_3;
   DATA_RECORD       : in VALUATOR_DATA_RECORD);
```

INITIALIZE VALUATOR

```
procedure INITIALIZE_VALUATOR
  (WS                : in WS_ID;
   DEVICE            : in VALUATOR_DEVICE_NUMBER;
   INITIAL_VALUE     : in VALUATOR_VALUE;
   ECHO_AREA         : in DC.RECTANGULAR_REGION_2;
   DATA_RECORD       : in VALUATOR_DATA_RECORD);
```

INITIALIZE CHOICE 3

```
procedure INITIALIZE_CHOICE
  (WS                : in WS_ID;
   DEVICE            : in CHOICE_DEVICE_NUMBER;
   INITIAL_STATUS    : in CHOICE_STATUS;
   INITIAL_CHOICE    : in CHOICE_NUMBER;
   ECHO_VOLUME       : in DC.RECTANGULAR_REGION_3;
   DATA_RECORD       : in CHOICE_DATA_RECORD);
```

INITIALIZE CHOICE

```
procedure INITIALIZE_CHOICE
  (WS                : in WS_ID;
   DEVICE            : in CHOICE_DEVICE_NUMBER;
   INITIAL_STATUS    : in CHOICE_STATUS;
   INITIAL_CHOICE    : in CHOICE_NUMBER;
   ECHO_AREA         : in DC.RECTANGULAR_REGION_2;
   DATA_RECORD       : in CHOICE_DATA_RECORD);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

---

INITIALIZE PICK 3

```
procedure INITIALIZE_PICK
  (WS             : in WS_ID;
   DEVICE         : in PICK_DEVICE_NUMBER;
   INITIAL_STATUS : in PICK_STATUS;
   INITIAL_PATH   : in PICK_PATH;
   ECHO_VOLUME    : in DC.RECTANGULAR_REGION_3;
   DATA_RECORD    : in PICK_DATA_RECORD;
   ORDER          : in PATH_ORDER);
```

INITIALIZE PICK

```
procedure INITIALIZE_PICK
  (WS             : in WS_ID;
   DEVICE         : in PICK_DEVICE_NUMBER;
   INITIAL_STATUS : in PICK_STATUS;
   INITIAL_PATH   : in PICK_PATH;
   ECHO_AREA      : in DC.RECTANGULAR_REGION_2;
   DATA_RECORD    : in PICK_DATA_RECORD;
   ORDER          : in PATH_ORDER);
```

INITIALIZE STRING 3

```
procedure INITIALIZE_STRING
  (WS             : in WS_ID;
   DEVICE         : in STRING_DEVICE_NUMBER;
   INITIAL_STRING : in PHIGS_STRING;
   ECHO_VOLUME    : in DC.RECTANGULAR_REGION_3;
   DATA_RECORD    : in STRING_DATA_RECORD);
```

INITIALIZE STRING

```
procedure INITIALIZE_STRING
  (WS             : in WS_ID;
   DEVICE         : in STRING_DEVICE_NUMBER;
   INITIAL_STRING : in PHIGS_STRING;
   ECHO_AREA      : in DC.RECTANGULAR_REGION_2;
   DATA_RECORD    : in STRING_DATA_RECORD);
```

SET LOCATOR MODE

```
procedure SET_LOCATOR_MODE
  (WS     : in WS_ID;
   DEVICE : in LOCATOR_DEVICE_NUMBER;
   MODE   : in OPERATING_MODE;
   SWITCH : in ECHO_SWITCH);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0575478 080 ■

SET STROKE MODE

```
procedure SET_STROKE_MODE
  (WS        : in WS_ID;
   DEVICE    : in STROKE_DEVICE_NUMBER;
   MODE      : in OPERATING_MODE;
   SWITCH    : in ECHO_SWITCH);
```

SET VALUATOR MODE

```
procedure SET_VALUATOR_MODE
  (WS        : in WS_ID;
   DEVICE    : in VALUATOR_DEVICE_NUMBER;
   MODE      : in OPERATING_MODE;
   SWITCH    : in ECHO_SWITCH);
```

SET CHOICE MODE

```
procedure SET_CHOICE_MODE
  (WS        : in WS_ID;
   DEVICE    : in CHOICE_DEVICE_NUMBER;
   MODE      : in OPERATING_MODE;
   SWITCH    : in ECHO_SWITCH);
```

SET PICK MODE

```
procedure SET_PICK_MODE
  (WS        : in WS_ID;
   DEVICE    : in PICK_DEVICE_NUMBER;
   MODE      : in OPERATING_MODE;
   SWITCH    : in ECHO_SWITCH);
```

SET STRING MODE

```
procedure SET_STRING_MODE
  (WS        : in WS_ID;
   DEVICE    : in STRING_DEVICE_NUMBER;
   MODE      : in OPERATING_MODE;
   SWITCH    : in ECHO_SWITCH);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

REQUEST LOCATOR 3

```
procedure REQUEST_LOCATOR
  (WS          : in WS_ID;
   DEVICE      : in LOCATOR_DEVICE_NUMBER;
   STATUS      : out INPUT_STATUS;
   VIEW_IND    : out VIEW_INDEX;
   POSITION    : out WC.POINT_3);
```

REQUEST LOCATOR

```
procedure REQUEST_LOCATOR
  (WS          : in WS_ID;
   DEVICE      : in LOCATOR_DEVICE_NUMBER;
   STATUS      : out INPUT_STATUS;
   VIEW_IND    : out VIEW_INDEX;
   POSITION    : out WC.POINT_2);
```

REQUEST STROKE 3

```
procedure REQUEST_STROKE
  (WS                : in WS_ID;
   DEVICE            : in STROKE_DEVICE_NUMBER;
   STATUS            : out INPUT_STATUS;
   VIEW_IND          : out VIEW_INDEX;
   STROKE_POINTS     : out WC.ACCESS_POINT_LIST_3);
```

REQUEST STROKE

```
procedure REQUEST_STROKE
  (WS                : in WS_ID;
   DEVICE            : in STROKE_DEVICE_NUMBER;
   STATUS            : out INPUT_STATUS;
   VIEW_IND          : out VIEW_INDEX;
   STROKE_POINTS     : out WC.ACCESS_POINT_LIST_2);
```

REQUEST VALUATOR

```
procedure REQUEST_VALUATOR
  (WS        : in WS_ID;
   DEVICE    : in VALUATOR_DEVICE_NUMBER;
   STATUS    : out INPUT_STATUS;
   VALUE     : out VALUATOR_VALUE);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0575480 739 ■

REQUEST CHOICE

```
procedure REQUEST_CHOICE
  (WS      : in  WS_ID;
   DEVICE  : in  CHOICE_DEVICE_NUMBER;
   STATUS  : out CHOICE_REQUEST_STATUS;
   CHOICE  : out CHOICE_NUMBER);
```

REQUEST PICK

```
procedure REQUEST_PICK
  (WS                 : in  WS_ID;
   DEVICE             : in  PICK_DEVICE_NUMBER;
   DEPTH_TO_RETURN    : in  PATH_DEPTH;
   STATUS             : out PICK_REQUEST_STATUS;
   PATH               : out PICK_PATH);
```

REQUEST STRING

```
procedure REQUEST_STRING
  (WS          : in  WS_ID;
   DEVICE      : in  STRING_DEVICE_NUMBER;
   STATUS      : out INPUT_STATUS;
   CHAR_STRING : out INPUT_STRING);
```

SAMPLE LOCATOR 3

```
procedure SAMPLE_LOCATOR
  (WS       : in  WS_ID;
   DEVICE   : in  LOCATOR_DEVICE_NUMBER;
   VIEW_IND : out VIEW_INDEX;
   POSITION : out WC.POINT_3);
```

SAMPLE LOCATOR

```
procedure SAMPLE_LOCATOR
  (WS       : in  WS_ID;
   DEVICE   : in  LOCATOR_DEVICE_NUMBER;
   VIEW_IND : out VIEW_INDEX;
   POSITION : out WC.POINT_2);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

9593 PT*3 90 ■ 4851903 0575481 675 ■

SAMPLE STROKE 3

```
procedure SAMPLE_STROKE
  (WS                : in  WS_ID;
   DEVICE            : in  STROKE_DEVICE_NUMBER;
   VIEW_IND          : out VIEW_INDEX;
   STROKE_POINTS     : out WC.ACCESS_POINT_LIST_3);
```

SAMPLE STROKE

```
procedure SAMPLE_STROKE
  (WS                : in  WS_ID;
   DEVICE            : in  STROKE_DEVICE_NUMBER;
   VIEW_IND          : out VIEW_INDEX;
   STROKE_POINTS     : out WC.ACCESS_POINT_LIST_2);
```

SAMPLE VALUATOR

```
procedure SAMPLE_VALUATOR
  (WS       : in  WS_ID;
   DEVICE   : in  VALUATOR_DEVICE_NUMBER;
   VALUE    : out VALUATOR_VALUE);
```

SAMPLE CHOICE

```
procedure SAMPLE_CHOICE
  (WS       : in  WS_ID;
   DEVICE   : in  CHOICE_DEVICE_NUMBER;
   STATUS   : out CHOICE_STATUS;
   CHOICE   : out CHOICE_NUMBER);
```

SAMPLE PICK

```
procedure SAMPLE_PICK
  (WS                : in  WS_ID;
   DEVICE            : in  PICK_DEVICE_NUMBER;
   DEPTH_TO_RETURN   : in  PATH_DEPTH;
   STATUS            : out PICK_STATUS;
   PATH              : out PICK_PATH);
```

SAMPLE STRING

```
procedure SAMPLE_STRING
  (WS                : in  WS_ID;
   DEVICE            : in  STRING_DEVICE_NUMBER;
   CHAR_STRING       : out INPUT_STRING);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

---

AWAIT EVENT

```
procedure AWAIT_EVENT
  (TIMEOUT : in DURATION;
   WS      : out WS_ID;
   DEVICE  : out EVENT_DEVICE_NUMBER);
```

FLUSH DEVICE EVENTS

```
procedure FLUSH_DEVICE_EVENTS
  (WS     : in WS_ID;
   DEVICE : in EVENT_QUEUE_DEVICE_NUMBER);
```

GET LOCATOR 3

```
procedure GET_LOCATOR
  (VIEW_IND : out VIEW_INDEX;
   POSITION : out WC.POINT_3);
```

GET LOCATOR

```
procedure GET_LOCATOR
  (VIEW_IND : out VIEW_INDEX;
   POSITION : out WC.POINT_2);
```

GET STROKE 3

```
procedure GET_STROKE
  (VIEW_IND      : out VIEW_INDEX;
   STROKE_POINTS : out WC.ACCESS_POINT_LIST_3);
```

GET STROKE

```
procedure GET_STROKE
  (VIEW_IND      : out VIEW_INDEX;
   STROKE_POINTS : out WC.ACCESS_POINT_LIST_2);
```

GET VALUATOR

```
procedure GET_VALUATOR
  (VALUE : out VALUATOR_VALUE);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

GET CHOICE

```
procedure GET_CHOICE
  (STATUS  : out CHOICE_STATUS;
   CHOICE  : out CHOICE_NUMBER);
```

GET PICK

```
procedure GET_PICK
  (DEPTH_TO_RETURN : in  PATH_DEPTH;
   STATUS          : out PICK_STATUS;
   PATH            : out PICK_PATH);
```

GET STRING

```
procedure GET_STRING
  (CHAR_STRING : out INPUT_STRING);
```

5.10 Metafile functions

WRITE ITEM TO METAFILE

```
procedure WRITE_ITEM_TO_METAFILE
  (WS    : in WS_ID;
   ITEM  : in METAFILE_DATA_RECORD);
```

GET ITEM TYPE FROM METAFILE

```
procedure GET_ITEM_TYPE_FROM_METAFILE
  (WS              : in  WS_ID;
   TYPE_OF_ITEM    : out METAFILE_ITEM_TYPE;
   LENGTH          : out METAFILE_ITEM_LENGTH);
```

READ ITEM FROM METAFILE

```
procedure READ_ITEM_FROM_METAFILE
  (WS              : in  WS_ID;
   MAX_LENGTH      : in  METAFILE_ITEM_LENGTH;
   ITEM            : out METAFILE_DATA_RECORD);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0575484 384 ■

INTERPRET ITEM

```
procedure INTERPRET_ITEM
  (ITEM : in METAFILE_DATA_RECORD);
```

5.11 Inquiry functions

INQUIRE SYSTEM STATE VALUE

```
procedure INQ_SYSTEM_STATE_VALUE
  (STATE_VALUE : out SYSTEM_STATE);
```

INQUIRE WORKSTATION STATE VALUE

```
procedure INQ_WS_STATE_VALUE
  (STATE_VALUE : out WS_STATE);
```

INQUIRE STRUCTURE STATE VALUE

```
procedure INQ_STRUCTURE_STATE_VALUE
  (STATE_VALUE : out STRUCTURE_STATE);
```

INQUIRE ARCHIVE STATE VALUE

```
procedure INQ_ARCHIVE_STATE_VALUE
  (STATE_VALUE : out ARCHIVE_STATE);
```

INQUIRE LIST OF AVAILABLE WORKSTATION TYPES

```
procedure INQ_LIST_OF_AVAILABLE_WS_TYPES
  (ERROR_INDICATOR  : out ERROR_NUMBER;
   LIST_OF_TYPES    : out WS_TYPES.LIST_OF);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

INQUIRE PHIGS FACILITIES

```
procedure INQ_PHIGS_FACILITIES
  (ERROR_INDICATOR                : out ERROR_NUMBER;
   MAX_SIMUL_OPEN_WS              : out PHIGS_POSITIVE;
   MAX_SIMUL_OPEN_ARCHIVES        : out PHIGS_POSITIVE;
   NUMBER_AVAIL_NAMES             : out PHIGS_POSITIVE;
   AVAIL_CHAR_SETS                : out CHAR_SETS.LIST_OF;
   MAX_ISS_NORMAL_FILTER_LIST     : out PHIGS_POSITIVE;
   MAX_ISS_INVERTED_FILTER_LIST : out PHIGS_POSITIVE);
```

INQUIRE GENERALIZED STRUCTURE ELEMENT FACILITIES

```
procedure INQ_GSE_FACILITIES
  (ERROR_INDICATOR        : out ERROR_NUMBER;
   LIST_AVAIL_GSES        : out GSE_IDS.LIST_OF;
   LIST_WS_DEPENDENCIES : out WS_DEPENDENCIES.LIST_OF);
```

INQUIRE MODELLING CLIPPING FACILITIES

```
procedure INQ_MODELLING_CLIPPING_FACILITIES
  (ERROR_INDICATOR          : out ERROR_NUMBER;
   NUMBER_DISTINCT_PLANES : out PHIGS_POSITIVE;
   LIST_OF_OPERATIONS       : out MODELLING_CLIP_OPERATION_TYPES.LIST_OF);
```

INQUIRE EDIT MODE

```
procedure INQ_EDIT_MODE
  (ERROR_INDICATOR : out ERROR_NUMBER;
   MODE            : out EDIT_MODE);
```

INQUIRE SET OF OPEN WORKSTATIONS

```
procedure INQ_SET_OF_OPEN_WS
  (ERROR_INDICATOR : out ERROR_NUMBER;
   OPEN_WS         : out WS_IDS.LIST_OF);
```

INQUIRE STRUCTURE IDENTIFIERS

```
procedure INQ_STRUCTURE_IDENTIFIERS
  (ERROR_INDICATOR     : out ERROR_NUMBER;
   LIST_OF_STRUCTURES : out STRUCTURE_IDS.LIST_OF);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0575486 157 ■

INQUIRE ARCHIVE FILES

```
procedure INQ_ARCHIVE_FILES
  (ERROR_INDICATOR          : out ERROR_NUMBER;
   LIST_ARCHIVE_IDENTIFIERS : out ARCHIVE_IDS.LIST_OF;
   LIST_ARCHIVE_FILES       : out VARIABLE_FILE_IDS.LIST_OF);
```

INQUIRE_CONFLICT_RESOLUTION

```
procedure INQ_CONFLICT_RESOLUTION
  (ERROR_INDICATOR     : out ERROR_NUMBER;
   ARCHIVAL_CONFLICT   : out CONFLICT_RESOLUTION;
   RETRIEVAL_CONFLICT  : out CONFLICT_RESOLUTION);
```

INQUIRE ALL CONFLICTING STRUCTURES

```
procedure INQ_ALL_CONFLICTING_STRUCTURES
  (ARCHIVE_IDENTIFIER  : in  ARCHIVE_ID;
   ERROR_INDICATOR     : out ERROR_NUMBER;
   LIST_OF_STRUCTURES  : out STRUCTURE_IDS.LIST_OF);
```

INQUIRE CONFLICTING STRUCTURES IN NETWORK

```
procedure INQ_CONFLICTING_STRUCTURES_IN_NETWORK
  (ARCHIVE_IDENTIFIER     : in  ARCHIVE_ID;
   STRUCTURE_IDENTIFIER   : in  STRUCTURE_ID;
   SOURCE                 : in  STRUCTURE_NETWORK_SOURCE;
   ERROR_INDICATOR        : out ERROR_NUMBER;
   LIST_OF_STRUCTURES     : out STRUCTURE_IDS.LIST_OF);
```

INQUIRE MORE SIMULTANEOUS EVENTS

```
procedure INQ_MORE_SIMULTANEOUS_EVENTS
  (ERROR_INDICATOR : out ERROR_NUMBER;
   EVENTS          : out MORE_EVENTS);
```

INQUIRE WORKSTATION CONNECTION AND TYPE

```
procedure INQ_WS_CONNECTION_AND_TYPE
  (WS              : in  WS_ID;
   ERROR_INDICATOR : out ERROR_NUMBER;
   CONNECTION      : out VARIABLE_CONNECTION_ID;
   TYPE_OF_WS      : out WS_TYPE);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

9593 PT*3 90 ■ 4851903 0575487 093 ■

INQUIRE LIST OF VIEW INDICES

```
procedure INQ_LIST_OF_VIEW_INDICES
  (WS                  : in  WS_ID;
   ERROR_INDICATOR     : out ERROR_NUMBER;
   DEFINED_VIEW_LIST   : out VIEW_INDICES.LIST_OF);
```

INQUIRE VIEW REPRESENTATION

```
procedure INQ_VIEW_REPRESENTATION
  (WS                                 : in  WS_ID;
   VIEW_IND                           : in  VIEW_INDEX;
   ERROR_INDICATOR                    : out ERROR_NUMBER;
   UPDATE                             : out UPDATE_STATE;
   REQUESTED_ORIENTATION_MATRIX       : out TRANSFORMATION_MATRIX_3;
   CURRENT_ORIENTATION_MATRIX         : out TRANSFORMATION_MATRIX_3;
   REQUESTED_MAPPING_MATRIX           : out TRANSFORMATION_MATRIX_3;
   CURRENT_MAPPING_MATRIX             : out TRANSFORMATION_MATRIX_3;
   REQUESTED_CLIPPING_LIMITS          : out NPC.RECTANGULAR_REGION_3;
   CURRENT_CLIPPING_LIMITS            : out NPC.RECTANGULAR_REGION_3;
   REQUESTED_XY_CLIPPING              : out CLIPPING_INDICATOR;
   CURRENT_XY_CLIPPING                : out CLIPPING_INDICATOR;
   REQUESTED_BACK_CLIPPING            : out CLIPPING_INDICATOR;
   CURRENT_BACK_CLIPPING              : out CLIPPING_INDICATOR;
   REQUESTED_FRONT_CLIPPING           : out CLIPPING_INDICATOR;
   CURRENT_FRONT_CLIPPING             : out CLIPPING_INDICATOR);
```

INQUIRE HLHSR MODE

```
procedure INQ_HLHSR_MODE
  (WS                  : in  WS_ID;
   ERROR_INDICATOR     : out ERROR_NUMBER;
   UPDATE              : out UPDATE_STATE;
   CURRENT_MODE        : out HLHSR_MODE;
   REQUESTED_MODE      : out HLHSR_MODE);
```

INQUIRE POSTED STRUCTURES

```
procedure INQ_POSTED_STRUCTURES
  (WS                  : in  WS_ID;
   ERROR_INDICATOR     : out ERROR_NUMBER;
   LIST_OF_STRUCTURES  : out POSTED_STRUCTURES.LIST_OF);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0575488 T2T ■

INQUIRE DISPLAY UPDATE STATE

```
procedure INQ_DISPLAY_UPDATE_STATE
  (WS                 : in  WS_ID;
   ERROR_INDICATOR    : out ERROR_NUMBER;
   DEFERRAL           : out DEFERRAL_MODE;
   MODIFICATION       : out MODIFICATION_MODE;
   EMPTINESS          : out DISPLAY_SURFACE_EMPTY;
   VISUAL_STATE       : out VISUAL_REPRESENTATION_STATE);
```

INQUIRE LIST OF POLYLINE INDICES

```
procedure INQ_LIST_OF_POLYLINE_INDICES
  (WS                 : in  WS_ID;
   ERROR_INDICATOR    : out ERROR_NUMBER;
   INDICES            : out POLYLINE_INDICES.LIST_OF);
```

INQUIRE POLYLINE REPRESENTATION

```
procedure INQ_POLYLINE_REPRESENTATION
  (WS                 : in  WS_ID;
   POLYLINE_IND       : in  POLYLINE_INDEX;
   RETURNED_VALUES    : in  RETURN_VALUE_TYPE;
   ERROR_INDICATOR    : out ERROR_NUMBER;
   TYPE_OF_LINE       : out LINETYPE;
   LINEWIDTH_SF       : out LINEWIDTH;
   LINE_COLOUR        : out COLOUR_INDEX);
```

INQUIRE LIST OF POLYMARKER INDICES

```
procedure INQ_LIST_OF_POLYMARKER_INDICES
  (WS                 : in  WS_ID;
   ERROR_INDICATOR    : out ERROR_NUMBER;
   INDICES            : out POLYMARKER_INDICES.LIST_OF);
```

INQUIRE POLYMARKER REPRESENTATION

```
procedure INQ_POLYMARKER_REPRESENTATION
  (WS                 : in  WS_ID;
   POLYMARKER_IND     : in  POLYMARKER_INDEX;
   RETURNED_VALUES    : in  RETURN_VALUE_TYPE;
   ERROR_INDICATOR    : out ERROR_NUMBER;
   TYPE_OF_MARKER     : out MARKER_TYPE;
   SIZE               : out MARKER_SIZE;
   MARKER_COLOUR      : out COLOUR_INDEX);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

INQUIRE LIST OF TEXT INDICES

```
procedure INQ_LIST_OF_TEXT_INDICES
  (WS                 : in  WS_ID;
   ERROR_INDICATOR    : out ERROR_NUMBER;
   INDICES            : out TEXT_INDICES.LIST_OF);
```

INQUIRE TEXT REPRESENTATION

```
procedure INQ_TEXT_REPRESENTATION
  (WS                 : in  WS_ID;
   TEXT_IND           : in  TEXT_INDEX;
   RETURNED_VALUES    : in  RETURN_VALUE_TYPE;
   ERROR_INDICATOR    : out ERROR_NUMBER;
   FONT               : out TEXT_FONT;
   PRECISION          : out TEXT_PRECISION;
   EXPANSION          : out CHAR_EXPANSION;
   SPACING            : out CHAR_SPACING;
   TEXT_COLOUR        : out COLOUR_INDEX);
```

INQUIRE LIST OF INTERIOR INDICES

```
procedure INQ_LIST_OF_INTERIOR_INDICES
  (WS                 : in  WS_ID;
   ERROR_INDICATOR    : out ERROR_NUMBER;
   INDICES            : out INTERIOR_INDICES.LIST_OF);
```

INQUIRE INTERIOR REPRESENTATION

```
procedure INQ_INTERIOR_REPRESENTATION
  (WS                 : in  WS_ID;
   INTERIOR_IND       : in  INTERIOR_INDEX;
   RETURNED_VALUES    : in  RETURN_VALUE_TYPE;
   ERROR_INDICATOR    : out ERROR_NUMBER;
   STYLE_OF_INTERIOR  : out INTERIOR_STYLE;
   STYLE_IND          : out STYLE_INDEX;
   INTERIOR_COLOUR    : out COLOUR_INDEX);
```

INQUIRE LIST OF EDGE INDICES

```
procedure INQ_LIST_OF_EDGE_INDICES
  (WS                 : in  WS_ID;
   ERROR_INDICATOR    : out ERROR_NUMBER;
   INDICES            : out EDGE_INDICES.LIST_OF);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0575490 688 ■

INQUIRE EDGE REPRESENTATION

```
procedure INQ_EDGE_REPRESENTATION
  (WS                  : in  WS_ID;
   EDGE_IND            : in  EDGE_INDEX;
   RETURNED_VALUES     : in  RETURN_VALUE_TYPE;
   ERROR_INDICATOR     : out ERROR_NUMBER;
   FLAG                : out EDGE_FLAG;
   TYPE_OF_EDGE        : out EDGETYPE;
   EDGEWIDTH_SF        : out EDGEWIDTH;
   EDGE_COLOUR         : out COLOUR_INDEX);
```

INQUIRE LIST OF PATTERN INDICES

```
procedure INQ_LIST_OF_PATTERN_INDICES
  (WS                  : in  WS_ID;
   ERROR_INDICATOR     : out ERROR_NUMBER;
   INDICES             : out PATTERN_INDICES.LIST_OF);
```

INQUIRE PATTERN REPRESENTATION

```
procedure INQ_PATTERN_REPRESENTATION
  (WS                  : in  WS_ID;
   PATTERN_IND         : in  PATTERN_INDEX;
   RETURNED_VALUES     : in  RETURN_VALUE_TYPE;
   ERROR_INDICATOR     : out ERROR_NUMBER;
   PATTERN             : out ACCESS_COLOUR_MATRIX);
```

INQUIRE COLOUR MODEL

```
procedure INQ_COLOUR_MODEL
  (WS                  : in  WS_ID;
   ERROR_INDICATOR     : out ERROR_NUMBER;
   MODEL               : out COLOUR_MODEL);
```

INQUIRE LIST OF COLOUR INDICES

```
procedure INQ_LIST_OF_COLOUR_INDICES
  (WS                  : in  WS_ID;
   ERROR_INDICATOR     : out ERROR_NUMBER;
   INDICES             : out COLOUR_INDICES.LIST_OF);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

INQUIRE COLOUR REPRESENTATION

```
procedure INQ_COLOUR_REPRESENTATION
  (WS                  : in  WS_ID;
   COLOUR_IND          : in  COLOUR_INDEX;
   RETURNED_VALUES     : in  RETURN_VALUE_TYPE;
   ERROR_INDICATOR     : out ERROR_NUMBER;
   COLOUR              : out COLOUR_REPRESENTATION);
```

INQUIRE HIGHLIGHTING FILTER

```
procedure INQ_HIGHLIGHTING_FILTER
  (WS                  : in  WS_ID;
   ERROR_INDICATOR     : out ERROR_NUMBER;
   HIGHLIGHTING        : out NAME_SET_FILTER);
```

INQUIRE INVISIBILITIY FILTER

```
procedure INQ_INVISIBILITY_FILTER
  (WS                  : in  WS_ID;
   ERROR_INDICATOR     : out ERROR_NUMBER;
   INVISIBILITY        : out NAME_SET_FILTER);
```

INQUIRE WORKSTATION TRANSFORMATION 3

```
procedure INQ_WS_TRANSFORMATION
  (WS                           : in  WS_ID;
   ERROR_INDICATOR              : out ERROR_NUMBER;
   UPDATE                       : out UPDATE_STATE;
   REQUESTED_WINDOW_LIMITS      : out NPC.RECTANGULAR_REGION_3;
   CURRENT_WINDOW_LIMITS        : out NPC.RECTANGULAR_REGION_3;
   REQUESTED_VIEWPORT_LIMITS    : out DC.RECTANGULAR_REGION_3;
   CURRENT_VIEWPORT_LIMITS      : out DC.RECTANGULAR_REGION_3);
```

INQUIRE WORKSTATION TRANSFORMATION

```
procedure INQ_WS_TRANSFORMATION
  (WS                           : in  WS_ID;
   ERROR_INDICATOR              : out ERROR_NUMBER;
   UPDATE                       : out UPDATE_STATE;
   REQUESTED_WINDOW_LIMITS      : out NPC.RECTANGULAR_REGION_2;
   CURRENT_WINDOW_LIMITS        : out NPC.RECTANGULAR_REGION_2;
   REQUESTED_VIEWPORT_LIMITS    : out DC.RECTANGULAR_REGION_2;
   CURRENT_VIEWPORT_LIMITS      : out DC.RECTANGULAR_REGION_2);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

---

INQUIRE LOCATOR DEVICE STATE 3

```
procedure INQ_LOCATOR_DEVICE_STATE
  (WS                 : in  WS_ID;
   DEVICE             : in  LOCATOR_DEVICE_NUMBER;
   RETURNED_VALUES    : in  RETURN_VALUE_TYPE;
   ERROR_INDICATOR    : out ERROR_NUMBER;
   MODE               : out OPERATING_MODE;
   SWITCH             : out ECHO_SWITCH;
   INITIAL_VIEW_IND   : out VIEW_INDEX;
   INITIAL_POSITION   : out WC.POINT_3;
   ECHO_VOLUME        : out DC.RECTANGULAR_REGION_3;
   DATA_RECORD        : out LOCATOR_DATA_RECORD);
```

INQUIRE LOCATOR DEVICE STATE

```
procedure INQ_LOCATOR_DEVICE_STATE
  (WS                 : in  WS_ID;
   DEVICE             : in  LOCATOR_DEVICE_NUMBER;
   RETURNED_VALUES    : in  RETURN_VALUE_TYPE;
   ERROR_INDICATOR    : out ERROR_NUMBER;
   MODE               : out OPERATING_MODE;
   SWITCH             : out ECHO_SWITCH;
   INITIAL_VIEW_IND   : out VIEW_INDEX;
   INITIAL_POSITION   : out WC.POINT_2;
   ECHO_AREA          : out DC.RECTANGULAR_REGION_2;
   DATA_RECORD        : out LOCATOR_DATA_RECORD);
```

INQUIRE STROKE DEVICE STATE 3

```
procedure INQ_STROKE_DEVICE_STATE
  (WS                 : in  WS_ID;
   DEVICE             : in  STROKE_DEVICE_NUMBER;
   RETURNED_VALUES    : in  RETURN_VALUE_TYPE;
   ERROR_INDICATOR    : out ERROR_NUMBER;
   MODE               : out OPERATING_MODE;
   SWITCH             : out ECHO_SWITCH;
   INITIAL_VIEW_IND   : out VIEW_INDEX;
   INITIAL_STROKE_POINTS : out WC.ACCESS_POINT_LIST_3;
   ECHO_VOLUME        : out DC.RECTANGULAR_REGION_3;
   DATA_RECORD        : out STROKE_DATA_RECORD);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

INQUIRE STROKE DEVICE STATE

```
procedure INQ_STROKE_DEVICE_STATE
  (WS                    : in WS_ID;
   DEVICE                : in STROKE_DEVICE_NUMBER;
   RETURNED_VALUES       : in RETURN_VALUE_TYPE;
   ERROR_INDICATOR       : out ERROR_NUMBER;
   MODE                  : out OPERATING_MODE;
   SWITCH                : out ECHO_SWITCH;
   INITIAL_VIEW_IND      : out VIEW_INDEX;
   INITIAL_STROKE_POINTS : out WC.ACCESS_POINT_LIST_2;
   ECHO_AREA             : out DC.RECTANGULAR_REGION_2;
   DATA_RECORD           : out STROKE_DATA_RECORD);
```

INQUIRE VALUATOR DEVICE STATE 3

```
procedure INQ_VALUATOR_DEVICE_STATE
  (WS                : in WS_ID;
   DEVICE            : in VALUATOR_DEVICE_NUMBER;
   ERROR_INDICATOR   : out ERROR_NUMBER;
   MODE              : out OPERATING_MODE;
   SWITCH            : out ECHO_SWITCH;
   INITIAL_VALUE     : out VALUATOR_VALUE;
   ECHO_VOLUME       : out DC.RECTANGULAR_REGION_3;
   DATA_RECORD       : out VALUATOR_DATA_RECORD);
```

INQUIRE VALUATOR DEVICE STATE

```
procedure INQ_VALUATOR_DEVICE_STATE
  (WS                : in WS_ID;
   DEVICE            : in VALUATOR_DEVICE_NUMBER;
   ERROR_INDICATOR   : out ERROR_NUMBER;
   MODE              : out OPERATING_MODE;
   SWITCH            : out ECHO_SWITCH;
   INITIAL_VALUE     : out VALUATOR_VALUE;
   ECHO_AREA         : out DC.RECTANGULAR_REGION_2;
   DATA_RECORD       : out VALUATOR_DATA_RECORD);
```

INQUIRE CHOICE DEVICE STATE 3

```
procedure INQ_CHOICE_DEVICE_STATE
  (WS                : in WS_ID;
   DEVICE            : in CHOICE_DEVICE_NUMBER;
   ERROR_INDICATOR   : out ERROR_NUMBER;
   MODE              : out OPERATING_MODE;
   SWITCH            : out ECHO_SWITCH;
   INITIAL_STATUS    : out CHOICE_STATUS;
   INITIAL_CHOICE    : out CHOICE_NUMBER;
   ECHO_VOLUME       : out DC.RECTANGULAR_REGION_3;
   DATA_RECORD       : out CHOICE_DATA_RECORD);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

---

INQUIRE CHOICE DEVICE STATE

```
procedure INQ_CHOICE_DEVICE_STATE
  (WS                : in WS_ID;
   DEVICE            : in CHOICE_DEVICE_NUMBER;
   ERROR_INDICATOR   : out ERROR_NUMBER;
   MODE              : out OPERATING_MODE;
   SWITCH            : out ECHO_SWITCH;
   INITIAL_STATUS    : out CHOICE_STATUS;
   INITIAL_CHOICE    : out CHOICE_NUMBER;
   ECHO_AREA         : out DC.RECTANGULAR_REGION_2;
   DATA_RECORD       : out CHOICE_DATA_RECORD);
```

INQUIRE PICK DEVICE STATE 3

```
procedure INQ_PICK_DEVICE_STATE
  (WS                : in WS_ID;
   DEVICE            : in PICK_DEVICE_NUMBER;
   RETURNED_VALUES   : in RETURN_VALUE_TYPE;
   ERROR_INDICATOR   : out ERROR_NUMBER;
   MODE              : out OPERATING_MODE;
   SWITCH            : out ECHO_SWITCH;
   PICKABILITY       : out NAME_SET_FILTER;
   INITIAL_STATUS    : out PICK_STATUS;
   INITIAL_PATH      : out PICK_PATH;
   ECHO_VOLUME       : out DC.RECTANGULAR_REGION_3;
   DATA_RECORD       : out PICK_DATA_RECORD;
   ORDER             : out PATH_ORDER);
```

INQUIRE PICK DEVICE STATE

```
procedure INQ_PICK_DEVICE_STATE
  (WS                : in WS_ID;
   DEVICE            : in PICK_DEVICE_NUMBER;
   RETURNED_VALUES   : in RETURN_VALUE_TYPE;
   ERROR_INDICATOR   : out ERROR_NUMBER;
   MODE              : out OPERATING_MODE;
   SWITCH            : out ECHO_SWITCH;
   PICKABILITY       : out NAME_SET_FILTER;
   INITIAL_STATUS    : out PICK_STATUS;
   INITIAL_PATH      : out PICK_PATH;
   ECHO_AREA         : out DC.RECTANGULAR_REGION_2;
   DATA_RECORD       : out PICK_DATA_RECORD;
   ORDER             : out PATH_ORDER);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

INQUIRE STRING DEVICE STATE 3

```
procedure INQ_STRING_DEVICE_STATE
  (WS                    : in WS_ID;
   DEVICE                : in STRING_DEVICE_NUMBER;
   ERROR_INDICATOR       : out ERROR_NUMBER;
   MODE                  : out OPERATING_MODE;
   SWITCH                : out ECHO_SWITCH;
   INITIAL_CHAR_STRING   : out INPUT_STRING;
   ECHO_VOLUME           : out DC.RECTANGULAR_REGION_3;
   DATA_RECORD           : out STRING_DATA_RECORD);
```

INQUIRE STRING DEVICE STATE

```
procedure INQ_STRING_DEVICE_STATE
  (WS                    : in WS_ID;
   DEVICE                : in STRING_DEVICE_NUMBER;
   ERROR_INDICATOR       : out ERROR_NUMBER;
   MODE                  : out OPERATING_MODE;
   SWITCH                : out ECHO_SWITCH;
   INITIAL_CHAR_STRING   : out INPUT_STRING;
   ECHO_AREA             : out DC.RECTANGULAR_REGION_2;
   DATA_RECORD           : out STRING_DATA_RECORD);
```

INQUIRE WORKSTATION CATEGORY

```
procedure INQ_WS_CATEGORY
  (TYPE_OF_WS        : in WS_TYPE;
   ERROR_INDICATOR   : out ERROR_NUMBER;
   CATEGORY          : out WS_CATEGORY);
```

INQUIRE DISPLAY SPACE SIZE 3

```
procedure INQ_DISPLAY_SPACE_SIZE
  (TYPE_OF_WS            : in WS_TYPE;
   ERROR_INDICATOR       : out ERROR_NUMBER;
   UNITS                 : out DC_UNITS;
   MAX_DC_UNIT_SIZE      : out DC.SIZE_3;
   MAX_RASTER_UNIT_SIZE  : out RASTER_UNIT_SIZE_3);
```

INQUIRE DISPLAY SPACE SIZE

```
procedure INQ_DISPLAY_SPACE_SIZE
  (TYPE_OF_WS            : in WS_TYPE;
   ERROR_INDICATOR       : out ERROR_NUMBER;
   UNITS                 : out DC_UNITS;
   MAX_DC_UNIT_SIZE      : out DC.SIZE_2;
   MAX_RASTER_UNIT_SIZE  : out RASTER_UNIT_SIZE_2);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0575496 0T6 ■

INQUIRE HLHSR FACILITIES

```
procedure INQ_HLHSR_FACILITIES
  (TYPE_OF_WS           : in  WS_TYPE;
   ERROR_INDICATOR      : out ERROR_NUMBER;
   LIST_OF_IDENTIFIERS  : out HLHSR_IDS.LIST_OF;
   LIST_OF_MODES        : out HLHSR_MODES.LIST_OF);
```

INQUIRE VIEW FACILITIES

```
procedure INQ_VIEW_FACILITIES
  (TYPE_OF_WS           : in  WS_TYPE;
   ERROR_INDICATOR      : out ERROR_NUMBER;
   NUMBER_OF_INDICES    : out PHIGS_POSITIVE);
```

INQUIRE PREDEFINED VIEW REPRESENTATION

```
procedure INQ_PREDEFINED_VIEW_REPRESENTATION
  (TYPE_OF_WS           : in  WS_TYPE;
   VIEW_IND             : in  VIEW_INDEX;
   ERROR_INDICATOR      : out ERROR_NUMBER;
   ORIENTATION_MATRIX   : out TRANSFORMATION_MATRIX_3;
   MAPPING_MATRIX       : out TRANSFORMATION_MATRIX_3;
   CLIPPING_LIMITS      : out NPC.RECTANGULAR_REGION_3;
   XY_CLIPPING          : out CLIPPING_INDICATOR;
   BACK_CLIPPING        : out CLIPPING_INDICATOR;
   FRONT_CLIPPING       : out CLIPPING_INDICATOR);
```

INQUIRE WORKSTATION CLASSIFICATION

```
procedure INQ_WS_CLASSIFICATION
  (TYPE_OF_WS           : in  WS_TYPE;
   ERROR_INDICATOR      : out ERROR_NUMBER;
   CLASS                : out DISPLAY_CLASS);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

INQUIRE DYNAMICS OF WORKSTATION ATTRIBUTES

```
procedure INQ_DYNAMICS_OF_WS_ATTRIBUTES
  (TYPE_OF_WS                    : in  WS_TYPE;
   ERROR_INDICATOR               : out ERROR_NUMBER;
   POLYLINE_REPRESENTATION       : out DYNAMIC_MODIFICATION;
   POLYMARKER_REPRESENTATION     : out DYNAMIC_MODIFICATION;
   TEXT_REPRESENTATION           : out DYNAMIC_MODIFICATION;
   INTERIOR_REPRESENTATION       : out DYNAMIC_MODIFICATION;
   EDGE_REPRESENTATION           : out DYNAMIC_MODIFICATION;
   PATTERN_REPRESENTATION        : out DYNAMIC_MODIFICATION;
   COLOUR_REPRESENTATION         : out DYNAMIC_MODIFICATION;
   VIEW_REPRESENTATION           : out DYNAMIC_MODIFICATION;
   WS_TRANSFORMATION             : out DYNAMIC_MODIFICATION;
   HIGHLIGHTING_FILTER           : out DYNAMIC_MODIFICATION;
   INVISIBILITY_FILTER           : out DYNAMIC_MODIFICATION;
   HLHSR_MODE_CHANGE             : out DYNAMIC_MODIFICATION);
```

INQUIRE DEFAULT DISPLAY UPDATE STATE

```
procedure INQ_DEFAULT_DISPLAY_UPDATE_STATE
  (TYPE_OF_WS          : in  WS_TYPE;
   ERROR_INDICATOR     : out ERROR_NUMBER;
   DEFERRAL            : out DEFERRAL_MODE;
   MODIFICATION        : out MODIFICATION_MODE);
```

INQUIRE POLYLINE FACILITIES

```
procedure INQ_POLYLINE_FACILITIES
  (TYPE_OF_WS          : in  WS_TYPE;
   ERROR_INDICATOR     : out ERROR_NUMBER;
   NUMBER_OF_TYPES     : out PHIGS_INTEGER;
   LIST_OF_TYPES       : out LINETYPES.LIST_OF;
   NUMBER_OF_WIDTHS    : out PHIGS_NATURAL;
   NOMINAL_WIDTH       : out DC.MAGNITUDE;
   RANGE_OF_WIDTHS     : out DC.RANGE_OF_MAGNITUDES;
   NUMBER_OF_INDICES   : out PHIGS_NATURAL);
```

INQUIRE PREDEFINED POLYLINE REPRESENTATION

```
procedure INQ_PREDEFINED_POLYLINE_REPRESENTATION
  (TYPE_OF_WS          : in  WS_TYPE;
   POLYLINE_IND        : in  POLYLINE_INDEX;
   ERROR_INDICATOR     : out ERROR_NUMBER;
   TYPE_OF_LINE        : out LINETYPE;
   LINEWIDTH_SF        : out LINEWIDTH;
   LINE_COLOUR         : out COLOUR_INDEX);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

9593 PT*3 90 ■ 4851903 0575498 979 ■

INQUIRE POLYMARKER FACILITIES

```
procedure INQ_POLYMARKER_FACILITIES
  (TYPE_OF_WS           : in WS_TYPE;
   ERROR_INDICATOR      : out ERROR_NUMBER;
   NUMBER_OF_TYPES      : out PHIGS_INTEGER;
   LIST_OF_TYPES        : out MARKER_TYPES.LIST_OF;
   NUMBER_OF_SIZES      : out PHIGS_NATURAL;
   NOMINAL_SIZE         : out DC.MAGNITUDE;
   RANGE_OF_SIZES       : out DC.RANGE_OF_MAGNITUDES;
   NUMBER_OF_INDICES    : out PHIGS_NATURAL);
```

INQUIRE PREDEFINED POLYMARKER REPRESENTATION

```
procedure INQ_PREDEFINED_POLYMARKER_REPRESENTATION
  (TYPE_OF_WS           : in WS_TYPE;
   POLYMARKER_IND       : in POLYMARKER_INDEX;
   ERROR_INDICATOR      : out ERROR_NUMBER;
   TYPE_OF_MARKER       : out MARKER_TYPE;
   SIZE                 : out MARKER_SIZE;
   MARKER_COLOUR        : out COLOUR_INDEX);
```

INQUIRE TEXT FACILITIES

```
procedure INQ_TEXT_FACILITIES
  (TYPE_OF_WS                    : in WS_TYPE;
   ERROR_INDICATOR               : out ERROR_NUMBER;
   LIST_OF_FONT_PRECISION_PAIRS  : out TEXT_FONT_PRECISIONS.LIST_OF;
   NUMBER_OF_HEIGHTS             : out PHIGS_NATURAL;
   RANGE_OF_HEIGHTS              : out DC.RANGE_OF_MAGNITUDES;
   NUMBER_OF_EXPANSIONS          : out PHIGS_NATURAL;
   RANGE_OF_EXPANSIONS           : out DC.RANGE_OF_MAGNITUDES;
   NUMBER_OF_INDICES             : out PHIGS_NATURAL);
```

INQUIRE PREDEFINED TEXT REPRESENTATION

```
procedure INQ_PREDEFINED_TEXT_REPRESENTATION
  (TYPE_OF_WS           : in WS_TYPE;
   TEXT_IND             : in TEXT_INDEX;
   ERROR_INDICATOR      : out ERROR_NUMBER;
   FONT                 : out TEXT_FONT;
   PRECISION            : out TEXT_PRECISION;
   EXPANSION            : out CHAR_EXPANSION;
   SPACING              : out CHAR_SPACING;
   TEXT_COLOUR          : out COLOUR_INDEX);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

INQUIRE ANNOTATION FACILITIES

```
procedure INQ_ANNOTATION_FACILITIES
  (TYPE_OF_WS                    : in WS_TYPE;
   ERROR_INDICATOR               : out ERROR_NUMBER;
   LIST_OF_ANNOTATION_STYLES     : out ANNOTATION_STYLES.LIST_OF;
   NUMBER_OF_CHAR_HEIGHTS        : out PHIGS_NATURAL;
   RANGE_OF_CHAR_HEIGHTS         : out DC.RANGE_OF_MAGNITUDES);
```

INQUIRE TEXT EXTENT

```
procedure INQ_TEXT_EXTENT
  (TYPE_OF_WS                    : in  WS_TYPE;
   FONT                          : in  TEXT_FONT;
   EXPANSION                     : in  CHAR_EXPANSION;
   SPACING                       : in  CHAR_SPACING;
   HEIGHT                        : in  MC.MAGNITUDE;
   PATH                          : in  TEXT_PATH;
   ALIGNMENT                     : in  TEXT_ALIGNMENT;
   CHAR_STRING                   : in  PHIGS_STRING;
   ERROR_INDICATOR               : out ERROR_NUMBER;
   EXTENT_RECTANGLE              : out MC.RECTANGULAR_REGION_2;
   CONCATENATION_OFFSET          : out MC.VECTOR_2);

-- This procedure using MC coordinates is for use with text primitives.
```

INQUIRE TEXT EXTENT

```
procedure INQ_TEXT_EXTENT
  (TYPE_OF_WS                    : in  WS_TYPE;
   FONT                          : in  TEXT_FONT;
   EXPANSION                     : in  CHAR_EXPANSION;
   SPACING                       : in  CHAR_SPACING;
   HEIGHT                        : in  NPC.MAGNITUDE;
   PATH                          : in  TEXT_PATH;
   ALIGNMENT                     : in  TEXT_ALIGNMENT;
   CHAR_STRING                   : in  PHIGS_STRING;
   ERROR_INDICATOR               : out ERROR_NUMBER;
   EXTENT_RECTANGLE              : out NPC.RECTANGULAR_REGION_2;
   CONCATENATION_OFFSET          : out NPC.VECTOR_2);

-- This procedure using NPC coordinates is for use with annotation text
-- primitives.
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0575500 357 ■

INQUIRE INTERIOR FACILITIES

```
procedure INQ_INTERIOR_FACILITIES
  (TYPE_OF_WS                  : in  WS_TYPE;
   ERROR_INDICATOR             : out ERROR_NUMBER;
   LIST_OF_INTERIOR_STYLES     : out INTERIOR_STYLES.LIST_OF;
   NUMBER_OF_HATCH_STYLES      : out PHIGS_NATURAL;
   LIST_OF_HATCH_STYLES        : out HATCH_STYLES.LIST_OF;
   NUMBER_OF_INDICES           : out PHIGS_NATURAL);
```

INQUIRE PREDEFINED INTERIOR REPRESENTATION

```
procedure INQ_PREDEFINED_INTERIOR_REPRESENTATION
  (TYPE_OF_WS          : in  WS_TYPE;
   INTERIOR_IND        : in  INTERIOR_INDEX;
   ERROR_INDICATOR     : out ERROR_NUMBER;
   STYLE_OF_INTERIOR   : out INTERIOR_STYLE;
   STYLE_IND           : out STYLE_INDEX;
   INTERIOR_COLOUR     : out COLOUR_INDEX);
```

INQUIRE EDGE FACILITIES

```
procedure INQ_EDGE_FACILITIES
  (TYPE_OF_WS          : in  WS_TYPE;
   ERROR_INDICATOR     : out ERROR_NUMBER;
   NUMBER_OF_TYPES     : out PHIGS_INTEGER;
   LIST_OF_TYPES       : out EDGETYPES.LIST_OF;
   NUMBER_OF_WIDTHS    : out PHIGS_NATURAL;
   NOMINAL_WIDTH       : out DC.MAGNITUDE;
   RANGE_OF_WIDTHS     : out DC.RANGE_OF_MAGNITUDES;
   NUMBER_OF_INDICES   : out PHIGS_NATURAL);
```

INQUIRE PREDEFINED EDGE REPRESENTATION

```
procedure INQ_PREDEFINED_EDGE_REPRESENTATION
  (TYPE_OF_WS          : in  WS_TYPE;
   EDGE_IND            : in  EDGE_INDEX;
   ERROR_INDICATOR     : out ERROR_NUMBER;
   FLAG                : out EDGE_FLAG;
   TYPE_OF_EDGE        : out EDGETYPE;
   EDGEWIDTH_SF        : out EDGEWIDTH;
   EDGE_COLOUR         : out COLOUR_INDEX);
```

INQUIRE PATTERN FACILITIES

```
procedure INQ_PATTERN_FACILITIES
  (TYPE_OF_WS          : in  WS_TYPE;
   ERROR_INDICATOR     : out ERROR_NUMBER;
   NUMBER_OF_INDICES   : out PHIGS_NATURAL);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

9593 PT*3 90 ■ 4851903 0575501 293 ■

INQUIRE PREDEFINED PATTERN REPRESENTATION

```
procedure INQ_PREDEFINED_PATTERN_REPRESENTATION
  (TYPE_OF_WS          : in WS_TYPE;
   PATTERN_IND         : in PATTERN_INDEX;
   ERROR_INDICATOR     : out ERROR_NUMBER;
   PATTERN             : out ACCESS_COLOUR_MATRIX);
```

INQUIRE COLOUR MODEL FACILITIES

```
procedure INQ_COLOUR_MODEL_FACILITIES
  (TYPE_OF_WS          : in WS_TYPE;
   ERROR_INDICATOR     : out ERROR_NUMBER;
   LIST_OF_MODELS      : out COLOUR_MODELS.LIST_OF;
   DEFAULT_MODEL       : out COLOUR_MODEL);
```

INQUIRE COLOUR FACILITIES

```
procedure INQ_COLOUR_FACILITIES
  (TYPE_OF_WS                : in WS_TYPE;
   ERROR_INDICATOR           : out ERROR_NUMBER;
   NUMBER_OF_COLOURS         : out PHIGS_NATURAL;
   AVAILABILITY_OF_COLOUR    : out COLOUR_AVAILABLE;
   NUMBER_OF_INDICES         : out PHIGS_NATURAL;
   PRIMARY_COLOURS           : out CHROMATICITY_COEFFICIENT_SET);
```

INQUIRE PREDEFINED COLOUR REPRESENTATION

```
procedure INQ_PREDEFINED_COLOUR_REPRESENTATION
  (TYPE_OF_WS          : in WS_TYPE;
   INDEX               : in COLOUR_INDEX;
   ERROR_INDICATOR     : out ERROR_NUMBER;
   COLOUR              : out COLOUR_REPRESENTATION);
```

INQUIRE LIST OF AVAILABLE GENERALIZED DRAWING PRIMITIVES 3

```
procedure INQ_LIST_OF_AVAILABLE_GDP
  (TYPE_OF_WS          : in WS_TYPE;
   ERROR_INDICATOR     : out ERROR_NUMBER;
   LIST_OF_GDP_3       : out GDP_3_IDS.LIST_OF);
```

INQUIRE LIST OF AVAILABLE GENERALIZED DRAWING PRIMITIVES

```
procedure INQ_LIST_OF_AVAILABLE_GDP
  (TYPE_OF_WS          : in WS_TYPE;
   ERROR_INDICATOR     : out ERROR_NUMBER;
   LIST_OF_GDP         : out GDP_IDS.LIST_OF);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0575502 12T ■

INQUIRE GENERALIZED DRAWING PRIMITIVE 3

```
procedure INQ_GDP
  (TYPE_OF_WS                    : in  WS_TYPE;
   GDP                           : in  GDP_3_ID;
   ERROR_INDICATOR               : out ERROR_NUMBER;
   LIST_OF_ATTRIBUTE_SETS_USED   : out ATTRIBUTES_USED.LIST_OF);
```

INQUIRE GENERALIZED DRAWING PRIMITIVE

```
procedure INQ_GDP
  (TYPE_OF_WS                    : in  WS_TYPE;
   GDP                           : in  GDP_ID;
   ERROR_INDICATOR               : out ERROR_NUMBER;
   LIST_OF_ATTRIBUTE_SETS_USED   : out ATTRIBUTES_USED.LIST_OF);
```

INQUIRE LIST OF AVAILABLE GENERALIZED STRUCTURE ELEMENTS

```
procedure INQ_LIST_OF_AVAILABLE_GSE
  (TYPE_OF_WS       : in  WS_TYPE;
   ERROR_INDICATOR  : out ERROR_NUMBER;
   LIST_OF_GSE      : out GSE_IDS.LIST_OF);
```

INQUIRE NUMBER OF DISPLAY PRIORITIES SUPPORTED

```
procedure INQ_NUMBER_OF_DISPLAY_PRIORITIES_SUPPORTED
  (TYPE_OF_WS            : in  WS_TYPE;
   ERROR_INDICATOR       : out ERROR_NUMBER;
   NUMBER_OF_PRIORITIES  : out PHIGS_NATURAL);
```

INQUIRE WORKSTATION STATE TABLE LENGTHS

```
procedure INQ_WS_STATE_TABLE_LENGTHS
  (TYPE_OF_WS                 : in  WS_TYPE;
   ERROR_INDICATOR            : out ERROR_NUMBER;
   MAX_POLYLINE_ENTRIES       : out PHIGS_NATURAL;
   MAX_POLYMARKER_ENTRIES     : out PHIGS_NATURAL;
   MAX_TEXT_ENTRIES           : out PHIGS_NATURAL;
   MAX_INTERIOR_ENTRIES       : out PHIGS_NATURAL;
   MAX_EDGE_ENTRIES           : out PHIGS_NATURAL;
   MAX_PATTERN_INDICES        : out PHIGS_NATURAL;
   MAX_COLOUR_INDICES         : out PHIGS_NATURAL;
   MAX_VIEW_INDICES           : out PHIGS_NATURAL);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

---

INQUIRE DYNAMICS OF STRUCTURES

```
procedure INQ_DYNAMICS_OF_STRUCTURES
  (TYPE_OF_WS                 : in  WS_TYPE;
   ERROR_INDICATOR            : out ERROR_NUMBER;
   CONTENT_MODIFICATION       : out DYNAMIC_MODIFICATION;
   POST                       : out DYNAMIC_MODIFICATION;
   UNPOST                     : out DYNAMIC_MODIFICATION;
   DELETE                     : out DYNAMIC_MODIFICATION;
   REFERENCE_MODIFICATION     : out DYNAMIC_MODIFICATION);
```

INQUIRE NUMBER OF AVAILABLE LOGICAL INPUT DEVICES

```
procedure INQ_NUMBER_OF_AVAILABLE_LOGICAL_INPUT_DEVICES
  (TYPE_OF_WS         : in  WS_TYPE;
   ERROR_INDICATOR    : out ERROR_NUMBER;
   LOCATOR            : out PHIGS_NATURAL;
   STROKE             : out PHIGS_NATURAL;
   VALUATOR           : out PHIGS_NATURAL;
   CHOICE             : out PHIGS_NATURAL;
   PICK               : out PHIGS_NATURAL;
   STRING             : out PHIGS_NATURAL);
```

INQUIRE DEFAULT LOCATOR DEVICE DATA 3

```
procedure INQ_DEFAULT_LOCATOR_DEVICE_DATA
  (TYPE_OF_WS                 : in  WS_TYPE;
   DEVICE                     : in  LOCATOR_DEVICE_NUMBER;
   ERROR_INDICATOR            : out ERROR_NUMBER;
   INITIAL_POSITION           : out WC.POINT_3;
   LIST_OF_PROMPT_ECHO_TYPES  : out LOCATOR_PROMPT_ECHO_TYPES.LIST_OF;
   ECHO_VOLUME                : out DC.RECTANGULAR_REGION_3;
   DATA_RECORD                : out LOCATOR_DATA_RECORD);
```

INQUIRE DEFAULT LOCATOR DEVICE DATA

```
procedure INQ_DEFAULT_LOCATOR_DEVICE_DATA
  (TYPE_OF_WS                 : in  WS_TYPE;
   DEVICE                     : in  LOCATOR_DEVICE_NUMBER;
   ERROR_INDICATOR            : out ERROR_NUMBER;
   INITIAL_POSITION           : out WC.POINT_2;
   LIST_OF_PROMPT_ECHO_TYPES  : out LOCATOR_PROMPT_ECHO_TYPES.LIST_OF;
   ECHO_AREA                  : out DC.RECTANGULAR_REGION_2;
   DATA_RECORD                : out LOCATOR_DATA_RECORD);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0575504 TT2 ■

INQUIRE DEFAULT STROKE DEVICE DATA 3

```
procedure INQ_DEFAULT_STROKE_DEVICE_DATA
  (TYPE_OF_WS                    : in  WS_TYPE;
   DEVICE                        : in  STROKE_DEVICE_NUMBER;
   ERROR_INDICATOR               : out ERROR_NUMBER;
   MAX_BUFFER_SIZE               : out PHIGS_NATURAL;
   LIST_OF_PROMPT_ECHO_TYPES     : out STROKE_PROMPT_ECHO_TYPES.LIST_OF;
   ECHO_VOLUME                   : out DC.RECTANGULAR_REGION_3;
   DATA_RECORD                   : out STROKE_DATA_RECORD);
```

INQUIRE DEFAULT STROKE DEVICE DATA

```
procedure INQ_DEFAULT_STROKE_DEVICE_DATA
  (TYPE_OF_WS                    : in  WS_TYPE;
   DEVICE                        : in  STROKE_DEVICE_NUMBER;
   ERROR_INDICATOR               : out ERROR_NUMBER;
   MAX_BUFFER_SIZE               : out PHIGS_NATURAL;
   LIST_OF_PROMPT_ECHO_TYPES     : out STROKE_PROMPT_ECHO_TYPES.LIST_OF;
   ECHO_AREA                     : out DC.RECTANGULAR_REGION_2;
   DATA_RECORD                   : out STROKE_DATA_RECORD);
```

INQUIRE DEFAULT VALUATOR DEVICE DATA 3

```
procedure INQ_DEFAULT_VALUATOR_DEVICE_DATA
  (TYPE_OF_WS                    : in  WS_TYPE;
   DEVICE                        : in  VALUATOR_DEVICE_NUMBER;
   ERROR_INDICATOR               : out ERROR_NUMBER;
   INITIAL_VALUE                 : out VALUATOR_VALUE;
   LIST_OF_PROMPT_ECHO_TYPES     : out VALUATOR_PROMPT_ECHO_TYPES.LIST_OF;
   ECHO_VOLUME                   : out DC.RECTANGULAR_REGION_3;
   DATA_RECORD                   : out VALUATOR_DATA_RECORD);
```

INQUIRE DEFAULT VALUATOR DEVICE DATA

```
procedure INQ_DEFAULT_VALUATOR_DEVICE_DATA
  (TYPE_OF_WS                    : in  WS_TYPE;
   DEVICE                        : in  VALUATOR_DEVICE_NUMBER;
   ERROR_INDICATOR               : out ERROR_NUMBER;
   INITIAL_VALUE                 : out VALUATOR_VALUE;
   LIST_OF_PROMPT_ECHO_TYPES     : out VALUATOR_PROMPT_ECHO_TYPES.LIST_OF;
   ECHO_AREA                     : out DC.RECTANGULAR_REGION_2;
   DATA_RECORD                   : out VALUATOR_DATA_RECORD);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

---

INQUIRE DEFAULT CHOICE DEVICE DATA 3

```
procedure INQ_DEFAULT_CHOICE_DEVICE_DATA
  (TYPE_OF_WS                    : in WS_TYPE;
   DEVICE                        : in CHOICE_DEVICE_NUMBER;
   ERROR_INDICATOR               : out ERROR_NUMBER;
   MAX_CHOICES                   : out CHOICE_NUMBER;
   LIST_OF_PROMPT_ECHO_TYPES     : out CHOICE_PROMPT_ECHO_TYPES.LIST_OF;
   ECHO_VOLUME                   : out DC.RECTANGULAR_REGION_3;
   DATA_RECORD                   : out CHOICE_DATA_RECORD);
```

INQUIRE DEFAULT CHOICE DEVICE DATA

```
procedure INQ_DEFAULT_CHOICE_DEVICE_DATA
  (TYPE_OF_WS                    : in WS_TYPE;
   DEVICE                        : in CHOICE_DEVICE_NUMBER;
   ERROR_INDICATOR               : out ERROR_NUMBER;
   MAX_CHOICES                   : out CHOICE_NUMBER;
   LIST_OF_PROMPT_ECHO_TYPES     : out CHOICE_PROMPT_ECHO_TYPES.LIST_OF;
   ECHO_AREA                     : out DC.RECTANGULAR_REGION_2;
   DATA_RECORD                   : out CHOICE_DATA_RECORD);
```

INQUIRE DEFAULT PICK DEVICE DATA 3

```
procedure INQ_DEFAULT_PICK_DEVICE_DATA
  (TYPE_OF_WS                    : in WS_TYPE;
   DEVICE                        : in PICK_DEVICE_NUMBER;
   ERROR_INDICATOR               : out ERROR_NUMBER;
   LIST_OF_PROMPT_ECHO_TYPES     : out PICK_PROMPT_ECHO_TYPES.LIST_OF;
   ECHO_VOLUME                   : out DC.RECTANGULAR_REGION_3;
   DATA_RECORD                   : out PICK_DATA_RECORD);
```

INQUIRE DEFAULT PICK DEVICE DATA

```
procedure INQ_DEFAULT_PICK_DEVICE_DATA
  (TYPE_OF_WS                    : in WS_TYPE;
   DEVICE                        : in PICK_DEVICE_NUMBER;
   ERROR_INDICATOR               : out ERROR_NUMBER;
   LIST_OF_PROMPT_ECHO_TYPES     : out PICK_PROMPT_ECHO_TYPES.LIST_OF;
   ECHO_AREA                     : out DC.RECTANGULAR_REGION_2;
   DATA_RECORD                   : out PICK_DATA_RECORD);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0575506 875 ■

INQUIRE DEFAULT STRING DEVICE DATA 3

```
procedure INQ_DEFAULT_STRING_DEVICE_DATA
  (TYPE_OF_WS                    : in  WS_TYPE;
   DEVICE                        : in  STRING_DEVICE_NUMBER;
   ERROR_INDICATOR               : out ERROR_NUMBER;
   MAX_STRING_BUFFER_SIZE        : out PHIGS_NATURAL;
   LIST_OF_PROMPT_ECHO_TYPES     : out STRING_PROMPT_ECHO_TYPES.LIST_OF;
   ECHO_VOLUME                   : out DC.RECTANGULAR_REGION_3;
   DATA_RECORD                   : out STRING_DATA_RECORD);
```

INQUIRE DEFAULT STRING DEVICE DATA

```
procedure INQ_DEFAULT_STRING_DEVICE_DATA
  (TYPE_OF_WS                    : in  WS_TYPE;
   DEVICE                        : in  STRING_DEVICE_NUMBER;
   ERROR_INDICATOR               : out ERROR_NUMBER;
   MAX_STRING_BUFFER_SIZE        : out PHIGS_NATURAL;
   LIST_OF_PROMPT_ECHO_TYPES     : out STRING_PROMPT_ECHO_TYPES.LIST_OF;
   ECHO_AREA                     : out DC.RECTANGULAR_REGION_2;
   DATA_RECORD                   : out STRING_DATA_RECORD);
```

INQUIRE SET OF WORKSTATIONS TO WHICH POSTED

```
procedure INQ_SET_OF_WS_TO_WHICH_POSTED
  (STRUCTURE_IDENTIFIER : in  STRUCTURE_ID;
   ERROR_INDICATOR      : out ERROR_NUMBER;
   LIST_OF_WS           : out WS_IDS.LIST_OF);
```

INQUIRE OPEN STRUCTURE

```
procedure INQ_OPEN_STRUCTURE
  (ERROR_INDICATOR      : out ERROR_NUMBER;
   STATUS               : out OPEN_STRUCTURE_STATUS;
   STRUCTURE_IDENTIFIER : out STRUCTURE_ID);
```

INQUIRE ELEMENT POINTER

```
procedure INQ_ELEMENT_POINTER
  (ERROR_INDICATOR : out ERROR_NUMBER;
   POSITION        : out RETURNED_ELEMENT_POSITION);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

INQUIRE CURRENT ELEMENT TYPE AND SIZE

```
procedure INQ_CURRENT_ELEMENT_TYPE_AND_SIZE
  (ERROR_INDICATOR : out ERROR_NUMBER;
   ELEMENT_TYPE    : out STRUCTURE_ELEMENT_TYPE;
   SIZE            : out PHIGS_NATURAL);
```

INQUIRE CURRENT ELEMENT CONTENT

```
procedure INQ_CURRENT_ELEMENT_CONTENT
  (ERROR_INDICATOR : out ERROR_NUMBER;
   ELEMENT_RECORD  : in out STRUCTURE_ELEMENT_RECORD);
```

INQUIRE ELEMENT TYPE AND SIZE

```
procedure INQ_ELEMENT_TYPE_AND_SIZE
  (STRUCTURE_IDENTIFIER : in STRUCTURE_ID;
   POSITION             : in ELEMENT_POSITION;
   ERROR_INDICATOR      : out ERROR_NUMBER;
   ELEMENT_TYPE         : out STRUCTURE_ELEMENT_TYPE;
   SIZE                 : out PHIGS_NATURAL);
```

INQUIRE ELEMENT CONTENT

```
procedure INQ_ELEMENT_CONTENT
  (STRUCTURE_IDENTIFIER : in     STRUCTURE_ID;
   POSITION             : in     ELEMENT_POSITION;
   ERROR_INDICATOR      :    out ERROR_NUMBER;
   ELEMENT_RECORD       : in out STRUCTURE_ELEMENT_RECORD);
```

INQUIRE STRUCTURE STATUS

```
procedure INQ_STRUCTURE_STATUS
  (STRUCTURE_IDENTIFIER : in STRUCTURE_ID;
   ERROR_INDICATOR      : out ERROR_NUMBER;
   STATUS               : out STRUCTURE_STATUS);
```

INQUIRE PATHS TO ANCESTORS

```
procedure INQ_PATHS_TO_ANCESTORS
  (STRUCTURE_IDENTIFIER : in STRUCTURE_ID;
   ORDER                : in PATH_ORDER;
   DEPTH                : in PATH_DEPTH;
   ERROR_INDICATOR      : out ERROR_NUMBER;
   LIST_OF_PATHS        : out REFERENCE_PATHS.LIST_OF);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0575508 648 ■

INQUIRE PATHS TO DESCENDANTS

```
procedure INQ_PATHS_TO_DESCENDANTS
  (STRUCTURE_IDENTIFIER : in STRUCTURE_ID;
   ORDER                : in PATH_ORDER;
   DEPTH                : in PATH_DEPTH;
   ERROR_INDICATOR      : out ERROR_NUMBER;
   LIST_OF_PATHS        : out REFERENCE_PATHS.LIST_OF);
```

ELEMENT SEARCH

```
procedure ELEMENT_SEARCH
  (STRUCTURE_IDENTIFIER : in STRUCTURE_ID;
   START_POSITION       : in ELEMENT_POSITION;
   DIRECTION            : in SEARCH_DIRECTION;
   INCLUSION_SET        : in ELEMENT_TYPES.LIST_OF;
   EXCLUSION_SET        : in ELEMENT_TYPES.LIST_OF;
   ERROR_INDICATOR      : out ERROR_NUMBER;
   SEARCH_STATUS        : out SEARCH_STATUS_INDICATOR;
   FOUND_POSITION       : out RETURNED_ELEMENT_POSITION);
```

INCREMENTAL SPATIAL SEARCH 3

```
procedure INCREMENTAL_SPATIAL_SEARCH
  (SEARCH_REFERENCE_POINT : in WC.POINT_3;
   SEARCH_DISTANCE        : in WC.MAGNITUDE;
   STARTING_PATH          : in REFERENCE_PATH;
   MODELLING_CLIPPING     : in CLIPPING_INDICATOR;
   SEARCH_CEILING_INDEX   : in ELEMENT_POSITION;
   LIST_OF_NORMAL_FILTERS : in NAME_SET_FILTERS.LIST_OF;
   LIST_OF_INVERTED_FILTERS : in NAME_SET_FILTERS.LIST_OF;
   ERROR_INDICATOR        : out ERROR_NUMBER;
   FOUND_PATH             : out REFERENCE_PATH);
```

INCREMENTAL SPATIAL SEARCH

```
procedure INCREMENTAL_SPATIAL_SEARCH
  (SEARCH_REFERENCE_POINT : in WC.POINT_2;
   SEARCH_DISTANCE        : in WC.MAGNITUDE;
   STARTING_PATH          : in REFERENCE_PATH;
   MODELLING_CLIPPING     : in CLIPPING_INDICATOR;
   SEARCH_CEILING_INDEX   : in ELEMENT_POSITION;
   LIST_OF_NORMAL_FILTERS : in NAME_SET_FILTERS.LIST_OF;
   LIST_OF_INVERTED_FILTERS : in NAME_SET_FILTERS.LIST_OF;
   ERROR_INDICATOR        : out ERROR_NUMBER;
   FOUND_PATH             : out REFERENCE_PATH);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

---

INQUIRE INPUT QUEUE OVERFLOW

```
procedure INQ_INPUT_QUEUE_OVERFLOW
  (ERROR_INDICATOR : out ERROR_NUMBER;
   WS              : out WS_ID;
   DEVICE          : out EVENT_QUEUE_DEVICE_NUMBER);
```

INQUIRE ERROR HANDLING MODE

```
procedure INQ_ERROR_HANDLING_MODE
  (ERROR_INDICATOR : out ERROR_NUMBER;
   MODE            : out ERROR_HANDLING_MODE);
```

5.12 Error control functions

EMERGENCY CLOSE PHIGS

```
procedure EMERGENCY_CLOSE_PHIGS;
```

ERROR HANDLING

```
procedure ERROR_HANDLING
  (ERROR_INDICATOR : in ERROR_NUMBER;
   SUBPROGRAM      : in SUBPROGRAM_NAME;
   ERROR_FILE      : in FILE_ID := PHIGS_STRING(DEFAULT_ERROR_FILE));

   -- The procedure ERROR_HANDLING is defined as a library
   -- subprogram, and is not declared within Package PHIGS.
```

ERROR LOGGING

```
procedure ERROR_LOGGING
  (ERROR_INDICATOR : in ERROR_NUMBER;
   SUBPROGRAM      : in SUBPROGRAM_NAME;
   ERROR_FILE      : in FILE_ID := PHIGS_STRING(DEFAULT_ERROR_FILE));
```

SET ERROR HANDLING MODE

```
procedure SET_ERROR_HANDLING_MODE
  (MODE : in ERROR_HANDLING_MODE);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0575510 2T6 ■

5.13 Special interface functions

ESCAPE

Escape functions are bound in Ada as separate procedures for each unique type of escape provided by the implementation, each with a formal parameter list appropriate to the procedure implemented. ESCAPE names and parameters are registered in the ISO International Register of Graphical Items which is maintained by the registration authority. All supported registered ESCAPE procedures will be in a library package named PHIGS_ESCAPE.

The minimal support for ESCAPE's required by this standard is that defined in the following specification for the PHIGS_ESCAPE package. This package is part of package PHIGS and shall be visible. The minimal support provides an interface to registered and unregistered ESCAPE's not supported by the implementation. Unsupported ESCAPE's may be invoked using the procedure GENERALIZED_ESC whose external specification is shown.

```
with PHIGS_TYPES;
use PHIGS_TYPES;
package PHIGS_ESCAPE is
  type ESC_INTEGER_ARRAY is array (SMALL_NATURAL range <>) of
                                  PHIGS_INTEGER;
  type ESC_FLOAT_ARRAY   is array (SMALL_NATURAL range <>) of
                                  GENERAL_FLOAT_PARAMETER;
  type ESC_STRING_ARRAY  is array (SMALL_NATURAL range <>) of
                                  PHIGS_STRING(1..80);

  type ESC_DATA_RECORD (NUM_OF_INTEGERS : SMALL_NATURAL := 0;
                        NUM_OF_REALS    : SMALL_NATURAL := 0;
                        NUM_OF_STRINGS  : SMALL_NATURAL := 0) is
~@

    record
      INTEGER_ARRAY : ESC_INTEGER_ARRAY (1..NUM_OF_INTEGERS);
      REAL_ARRAY    : ESC_FLOAT_ARRAY   (1..NUM_OF_REALS);
      ESC_STRINGS   : ESC_STRING_ARRAY  (1..NUM_OF_STRINGS);
    end record;


  procedure GENERALIZED_ESC
    (ESC_NAME      : in ESCAPE_ID;
     ESC_DATA_IN   : in ESC_DATA_RECORD;
     ESC_DATA_OUT  : out ESC_DATA_RECORD);

end PHIGS_ESCAPE;
```

Each registered ESCAPE procedure supported by the implementation will be included in package PHIGS_ESCAPE (which is part of package PHIGS) as a separate procedure using the following naming convention which results in references to the ESCAPE by the expression PHIGS_ESCAPE.<name of the ESCAPE procedure>. Specific names will be assigned when the ESCAPE is registered.

```
procedure <name of the ESCAPE procedure> (<parameters as required>);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

9593 PT*3 90 ■ 4851903 0575511 132 ■

Each unregistered ESCAPE procedure will be a library package using the following naming convention:

```
package PHIGS_UESC_<name of the escape procedure> is
  procedure ESC;
    -- Ada code for UESC procedure

  -- The only procedure name used in the package
  -- will be ESC

end PHIGS_UESC_<name of the escape procedure>;
```

5.14 Additional Functions

5.14.1 Subprograms for Manipulating Input Data Records

The procedures and functions defined in this section are those that are necessary for constructing and inquiring the input data records, declared as private types in this binding for each of the six classes of input devices defined by the PHIGS specification -- the Locator, Stroke, Valuator, Choice, Pick and String logical devices. The procedures listed here are used to construct the data records for each of the registered prompt and echo types of a device class to be used for initializing a particular input device. Assorted functions are also provided so that an application may examine the parts of the data record which are defined by PHIGS. Any implementation specific information in the data records is kept private and unavailable.

A PHIGS_ERROR is generated if any of the below procedures are used incorrectly. For example, if an illegal prompt and echo type is used for a build procedure, then error number 2500 is logged onto the error file.

To implement implementation dependent and registered items an implementation may provide additional overloaded versions of the BUILD procedures in this section and additional functions for extracting information from the private data records.

-- Locator Data Record Operations

```
procedure BUILD_LOCATOR_DATA_RECORD
  (PROMPT_ECHO_TYPE : in LOCATOR_PROMPT_ECHO_TYPE;
   DATA_RECORD      : out LOCATOR_DATA_RECORD);

-- Constructs and returns a locator data record for the
-- locator prompt and echo types 1, 2, 3, and 6.

procedure BUILD_LOCATOR_DATA_RECORD
  (PROMPT_ECHO_TYPE : in LOCATOR_PROMPT_ECHO_TYPE;
   CONTENTS         : in LINE_DATA;
   DATA_RECORD      : out LOCATOR_DATA_RECORD);

-- Constructs and returns a locator data record for prompt and
-- echo type 4 or for echo type 5 when its attributes are
-- specified by Polyline attributes.

procedure BUILD_LOCATOR_DATA_RECORD
  (PROMPT_ECHO_TYPE : in LOCATOR_PROMPT_ECHO_TYPE;
   CONTENTS         : in INTERIOR_DATA;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0575512 079 ■


```
    DATA_RECORD          : out LOCATOR_DATA_RECORD);

-- Constructs and returns a locator data record for the locator
-- prompt and echo type 5 when its attributes are specified only by
-- interior attributes.

procedure BUILD_LOCATOR_DATA_RECORD
  (PROMPT_ECHO_TYPE  : in LOCATOR_PROMPT_ECHO_TYPE;
   INTERIOR_CONTENTS : in INTERIOR_DATA;
   EDGE_CONTENTS     : in EDGE_DATA;
   DATA_RECORD       : out LOCATOR_DATA_RECORD);

-- Constructs and returns a locator data record for the locator
-- prompt and echo type 5 when its attributes are specified by
-- both interior and edge attributes.

function LOCATOR_ATTRIBUTES_USED
          (DATA_RECORD : in LOCATOR_DATA_RECORD)
  return POLYLINE_FILL_AREA_CONTROL_FLAG;

-- Returns which attribute sets are stored in the data record for
-- locator prompt and echo type 5.

function LINE_ATTRIBUTES (DATA_RECORD : in LOCATOR_DATA_RECORD)
  return LINE_DATA;

-- Returns the Polyline attribute information stored in the data
-- record for locator prompt and echo types 4 or 5.

function INTERIOR_ATTRIBUTES (DATA_RECORD : in  LOCATOR_DATA_RECORD)
  return INTERIOR_DATA;

-- Returns the Interior attribute information stored in the data
-- record for locator prompt and echo type 5.

function EDGE_ATTRIBUTES (DATA_RECORD : in  LOCATOR_DATA_RECORD)
  return EDGE_DATA;

-- Returns the Edge attribute information stored in the data
-- record for locator prompt and echo type 5.

-- Stroke Data Record Operations

procedure BUILD_STROKE_DATA_RECORD
  (PROMPT_ECHO_TYPE : in STROKE_PROMPT_ECHO_TYPE;
   BUFFER_SIZE      : in PHIGS_POSITIVE;
   POSITION         : in PHIGS_POSITIVE;
   INTERVAL         : in WC.SIZE_2;
   TIME             : in DURATION;
   DATA_RECORD      : out STROKE_DATA_RECORD);

-- Constructs and returns a stroke data record for the stroke
-- prompt and echo types 1 and 2.

procedure BUILD_STROKE_DATA_RECORD
  (PROMPT_ECHO_TYPE : in STROKE_PROMPT_ECHO_TYPE;
   BUFFER_SIZE      : in PHIGS_POSITIVE;
   POSITION         : in PHIGS_POSITIVE;
   INTERVAL         : in WC.SIZE_2;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

9593 PT*3 90 ■ 4851903 0575513 T05 ■


```
    TIME                : in DURATION;
    CONTENTS            : in MARKER_DATA;
    DATA_RECORD         : out STROKE_DATA_RECORD);

  -- Constructs and returns a stroke data record for the stroke
  -- prompt and echo type 3.

  procedure BUILD_STROKE_DATA_RECORD
    (PROMPT_ECHO_TYPE : in STROKE_PROMPT_ECHO_TYPE;
     BUFFER_SIZE      : in PHIGS_POSITIVE;
     POSITION         : in PHIGS_POSITIVE;
     INTERVAL         : in WC.SIZE_2;
     TIME             : in DURATION;
     CONTENTS         : in LINE_DATA;
     DATA_RECORD      : out STROKE_DATA_RECORD);

  -- Constructs and returns a stroke data record for the stroke
  -- prompt and echo type 4.

  function BUFFER_SIZE (DATA_RECORD : in STROKE_DATA_RECORD)
    return PHIGS_POSITIVE;

  -- Returns the size of the input stroke buffer stored in the data
  -- record for the stroke prompt and echo types 1, 2, 3, and 4.

  function POSITION (DATA_RECORD : in STROKE_DATA_RECORD)
    return PHIGS_POSITIVE;

  -- Returns the editing position within the stroke input buffer
  -- stored in the data record for the stroke prompt and echo
  -- types 1, 2, 3, and 4.

  function INTERVAL (DATA_RECORD : in STROKE_DATA_RECORD)
    return WC.SIZE_2;

  -- Returns the interval value stored in the stroke data record
  -- for the prompt and echo types 1, 2, 3, and 4.

  function TIME (DATA_RECORD : in STROKE_DATA_RECORD)
    return DURATION;

  -- Returns the measuring time for sampling stroke input stored in
  -- the stroke data record for the stroke prompt and echo types
  -- 1, 2, 3, and 4.

  function MARKER_ATTRIBUTES (DATA_RECORD : in STROKE_DATA_RECORD)
    return MARKER_DATA;

  -- Returns the Polymarker attributes used to echo the stroke input
  -- stored in the data record for prompt and echo type 3.

  function LINE_ATTRIBUTES (DATA_RECORD : in STROKE_DATA_RECORD)
    return LINE_DATA;

  -- Returns the Polyline attributes used to echo the stroke input
  -- stored in the data record for stroke prompt and echo type 4.

-- Valuator Data Record Operations
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
procedure BUILD_VALUATOR_DATA_RECORD
  (PROMPT_ECHO_TYPE : in VALUATOR_PROMPT_ECHO_TYPE;
   LOW_VALUE        : in VALUATOR_VALUE;
   HIGH_VALUE       : in VALUATOR_VALUE;
   DATA_RECORD      : out VALUATOR_DATA_RECORD);

-- Constructs and returns a valuator data record for valuator
-- prompt and echo type 1.

function HIGH_VALUE (DATA_RECORD : in VALUATOR_DATA_RECORD)
  return VALUATOR_VALUE;

-- Returns the high value for the valuator stored in the valuator
-- data record for the prompt and echo types 1, 2, and 3.

function LOW_VALUE (DATA_RECORD : in VALUATOR_DATA_RECORD)
  return VALUATOR_VALUE;

-- Returns the low value for the valuator stored in the valuator
-- data record for the prompt and echo types 1, 2, and 3.

-- Choice Data Record Operations

procedure BUILD_CHOICE_DATA_RECORD
  (PROMPT_ECHO_TYPE : in CHOICE_PROMPT_ECHO_TYPE;
   DATA_RECORD      : out CHOICE_DATA_RECORD);

-- Constructs and returns a choice data record for the choice
-- prompt and echo type 1.

procedure BUILD_CHOICE_DATA_RECORD
  (PROMPT_ECHO_TYPE : in CHOICE_PROMPT_ECHO_TYPE;
   ARRAY_OF_PROMPTS : in CHOICE_PROMPTS.LIST_OF;
   DATA_RECORD      : out CHOICE_DATA_RECORD);

-- Constructs and returns a choice data record for the choice
-- prompt and echo type 2.

procedure BUILD_CHOICE_DATA_RECORD
  (PROMPT_ECHO_TYPE : in CHOICE_PROMPT_ECHO_TYPE;
   ARRAY_OF_STRINGS : in CHOICE_PROMPT_STRING_LIST;
   DATA_RECORD      : out CHOICE_DATA_RECORD);

-- Constructs and returns a choice data record for the choice
-- prompt and echo types 3 and 4.

procedure BUILD_CHOICE_DATA_RECORD
  (PROMPT_ECHO_TYPE     : in CHOICE_PROMPT_ECHO_TYPE;
   STRUCTURE_IDENTIFIER : in STRUCTURE_ID;
   LIST_OF_PICK_IDS     : in PICK_IDS.LIST_OF;
   DATA_RECORD          : out CHOICE_DATA_RECORD);

-- Constructs and returns a choice data record for the choice
-- prompt and echo type 5.

function ARRAY_OF_PROMPTS (DATA_RECORD : in CHOICE_DATA_RECORD)
  return CHOICE_PROMPTS.LIST_OF;

-- Returns the array of prompts stored in the choice data record
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

9593 PT*3 90 ■ 4851903 0575515 888 ■


```
      -- for the prompt and echo type 2.

      function ARRAY_OF_STRINGS (DATA_RECORD : in CHOICE_DATA_RECORD)
        return CHOICE_PROMPT_STRING_LIST;

      -- Returns the array of prompt strings stored in the choice data
      -- record for the prompt and echo types 3 and 4.

      function STRUCTURE_IDENTIFIER (DATA_RECORD : in CHOICE_DATA_RECORD)
        return STRUCTURE_ID;

      -- Returns the structure identifier stored in the choice data
      -- record for the prompt and echo type 5.

      function LIST_OF_PICK_IDS (DATA_RECORD : in CHOICE_DATA_RECORD)
        return PICK_IDS.LIST_OF;

      -- Returns the list of pick identifiers stored in the choice data
      -- record for the prompt and echo type 5.

-- Pick Data Record Operations.

      procedure BUILD_PICK_DATA_RECORD
        (PROMPT_ECHO_TYPE : in PICK_PROMPT_ECHO_TYPE;
         DATA_RECORD      : out PICK_DATA_RECORD);

      -- Constructs and returns a pick data record for the pick
      -- prompt and echo types 1, 2, and 3.

-- String Data Record Operations

      procedure BUILD_STRING_DATA_RECORD
        (PROMPT_ECHO_TYPE       : in STRING_PROMPT_ECHO_TYPE;
         INPUT_BUFFER_SIZE      : in PHIGS_NATURAL;
         INITIAL_CURSOR_POSITION : in PHIGS_NATURAL;
         DATA_RECORD            : out STRING_DATA_RECORD);

      -- Constructs and returns a string data record for the string
      -- prompt and echo type 1.

      function INPUT_BUFFER_SIZE (DATA_RECORD : in STRING_DATA_RECORD)
        return PHIGS_NATURAL;

      -- Returns the size of the buffer used for storing string input
      -- stored in the string data record for prompt and echo type 1.

      function INITIAL_CURSOR_POSITION (DATA_RECORD : in STRING_DATA_RECORD)
        return PHIGS_NATURAL;

      -- Returns the initial cursor position for string input stored in
      -- the string data record for prompt and echo type 1.
```

5.14.2 PHIGS Generic Coordinate System Package

The generic package declared in this section is the specification of a generic Cartesian Coordinate System for PHIGS. This package is instantiated in package PHIGS_TYPES once for each of Modelling Coordinates,

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0575516 714 ■

World Coordinates, View Reference Coordinates, Normalized Projection Coordinates, and Device Coordinates. The package defines the representation of the following data types:

POINT_3
POINT_2
POINT_LIST_3
POINT_LIST_2
ACCESS_POINT_LIST_3
ACCESS_POINT_LIST_2
LIST_OF_POINT_LIST_3
LIST_OF_POINT_LIST_2
ACCESS_LIST_OF_POINT_LIST_3
ACCESS_LIST_OF_POINT_LIST_2
VECTOR_3
VECTOR_PAIR_3
VECTOR_2
VECTOR_PAIR_2
RECTANGULAR_REGION_3
RECTANGULAR_REGION_2
HALF_SPACE_3
HALF_SPACE_2
HALF_SPACE_LIST_3
HALF_SPACE_LIST_2
ACCESS_HALF_SPACE_LIST_3
ACCESS_HALF_SPACE_LIST_2

Also defined is a MAGNITUDE type for measuring lengths within a coordinate space. The type SIZE measures lengths parallel to both the axes, and the RANGE_OF_MAGNITUDES type specifies two lengths within a coordinate system, a minimum and maximum for values such as the range of Character Heights available on a device.

```
generic

  type COORDINATE_COMPONENT_TYPE is digits <>;

package PHIGS_COORDINATE_SYSTEM is


  type POINT_3 is
    record
      X : COORDINATE_COMPONENT_TYPE;
      Y : COORDINATE_COMPONENT_TYPE;
      Z : COORDINATE_COMPONENT_TYPE;
    end record;


  type POINT_2 is
    record
      X : COORDINATE_COMPONENT_TYPE;
      Y : COORDINATE_COMPONENT_TYPE;
    end record;

  type POINT_LIST_3 is array (POSITIVE range <>) of POINT_3;

  type POINT_LIST_2 is array (POSITIVE range <>) of POINT_2;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
type ACCESS_POINT_LIST_3 is access POINT_LIST_3;

type ACCESS_POINT_LIST_2 is access POINT_LIST_3;


type LIST_OF_POINT_LIST_3 is
  array (POSITIVE range < >) of ACCESS_POINT_LIST_3;

type LIST_OF_POINT_LIST_2 is
  array (POSITIVE range < >) of ACCESS_POINT_LIST_2;


type ACCESS_LIST_OF_POINT_LIST_3 is access LIST_OF_POINT_LIST_3;

type ACCESS_LIST_OF_POINT_LIST_2 is access LIST_OF_POINT_LIST_2;


type VECTOR_3 is new POINT_3;

type VECTOR_PAIR_3 is array (1..2) of VECTOR_3;

type VECTOR_2 is new POINT_2;

type VECTOR_PAIR_2 is array (1..2) of VECTOR_2;

type RECTANGULAR_REGION_3 is
  record
    XMIN  : COORDINATE_COMPONENT_TYPE;
    XMAX  : COORDINATE_COMPONENT_TYPE;
    YMIN  : COORDINATE_COMPONENT_TYPE;
    YMAX  : COORDINATE_COMPONENT_TYPE;
    ZMIN  : COORDINATE_COMPONENT_TYPE;
    ZMAX  : COORDINATE_COMPONENT_TYPE;
  end record;


type RECTANGULAR_REGION_2 is
  record
    XMIN  : COORDINATE_COMPONENT_TYPE;
    XMAX  : COORDINATE_COMPONENT_TYPE;
    YMIN  : COORDINATE_COMPONENT_TYPE;
    YMAX  : COORDINATE_COMPONENT_TYPE;
  end record;


type HALF_SPACE_3 is
  record
    REFERENCE_POSITION   : POINT_3;
    ACCEPTANCE_NORMAL : VECTOR_3;
  end record;


type HALF_SPACE_2 is
  record
    REFERENCE_POSITION   : POINT_2;
    ACCEPTANCE_NORMAL : VECTOR_2;
  end record;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
type HALF_SPACE_LIST_3 is
  array (POSITIVE range <>) of HALF_SPACE_3;

type HALF_SPACE_LIST_2 is
  array (POSITIVE range <>) of HALF_SPACE__2;


type ACCESS_HALF_SPACE_LIST_3 is access HALF_SPACE_LIST_3;

type ACCESS_HALF_SPACE_LIST_2 is access HALF_SPACE_LIST_2;


type MAGNITUDE_BASE_TYPE is digits PHIGS_PRECISION;

subtype MAGNITUDE is
  MAGNITUDE_BASE_TYPE range
      COORDINATE_COMPONENT_TYPE'SAFE_SMALL ..
      COORDINATE_COMPONENT_TYPE'SAFE_LARGE;


type SIZE_3 is
  record
    XAXIS : MAGNITUDE;
    YAXIS : MAGNITUDE;
    ZAXIS : MAGNITUDE;
  end record;


type SIZE_2 is
  record
    XAXIS : MAGNITUDE;
    YAXIS : MAGNITUDE;
  end record;


type RANGE_OF_MAGNITUDES is
  record
    MIN  : MAGNITUDE;
    MAX : MAGNITUDE;
  end record;

end PHIGS_COORDINATE_SYSTEM;
```

5.14.3 PHIGS Generic List Utility Package

The generic package PHIGS_LIST_UTILITIES is instantiated in the PHIGS_TYPES package once for each of several LIST_OF types and their manipulation subprograms. Each LIST_OF type contains different element type values.

The LIST_OF type is declared as a private type in PHIGS_LIST_UTILITIES to restrict the operations on the LIST_OF type that are available to outside program units. The LIST_OF private type declaration includes a discriminant part that defines the current size of the lists. LIST_OF objects are declared as unconstrained objects (by using the default discriminant value) to allow dynamic modification of the list size.

A LIST_OF object is a sequence of element type values. Each element type value is associated with an index. Index values begin at one and increase in steps of one.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

The size of a LIST_OF object is the number of element type values stored within it. A single element type value may be stored more than onece within a LIST_OF object. A LIST_OF object may be empty. The size of an empty LIST_OF object is zero. The maximum size of a LIST_OF object is given by the MAX_LIST_SIZE generic parameter. If this parameter is not specified in the instantiation, an implementation dependent value is used.

An adequate package specification is shown below. The comments within the package describe the functionality of the manipulation subprograms:

```
generic
   type ELEMENT_TYPE is private;

   MAX_LIST_SIZE : NATURAL := PHIGS_CONFIGURATION.DEFAULT_LIST_SIZE;


package PHIGS_LIST_UTILITIES is

   subtype LIST_SIZE is NATURAL range 0..MAX_LIST_SIZE;

   type LIST_OF (SIZE : LIST_SIZE := 0) is limited private;

   type LIST_VALUES is ARRAY (POSITIVE range <>) of ELEMENT_TYPE;


   function NULL_LIST return LIST_OF;

      -- This function returns an empty LIST_OF object. This list is
      -- intended primarily for use by implementors.


   function SIZE_OF_LIST (LIST : in LIST_OF) return NATURAL;

      -- This function returns the number of element type values stored
      -- in the list object.


   function IS_IN_LIST (ELEMENT  : in ELEMENT_TYPE;
                        LIST     : in LIST_OF) return BOOLEAN;

      -- This function returns the value TRUE if the element parameter
      -- value is in the list object. Otherwise it returns the value
      -- FALSE.


   function LIST_ELEMENT (INDEX  : in POSITIVE;
                          LIST   : in  LIST_OF) return ELEMENT_TYPE;

      -- This function returns the element value in the list object that
      -- has an associated index value equal to the index parameter. The
      -- PHIGS_ERROR 2502 is generated if the index parameter exceeds the
      -- current size of the list parameter object.
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
function LIST (VALUES : in LIST_VALUES) return LIST_OF;

   -- This function returns a valid LIST_OF object. If the VALUES
   -- parameter is a null array, an empty LIST_OF object is returned.
   -- If the values parameter is not null, this function returns a
   -- LIST_OF object containing all the values in the VALUES parameter.
   -- The PHIGS_ERROR 2502 is generated if the number of element values
   -- exceeds the specified maximum size of the LIST_OF object.


procedure ADD_TO_LIST (ELEMENT  : in ELEMENT_TYPE;
                       LIST     : in out LIST_OF);

   -- This procedure stores the element parameter value in the list
   -- parameter object, and increases the size of the list object
   -- by one. An index value equal to the incremented list size
   -- is associated with the stored element value. The ADD_TO_LIST
   -- procedure will generate PHIGS_ERROR 2502 if it is called when
   -- the list parameter has a size equal to the specified maximum
   -- size. If desired, the user can ensure duplicate values are not
   -- stored. This is accomplised by calling ADD_TO_LIST with a
   -- particular element value only if the function IS_IN_LIST returns
   -- false for that element value.


procedure DELETE_FROM_LIST (ELEMENT  : in ELEMENT_TYPE;
                            LIST     : in out LIST_OF);

   -- If the list parameter object does not contain the element
   -- parameter value, this procedure has no effect. Otherwise, the
   -- first occurrence of the element value is deleted. The size of
   -- the list object is decreased by one, and the indices associated
   -- with the remaining element values are adjusted so that the indices
   -- begin at one and increment in steps of one. If desired, the user
   -- can delete all occurrences of an element value. This is accomplished
   -- by calling DELETE_FROM_LIST repeatedly with a particular element
   -- value while the function IS_IN_LIST returns TRUE for that value.


private

   -- The declaration of the LIST_OF type is implementation dependent.
   -- However, the operations implicitly declared by the LIST_OF declaration,
   -- including both assignment and the predefined comparison for equality
   -- and inequality, shall produce the correct results. This requirement
   -- precludes the use of access types for the implementation of the
   -- LIST_OF type. The recommended implementation is given below:
   --
   --       type LIST_OF (SIZE : LIST_SIZE := 0) is
   --         record
   --           ELEMENTS : LIST_VALUES(1..SIZE);
   --         end record;
   --
   -- Note that declaring unconstrained LIST_OF objects by using the default
   -- discriminant value allows dynamic modification of the size of the
   -- element array.

end PHIGS_LIST_UTILITIES;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

5.14.4 PHIGS Name Set Facility Package

The types, procedures, and functions defined in this section are those necessary for constructing and inquiring name sets, declared as private types in this binding. The procedures listed here are used to construct the data items of type NAME_SET and NAME_SET_FILTER and to extract information about the classes contained included in NAME_SET data elements. Assorted functions are also provided so that an application may obtain information about the characteristics of one or more name sets.

The procedure COPY_NAME_SET and the function IS_EQUAL are provided as replacements for the normal Ada assignment and equality operators for NAME_SET objects. Conforming programs shall use these replacement functions. This allows the use of Ada access types to implement name sets at the discretion of the implementation.

PHIGS_ERROR 2501 is generated if any of the below procedures are used incorrectly.

```
with PHIGS_CONFIGURATION, PHIGS_LIST_UTILITIES;

package PHIGS_NAME_SET_FACILITY is

  -- PHIGS_CONFIGURATION.MAX_NAME_SUPPORTED

    -- An implementation dependent value which defines the maximum
    -- name set class supported.

  -- NAME SET

  type NAME_SET is private;

    -- Defines a name set data element.

  -- NAME_SET_ACCEPTANCE

  type NAME_SET_ACCEPTANCE is (REJECTED, ACCEPTED);

    -- Used to indicate the results of applying a name set
    -- against a filter pair.

  -- NAME

  type NAME is range 0..PHIGS_CONFIGURATION.MAX_NAME_SUPPORTED;

    -- Provides for names for each of the name set classes.


  -- NAMES

  package NAMES is
    new PHIGS_LIST_UTILITIES (NAME);

    -- Provides for lists of names.
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0575522 T18 ■


```
-- NAME_SET_FILTER

type NAME_SET_FILTER is
  record
    INCLUSION   : NAME_SET;
    EXCLUSION   : NAME_SET;
  end record;

  -- Used to define name set pairs used as filters.


-- NAME_SET_MEMBERSHIP

type NAME_SET_MEMBERSHIP is (NON_MEMBER, MEMBER);

  -- Provides for indicating whether a name is a member of a
  -- name set.


-- Subprograms for Manipulating Name Sets

procedure BUILD_NAME_SET
  (MEMBERS : in  NAMES.LIST_OF;
   SET     : out NAME_SET);

  -- Constructs a name set from a list of names.


procedure BUILD_NAME_SET_UNION
  (LEFT_SET  : in  NAME_SET;
   RIGHT_SET : in  NAME_SET;
   UNION     : out NAME_SET);

  -- Constructs the logical union of two name sets.


procedure BUILD_NAME_SET_INTERSECTION
  (LEFT_SET    : in  NAME_SET;
   RIGHT_SET   : in  NAME_SET;
   INTERSECTION : out NAME_SET);

  -- Constructs the logical intersection of two name sets.


procedure BUILD_NAME_SET_DIFFERENCE
  (LEFT_SET   : in  NAME_SET;
   RIGHT_SET  : in  NAME_SET;
   DIFFERENCE : out NAME_SET);

  -- Constructs the logical difference of two name sets.


procedure COPY_NAME_SET
  (ORIGINAL : in  NAME_SET;
   COPY     : out NAME_SET);

  -- Duplicates the provided name set.
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
function IS_EQUAL
  (LEFT_SET   : in NAME_SET;
   RIGHT_SET  : in NAME_SET)
  return BOOLEAN;

  -- Compares two name sets.


procedure EXTRACT_NAMES
  (SET         : in  NAME_SET;
   MEMBERS     : out NAMES.LIST_OF);

  -- Extracts the names from the provided name set.


procedure ADD_NAMES
  (MEMBERS : in     NAMES.LIST_OF;
   SET     : in out NAME_SET);

  -- Adds names to the name set if not already there.


procedure REMOVE_NAMES
  (MEMBERS : in     NAMES.LIST_OF;
   SET     : in out NAME_SET);

  -- Removes the names from the name set if they are there.


function APPLY_FILTER
  (SET      : in NAME_SET;
   FILTER   : in NAME_SET_FILTER)
  return NAME_SET_ACCEPTANCE;

  -- Applies a filter to a name set and indicates whether the name
  -- set passes through the filter.


function INQUIRE_MEMBERSHIP
  (MEMBER  : in NAME;
   SET     : in NAME_SET)
  return NAME_SET_MEMBERSHIP;

  -- Returns an indication of whether the indicated name is a member
  -- of the provided name set.


procedure DEALLOCATE
  (OBJECT  : in out NAME_SET);

  -- Deallocates name set objects.
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0575524 890 ■


```
private

   -- The following types define the specifications for private
   -- types used in the PHIGS_NAME_SET_FACILITY. Null records are
   -- used solely for the purposes of demonstrating compilability.
   -- An implementation would use some more appropriate declaration.

   type NAME_SET is
     record
       null;
     end record;

end PHIGS_NAME_SET_FACILITY;
```

5.14.5 Deallocation of structure element records

The contents of objects of type STRUCTURE_ELEMENT_RECORD and the various private types may contain data allocated from heap space. For example, objects of type STRUCTURE_ELEMENT_RECORD are used to store definitions of structure elements. The data types of these elements differ and vary according to the type of structure element. Some elements are as complex as variable length lists of variable length lists of points. The size and dimension of such lists is known only by the implementation at the time an application invokes the inquiry functions which use the STRUCTURE_ELEMENT_RECORD type. Similarly, objects of the various private types may also contain allocated data space which represent variable length lists. The implementation can allocate the exact amount of memory needed to return the variable length lists.

To avoid requiring that an application provide cumbersome code to deallocate all of the various possible cases, deallocation functions for each such data type are provided. These functions are all named DEALLOCATE and may be used by either the application or the implementation to free the memory space used by the corresponding records. For example, the implementation uses the DEALLOCATE function for objects of type STRUCTURE_ELEMENT_RECORD on successive calls to either INQ_ELEMENT_CONTENT or INQ_CURRENT_ELEMENT_CONTENT to free the space which may have been allocated in a previous call with the same object. The application will use the function to free memory after the last invocation of these inquiry functions. Once the memory has been released, the access variables will be set to the null value.

The REQUEST_STROKE, SAMPLE_STROKE, GET_STROKE, INQ_PATTERN_REPRESENTATION, INQ_STROKE_DEVICE_STATE, and INQ_PREDEFINED_PATTERN_REPRESENTATION functions also return data in the form of access types. Since the space for this data can easily be deallocated by the application by using the Ada predefined generic library procedure UNCHECKED_DEALLOCATION, no additional functions are provided in this binding.

DEALLOCATE

```
procedure DEALLOCATE
  (OBJECT : in out CHOICE_DATA_RECORD);
```

DEALLOCATE

```
procedure DEALLOCATE
  (OBJECT : in out ESCAPE_DATA_RECORD);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

---

DEALLOCATE

```
procedure DEALLOCATE
  (OBJECT : in out GDP_3_RECORD);
```

DEALLOCATE

```
procedure DEALLOCATE
  (OBJECT : in out GDP_RECORD);
```

DEALLOCATE

```
procedure DEALLOCATE
  (OBJECT : in out GSE_RECORD);
```

DEALLOCATE

```
procedure DEALLOCATE
  (OBJECT : in out LOCATOR_DATA_RECORD);
```

DEALLOCATE

```
procedure DEALLOCATE
  (OBJECT : in out METAFILE_DATA_RECORD);
```

DEALLOCATE

```
procedure DEALLOCATE
  (OBJECT : in out NAME_SET)
  renames PHIGS_NAME_SET_FACILITY.DEALLOCATE;
```

DEALLOCATE

```
procedure DEALLOCATE
  (OBJECT : in out PICK_DATA_RECORD);
```

DEALLOCATE

```
procedure DEALLOCATE
  (OBJECT : in out STRING_DATA_RECORD);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

DEALLOCATE

```
procedure DEALLOCATE
  (OBJECT : in out STROKE_DATA_RECORD);
```

DEALLOCATE

```
procedure DEALLOCATE
  (OBJECT : in out STRUCTURE_ELEMENT_RECORD);
```

DEALLOCATE

```
procedure DEALLOCATE
  (OBJECT : in out VALUATOR_DATA_RECORD);
```

5.14.6 Metafile Function Utilities

Metafile item data records are complex and are dependent on the encoding for the specified workstation. Record length depends on the number of data elements. PHIGS defines that the format is implementation defined. The item data record type should be private to allow direct manipulation of the record contents in order to have them efficiently processed.

The application programmer shall be able to write non-graphical data into the metafile. This can be provided by allowing character strings to be output. Numeric data can be converted to a string prior to output. A function is provided as a means to convert item data records into strings.

BUILD NEW METAFILE DATA RECORD

```
procedure BUILD_NEW_METAFILE_DATA_RECORD
  (TYPE_OF_ITEM  : in METAFILE_ITEM_TYPE;
   ITEM_DATA     : in PHIGS_STRING;
   ITEM          : out METAFILE_DATA_RECORD);
```

ITEM DATA RECORD STRING

```
function ITEM_DATA_RECORD_STRING
  (ITEM : in METAFILE_DATA_RECORD) return ACCESS_STRING;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

5.15 Conformal Variants

Since no subsets or supersets of the Ada language are allowed, PHIGS/Ada has no conformal variants. Furthermore, this binding does not require the use of any Ada language feature for which support of that feature is implementation-dependent.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS

Not for Resale

9593 PT*3 90 ■ 4851903 0575528 436 ■

Annex A

(informative)

Compilable PHIGS Specification

This annex is intended to show one example of a compilable PHIGS binding to Ada. It is not the only possible such example. It is expected that implementations of a PHIGS binding to Ada will need to add additional constructs to the various packages or rearrange the constructs in a manner suitable to the implementation. However, all names defined in this binding shall be visible as dictated in this standard.

-- **PHIGS Configuration Package**

```
package PHIGS_CONFIGURATION is

-- The following define example values for the PHIGS configuration names.
-- Note that the values specified are implementation dependent. The values
-- shown below are arbitrary and have been chosen solely as examples. More
-- appropriate values should be chosen by an implementation. Note also that
-- only the technique of defining the names as constants was used in this
-- example. An implementation is free to choose the most appropriate means
-- of providing the value as discussed in 4.2.5.

  DEFAULT_ERROR_FILE                          : constant STRING := "ERROR.FIL";
  DEFAULT_LIST_SIZE                           : constant := 256;
  DEFAULT_MEMORY_UNITS                        : constant := 0;
  MAX_ANNOTATION_STYLES_SUPPORTED             : constant := 2;
  MAX_APPLICATION_DATA                        : constant := 256;
  MAX_ARCHIVE_IDS_SUPPORTED                   : constant := 4;
  MAX_CHAR_SETS_SUPPORTED                     : constant := 1;
  MAX_CHOICE_PROMPTS_SUPPORTED                : constant := 32;
  MAX_CHOICE_PROMPT_ECHO_TYPES_SUPPORTED      : constant := 10;
  MAX_CHOICE_SMALL_NATURAL                    : constant := 64;
  MAX_COLOUR_COEFFICIENTS                     : constant := 4;
  MAX_COLOUR_INDICES_SUPPORTED                : constant := 4096;
  MAX_COLOUR_MATRIX_SMALL_NATURAL             : constant := 128;
  MAX_COLOUR_MODELS_SUPPORTED                 : constant := 2;
  MAX_EDGE_INDICES_SUPPORTED                  : constant := 5;
  MAX_EDGETYPES_SUPPORTED                     : constant := 1;
  MAX_ESCAPE_IDS_SUPPORTED                    : constant := 16;
  MAX_FILE_IDS_SUPPORTED                      : constant := 10;
  MAX_FILE_SMALL_NATURAL                      : constant := 80;
  MAX_GDP_3_IDS_SUPPORTED                     : constant := 24;
  MAX_GDP_IDS_SUPPORTED                       : constant := 24;
  MAX_GSE_IDS_SUPPORTED                       : constant := 8;
  MAX_HATCH_STYLES_SUPPORTED                  : constant := 64;
  MAX_HLHSR_IDS_SUPPORTED                     : constant := 128;
  MAX_HLHSR_MODES_SUPPORTED                   : constant := 4;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
   MAX_INPUT_STRING_SMALL_NATURAL                          : constant := 80;
   MAX_INTERIOR_INDICES_SUPPORTED                          : constant := 5;
   MAX_LINETYPES_SUPPORTED                                 : constant := 5;
   MAX_LOCATOR_PROMPT_ECHO_TYPES_SUPPORTED                 : constant := 8;
   MAX_MARKER_TYPES_SUPPORTED                              : constant := 256;
   MAX_MEMORY_UNITS                                        : constant := 0;
   MAX_METAFILE_ITEM_LENGTH                                : constant := 1024;
   MAX_METAFILE_ITEM_TYPE                                  : constant := 256;
   MAX_MODELLING_CLIP_OPERATION_TYPES_SUPPORTED            : constant := 2;
   MAX_MODELLING_CLIP_DISTINCT_PLANES_SUPPORTED            : constant := 6;
   MAX_NAME_SUPPORTED                                      : constant := 63;
   MAX_NAME_SET_FILTER_LIST_LENGTH_SUPPORTED               : constant := 4;
   MAX_OPEN_WS_SUPPORTED                                   : constant := 4;
   MAX_PATH_DEPTH_SUPPORTED                                : constant := 64;
   MAX_PATTERN_INDICES_SUPPORTED                           : constant := 4;
   MAX_PICK_IDS_SUPPORTED                                  : constant := 64536;
   MAX_PICK_PROMPT_ECHO_TYPES_SUPPORTED                    : constant := 2;
   MAX_POLYLINE_INDICES_SUPPORTED                          : constant := 5;
   MAX_POLYMARKER_INDICES_SUPPORTED                        : constant := 5;
   MAX_POSTED_STRUCTURES_SUPPORTED                         : constant := 4096;
   MAX_REFERENCE_PATHS_SUPPORTED                           : constant := 64;
   MAX_SMALL_NATURAL                                       : constant := 255;
   MAX_STRING_PROMPT_ECHO_TYPES_SUPPORTED                  : constant := 2;
   MAX_STRING_SMALL_NATURAL                                : constant := 255;
   MAX_STROKE_PROMPT_ECHO_TYPES_SUPPORTED                  : constant := 2;
   MAX_STRUCTURE_IDS_SUPPORTED                             : constant := 65535;
   MAX_TEXT_FONT_PRECISION_PAIRS_SUPPORTED                 : constant := 32;
   MAX_TEXT_INDICES_SUPPORTED                              : constant := 5;
   MAX_VALUATOR_PROMPT_ECHO_TYPES_SUPPORTED                : constant := 4;
   MAX_VIEW_INDICES_SUPPORTED                              : constant := 20;
   MAX_WS_IDS_SUPPORTED                                    : constant := 16383;
   MAX_WS_TYPES_SUPPORTED                                  : constant := 64;
   MIN_METAFILE_ITEM_TYPE                                  : constant := 0;
   MIN_INTEGER_VALUE                                       : constant := -2147483648;
   MAX_INTEGER_VALUE                                       : constant := 2147483647;
   PHIGS_PRECISION                                         : constant := 7;

end PHIGS_CONFIGURATION;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
-- PHIGS List Utilities Generic Package

with PHIGS_CONFIGURATION;

generic

   type ELEMENT_TYPE is private;

   MAX_LIST_SIZE : NATURAL := PHIGS_CONFIGURATION.DEFAULT_LIST_SIZE;

package PHIGS_LIST_UTILITIES is

   subtype LIST_SIZE is NATURAL range 0..MAX_LIST_SIZE;

   type LIST_OF (SIZE : LIST_SIZE := 0) is private;

   type LIST_VALUES is ARRAY (POSITIVE range <>) of ELEMENT_TYPE;

   function NULL_LIST return LIST_OF;

      -- This function returns an empty LIST_OF object. This list is
      -- intended primarily for use by implementors.

   function SIZE_OF_LIST (LIST : in LIST_OF) return NATURAL;

      -- This function returns the number of element type values stored
      -- in the list object.

   function IS_IN_LIST (ELEMENT  : in ELEMENT_TYPE;
                        LIST     : in LIST_OF) return BOOLEAN;

      -- This function returns the value TRUE if the element parameter
      -- value is in the list object. Otherwise it returns the value
      -- FALSE.

   function LIST_ELEMENT (INDEX  : in POSITIVE;
                          LIST   : in  LIST_OF) return ELEMENT_TYPE;

      -- This function returns the element value in the list object that
      -- has an associated index value equal to the index parameter. The
      -- PHIGS_ERROR 2502 is generated if the index parameter exceeds the
      -- current size of the list parameter object.

   function LIST (VALUES : in LIST_VALUES) return LIST_OF;

      -- This function returns a valid LIST_OF object. If the VALUES
      -- parameter is a null array, an empty LIST_OF object is returned.
      -- If the values parameter is not null, this function returns a
      -- LIST_OF object containing all the values in the VALUES parameter.
      -- The PHIGS_ERROR 2502 is generated if the number of element values
      -- exceeds the specified maximum size of the LIST_OF object.

   procedure ADD_TO_LIST (ELEMENT  : in ELEMENT_TYPE;
                          LIST     : in out LIST_OF);

      -- This procedure stores the element parameter value in the list
      -- parameter object, and increases the size of the list object
      -- by one. An index value equal to the incremented list size
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
    -- is associated with the stored element value. The ADD_TO_LIST
    -- procedure will generate PHIGS_ERROR 2502 if it is called when
    -- the list parameter has a size equal to the specified maximum
    -- size. If desired, the user can ensure duplicate values are not
    -- stored. This is accomplised by calling ADD_TO_LIST with a
    -- particular element value only if the function IS_IN_LIST returns
    -- false for that element value.

  procedure DELETE_FROM_LIST (ELEMENT : in ELEMENT_TYPE;
                              LIST    : in out LIST_OF);

    -- If the list parameter object does not contain the element
    -- parameter value, this procedure has no effect. Otherwise, the
    -- first occurrence of the element value is deleted. The size of
    -- the list object is decreased by one, and the indices associated
    -- with the remaining element values are adjusted so that the indices
    -- begin at one and increment in steps of one. If desired, the user
    -- can delete all occurrences of an element value. This is accomplished
    -- by calling DELETE_FROM_LIST repeatedly with a particular element
    -- value while the function IS_IN_LIST returns TRUE for that value.

private

  -- The declaration of the LIST_OF type is implementation dependent.
  -- However, the operations implicitly declared by the LIST_OF declaration,
  -- including both assignment and the predefined comparison for equality
  -- and inequality, shall produce the correct results. This requirement
  -- precludes the use of access types for the implementation of the
  -- LIST_OF type. The recommended implementation is given below:
  --
  type LIST_OF (SIZE : LIST_SIZE := 0) is
    record
      ELEMENTS : LIST_VALUES(1..SIZE);
    end record;
  --
  -- Note that declaring unconstrained LIST_OF objects by using the default
  -- discriminant value allows dynamic modification of the size of the
  -- element array.

end PHIGS_LIST_UTILITIES;

package body PHIGS_LIST_UTILITIES is

  -- This package body contains stubs for each of the subprograms
  -- contained in the PHIGS_LIST_UTILITIES package. The purpose of
  -- these stubs is to show compilability for the PHIGS/Ada binding.
  -- An implementation would replace these stubs with code appropriate
  -- to implement the functionality.

  function NULL_LIST return LIST_OF is
    EMPTY_LIST : LIST_OF;
  begin
    return EMPTY_LIST;
  end NULL_LIST;

  function SIZE_OF_LIST (LIST : in LIST_OF) return NATURAL is
  begin
    return 0;
  end SIZE_OF_LIST;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
   function IS_IN_LIST (ELEMENT : in ELEMENT_TYPE;
                        LIST    : in LIST_OF) return BOOLEAN is
   begin
     return FALSE;
   end IS_IN_LIST;

   function LIST_ELEMENT (INDEX : in POSITIVE;
                          LIST  : in  LIST_OF) return ELEMENT_TYPE is
   EXAMPLE_ELEMENT : ELEMENT_TYPE;
   begin
     return EXAMPLE_ELEMENT;
   end LIST_ELEMENT;

   function LIST (VALUES : in LIST_VALUES) return LIST_OF is
   begin
     return NULL_LIST;
   end LIST;

   procedure ADD_TO_LIST (ELEMENT : in ELEMENT_TYPE;
                          LIST    : in out LIST_OF) is
   begin
     null;
   end ADD_TO_LIST;

   procedure DELETE_FROM_LIST (ELEMENT : in ELEMENT_TYPE;
                               LIST    : in out LIST_OF) is
   begin
     null;
   end DELETE_FROM_LIST;

end PHIGS_LIST_UTILITIES;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
-- PHIGS Name Set Facility Package

with PHIGS_CONFIGURATION, PHIGS_LIST_UTILITIES;

package PHIGS_NAME_SET_FACILITY is

   -- PHIGS_CONFIGURATION.MAX_NAME_SUPPORTED

   -- An implementation dependent constant which defines the maximum
   -- name supported. The minimum value allowed by the PHIGS
   -- specification is 63 thus supporting 64 classes. A larger
   -- value is strongly encouraged.

   type NAME_SET is private;

   -- Defines a name set data element.

   -- NAME_SET_ACCEPTANCE

   type NAME_SET_ACCEPTANCE is (REJECTED, ACCEPTED);

   -- Used to indicate the results of applying a name set
   -- against a filter pair.

   -- NAME

   type NAME is
         range 0..PHIGS_CONFIGURATION.MAX_NAME_SUPPORTED;

   -- Provides for names for each of the name set classes.

   -- NAMES

   package NAMES is
            new PHIGS_LIST_UTILITIES (NAME);

   -- Provides for lists of names.

   -- NAME_SET_FILTER

   type NAME_SET_FILTER is
     record
       INCLUSION   : NAME_SET;
       EXCLUSION   : NAME_SET;
     end record;

     -- Used to define name set pairs used as filters.

   -- NAME_SET_MEMBERSHIP

   type NAME_SET_MEMBERSHIP is (NON_MEMBER, MEMBER);

   -- Provides for indicating whether a name is a member of a
   -- name set.

   -- Subprograms for Manipulating Name Sets

   procedure BUILD_NAME_SET
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
    (MEMBERS : in   NAMES.LIST_OF;
     SET         : out NAME_SET);

  -- Constructs a name set from a list of names.

  procedure BUILD_NAME_SET_UNION
    (LEFT_SET   : in  NAME_SET;
     RIGHT_SET  : in  NAME_SET;
     UNION      : out NAME_SET);

    -- Constructs the logical union of two name sets.

  procedure BUILD_NAME_SET_INTERSECTION
    (LEFT_SET        : in  NAME_SET;
     RIGHT_SET       : in  NAME_SET;
     INTERSECTION    : out NAME_SET);

    -- Constructs the logical intersection of two name sets.

  procedure BUILD_NAME_SET_DIFFERENCE
    (LEFT_SET       : in  NAME_SET;
     RIGHT_SET      : in  NAME_SET;
     DIFFERENCE     : out NAME_SET);

    -- Constructs the logical difference of two name sets.

  procedure COPY_NAME_SET
    (ORIGINAL  : in  NAME_SET;
     COPY      : out NAME_SET);

    -- Duplicates the provided name set.

  function IS_EQUAL
    (LEFT_SET   : in NAME_SET;
     RIGHT_SET  : in NAME_SET)
    return BOOLEAN;

    -- Compares two name sets.

  procedure EXTRACT_NAMES
    (SET         : in  NAME_SET;
     MEMBERS     : out NAMES.LIST_OF);

  -- Extracts the names from the provided name set.

  procedure ADD_NAMES
    (MEMBERS : in     NAMES.LIST_OF;
     SET         : in out NAME_SET);

  -- Adds names to the name set if not already there.

  procedure REMOVE_NAMES
    (MEMBERS : in     NAMES.LIST_OF;
     SET         : in out NAME_SET);

  -- Removes names from the name set if they are there.

  function APPLY_FILTER
    (SET       : in NAME_SET;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
      FILTER  : in NAME_SET_FILTER)
    return NAME_SET_ACCEPTANCE;

  -- Applies a filter to a name set and indicates whether the name
  -- set passes through the filter.

  function INQUIRE_MEMBERSHIP
    (MEMBER : in NAME;
     SET    : in NAME_SET)
    return NAME_SET_MEMBERSHIP;

  -- Returns an indication of whether the indicated name is a member
  -- of the provided name set.

  procedure DEALLOCATE
    (OBJECT  : in out NAME_SET);

    -- Deallocates name set objects.

  -- PRIVATE TYPE DEFINITIONS

private

  -- The following types define the specifications for private
  -- types used in the PHIGS_NAME_SET_FACILITY.  Null records are
  -- used soley for the purposes of demonstrating compilability.
  -- An implementation would use some more appropriate declaration.

  type NAME_SET is
    record
      null;
    end record;

end PHIGS_NAME_SET_FACILITY;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

-- **The PHIGS Types Package**

```
with PHIGS_CONFIGURATION, PHIGS_LIST_UTILITIES, PHIGS_NAME_SET_FACILITY;

package PHIGS_TYPES is

-- This package contains all the data type definitions used to define
-- the Ada binding to PHIGS.

-- Configuration names

use PHIGS_CONFIGURATION;


-- Basic type definitions

-- BASE_TYPES

    package BASE_TYPES is

    -- This package is used to encapsulate the base derived types of PHIGS
    -- since they are used as the parent of several other derived types.
    -- In Ada, if the parent of a derived type is itself a derived type,
    -- then this parent type cannot be declared immediately in the visible
    -- part of the same package.

    -- PHIGS_INTEGER

    type PHIGS_INTEGER is range MIN_INTEGER_VALUE .. MAX_INTEGER_VALUE;

    --  Base type for PHIGS integer types.

    -- PHIGS_NATURAL

    subtype PHIGS_NATURAL is
                PHIGS_INTEGER range 0..PHIGS_INTEGER'last;

    --  Base type for PHIGS natural types.

    -- PHIGS POSITIVE

    subtype PHIGS_POSITIVE is
                PHIGS_INTEGER range 1..PHIGS_INTEGER'last;

    --  Base type for PHIGS positive types.

    -- SCALE_FACTOR

    type SCALE_FACTOR is digits PHIGS_PRECISION;

    -- The type used for unitless scaling factors.

    -- PHIGS_STRING

    type PHIGS_STRING is
            array (PHIGS_INTEGER range < >) of CHARACTER;

    -- Base type for PHIGS string types.
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
end BASE_TYPES;

subtype PHIGS_INTEGER is BASE_TYPES.PHIGS_INTEGER;
subtype PHIGS_NATURAL is BASE_TYPES.PHIGS_NATURAL;
subtype PHIGS_POSITIVE is BASE_TYPES.PHIGS_POSITIVE;
subtype PHIGS_STRING is BASE_TYPES.PHIGS_STRING;
subtype SCALE_FACTOR is BASE_TYPES.SCALE_FACTOR;

-- Make BASE_TYPES visible externally.

-- SMALL_NATURAL

subtype SMALL_NATURAL is
        PHIGS_NATURAL range 0..MAX_SMALL_NATURAL;

-- This is a subtype declaration which allows for unconstrained record
-- objects for various record types without causing the exception
-- STORAGE_ERROR to be raised.

-- CHOICE_SMALL_NATURAL

subtype CHOICE_SMALL_NATURAL is
        PHIGS_NATURAL range 0..MAX_CHOICE_SMALL_NATURAL;

-- This is a subtype declaration which allows for
-- unconstrained record objects for CHOICE_PROMPT_STRING_LIST
-- type without causing the exception STORAGE_ERROR to be raised.

-- COLOUR_MATRIX_SMALL_NATURAL

subtype COLOUR_MATRIX_SMALL_NATURAL is
        PHIGS_NATURAL range 0..MAX_COLOUR_MATRIX_SMALL_NATURAL;

-- This is a subtype declaration which allows for unconstrained record
-- objects of VARIABLE_COLOUR_MATRIX type without causing the
-- exception STORAGE_ERROR to be raised.

-- FILE_SMALL_NATURAL

subtype FILE_SMALL_NATURAL is
        PHIGS_NATURAL range 0..MAX_FILE_SMALL_NATURAL;

-- This is a subtype declaration which allows for
-- unconstrained record objects for VARIABLE_FILE_ID type
-- without causing the exception STORAGE_ERROR to be raised.

-- INPUT_STRING_SMALL_NATURAL

subtype INPUT_STRING_SMALL_NATURAL is
        PHIGS_NATURAL range 0..MAX_INPUT_STRING_SMALL_NATURAL;

-- This is a subtype declaration which allows for
-- unconstrained record objects for INPUT_STRING type
-- without causing the exception STORAGE_ERROR to be raised.

-- STRING_SMALL_NATURAL

subtype STRING_SMALL_NATURAL is
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
            PHIGS_NATURAL range 0..MAX_STRING_SMALL_NATURAL;

    -- This is a subtype declaration which allows for
    -- unconstrained record objects for various string record
    -- types without causing the exception STORAGE_ERROR to be
    -- raised.

    generic

     type COORDINATE_COMPONENT_TYPE is digits <>;

     package PHIGS_COORDINATE_SYSTEM is

        type POINT_3 is
          record
            X : COORDINATE_COMPONENT_TYPE;
            Y : COORDINATE_COMPONENT_TYPE;
            Z : COORDINATE_COMPONENT_TYPE;
          end record;

        type POINT_2 is
          record
            X : COORDINATE_COMPONENT_TYPE;
            Y : COORDINATE_COMPONENT_TYPE;
          end record;

        type POINT_LIST_3 is array (POSITIVE range <>) of POINT_3;

        type POINT_LIST_2 is array (POSITIVE range <>) of POINT_2;

        type ACCESS_POINT_LIST_3 is access POINT_LIST_3;

        type ACCESS_POINT_LIST_2 is access POINT_LIST_2;

        type LIST_OF_POINT_LIST_3 is array (POSITIVE range <>) of
                                          ACCESS_POINT_LIST_3;

        type LIST_OF_POINT_LIST_2 is array (POSITIVE range <>) of
                                          ACCESS_POINT_LIST_2;

        type ACCESS_LIST_OF_POINT_LIST_3 is access LIST_OF_POINT_LIST_3;

        type ACCESS_LIST_OF_POINT_LIST_2 is access LIST_OF_POINT_LIST_2;

        type VECTOR_3 is new POINT_3;

        type VECTOR_PAIR_3 is array (1..2) of VECTOR_3;

        type VECTOR_2 is new POINT_2;

        type VECTOR_PAIR_2 is array (1..2) of VECTOR_2;

        type RECTANGULAR_REGION_3 is
          record
            XMIN  : COORDINATE_COMPONENT_TYPE;
            XMAX  : COORDINATE_COMPONENT_TYPE;
            YMIN  : COORDINATE_COMPONENT_TYPE;
            YMAX  : COORDINATE_COMPONENT_TYPE;
            ZMIN  : COORDINATE_COMPONENT_TYPE;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
      ZMAX  : COORDINATE_COMPONENT_TYPE;
    end record;

  type RECTANGULAR_REGION_2 is
    record
      XMIN  : COORDINATE_COMPONENT_TYPE;
      XMAX  : COORDINATE_COMPONENT_TYPE;
      YMIN  : COORDINATE_COMPONENT_TYPE;
      YMAX  : COORDINATE_COMPONENT_TYPE;
    end record;

  type HALF_SPACE_3 is
    record
      REFERENCE_POSITION   : POINT_3;
      ACCEPTANCE_NORMAL : VECTOR_3;
  end record;

  type HALF_SPACE_2 is
    record
      REFERENCE_POSITION   : POINT_2;
      ACCEPTANCE_NORMAL : VECTOR_2;
  end record;

  type HALF_SPACE_LIST_3 is
    array (POSITIVE range <>) of HALF_SPACE_3;

  type HALF_SPACE_LIST_2 is
    array (POSITIVE range <>) of HALF_SPACE_2;

  type ACCESS_HALF_SPACE_LIST_3 is access HALF_SPACE_LIST_3;

  type ACCESS_HALF_SPACE_LIST_2 is access HALF_SPACE_LIST_2;

  type MAGNITUDE_BASE_TYPE is digits PHIGS_PRECISION;

  subtype MAGNITUDE is MAGNITUDE_BASE_TYPE range
    COORDINATE_COMPONENT_TYPE'SAFE_SMALL ..
    COORDINATE_COMPONENT_TYPE'SAFE_LARGE;

  type SIZE_3 is
    record
      XAXIS  : MAGNITUDE;
      YAXIS  : MAGNITUDE;
      ZAXIS  : MAGNITUDE;
    end record;

  type SIZE_2 is
    record
      XAXIS  : MAGNITUDE;
      YAXIS  : MAGNITUDE;
    end record;

  type RANGE_OF_MAGNITUDES is
    record
      MIN   : MAGNITUDE;
      MAX   : MAGNITUDE;
    end record;

end PHIGS_COORDINATE_SYSTEM;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
-- MC_TYPE

type MC_TYPE is digits PHIGS_PRECISION;

-- Defines the type of coordinate in the Modeling Coordinate
-- System.

-- MC

package MC is new PHIGS_COORDINATE_SYSTEM (MC_TYPE);

-- Defines the Modeling Coordinate System.

-- WC_TYPE

type WC_TYPE is digits PHIGS_PRECISION;

-- Defines the type of coordinate in the World Coordinate
-- System.

-- WC

package WC is new PHIGS_COORDINATE_SYSTEM (WC_TYPE);

-- Defines the World Coordinate System.

-- VRC_TYPE

type VRC_TYPE is digits PHIGS_PRECISION;

-- Defines the type of coordinate in the View Reference
-- Coordinate System.

-- VRC

package VRC is new PHIGS_COORDINATE_SYSTEM (VRC_TYPE);

-- Defines the viewing coordinate system.

-- NPC_TYPE

type NPC_TYPE is digits PHIGS_PRECISION;

-- Defines the type of a coordinate in the Normalized
-- Projection Coordinate System.

-- NPC

package NPC is new PHIGS_COORDINATE_SYSTEM (NPC_TYPE);

-- Defines the Normalized Projection Coordinate System.

-- DC_TYPE

type DC_TYPE is digits PHIGS_PRECISION;

-- The type of a coordinate component in the Device Coordinate
-- System.
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
-- DC

package DC is new PHIGS_COORDINATE_SYSTEM (DC_TYPE);

-- Defines the Device Coordinate System.

-- GENERAL_FLOAT_PARAMETER

type GENERAL_FLOAT_PARAMETER is digits PHIGS_PRECISION;

-- Defines a type for specifying implementation-specific real
-- data in GDP, GSE, and ESCAPE data records.

-- OFF_ON

type OFF_ON is (OFF,
                ON);

-- Generic type for off/on switches.

-- ANGLE

type ANGLE is digits PHIGS_PRECISION;

-- Values used in the modeling transformation utility functions
-- for specifying angles of rotation in radians. Positive
-- indicates a counterclockwise direction.

-- ANNOTATION_STYLE

type ANNOTATION_STYLE is new PHIGS_INTEGER;

-- Defines the annotation styles for annotation primitives.

-- ANNOTATION_STYLES
package ANNOTATION_STYLES is
        new PHIGS_LIST_UTILITIES (ANNOTATION_STYLE,
                                  MAX_ANNOTATION_STYLES_SUPPORTED);

-- Provides for lists of annotation styles.

-- APPLICATION_DATA_RANGE

subtype APPLICATION_DATA_RANGE is
        PHIGS_NATURAL range 0..MAX_APPLICATION_DATA;

-- Defines range of lengths for application data objects.

-- APPLICATION DATA RECORD

type APPLICATION_DATA_RECORD (LENGTH : APPLICATION_DATA_RANGE := 0) is
  record
    DATA : PHIGS_STRING (1..LENGTH);
  end record;

-- Defines type for defining application data objects.

-- ARCHIVE_ID
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
type ARCHIVE_ID is new PHIGS_NATURAL;

-- Provides for an application specified pointer to archive
-- files separate from the file identifier.

-- ARCHIVE_IDS

package ARCHIVE_IDS is new PHIGS_LIST_UTILITIES (ARCHIVE_ID,
                                                 MAX_ARCHIVE_IDS_SUPPORTED);

-- Provides for lists of archive identifiers.

-- ARCHIVE_STATE

type ARCHIVE_STATE is (ARCL,
                       AROP);

-- The type used to return the archive state.

-- ASF

type ASF is (BUNDLED,
             INDIVIDUAL);

-- This type defines an aspect source flag whose
-- value indicates whether an aspect of a primitive
-- should be set from a bundle table or from an
-- individual attribute.

-- ASPECT

type ASPECT is (TYPE_OF_LINE,
                LINEWIDTH_SF,
                LINE_COLOUR,
                TYPE_OF_MARKER,
                SIZE,
                MARKER_COLOUR,
                FONT,
                PRECISION,
                EXPANSION,
                SPACING,
                TEXT_COLOUR,
                STYLE_OF_INTERIOR,
                STYLE_IND,
                INTERIOR_COLOUR,
                FLAG,
                TYPE_OF_EDGE,
                EDGEWIDTH_SF,
                EDGE_COLOUR);

-- This type lists the aspects for which an aspect source
-- flag exists.

-- ATTRIBUTES_USED_TYPE

type ATTRIBUTES_USED_TYPE is (POLYLINE_ATTRIBUTES,
                              POLYMARKER_ATTRIBUTES,
                              TEXT_ATTRIBUTES,
                              INTERIOR_ATTRIBUTES,
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
                                        EDGE_ATTRIBUTES);

-- The types of attributes which may be used in generating
-- output for a GDP and in generating prompt and echo
-- information for certain prompt and echo types of certain
-- classes of input devices.

-- ATTRIBUTES_USED

package ATTRIBUTES_USED is
   new PHIGS_LIST_UTILITIES
         (ATTRIBUTES_USED_TYPE,
          ATTRIBUTES_USED_TYPE'POS(ATTRIBUTES_USED_TYPE'LAST)+1);

-- Provides for a list of the attributes used which is returned
-- by the INQ_GDP and LOCATOR_ATTRIBUTES_USED functions.

-- CHAR_EXPANSION

type CHAR_EXPANSION is new SCALE_FACTOR range
  SCALE_FACTOR'SAFE_SMALL..SCALE_FACTOR'LAST;

-- Defines a character expansion factor.

-- CHAR_SET

type CHAR_SET is new PHIGS_NATURAL;

-- Provides identifications for the available character
-- sets. ISO 646 character set maps to value zero.

-- CHAR_SETS

package CHAR_SETS is new PHIGS_LIST_UTILITIES (CHAR_SET,
                                               MAX_CHAR_SETS_SUPPORTED);

-- Provides for lists of character set identifications.

-- CHAR_SPACING

type CHAR_SPACING is new SCALE_FACTOR;

-- Defines a character spacing factor. The factors are
-- unitless. A positive value indicates the amount of extra
-- space between character boxes in a text string, and a
-- negative value indicates the amount of overlap between
-- character boxes in a text string.

-- CHOICE_PROMPT_ECHO_TYPE

type CHOICE_PROMPT_ECHO_TYPE is new PHIGS_INTEGER;

-- Defines the choice prompt and echo type.

-- CHOICE_PROMPT_ECHO_TYPES

package CHOICE_PROMPT_ECHO_TYPES is
   new PHIGS_LIST_UTILITIES (CHOICE_PROMPT_ECHO_TYPE,
                             MAX_CHOICE_PROMPT_ECHO_TYPES_SUPPORTED);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
-- Provides for lists of choice prompt and echo types.

-- DEVICE_NUMBER

subtype DEVICE_NUMBER is PHIGS_POSITIVE;

-- Defines the base type for input device numbers.

-- CHOICE_DEVICE_NUMBER

type CHOICE_DEVICE_NUMBER is new DEVICE_NUMBER;

-- Provides for choice device identifiers.

-- CHOICE_NUMBER

type CHOICE_NUMBER is new PHIGS_POSITIVE;

-- Defines the choice numbers available on an implementation.

-- CHOICE_PROMPT

type CHOICE_PROMPT is new OFF_ON;

-- Indicates whether a specified choice prompt is to be displayed
-- or not.

-- CHOICE_PROMPTS

package CHOICE_PROMPTS is
        new PHIGS_LIST_UTILITIES (CHOICE_PROMPT,
                                  MAX_CHOICE_PROMPTS_SUPPORTED);

-- Provides for lists of choice prompts.

-- CHOICE_PROMPT_STRING

type CHOICE_PROMPT_STRING (LENGTH : STRING_SMALL_NATURAL := 0) is
  record
    CONTENTS : PHIGS_STRING (1..LENGTH);
  end record;

-- Provides for a variable length prompt.  Objects of this type
-- should be declared unconstrained to allow for dynamic
-- modification of the length.

-- CHOICE_PROMPT_STRING_ARRAY

type CHOICE_PROMPT_STRING_ARRAY is array (PHIGS_POSITIVE range <>)
                                   of CHOICE_PROMPT_STRING;

-- Provides for an array of prompt strings.

-- CHOICE_PROMPT_STRING_LIST

type CHOICE_PROMPT_STRING_LIST
  (LENGTH : CHOICE_SMALL_NATURAL := 0) is
     record
       LIST : CHOICE_PROMPT_STRING_ARRAY (1..LENGTH);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
        end record;

    -- Provides for lists of prompt strings.

    -- CHOICE_REQUEST_STATUS

    type CHOICE_REQUEST_STATUS is (OK,
                                   NOCHOICE,
                                   NONE);

    -- Defines the status of a choice input operation for the
    -- request function.

    -- CHOICE_STATUS

    subtype CHOICE_STATUS is CHOICE_REQUEST_STATUS range OK..NOCHOICE;

    -- Indicates if a choice was made by the operator for the
    -- sample, get, and inquiry functions.

    -- CLIPPING_INDICATOR

    type CLIPPING_INDICATOR is (CLIP,
                                NOCLIP);

    -- Indicates whether or not clipping is to be performed.

    -- COLOUR_AVAILABLE

    type COLOUR_AVAILABLE is (COLOUR,
                              MONOCHROME);

    -- Indicates whether colour output is available on a
    -- workstation.

    -- COLOUR_COMPONENTS

    type COLOUR_COMPONENTS is range 1..MAX_COLOUR_COEFFICIENTS;

    -- Defines a type for the size of colour component arrays.

    -- COLOUR_INDEX

    type COLOUR_INDEX is new PHIGS_NATURAL;

    -- Indices into colour tables are of this type.

    -- COLOUR_INDICES

    package COLOUR_INDICES is
            new PHIGS_LIST_UTILITIES (COLOUR_INDEX,
                                      MAX_COLOUR_INDICES_SUPPORTED);

    -- Provides for a set of colour indices which are available
    -- on a particular workstation.

    -- COLOUR_MATRIX

    type COLOUR_MATRIX is array (COLOUR_MATRIX_SMALL_NATURAL range <>,
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
                                    COLOUR_MATRIX_SMALL_NATURAL range <>)
                                of COLOUR_INDEX;

-- Provides for matrices containing colour indices corresponding
-- to a cell array or pattern array.

-- COLOUR_MODEL

type COLOUR_MODEL is new PHIGS_INTEGER;

-- Indicates the colour models available in PHIGS.

-- COLOUR_MODELS

package COLOUR_MODELS is
        new PHIGS_LIST_UTILITIES (COLOUR_MODEL,
                                  MAX_COLOUR_MODELS_SUPPORTED);

-- Provides for lists of colour models.

-- The following constants define the colour models specified by PHIGS:
RGB     : constant COLOUR_MODEL := 1;
CIELUV  : constant COLOUR_MODEL := 2;
HSV     : constant COLOUR_MODEL := 3;
HLS     : constant COLOUR_MODEL := 4;

-- COLOUR_COEFFICIENT

type COLOUR_COEFFICIENT is digits PHIGS_PRECISION;

-- Defines a general type for colour components.

-- COLOUR_COEFFICIENT_ARRAY

type COLOUR_COEFFICIENT_ARRAY is
  array (COLOUR_COMPONENTS) of COLOUR_COEFFICIENT;

-- Defines an array type for colour components.

-- COLOUR_COEFFICIENTS

package COLOUR_COEFFICIENTS is
  new PHIGS_LIST_UTILITIES (COLOUR_COEFFICIENT,
                            MAX_COLOUR_COEFFICIENTS);

-- Provides for a lists of colour coefficients for the generalized
-- colour model.

-- INTENSITY

subtype INTENSITY is COLOUR_COEFFICIENT range 0.0..1.0;

-- Defines the restricted range of colour components required
-- by some colour models.

-- COLOUR_REPRESENTATION

type COLOUR_REPRESENTATION (MODEL : COLOUR_MODEL := RGB) is
  record
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
    case MODEL is

      when RGB = >
        RED_RGB                 : INTENSITY;
        GREEN_RGB               : INTENSITY;
        BLUE_RGB                : INTENSITY;

      when CIELUV = >
        L_STAR_CIE              : COLOUR_COEFFICIENT;
        U_STAR_CIE              : COLOUR_COEFFICIENT;
        V_STAR_CIE              : COLOUR_COEFFICIENT;

      when HSV = >
        HUE_HSV                 : INTENSITY;
        SATURATION_HSV          : INTENSITY;
        VALUE_HSV               : INTENSITY;

      when HLS = >
        HUE_HLS                 : INTENSITY;
        LIGHTNESS_HLS           : INTENSITY;
        SATURATION_HLS          : INTENSITY;

      when others = >
        GENERIC_COMPONENTS : COLOUR_COEFFICIENT_ARRAY;
    end case;
  end record;

-- Defines the representation of a colour as a combination of three
-- components whose meaning depends on the colour model. Additional
-- variants may be included when additional registered or unregistered
-- colour models are supported by an implementation.

-- CHROMATICITY_COEFFICIENT

type CHROMATICITY_COEFFICIENT is
  record
    X : COLOUR_COEFFICIENT;
    Y : COLOUR_COEFFICIENT;
  end record;

-- Defines a CIE colour coefficient pair.

-- CHROMATICITY_COEFFICIENT_SET

type CHROMATICITY_COEFFICIENT_SET is
  record
    R : CHROMATICITY_COEFFICIENT;
    G : CHROMATICITY_COEFFICIENT;
    B : CHROMATICITY_COEFFICIENT;
  end record;

-- Defines a set of CIE primary colour chromaticity coefficients.

-- COMPOSITION_TYPE

type COMPOSITION_TYPE is (PRECONCATENATE,
                          POSTCONCATENATE,
                          REPLACE);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
-- Defines the type of matrix composition allowed between
-- modeling transformations.

-- CONFLICT_RESOLUTION

type CONFLICT_RESOLUTION is (MAINTAIN,
                             ABANDON,
                             UPDATE);

-- Defines a type for specifying the conflict resolution mode.

-- CONNECTION_ID

type CONNECTION_ID is new PHIGS_STRING;

-- Provides for specifying connection identifiers.

-- CONTROL_FLAG

type CONTROL_FLAG is (CONDITIONALLY,
                      ALWAYS);

-- The control flag is used to indicate the conditions under
-- which the display surface should be cleared.

-- DC_UNITS

type DC_UNITS is (METRES,
                  OTHER);

-- Device coordinate units for a particular workstation
-- may be in metres, or some other unit (such as inches).

-- DEFERRAL_MODE

type DEFERRAL_MODE is (ASAP,
                       BNIG,
                       BNIL,
                       ASTI,
                       WAIT);

-- Defines the five PHIGS deferral modes.

-- DISPLAY_CLASS

type DISPLAY_CLASS is (VECTOR_DISPLAY,
                       RASTER_DISPLAY,
                       OTHER_DISPLAY);

-- The classification of a workstation of category OUTPUT
-- or OUTIN.

-- DISPLAY_SURFACE_EMPTY

type DISPLAY_SURFACE_EMPTY is (EMPTY,
                               NOTEMPTY);

-- Indicates whether there is graphical output on the display
-- surface.
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
-- DYNAMIC_MODIFICATION

type DYNAMIC_MODIFICATION is (IRG,
                              IMM,
                              CBS);

-- Indicates whether an update to the state list is performed
-- immediately (IMM), requires implicit regeneration (IRG), or
-- can be simulated (CBS).

-- ECHO_SWITCH

type ECHO_SWITCH is (NOECHO,
                     ECHO);

-- Indicates whether or not echoing of the measure is
-- performed.

-- EDIT_MODE

type EDIT_MODE is (INSERT,
                   REPLACE);

-- Defines a type for specifying the mode in which editing
-- takes place.

-- ELEMENT_POSITION

type ELEMENT_POSITION is new PHIGS_INTEGER;

-- Base type for element pointer values and offsets.

-- RETURNED_ELEMENT_POSITION

subtype RETURNED_ELEMENT_POSITION is
  ELEMENT_POSITION range 0..ELEMENT_POSITION'LAST;

-- Defines a type for returning element pointer values.

-- ELEMENT_TYPE

type ELEMENT_TYPE is (ALL_ELEMENT_TYPES,
                      NIL,

                      POLYLINE_3,
                      POLYLINE,
                      POLYMARKER_3,
                      POLYMARKER,
                      TEXT_3,
                      TEXT,
                      ANNOTATION_TEXT_RELATIVE_3,
                      ANNOTATION_TEXT_RELATIVE,
                      FILL_AREA_3,
                      FILL_AREA,
                      FILL_AREA_SET_3,
                      FILL_AREA_SET,
                      CELL_ARRAY_3,
                      CELL_ARRAY,
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
    GDP_3,
    GDP,

    SET_POLYLINE_INDEX,
    SET_POLYMARKER_INDEX,
    SET_TEXT_INDEX,
    SET_INTERIOR_INDEX,
    SET_EDGE_INDEX,

    SET_LINETYPE,
    SET_LINEWIDTH_SCALE_FACTOR,
    SET_POLYLINE_COLOUR_INDEX,

    SET_MARKER_TYPE,
    SET_MARKER_SIZE_SCALE_FACTOR,
    SET_POLYMARKER_COLOUR_INDEX,

    SET_TEXT_FONT,
    SET_TEXT_PRECISION,
    SET_CHAR_EXPANSION_FACTOR,
    SET_CHAR_SPACING,
    SET_TEXT_COLOUR_INDEX,
    SET_CHAR_HEIGHT,
    SET_CHAR_UP_VECTOR,
    SET_TEXT_PATH,
    SET_TEXT_ALIGNMENT,

    SET_ANNOTATION_TEXT_CHAR_HEIGHT,
    SET_ANNOTATION_TEXT_CHAR_UP_VECTOR,
    SET_ANNOTATION_TEXT_PATH,
    SET_ANNOTATION_TEXT_ALIGNMENT,
    SET_ANNOTATION_STYLE,

    SET_INTERIOR_STYLE,
    SET_INTERIOR_STYLE_INDEX,
    SET_INTERIOR_COLOUR_INDEX,

    SET_EDGE_FLAG,
    SET_EDGETYPE,
    SET_EDGEWIDTH_SCALE_FACTOR,
    SET_EDGE_COLOUR_INDEX,

    SET_PATTERN_SIZE,
    SET_PATTERN_REFERENCE_POINT_AND_VECTORS,
    SET_PATTERN_REFERENCE_POINT,

    ADD_NAMES_TO_SET,
    REMOVE_NAMES_FROM_SET,

    SET_INDIVIDUAL_ASF,
    SET_HLHSR_IDENTIFIER,

    SET_LOCAL_TRANSFORMATION_3,
    SET_LOCAL_TRANSFORMATION,
    SET_GLOBAL_TRANSFORMATION_3,
    SET_GLOBAL_TRANSFORMATION,
    SET_MODELLING_CLIPPING_VOLUME_3,
    SET_MODELLING_CLIPPING_VOLUME,
    SET_MODELLING_CLIPPING_INDICATOR,
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS

Not for Resale

```
                    RESTORE_MODELLING_CLIPPING_VOLUME,
                    SET_VIEW_INDEX,

                    EXECUTE_STRUCTURE,

                    LABEL,
                    APPLICATION_DATA,
                    GSE,
                    SET_PICK_IDENTIFIER);

-- Provides for specification of element types.

-- ELEMENT_TYPES

package ELEMENT_TYPES is
        new PHIGS_LIST_UTILITIES (ELEMENT_TYPE,
                                  ELEMENT_TYPE'POS(ELEMENT_TYPE'LAST)+1);

-- Provides for lists of element types.

-- ERROR_HANDLING_MODE

type ERROR_HANDLING_MODE is new OFF_ON;

-- Type for inquiring the error handling mode.

-- ERROR_NUMBER

type ERROR_NUMBER is new PHIGS_INTEGER;

-- Defines the type for error indicator values.

-- ESCAPE_ID

type ESCAPE_ID is new PHIGS_INTEGER;

-- Defines a type for identifying different escapes.

-- ESCAPE_IDS

package ESCAPE_IDS is
        new PHIGS_LIST_UTILITIES (ESCAPE_ID,
                                  MAX_ESCAPE_IDS_SUPPORTED);

-- Provides for lists of Escape ID's.

-- ESCAPE_DATA_RECORD

type ESCAPE_DATA_RECORD (ESCAPE : ESCAPE_ID := 0) is
  record
    case ESCAPE is

      -- For each ESCAPE which needs a record, an implementation
      -- dependent entry is defined.  For others, a null record
      -- is defined.

      when others = >
         null;
    end case;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
  end record;

-- Provides for the definition of ESCAPE data records.

-- FILE_ID

subtype FILE_ID is PHIGS_STRING;

-- Provides for file identifiers.

-- VARIABLE_FILE_ID

type VARIABLE_FILE_ID (LENGTH : FILE_SMALL_NATURAL := 0) is
  record
    NAME : FILE_ID (1..LENGTH);
  end record;

-- Provides for variable length file identifiers.

-- VARIABLE_FILE_IDS

package VARIABLE_FILE_IDS is
   new PHIGS_LIST_UTILITIES (VARIABLE_FILE_ID,
                             MAX_FILE_IDS_SUPPORTED);

-- This type is used to define lists of file identifiers.

-- GDP_3_ID

type GDP_3_ID is new PHIGS_INTEGER;

-- Defines a type for selecting a Generalized Drawing Primitive.

-- GDP_3_IDS

package GDP_3_IDS is
          new PHIGS_LIST_UTILITIES (GDP_3_ID,
                                    MAX_GDP_3_IDS_SUPPORTED);

-- Provides for lists of 3D Generalized Drawing Primitive ID's.

-- GDP_3_RECORD

type GDP_3_RECORD (GDP : GDP_3_ID := 0) is
  record
    case GDP is

      -- For each supported GDP_3 which needs a record, an implementation
      -- dependent entry is defined.  For others, a generic record
      -- is defined.

      when others = >
         GDP_INFO : APPLICATION_DATA_RECORD;
    end case;
  end record;

-- Provides for the definition of GDP_3 data records.

-- GDP_ID
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
type GDP_ID is new PHIGS_INTEGER;

-- Defines a type for selecting a Generalized Drawing Primitive.

-- GDP_IDS

package GDP_IDS is
        new PHIGS_LIST_UTILITIES (GDP_ID,
                                  MAX_GDP_IDS_SUPPORTED);

-- Provides for lists of Generalized Drawing Primitive ID's.

-- GDP_RECORD

type GDP_RECORD (GDP : GDP_ID := 0) is
  record
    case GDP is

      -- For each supported GDP which needs a record, an implementation
      -- dependent entry is defined. For others, a generic record
      -- is defined.

      when others = >
        GDP_INFO : APPLICATION_DATA_RECORD;
    end case;
  end record;

-- Provides for the definition of GDP data records.

-- GSE_ID

type GSE_ID is new PHIGS_INTEGER;

-- Defines a type for selecting a Generalized Structure Element.

-- GSE_IDS

package GSE_IDS is new PHIGS_LIST_UTILITIES (GSE_ID,
                                             MAX_GSE_IDS_SUPPORTED);

-- Provides for lists of Generalized Structure Element ID's.

-- GSE_RECORD

type GSE_RECORD (GSE : GSE_ID := 0) is
  record
    case GSE is

      -- For each supported GSE which needs a record, an implementation
      -- dependent entry is defined. For others, a generic record
      -- is defined.

      when others = >
        GSE_INFO : APPLICATION_DATA_RECORD;
    end case;
  end record;

-- Provides for the definition of GSE data records.
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
-- STYLE_INDEX

type STYLE_INDEX is new PHIGS_INTEGER;

-- Defines an interior style index.

-- HATCH_STYLE

subtype HATCH_STYLE is STYLE_INDEX;

-- Defines the interior hatch styles type.

-- HATCH_STYLES

package HATCH_STYLES is
   new PHIGS_LIST_UTILITIES (HATCH_STYLE,
                             MAX_HATCH_STYLES_SUPPORTED);


-- Provides for lists of hatch styles.

-- HLHSR_ID

type HLHSR_ID is new PHIGS_INTEGER;

-- A number used as an attribute for workstation independent
-- HLHSR factors.

-- HLHSR_IDS

package HLHSR_IDS is new PHIGS_LIST_UTILITIES (HLHSR_ID,
   MAX_HLHSR_IDS_SUPPORTED);

-- Provides for lists of HLHSR identifiers.

-- HLHSR_MODE

type HLHSR_MODE is new PHIGS_INTEGER;

-- A number used for controlling workstation dependent
-- HLHSR factors.

-- HLHSR_MODES

package HLHSR_MODES is new PHIGS_LIST_UTILITIES (HLHSR_MODE,
                                                 MAX_HLHSR_MODES_SUPPORTED);

-- Provides for lists of HLHSR modes.

-- HORIZONTAL_ALIGNMENT

type HORIZONTAL_ALIGNMENT is (NORMAL,
                              LEFT,
                              CENTRE,
                              RIGHT);

-- The alignment of the text extent rectangle with respect to
-- horizontal positioning of the text.
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
-- INPUT_CLASS

type INPUT_CLASS is (NONE,
                     LOCATOR_INPUT,
                     STROKE_INPUT,
                     VALUATOR_INPUT,
                     CHOICE_INPUT,
                     PICK_INPUT,
                     STRING_INPUT);

-- Defines the input device classifications for workstations
-- of category INPUT or OUTIN.

-- INPUT_QUEUE_CLASS

subtype INPUT_QUEUE_CLASS is
  INPUT_CLASS range LOCATOR_INPUT .. STRING_INPUT;

-- Defines the input device classifications for workstations
-- of category INPUT and OUTIN that do not return a NONE
-- classification.

-- INPUT_STATUS

type INPUT_STATUS is (OK,
                      NONE);

-- Defines the status of a locator, stroke, valuator, or
-- string operation.

-- INPUT_STRING

type INPUT_STRING (LENGTH : INPUT_STRING_SMALL_NATURAL := 0) is
  record
    CONTENTS : PHIGS_STRING (1..LENGTH);
  end record;

-- Provides a variable length input string. Objects of this type
-- should be declared unconstrained to allow for dynamic modification
-- of the length.

-- INTERIOR_INDEX

type INTERIOR_INDEX is new PHIGS_POSITIVE;

-- Defines interior bundle table indices.

-- INTERIOR_INDICES

package INTERIOR_INDICES is
  new PHIGS_LIST_UTILITIES (INTERIOR_INDEX,
                            MAX_INTERIOR_INDICES_SUPPORTED);

-- Provides for lists of interior bundle table indices.

-- INTERIOR_STYLE

type INTERIOR_STYLE is (HOLLOW,
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
                        SOLID,
                        PATTERN,
                        HATCH,
                        EMPTY);

-- Defines the interior styles for filled areas.

-- INTERIOR_STYLES

package INTERIOR_STYLES is
   new PHIGS_LIST_UTILITIES (INTERIOR_STYLE,
                             INTERIOR_STYLE'POS(INTERIOR_STYLE'LAST) + 1);

-- Provides for lists of interior styles.

-- INTERIOR_DATA

type INTERIOR_DATA is
  record
    STYLE_OF_INTERIOR_ASF : ASF;
    STYLE_IND_ASF         : ASF;
    INTERIOR_COLOUR_ASF   : ASF;
    INTERIOR_IND          : INTERIOR_INDEX;
    STYLE_OF_INTERIOR     : INTERIOR_STYLE;
    STYLE_IND             : STYLE_INDEX;
    INTERIOR_COLOUR       : COLOUR_INDEX;
  end record;

-- A record containing information needed to specify the
-- appearance of a filled area interior.

-- INVALID_VALUES_INDICATOR

type INVALID_VALUES_INDICATOR is (ABSENT,
                                  PRESENT);

-- Indicates whether invalid values are contained in a colour
-- array or matrix.

-- LABEL_ID

type LABEL_ID is new PHIGS_INTEGER;

-- Defines a type for specifying structure element labels.

-- POLYLINE_INDEX

type POLYLINE_INDEX is new PHIGS_POSITIVE;

-- Defines the range of polyline indices.

-- POLYLINE_INDICES

package POLYLINE_INDICES is
   new PHIGS_LIST_UTILITIES (POLYLINE_INDEX,
                             MAX_POLYLINE_INDICES_SUPPORTED);

-- Provides for lists of polyline indices.
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
-- LINETYPE

type LINETYPE is new PHIGS_INTEGER;

-- Defines the types of line styles provided by PHIGS.

-- LINETYPES

package LINETYPES is new PHIGS_LIST_UTILITIES (LINETYPE,
                                               MAX_LINETYPES_SUPPORTED);

-- Provides for lists of linetypes.

-- LINEWIDTH

type LINEWIDTH is new SCALE_FACTOR range 0.0..SCALE_FACTOR'LAST;

-- The width of a line is indicated by a scale factor.

-- LINE_DATA

type LINE_DATA is
  record
    TYPE_OF_LINE_ASF  : ASF;
    WIDTH_ASF         : ASF;
    LINE_COLOUR_ASF   : ASF;
    POLYLINE_IND      : POLYLINE_INDEX;
    TYPE_OF_LINE      : LINETYPE;
    WIDTH             : LINEWIDTH;
    LINE_COLOUR       : COLOUR_INDEX;
  end record;

-- A record containing information needed to specify the
-- appearance of a line for input data records.

-- EDGE_FLAG

type EDGE_FLAG is new OFF_ON;

-- Defines a type for indicating whether or not to depict edges
-- on fill area set and related primitives.

-- EDGE_INDEX

type EDGE_INDEX is new PHIGS_POSITIVE;

-- Defines the edge bundle table indices.

-- EDGE_INDICES

package EDGE_INDICES is new PHIGS_LIST_UTILITIES (EDGE_INDEX,
                                                  MAX_EDGE_INDICES_SUPPORTED);

-- Provides for a list of edge indices.

-- EDGETYPE

type EDGETYPE is new PHIGS_INTEGER;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
-- Defines a type for describing the form of edges.

-- EDGETYPES

package EDGETYPES is new PHIGS_LIST_UTILITIES (EDGETYPE,
                                               MAX_EDGETYPES_SUPPORTED);

-- Provides for lists of edgetypes.

-- EDGEWIDTH

type EDGEWIDTH is new SCALE_FACTOR range 0.0..SCALE_FACTOR'LAST;

-- Defines a type for describing the width scale factor of edges.

-- EDGE_DATA

type EDGE_DATA is
  record
    FLAG_ASF             : ASF;
    TYPE_OF_EDGE_ASF     : ASF;
    EDGEWIDTH_SF_ASF     : ASF;
    EDGE_COLOUR_ASF      : ASF;
    FLAG                 : EDGE_FLAG;
    TYPE_OF_EDGE         : EDGETYPE;
    EDGEWIDTH_SF         : EDGEWIDTH;
    EDGE_COLOUR          : COLOUR_INDEX;
  end record;

-- A record containing information needed to specify the
-- appearance of a fill area set edge.

-- LOCATOR_DEVICE_NUMBER

type LOCATOR_DEVICE_NUMBER is new DEVICE_NUMBER;

-- Provides for locator device identifiers.

-- LOCATOR_PROMPT_ECHO_TYPE

type LOCATOR_PROMPT_ECHO_TYPE is new PHIGS_INTEGER;

-- Defines the locator prompt and echo types supported by
-- the implementation.

-- LOCATOR_PROMPT_ECHO_TYPES

package LOCATOR_PROMPT_ECHO_TYPES is
   new PHIGS_LIST_UTILITIES (LOCATOR_PROMPT_ECHO_TYPE,
                             MAX_LOCATOR_PROMPT_ECHO_TYPES_SUPPORTED);

-- Provides for lists of locator prompt and echo types.

-- POLYMARKER_INDEX

type POLYMARKER_INDEX is new PHIGS_POSITIVE;

-- Defines the range of polymarker bundle table indices.
```


--`,,`-`-`,,`,,`,`,,`---

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS

Not for Resale

```
-- POLYMARKER_INDICES

package POLYMARKER_INDICES is
   new PHIGS_LIST_UTILITIES (POLYMARKER_INDEX,
                             MAX_POLYMARKER_INDICES_SUPPORTED);

-- Provides for lists of polymarker indices.

-- MARKER_SIZE

type MARKER_SIZE is new SCALE_FACTOR;

-- The size of a marker is indicated by a scale factor.

-- MARKER_TYPE

type MARKER_TYPE is new PHIGS_INTEGER;

-- Defines the type for markers provided by PHIGS.

-- MARKER_TYPES

package MARKER_TYPES is
        new PHIGS_LIST_UTILITIES (MARKER_TYPE,
                                  MAX_MARKER_TYPES_SUPPORTED);


-- Provides for lists of marker types.

-- MARKER_DATA

type MARKER_DATA is
  record
    TYPE_OF_MARKER_ASF : ASF;
    SIZE_ASF           : ASF;
    MARKER_COLOUR_ASF  : ASF;
    POLYMARKER_IND     : POLYMARKER_INDEX;
    TYPE_OF_MARKER     : MARKER_TYPE;
    SIZE               : MARKER_SIZE;
    MARKER_COLOUR      : COLOUR_INDEX;
  end record;

-- A record containing information needed to specify the
-- appearance of a marker.

-- MEMORY_UNITS

type MEMORY_UNITS is range 0..MAX_MEMORY_UNITS;

-- Defines the type for units of memory that may be allocated
-- by PHIGS.

-- METAFILE_ITEM_TYPE

type METAFILE_ITEM_TYPE is
      new PHIGS_NATURAL
            range MIN_METAFILE_ITEM_TYPE..MAX_METAFILE_ITEM_TYPE;

-- The type of an item contained in a metafile.
```


--`,,`-`-`,,`,,`,`,,`---

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS

Not for Resale

```
-- METAFILE_ITEM_LENGTH

type METAFILE_ITEM_LENGTH is
        new PHIGS_NATURAL range 0..MAX_METAFILE_ITEM_LENGTH;

-- The length of an item contained in a metafile.

-- MODELLING_CLIP_OPERATION_TYPE

type MODELLING_CLIP_OPERATION_TYPE is new PHIGS_INTEGER;

-- This type provides for specifying modelling clip operation
-- types.

-- MODELLING_CLIP_OPERATION_TYPES

package MODELLING_CLIP_OPERATION_TYPES is
   new PHIGS_LIST_UTILITIES
         (MODELLING_CLIP_OPERATION_TYPE,
          MAX_MODELLING_CLIP_OPERATION_TYPES_SUPPORTED);

-- This type provides for lists of modelling clip operation types.

-- MODIFICATION_MODE

type MODIFICATION_MODE is (NIVE,
                           UWOR,
                           UQUM);

-- Defines type for specifying modification modes.

-- MORE_EVENTS

type MORE_EVENTS is (NOMORE,
                     MORE);

-- Indicates whether more simultaneous events are contained in
-- the input queue.

-- NAME_SET

subtype NAME_SET is PHIGS_NAME_SET_FACILITY.NAME_SET;

-- Used to define name sets. The internal structure of
-- a name set is implementation-dependent. "Build" functions
-- contained in the package PHIGS_NAME_SET_FACILITIES shall be
-- used to create and manipulate name sets (See 5.14.4).

-- NAME_SET_ACCEPTANCE

subtype NAME_SET_ACCEPTANCE is PHIGS_NAME_SET_FACILITY.NAME_SET_ACCEPTANCE;

-- Used to indicate the results of applying a name set
-- against a filter pair. See 5.14.4.

-- NAME

subtype NAME is PHIGS_NAME_SET_FACILITY.NAME;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
-- Provides for names for each of the name set classes.
-- See 5.14.4.

-- NAMES

package NAMES renames
   PHIGS_NAME_SET_FACILITY.NAMES;

-- Provides for lists of name set classes. See 5.14.4.

-- NAME_SET_FILTER

subtype NAME_SET_FILTER is PHIGS_NAME_SET_FACILITY.NAME_SET_FILTER;

-- Used to define name set pairs used as filters.
-- See 5.14.4.

-- NAME_SET_FILTERS

package NAME_SET_FILTERS is
            new PHIGS_LIST_UTILITIES (NAME_SET_FILTER,
                                      MAX_NAME_SET_FILTER_LIST_LENGTH_SUPPORTED);

-- Used to provide lists of name set filters.
-- See 5.14.4.

-- NAME_SET_MEMBERSHIP

subtype NAME_SET_MEMBERSHIP is
   PHIGS_NAME_SET_FACILITY.NAME_SET_MEMBERSHIP;

-- Provides for indicating whether a class is contained in a
-- name set. See 5.14.4.

-- OPEN_STRUCTURE_STATUS

type OPEN_STRUCTURE_STATUS is (NONE,
                               OPEN);

-- Returns the status of the open structure.

-- OPERATING_MODE

type OPERATING_MODE is (REQUEST_MODE,
                        SAMPLE_MODE,
                        EVENT_MODE);

-- Defines the operating modes of an input device.

-- PATH_DEPTH

subtype PATH_DEPTH is
            PHIGS_NATURAL range 0..MAX_PATH_DEPTH_SUPPORTED;


-- Defines a type for specifying the depth of a traversal path.

-- PATH_ORDER
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

9593 PT*3 90 ■ 4851903 0575562 6T4 ■


```
type PATH_ORDER is (TOPFIRST,
                    BOTTOMFIRST);

-- Provides for specifying the order of the pick and
-- reference paths.

-- PATTERN_INDEX

subtype PATTERN_INDEX is STYLE_INDEX range 1..STYLE_INDEX'LAST;

-- Defines the range of pattern table indices.

-- PATTERN_INDICES

package PATTERN_INDICES is
   new PHIGS_LIST_UTILITIES (PATTERN_INDEX,
                             MAX_PATTERN_INDICES_SUPPORTED);

-- Provides for lists of pattern table indices.

-- PICK_DEVICE_NUMBER

type PICK_DEVICE_NUMBER is new DEVICE_NUMBER;

-- Provides for pick device identifiers.

-- PICK_ID

type PICK_ID is new PHIGS_INTEGER;

-- Defines the range of pick identifiers available on an
-- implementation.

-- PICK_IDS

package PICK_IDS is
        new PHIGS_LIST_UTILITIES (PICK_ID,
                                  MAX_PICK_IDS_SUPPORTED);

-- Provides for a list of pick identifiers.

-- PICK_PROMPT_ECHO_TYPE

type PICK_PROMPT_ECHO_TYPE is new PHIGS_INTEGER;

-- Defines the pick prompt and echo type.

-- PICK_PROMPT_ECHO_TYPES

package PICK_PROMPT_ECHO_TYPES is
   new PHIGS_LIST_UTILITIES (PICK_PROMPT_ECHO_TYPE,
        MAX_PICK_PROMPT_ECHO_TYPES_SUPPORTED);

-- Provides for lists of pick prompt and echo types.

-- PICK_REQUEST_STATUS

type PICK_REQUEST_STATUS is (OK,
                             NOPICK,
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
                                        NONE);

-- Defines the status of a pick input operation for the
-- request function.

-- PICK_STATUS

subtype PICK_STATUS is PICK_REQUEST_STATUS range OK..NOPICK;

-- Defines the status of pick input operation for the sample,
-- get, and inquiry functions.

-- STRUCTURE_ID

type STRUCTURE_ID is new PHIGS_NATURAL;

-- Defines a type for structure identifiers.

-- STRUCTURE_IDS

package STRUCTURE_IDS is
   new PHIGS_LIST_UTILITIES (STRUCTURE_ID,
                             MAX_STRUCTURE_IDS_SUPPORTED);

-- Provides for lists of structure identifiers.

-- PICK_PATH_ENTRY

type PICK_PATH_ENTRY is
  record
    STRUCTURE_IDENTIFIER : STRUCTURE_ID;
    PICK_IDENTIFIER      : PICK_ID;
    ELEMENT_NUMBER       : ELEMENT_POSITION;
  end record;

-- Defines the entries of a pick path.

-- PICK_PATH_ARRAY

type PICK_PATH_ARRAY is array (PATH_DEPTH range <>) of
                             PICK_PATH_ENTRY;

-- Provides for the specification of an unconstrained array of pick
-- path entries.

-- PICK_PATH

type PICK_PATH (DEPTH : PATH_DEPTH := 0) is
        record
          PATH : PICK_PATH_ARRAY (0..DEPTH);
        end record;

-- Provides for the specification of a pick path.

-- POLYLINE_FILL_AREA_CONTROL_FLAG

type POLYLINE_FILL_AREA_CONTROL_FLAG is (POLYLINE,
                                         FILL_AREA,
                                         FILL_AREA_SET);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
-- Provides for indicating the type of primitive used
-- to produce locator prompt and echo type 5.

-- PROJECTION_TYPE

type PROJECTION_TYPE is (PARALLEL,
                         PERSPECTIVE);

-- Defines the type of projections supported by PHIGS.

-- RASTER_UNITS

type RASTER_UNITS is new PHIGS_POSITIVE;

-- Defines the range of raster units.

-- RASTER_UNIT_SIZE_3

type RASTER_UNIT_SIZE_3 is
  record
    X : RASTER_UNITS;
    Y : RASTER_UNITS;
    Z : RASTER_UNITS;
  end record;

-- Defines the size of an object in raster units on a raster
-- device.

-- RASTER_UNIT_SIZE_2

type RASTER_UNIT_SIZE_2 is
  record
    X : RASTER_UNITS;
    Y : RASTER_UNITS;
  end record;

-- Defines the size of an object in raster units on a raster
-- device.


-- REFERENCE_HANDLING_FLAG

type REFERENCE_HANDLING_FLAG is (DELETE,
                                 KEEP);

-- Provides for specification of the handling of deleted
-- structure networks.

-- REFERENCE_PATH_ENTRY

type REFERENCE_PATH_ENTRY is
  record
    STRUCTURE_IDENTIFIER : STRUCTURE_ID;
    ELEMENT_NUMBER       : ELEMENT_POSITION;
  end record;

-- Defines the entries of a reference path.

-- REFERENCE_PATH_ARRAY
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
type REFERENCE_PATH_ARRAY is array (PATH_DEPTH range <>) of
  REFERENCE_PATH_ENTRY;

-- Provides for the specification of a basic reference path.

-- REFERENCE_PATH

type REFERENCE_PATH (DEPTH : PATH_DEPTH := 0) is
  record
    PATH : REFERENCE_PATH_ARRAY (0..DEPTH);
  end record;

-- Provides for the specification of a variable size
-- reference path.

-- REFERENCE_PATHS

package REFERENCE_PATHS is
   new PHIGS_LIST_UTILITIES (REFERENCE_PATH,
                             MAX_REFERENCE_PATHS_SUPPORTED);

-- Provides for the specification of lists of reference
-- paths.

-- RELATIVE_PRIORITY

type RELATIVE_PRIORITY is (HIGHER,
                           LOWER);

-- Indicates the relative input priority between two views.

-- RETURN_VALUE_TYPE

type RETURN_VALUE_TYPE is (SET,
                           REALIZED);

-- Indicates whether the returned values should be as they
-- were set by the program or as they were actually realized
-- on the device.

-- SEARCH_DIRECTION

type SEARCH_DIRECTION is (BACKWARD,
                          FORWARD);

-- This type provides for indicating search direction.

-- SEARCH_STATUS_INDICATOR

type SEARCH_STATUS_INDICATOR is (FAILURE,
                                 SUCCESS);

-- This type provides for indicating the result of a search.

-- STRING_DEVICE_NUMBER

type STRING_DEVICE_NUMBER is new DEVICE_NUMBER;

-- Provides for string device identifiers.
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
-- STRING_PROMPT_ECHO_TYPE

type STRING_PROMPT_ECHO_TYPE is new PHIGS_INTEGER;

-- Defines the string prompt and echo types.

-- STRING_PROMPT_ECHO_TYPES

package STRING_PROMPT_ECHO_TYPES is
   new PHIGS_LIST_UTILITIES (STRING_PROMPT_ECHO_TYPE,
                             MAX_STRING_PROMPT_ECHO_TYPES_SUPPORTED);

-- Provides for lists of string prompt and echo types.

-- STROKE DEVICE NUMBER

type STROKE_DEVICE_NUMBER is new DEVICE_NUMBER;

-- Provides for stroke device identifiers.

-- STROKE_PROMPT_ECHO_TYPE

type STROKE_PROMPT_ECHO_TYPE is new PHIGS_INTEGER;

-- Defines the stroke prompt and echo types.

-- STROKE_PROMPT_ECHO_TYPES

package STROKE_PROMPT_ECHO_TYPES is
   new PHIGS_LIST_UTILITIES (STROKE_PROMPT_ECHO_TYPE,
                             MAX_STROKE_PROMPT_ECHO_TYPES_SUPPORTED);

-- Provides for lists of stroke prompt and echo types

-- STRUCTURE_ELEMENT_TYPE

subtype STRUCTURE_ELEMENT_TYPE is
        ELEMENT_TYPE range NIL..ELEMENT_TYPE'LAST;

-- Provides for identifying the types of individual
-- structure elements.

-- STRUCTURE NETWORK SOURCE

type STRUCTURE_NETWORK_SOURCE is (CSS,
                                  ARCHIVE);

-- Provides for indicating source for check of conflicting
-- structure identifiers.

-- DISPLAY_PRIORITY

type DISPLAY_PRIORITY is digits PHIGS_PRECISION range 0.0..1.0;

-- Defines type for specifying structure display priority.


-- STRUCTURE_STATE
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
type STRUCTURE_STATE is (STCL,
                         STOP);

-- The type used to return the structure state value.

-- STRUCTURE_STATUS

type STRUCTURE_STATUS is (NON_EXISTENT,
                          EMPTY,
                          NOTEMPTY);

-- The type used to return the structure status.

-- POSTED_STRUCTURE

type POSTED_STRUCTURE is
  record
    STRUCTURE_IDENTIFIER : STRUCTURE_ID;
    PRIORITY             : DISPLAY_PRIORITY;
  end record;

-- Defines a data type for describing posted structure networks.

-- POSTED_STRUCTURES

package POSTED_STRUCTURES is
        new PHIGS_LIST_UTILITIES (POSTED_STRUCTURE,
                                  MAX_POSTED_STRUCTURES_SUPPORTED);

-- Provides for the specification of lists of posted structure
-- networks.


-- SUBPROGRAM_NAME

subtype SUBPROGRAM_NAME is PHIGS_STRING;

-- Defines the name of a PHIGS function detecting an error.

-- SYSTEM_STATE

type SYSTEM_STATE is (PHCL,
                      PHOP);

-- The type used to return the system state value.

-- VERTICAL_ALIGNMENT

type VERTICAL_ALIGNMENT is (NORMAL,
                            TOP,
                            CAP,
                            HALF,
                            BASE,
                            BOTTOM);

-- The alignment of the text extent parallelogram with respect
-- to the vertical positioning of the text.

-- TEXT_ALIGNMENT
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
type TEXT_ALIGNMENT is
  record
    HORIZONTAL : HORIZONTAL_ALIGNMENT;
    VERTICAL   : VERTICAL_ALIGNMENT;
  end record;

-- The type of the attribute controlling the positioning of
-- the text extent parallelogram in relation to the text
-- position, having horizontal and vertical components as
-- defined above.


-- TEXT_FONT

type TEXT_FONT is new PHIGS_INTEGER;

-- Defines the types of fonts provided by the implementation.

-- TEXT_PRECISION

type TEXT_PRECISION is (STRING_PRECISION,
                        CHAR_PRECISION,
                        STROKE_PRECISION);

-- The precision with which text appears.

-- TEXT_FONT_PRECISION

type TEXT_FONT_PRECISION is
  record
    FONT      : TEXT_FONT;
    PRECISION : TEXT_PRECISION;
  end record;

-- This type defines a record describing the text font and
-- precision supported.

-- TEXT_FONT_PRECISIONS

package TEXT_FONT_PRECISIONS is
   new PHIGS_LIST_UTILITIES (TEXT_FONT_PRECISION,
                             MAX_TEXT_FONT_PRECISION_PAIRS_SUPPORTED);

-- Provides for lists of text font and precision pairs.

-- TEXT_INDEX

type TEXT_INDEX is new PHIGS_POSITIVE;

-- Defines the range of text bundle table indices.

-- TEXT_INDICES

package TEXT_INDICES is
        new PHIGS_LIST_UTILITIES (TEXT_INDEX,
                                  MAX_TEXT_INDICES_SUPPORTED);

-- Provides for lists of text indices.
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
-- TEXT_PATH

type TEXT_PATH is (RIGHT,
                   LEFT,
                   UP,
                   DOWN);

-- The direction taken by a text string.

-- TRANSFORMATION_MATRIX_2

type TRANSFORMATION_MATRIX_2 is
      array (1..3, 1..3) of MC_TYPE;

-- For 2D modeling and view transformation matrices.

-- TRANSFORMATION_MATRIX_3

type TRANSFORMATION_MATRIX_3 is
      array (1..4, 1..4) of MC_TYPE;

-- For 3D modeling and view transformation matrices.

-- UPDATE_REGENERATION_FLAG

type UPDATE_REGENERATION_FLAG is (PERFORM,
                                  POSTPONE);

-- Flag indicating regeneration action on display.

-- UPDATE_STATE

type UPDATE_STATE is (NOTPENDING,
                      PENDING);

-- Indicates whether or not a workstation transformation
-- change has been requested and not yet provided.

-- VALUATOR_DEVICE_NUMBER

type VALUATOR_DEVICE_NUMBER is new DEVICE_NUMBER;

-- Provides for valuator device identifiers.

-- VALUATOR_PROMPT_ECHO_TYPE

type VALUATOR_PROMPT_ECHO_TYPE is new PHIGS_INTEGER;

-- Defines the possible range of valuator prompt and
-- echo types.

-- VALUATOR_PROMPT_ECHO_TYPES

package VALUATOR_PROMPT_ECHO_TYPES is
  new PHIGS_LIST_UTILITIES (VALUATOR_PROMPT_ECHO_TYPE,
                            MAX_VALUATOR_PROMPT_ECHO_TYPES_SUPPORTED);

-- Provides for lists of valuator prompt and echo types.
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
-- VALUATOR_VALUE

type VALUATOR_VALUE is digits PHIGS_PRECISION;

-- Defines the range of accuracy of valuator input values on
-- an implementation.


-- EVENT_DEVICE_NUMBER

type EVENT_DEVICE_NUMBER (CLASS : INPUT_CLASS := NONE) is
  record
    case CLASS is

      when NONE = > null;

      when LOCATOR_INPUT
        = > LOCATOR_EVENT_DEVICE : LOCATOR_DEVICE_NUMBER;

      when STROKE_INPUT
        = > STROKE_EVENT_DEVICE : STROKE_DEVICE_NUMBER;

      when VALUATOR_INPUT
        = > VALUATOR_EVENT_DEVICE : VALUATOR_DEVICE_NUMBER;

      when CHOICE_INPUT
        = > CHOICE_EVENT_DEVICE : CHOICE_DEVICE_NUMBER;

      when PICK_INPUT
        = > PICK_EVENT_DEVICE : PICK_DEVICE_NUMBER;

      when STRING_INPUT
        = > STRING_EVENT_DEVICE : STRING_DEVICE_NUMBER;

    end case;
  end record;

-- Provides for returning any class of device number from the
-- event queue.

-- EVENT_QUEUE_DEVICE_NUMBER

type EVENT_QUEUE_DEVICE_NUMBER
  (CLASS : INPUT_QUEUE_CLASS := LOCATOR_INPUT) is
      record
        case CLASS is

          when LOCATOR_INPUT
            = > LOCATOR_EVENT_DEVICE : LOCATOR_DEVICE_NUMBER;

          when STROKE_INPUT
            = > STROKE_EVENT_DEVICE : STROKE_DEVICE_NUMBER;

          when VALUATOR_INPUT
            = > VALUATOR_EVENT_DEVICE : VALUATOR_DEVICE_NUMBER;

          when CHOICE_INPUT
            = > CHOICE_EVENT_DEVICE : CHOICE_DEVICE_NUMBER;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
            when PICK_INPUT
               => PICK_EVENT_DEVICE : PICK_DEVICE_NUMBER;

            when STRING_INPUT
               => STRING_EVENT_DEVICE : STRING_DEVICE_NUMBER;
         end case;
      end record;

-- Allows EVENT_DEVICE_NUMBERs which cannot have value NONE.  This is
-- used when returning only real input classes as causes of event
-- queue overflow and for flushing the queue.

-- ACCESS_COLOUR_MATRIX

type ACCESS_COLOUR_MATRIX is access COLOUR_MATRIX;

-- Provides for returning variable sized matrices containing colour
-- indices corresponding to a cell array or pattern array.

-- VARIABLE_CONNECTION_ID

type VARIABLE_CONNECTION_ID (LENGTH : STRING_SMALL_NATURAL := 0) is
  record
    CONNECT : CONNECTION_ID (1..LENGTH);
  end record;

-- Defines a variable length connection identifier for
-- INQ_WS_CONNECTION_AND_TYPE.


-- ACCESS_STRING

type ACCESS_STRING is access PHIGS_STRING;

--  Defines a variable length string needed to store strings in
--  structure element records.


-- VIEW_INDEX

type VIEW_INDEX is new PHIGS_NATURAL;

-- Defines a type for specifying view indices.

-- VIEW_INDICES

package VIEW_INDICES is
         new PHIGS_LIST_UTILITIES (VIEW_INDEX,
                                   MAX_VIEW_INDICES_SUPPORTED);


--  Provides for lists of view indices.

-- VISUAL REPRESENTATION STATE

type VISUAL_REPRESENTATION_STATE is (CORRECT,
                                     DEFERRED,
                                     SIMULATED);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
-- Specifies state of image on display surface.

-- WS_CATEGORY

type WS_CATEGORY is (OUTPUT,
                     INPUT,
                     OUTIN,
                     MO,
                     MI);

-- Type for PHIGS workstation categories.

-- WS_DEPENDENCY

type WS_DEPENDENCY is (WS_INDEPENDENT,
                       WS_DEPENDENT);

-- Type for indications of workstation dependency.

-- WS_DEPENDENCIES

package WS_DEPENDENCIES is
  new PHIGS_LIST_UTILITIES (WS_DEPENDENCY,
                            MAX_GSE_IDS_SUPPORTED);

-- Provides for lists of workstation dependencies.

-- WS_ID

type WS_ID is new PHIGS_POSITIVE;

-- Defines the range of workstation identifiers.

-- WS_IDS

package WS_IDS is new PHIGS_LIST_UTILITIES (WS_ID,
                                            MAX_OPEN_WS_SUPPORTED);

-- Provides for lists of workstation identifiers.

-- WS_STATE

type WS_STATE is (WSCL,
                  WSOP);

-- The type used to return the workstation state value.

-- WS_TYPE

type WS_TYPE is new PHIGS_POSITIVE;

-- Range of values corresponding to valid workstation types.
-- Constants specifying names for the various types of
-- workstations should be provided by an implementation.

-- WS_TYPES

package WS_TYPES is new PHIGS_LIST_UTILITIES (WS_TYPE,
                                              MAX_WS_TYPES_SUPPORTED);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
-- Provides for lists of workstation types.

-- STRUCTURE_ELEMENT_RECORD

type STRUCTURE_ELEMENT_RECORD
  (ELEMENT_TYPE : STRUCTURE_ELEMENT_TYPE := NIL) is
      record
        case ELEMENT_TYPE is

          -- The empty element

          when NIL = >
            null;

          -- Primitive elements

          when POLYLINE_3 = >
            POLYLINE_3_POINTS : MC.ACCESS_POINT_LIST_3;

          when POLYLINE = >
            POLYLINE_POINTS : MC.ACCESS_POINT_LIST_2;

          when POLYMARKER_3 = >
            POLYMARKER_3_POINTS : MC.ACCESS_POINT_LIST_3;

          when POLYMARKER = >
            POLYMARKER_POINTS : MC.ACCESS_POINT_LIST_2;

          when TEXT_3 = >
            TEXT_3_POINT : MC.POINT_3;
            TEXT_DIRECTION_VECTORS : MC.VECTOR_PAIR_3;
            TEXT_3_CHAR_STRING : ACCESS_STRING;

          when TEXT = >
            TEXT_POINT : MC.POINT_2;
            TEXT_CHAR_STRING : ACCESS_STRING;

          when ANNOTATION_TEXT_RELATIVE_3 = >
            ANNOTATION_TEXT_RELATIVE_3_REF_POINT : MC.POINT_3;
            ANNOTATION_TEXT_RELATIVE_3_OFFSET : NPC.POINT_3;
            ANNOTATION_TEXT_RELATIVE_3_CHAR_STRING : ACCESS_STRING;

          when ANNOTATION_TEXT_RELATIVE = >
            ANNOTATION_TEXT_RELATIVE_REF_POINT : MC.POINT_2;
            ANNOTATION_TEXT_RELATIVE_OFFSET : NPC.POINT_2;
            ANNOTATION_TEXT_RELATIVE_CHAR_STRING : ACCESS_STRING;

          when FILL_AREA_3 = >
            FILL_AREA_3_POINTS : MC.ACCESS_POINT_LIST_3;

          when FILL_AREA = >
            FILL_AREA_POINTS : MC.ACCESS_POINT_LIST_2;

          when FILL_AREA_SET_3 = >
            FILL_AREA_SET_3_POINT_LISTS : MC.ACCESS_LIST_OF_POINT_LIST_3;

          when FILL_AREA_SET = >
            FILL_AREA_SET_POINT_LISTS : MC.ACCESS_LIST_OF_POINT_LIST_2;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
when CELL_ARRAY_3 = >
   CORNER_P_3 : MC.POINT_3;
   CORNER_Q_3 : MC.POINT_3;
   CORNER_R_3 : MC.POINT_3;
   CELL_ARRAY_3_CELLS : ACCESS_COLOUR_MATRIX;

when CELL_ARRAY = >
   CORNER_P : MC.POINT_2;
   CORNER_Q : MC.POINT_2;
   CELL_ARRAY_CELLS : ACCESS_COLOUR_MATRIX;

when GDP_3 = >
   GDP_3_POINTS : MC.ACCESS_POINT_LIST_3;
   GDP_3_DATA : GDP_3_RECORD;

when GDP = >
   GDP_POINTS : MC.ACCESS_POINT_LIST_2;
   GDP_DATA : GDP_RECORD;

-- Bundle index elements

when SET_POLYLINE_INDEX = >
   POLYLINE_IND : POLYLINE_INDEX;

when SET_POLYMARKER_INDEX = >
   POLYMARKER_IND : POLYMARKER_INDEX;

when SET_TEXT_INDEX = >
   TEXT_IND : TEXT_INDEX;

when SET_INTERIOR_INDEX = >
   INTERIOR_IND : INTERIOR_INDEX;

when SET_EDGE_INDEX = >
   EDGE_IND : EDGE_INDEX;

-- Individual aspect elements

when SET_LINETYPE = >
   TYPE_OF_LINE : LINETYPE;

when SET_LINEWIDTH_SCALE_FACTOR = >
   LINEWIDTH_SF : LINEWIDTH;

when SET_POLYLINE_COLOUR_INDEX = >
   LINE_COLOUR : COLOUR_INDEX;

when SET_MARKER_TYPE = >
   TYPE_OF_MARKER : MARKER_TYPE;

when SET_MARKER_SIZE_SCALE_FACTOR = >
   SIZE : MARKER_SIZE;

when SET_POLYMARKER_COLOUR_INDEX = >
   MARKER_COLOUR : COLOUR_INDEX;

when SET_TEXT_FONT = >
   FONT : TEXT_FONT;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
when SET_TEXT_PRECISION =>
   PRECISION : TEXT_PRECISION;

when SET_CHAR_EXPANSION_FACTOR =>
   EXPANSION : CHAR_EXPANSION;

when SET_CHAR_SPACING =>
   SPACING : CHAR_SPACING;

when SET_TEXT_COLOUR_INDEX =>
   TEXT_COLOUR : COLOUR_INDEX;

when SET_CHAR_HEIGHT =>
   HEIGHT : MC.MAGNITUDE;

when SET_CHAR_UP_VECTOR =>
   CHAR_UP_VECTOR : MC.VECTOR_2;

when SET_TEXT_PATH =>
   PATH : TEXT_PATH;

when SET_TEXT_ALIGNMENT =>
   ALIGNMENT : TEXT_ALIGNMENT;

when SET_ANNOTATION_TEXT_CHAR_HEIGHT =>
   ANNOTATION_HEIGHT : NPC.MAGNITUDE;

when SET_ANNOTATION_TEXT_CHAR_UP_VECTOR =>
   ANNOTATION_CHAR_UP_VECTOR : NPC.VECTOR_2;

when SET_ANNOTATION_TEXT_PATH =>
   ANNOTATION_PATH : TEXT_PATH;

when SET_ANNOTATION_TEXT_ALIGNMENT =>
   ANNOTATION_ALIGNMENT : TEXT_ALIGNMENT;

when SET_ANNOTATION_STYLE =>
   STYLE_OF_ANNOTATION : ANNOTATION_STYLE;

when SET_INTERIOR_STYLE =>
   STYLE_OF_INTERIOR : INTERIOR_STYLE;

when SET_INTERIOR_STYLE_INDEX =>
   STYLE_IND : STYLE_INDEX;

when SET_INTERIOR_COLOUR_INDEX =>
   INTERIOR_COLOUR : COLOUR_INDEX;

when SET_EDGE_FLAG =>
   FLAG : EDGE_FLAG;

when SET_EDGETYPE =>
   TYPE_OF_EDGE : EDGETYPE;

when SET_EDGEWIDTH_SCALE_FACTOR =>
   EDGEWIDTH_SF : EDGEWIDTH;

when SET_EDGE_COLOUR_INDEX =>
   EDGE_COLOUR : COLOUR_INDEX;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
-- Pattern attribute elements

when SET_PATTERN_SIZE = >
   PATTERN_SIZE : MC.SIZE_2;

when SET_PATTERN_REFERENCE_POINT_AND_VECTORS = >
   PATTERN_REFERENCE_POINT_3 : MC.POINT_3;
   PATTERN_REFERENCE_VECTORS : MC.VECTOR_PAIR_3;

when SET_PATTERN_REFERENCE_POINT = >
   PATTERN_REFERENCE_POINT : MC.POINT_2;

-- Name set elements

when ADD_NAMES_TO_SET = >
   NAMES_TO_ADD : NAME_SET;

when REMOVE_NAMES_FROM_SET = >
   NAMES_TO_REMOVE : NAME_SET;

-- ASF elements

when SET_INDIVIDUAL_ASF = >
   ATTRIBUTE_ID : ASPECT;
   SOURCE_FLAG : ASF;

-- HLHSR elements

when SET_HLHSR_IDENTIFIER = >
   HLHSR_IDENTIFIER : HLHSR_ID;

-- Transformation elements

when SET_LOCAL_TRANSFORMATION_3 = >
   LOCAL_MATRIX_3 : TRANSFORMATION_MATRIX_3;
   HOW_APPLIED_3 : COMPOSITION_TYPE;

when SET_LOCAL_TRANSFORMATION = >
   LOCAL_MATRIX : TRANSFORMATION_MATRIX_2;
   HOW_APPLIED : COMPOSITION_TYPE;

when SET_GLOBAL_TRANSFORMATION_3 = >
   GLOBAL_MATRIX_3 : TRANSFORMATION_MATRIX_3;

when SET_GLOBAL_TRANSFORMATION = >
   GLOBAL_MATRIX : TRANSFORMATION_MATRIX_2;

when SET_MODELLING_CLIPPING_VOLUME_3 = >
   MODELLING_CLIPPING_OPERATOR_3 : MODELLING_CLIP_OPERATION_TYPE;
   MODELLING_CLIPPING_LIMITS_3 : MC.ACCESS_HALF_SPACE_LIST_3;

when SET_MODELLING_CLIPPING_VOLUME = >
   MODELLING_CLIPPING_OPERATOR : MODELLING_CLIP_OPERATION_TYPE;
   MODELLING_CLIPPING_LIMITS : MC.ACCESS_HALF_SPACE_LIST_2;

when SET_MODELLING_CLIPPING_INDICATOR = >
   MODELLING_CLIPPING_INDICATOR : CLIPPING_INDICATOR;

when RESTORE_MODELLING_CLIPPING_VOLUME = >
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
            null;

         when SET_VIEW_INDEX = >
            VIEW_IND : VIEW_INDEX;

         -- Invocation elements

         when EXECUTE_STRUCTURE = >
            STRUCTURE_IDENTIFIER : STRUCTURE_ID;

         -- Structure content identification elements

         when LABEL = >
            LABEL_IDENTIFIER : LABEL_ID;

         when APPLICATION_DATA = >
            DATA : APPLICATION_DATA_RECORD;

         when GSE = >
            GSE_DATA : GSE_RECORD;

         when SET_PICK_IDENTIFIER = >
            PICK_IDENTIFIER : PICK_ID;

      end case;
   end record;

-- This type defines the format of structure contents and is used to
-- return structure elements during inquiry.

PHIGS_ERROR : exception;

-- Exception to notify the application program of an error in a
-- PHIGS procedure should the version of ERROR_HANDLING which raises
-- an exception be used.

-- This section contains the constants that provide the PHIGS standard
-- values defined for some PHIGS/Ada types. The constants for colour
-- models are defined earlier.

-- The following constants define the PHIGS standard line types:

SOLID_LINE                  : constant LINETYPE := 1;
DASHED_LINE                 : constant LINETYPE := 2;
DOTTED_LINE                 : constant LINETYPE := 3;
DASHED_DOTTED_LINE          : constant LINETYPE := 4;

-- The following constants define the PHIGS standard marker types:

DOT_MARKER                  : constant MARKER_TYPE := 1;
PLUS_MARKER                 : constant MARKER_TYPE := 2;
STAR_MARKER                 : constant MARKER_TYPE := 3;
ZERO_MARKER                 : constant MARKER_TYPE := 4;
X_MARKER                    : constant MARKER_TYPE := 5;

-- The following constants define the PHIGS standard edgetypes:

SOLID_EDGE                  : constant EDGETYPE := 1;
DASHED_EDGE                 : constant EDGETYPE := 2;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
    DOTTED_EDGE                     : constant EDGETYPE := 3;
    DASHED_DOTTED_EDGE              : constant EDGETYPE := 4;

    -- The following constants define the PHIGS standard annotation styles:

    UNCONNECTED_ANNOTATION          : constant ANNOTATION_STYLE := 1;
    LEAD_LINE_ANNOTATION            : constant ANNOTATION_STYLE := 2;

    -- The following constants are used for defining the modelling clipping
    -- operators specified by PHIGS:

    REPLACE_VOLUME                  : constant MODELLING_CLIP_OPERATION_TYPE := 1;
    INTERSECT_VOLUME                : constant MODELLING_CLIP_OPERATION_TYPE := 2;

    -- The following constants define the prompt and echo types supported
    -- by PHIGS:

    DEFAULT_LOCATOR                 : constant LOCATOR_PROMPT_ECHO_TYPE := 1;
    CROSS_HAIR_LOCATOR              : constant LOCATOR_PROMPT_ECHO_TYPE := 2;
    TRACKING_CROSS_LOCATOR          : constant LOCATOR_PROMPT_ECHO_TYPE := 3;
    RUBBER_BAND_LINE_LOCATOR        : constant LOCATOR_PROMPT_ECHO_TYPE := 4;
    RECTANGLE_LOCATOR               : constant LOCATOR_PROMPT_ECHO_TYPE := 5;
    DIGITAL_LOCATOR                 : constant LOCATOR_PROMPT_ECHO_TYPE := 6;

    DEFAULT_STROKE                  : constant STROKE_PROMPT_ECHO_TYPE := 1;
    DIGITAL_STROKE                  : constant STROKE_PROMPT_ECHO_TYPE := 2;
    MARKER_STROKE                   : constant STROKE_PROMPT_ECHO_TYPE := 3;
    LINE_STROKE                     : constant STROKE_PROMPT_ECHO_TYPE := 4;

    DEFAULT_VALUATOR                : constant VALUATOR_PROMPT_ECHO_TYPE := 1;
    GRAPHICAL_VALUATOR              : constant VALUATOR_PROMPT_ECHO_TYPE := 2;
    DIGITAL_VALUATOR                : constant VALUATOR_PROMPT_ECHO_TYPE := 3;

    DEFAULT_CHOICE                  : constant CHOICE_PROMPT_ECHO_TYPE := 1;
    PROMPT_ECHO_CHOICE              : constant CHOICE_PROMPT_ECHO_TYPE := 2;
    STRING_PROMPT_CHOICE            : constant CHOICE_PROMPT_ECHO_TYPE := 3;
    STRING_INPUT_CHOICE             : constant CHOICE_PROMPT_ECHO_TYPE := 4;
    STRUCTURE_CHOICE                : constant CHOICE_PROMPT_ECHO_TYPE := 5;

    DEFAULT_PICK                    : constant PICK_PROMPT_ECHO_TYPE := 1;
    GROUP_HIGHLIGHT_PICK            : constant PICK_PROMPT_ECHO_TYPE := 2;
    STRUCTURE_HIGHLIGHT_PICK        : constant PICK_PROMPT_ECHO_TYPE := 3;

end PHIGS_TYPES;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
-- The PHIGS Package

with PHIGS_CONFIGURATION, PHIGS_TYPES, PHIGS_NAME_SET_FACILITY;
use PHIGS_CONFIGURATION, PHIGS_TYPES;

package PHIGS is

 -- The package PHIGS contains all of the procedures that are required
 -- to implement PHIGS.

 -- Private Type Declarations

 -- The following data types are the PHIGS private types and are
 -- included in package PHIGS for ease of manipulation.

 -- METAFILE_DATA_RECORD

 type METAFILE_DATA_RECORD (TYPE_OF_ITEM : METAFILE_ITEM_TYPE := 0) is private;

 --  A data record for metafiles.

 -- CHOICE_DATA_RECORD

 type CHOICE_DATA_RECORD
   (PROMPT_ECHO_TYPE : CHOICE_PROMPT_ECHO_TYPE := 1) is private;

 -- Defines a choice input data record.

 -- LOCATOR_DATA_RECORD

 type LOCATOR_DATA_RECORD
   (PROMPT_ECHO_TYPE : LOCATOR_PROMPT_ECHO_TYPE := 1) is private;

 -- Defines a locator input data record.

 -- PICK_DATA_RECORD

 type PICK_DATA_RECORD
   (PROMPT_ECHO_TYPE : PICK_PROMPT_ECHO_TYPE := 1) is private;

 -- Defines a pick input data record.

 -- STRING_DATA_RECORD

 type STRING_DATA_RECORD
   (PROMPT_ECHO_TYPE : STRING_PROMPT_ECHO_TYPE := 1) is private;

 -- Defines a string input data record.

 -- STROKE_DATA_RECORD

 type STROKE_DATA_RECORD
   (PROMPT_ECHO_TYPE : STROKE_PROMPT_ECHO_TYPE := 1) is private;

 -- Defines a stroke input data record.

 -- VALUATOR_DATA_RECORD
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

9593 PT*3 90 ■ 4851903 0575580 61T ■


```
type VALUATOR_DATA_RECORD
  (PROMPT_ECHO_TYPE : VALUATOR_PROMPT_ECHO_TYPE := 1) is private;

-- Defines a valuator data record.

-- PHIGS Procedures

-- CONTROL FUNCTIONS

procedure OPEN_PHIGS
  (ERROR_FILE             : in FILE_ID := PHIGS_STRING(DEFAULT_ERROR_FILE);
   AMOUNT_OF_MEMORY       : in PHIGS_NATURAL := DEFAULT_MEMORY_UNITS);

procedure CLOSE_PHIGS;

procedure OPEN_WS
  (WS                     : in WS_ID;
   CONNECTION             : in CONNECTION_ID;
   TYPE_OF_WS             : in WS_TYPE);

procedure CLOSE_WS
  (WS : in WS_ID);

procedure REDRAW_ALL_STRUCTURES
  (WS     : in WS_ID;
   FLAG   : in CONTROL_FLAG);

procedure UPDATE_WS
  (WS                     : in WS_ID;
   REGENERATION           : in UPDATE_REGENERATION_FLAG);

procedure SET_DISPLAY_UPDATE_STATE
  (WS                     : in WS_ID;
   DEFERRAL               : in DEFERRAL_MODE;
   MODIFICATION           : in MODIFICATION_MODE);

procedure MESSAGE
  (WS                     : in WS_ID;
   CONTENTS               : in PHIGS_STRING);

-- OUTPUT FUNCTIONS

procedure POLYLINE
  (POINTS : in MC.POINT_LIST_3);

procedure POLYLINE
  (POINTS : in MC.POINT_LIST_2);

procedure POLYMARKER
  (POINTS : in MC.POINT_LIST_3);

procedure POLYMARKER
  (POINTS : in MC.POINT_LIST_2);

procedure TEXT
  (POSITION               : in MC.POINT_3;
   DIRECTION_VECTORS      : in MC.VECTOR_PAIR_3;
   CHAR_STRING            : in PHIGS_STRING);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
procedure TEXT
  (POSITION      : in MC.POINT_2;
   CHAR_STRING   : in PHIGS_STRING);


procedure ANNOTATION_TEXT_RELATIVE
  (REFERENCE_POINT    : in MC.POINT_3;
   ANNOTATION_OFFSET  : in NPC.POINT_3;
   CHAR_STRING        : in PHIGS_STRING);

procedure ANNOTATION_TEXT_RELATIVE
  (REFERENCE_POINT    : in MC.POINT_2;
   ANNOTATION_OFFSET  : in NPC.POINT_2;
   CHAR_STRING        : in PHIGS_STRING);

procedure FILL_AREA
  (POINTS : in MC.POINT_LIST_3);

procedure FILL_AREA
  (POINTS : in MC.POINT_LIST_2);

procedure FILL_AREA_SET
  (POINT_LISTS : in MC.LIST_OF_POINT_LIST_3);

procedure FILL_AREA_SET
  (POINT_LISTS : in MC.LIST_OF_POINT_LIST_2);

procedure CELL_ARRAY
  (CORNER_P  : in MC.POINT_3;
   CORNER_Q  : in MC.POINT_3;
   CORNER_R  : in MC.POINT_3;
   CELLS     : in COLOUR_MATRIX);

procedure CELL_ARRAY
  (CORNER_P  : in MC.POINT_2;
   CORNER_Q  : in MC.POINT_2;
   CELLS     : in COLOUR_MATRIX);

-- OUTPUT ATTRIBUTE FUNCTIONS

procedure SET_POLYLINE_INDEX
  (POLYLINE_IND : in POLYLINE_INDEX);

procedure SET_POLYMARKER_INDEX
  (POLYMARKER_IND : in POLYMARKER_INDEX);

procedure SET_TEXT_INDEX
  (TEXT_IND : in TEXT_INDEX);

procedure SET_INTERIOR_INDEX
  (INTERIOR_IND : in INTERIOR_INDEX);

procedure SET_EDGE_INDEX
  (EDGE_IND : in EDGE_INDEX);

procedure SET_LINETYPE
  (TYPE_OF_LINE : in LINETYPE);

procedure SET_LINEWIDTH_SCALE_FACTOR
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
    (LINEWIDTH_SF : in LINEWIDTH);

  procedure SET_POLYLINE_COLOUR_INDEX
    (LINE_COLOUR : in COLOUR_INDEX);

  procedure SET_MARKER_TYPE
    (TYPE_OF_MARKER : in MARKER_TYPE);

  procedure SET_MARKER_SIZE_SCALE_FACTOR
    (SIZE : in MARKER_SIZE);

  procedure SET_POLYMARKER_COLOUR_INDEX
    (MARKER_COLOUR : in COLOUR_INDEX);

  procedure SET_TEXT_FONT
    (FONT : in TEXT_FONT);

  procedure SET_TEXT_PRECISION
    (PRECISION : in TEXT_PRECISION);

  procedure SET_CHAR_EXPANSION_FACTOR
    (EXPANSION : in CHAR_EXPANSION);

  procedure SET_CHAR_SPACING
    (SPACING : in CHAR_SPACING);

  procedure SET_TEXT_COLOUR_INDEX
    (TEXT_COLOUR : in COLOUR_INDEX);

  procedure SET_CHAR_HEIGHT
    (HEIGHT : in MC.MAGNITUDE);

  procedure SET_CHAR_UP_VECTOR
    (CHAR_UP_VECTOR : in MC.VECTOR_2);

  procedure SET_TEXT_PATH
    (PATH : in TEXT_PATH);

  procedure SET_TEXT_ALIGNMENT
    (ALIGNMENT : in TEXT_ALIGNMENT);

  procedure SET_ANNOTATION_TEXT_CHAR_HEIGHT
    (ANNOTATION_HEIGHT : in NPC.MAGNITUDE);

  procedure SET_ANNOTATION_TEXT_CHAR_UP_VECTOR
    (ANNOTATION_CHAR_UP_VECTOR : in NPC.VECTOR_2);

  procedure SET_ANNOTATION_TEXT_PATH
    (ANNOTATION_PATH : in TEXT_PATH);

  procedure SET_ANNOTATION_TEXT_ALIGNMENT
    (ANNOTATION_ALIGNMENT : in TEXT_ALIGNMENT);

  procedure SET_ANNOTATION_STYLE
    (STYLE_OF_ANNOTATION : in ANNOTATION_STYLE);

  procedure SET_INTERIOR_STYLE
    (STYLE_OF_INTERIOR : in INTERIOR_STYLE);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
procedure SET_INTERIOR_STYLE_INDEX
  (STYLE_IND : in STYLE_INDEX);

procedure SET_INTERIOR_COLOUR_INDEX
  (INTERIOR_COLOUR : in COLOUR_INDEX);

procedure SET_EDGE_FLAG
  (FLAG : in EDGE_FLAG);

procedure SET_EDGETYPE
  (TYPE_OF_EDGE : in EDGETYPE);

procedure SET_EDGEWIDTH_SCALE_FACTOR
  (EDGEWIDTH_SF : in EDGEWIDTH);

procedure SET_EDGE_COLOUR_INDEX
  (EDGE_COLOUR : in COLOUR_INDEX);

procedure SET_PATTERN_SIZE
  (SIZE : in MC.SIZE_2);

procedure SET_PATTERN_REFERENCE_POINT_AND_VECTORS
  (REFERENCE_POINT    : in MC.POINT_3;
   REFERENCE_VECTORS  : in MC.VECTOR_PAIR_3);

procedure SET_PATTERN_REFERENCE_POINT
  (REFERENCE_POINT : MC.POINT_2);

procedure ADD_NAMES_TO_SET
  (NAMES_TO_ADD : in NAME_SET);

procedure REMOVE_NAMES_FROM_SET
  (NAMES_TO_REMOVE : in NAME_SET);

procedure SET_INDIVIDUAL_ASF
  (ASPECT_ID      : in ASPECT;
   SOURCE_FLAG    : in ASF);

procedure SET_POLYLINE_REPRESENTATION
  (WS              : in WS_ID;
   POLYLINE_IND    : in POLYLINE_INDEX;
   TYPE_OF_LINE    : in LINETYPE;
   LINEWIDTH_SF    : in LINEWIDTH;
   LINE_COLOUR     : in COLOUR_INDEX);

procedure SET_POLYMARKER_REPRESENTATION
  (WS               : in WS_ID;
   POLYMARKER_IND   : in POLYMARKER_INDEX;
   TYPE_OF_MARKER   : in MARKER_TYPE;
   SIZE             : in MARKER_SIZE;
   MARKER_COLOUR    : in COLOUR_INDEX);

procedure SET_TEXT_REPRESENTATION
  (WS               : in WS_ID;
   TEXT_IND         : in TEXT_INDEX;
   FONT             : in TEXT_FONT;
   PRECISION        : in TEXT_PRECISION;
   EXPANSION        : in CHAR_EXPANSION;
   SPACING          : in CHAR_SPACING;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
    TEXT_COLOUR  : in COLOUR_INDEX);

procedure SET_INTERIOR_REPRESENTATION
  (WS                  : in WS_ID;
   INTERIOR_IND        : in INTERIOR_INDEX;
   STYLE_OF_INTERIOR   : in INTERIOR_STYLE;
   STYLE_IND           : in STYLE_INDEX;
   INTERIOR_COLOUR     : in COLOUR_INDEX);

procedure SET_EDGE_REPRESENTATION
  (WS                  : in WS_ID;
   EDGE_IND            : in EDGE_INDEX;
   FLAG                : in EDGE_FLAG;
   TYPE_OF_EDGE        : in EDGETYPE;
   EDGEWIDTH_SF        : in EDGEWIDTH;
   EDGE_COLOUR         : in COLOUR_INDEX);

procedure SET_PATTERN_REPRESENTATION
  (WS                  : in WS_ID;
   PATTERN_IND         : in PATTERN_INDEX;
   PATTERN             : in COLOUR_MATRIX);

procedure SET_COLOUR_REPRESENTATION
  (WS                  : in WS_ID;
   COLOUR_IND          : in COLOUR_INDEX;
   COLOUR              : in COLOUR_REPRESENTATION);

procedure SET_HIGHLIGHTING_FILTER
  (WS                  : in WS_ID;
   HIGHLIGHTING        : in NAME_SET_FILTER);

procedure SET_INVISIBILITY_FILTER
  (WS                  : in WS_ID;
   INVISIBILITY        : in NAME_SET_FILTER);

procedure SET_COLOUR_MODEL
  (WS                  : in WS_ID;
   MODEL               : in COLOUR_MODEL);

procedure SET_HLHSR_IDENTIFIER
  (HLHSR_IDENTIFIER    : in HLHSR_ID);

procedure SET_HLHSR_MODE
  (WS                  : in WS_ID;
   MODE                : in HLHSR_MODE);

-- TRANSFORMATION FUNCTIONS

procedure SET_LOCAL_TRANSFORMATION
  (MATRIX              : in TRANSFORMATION_MATRIX_3;
   HOW_APPLIED         : in COMPOSITION_TYPE);

procedure SET_LOCAL_TRANSFORMATION
  (MATRIX              : in TRANSFORMATION_MATRIX_2;
   HOW_APPLIED         : in COMPOSITION_TYPE);

procedure SET_GLOBAL_TRANSFORMATION
  (MATRIX : in TRANSFORMATION_MATRIX_3);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
procedure SET_GLOBAL_TRANSFORMATION
  (MATRIX : in TRANSFORMATION_MATRIX_2);

procedure SET_MODELLING_CLIPPING_VOLUME
  (OPERATOR     : in MODELLING_CLIP_OPERATION_TYPE;
   HALF_SPACES  : in MC.HALF_SPACE_LIST_3);

procedure SET_MODELLING_CLIPPING_VOLUME
  (OPERATOR     : in MODELLING_CLIP_OPERATION_TYPE;
   HALF_SPACES  : in MC.HALF_SPACE_LIST_2);

procedure SET_MODELLING_CLIPPING_INDICATOR
  (MODELLING_CLIPPING : in CLIPPING_INDICATOR);

procedure RESTORE_MODELLING_CLIPPING_VOLUME;

procedure SET_VIEW_INDEX
  (VIEW_IND : in VIEW_INDEX);

procedure SET_VIEW_REPRESENTATION
  (WS                  : in WS_ID;
   VIEW_IND            : in VIEW_INDEX;
   ORIENTATION_MATRIX  : in TRANSFORMATION_MATRIX_3;
   MAPPING_MATRIX      : in TRANSFORMATION_MATRIX_3;
   CLIPPING_LIMITS     : in NPC.RECTANGULAR_REGION_3;
   XY_CLIPPING         : in CLIPPING_INDICATOR;
   BACK_CLIPPING       : in CLIPPING_INDICATOR;
   FRONT_CLIPPING      : in CLIPPING_INDICATOR);

procedure SET_VIEW_REPRESENTATION
  (WS                  : in WS_ID;
   VIEW_IND            : in VIEW_INDEX;
   ORIENTATION_MATRIX  : in TRANSFORMATION_MATRIX_2;
   MAPPING_MATRIX      : in TRANSFORMATION_MATRIX_2;
   CLIPPING_LIMITS     : in NPC.RECTANGULAR_REGION_2;
   XY_CLIPPING         : in CLIPPING_INDICATOR);

procedure SET_VIEW_TRANSFORMATION_INPUT_PRIORITY
  (WS                  : in WS_ID;
   VIEW_IND            : in VIEW_INDEX;
   REFERENCE_INDEX     : in VIEW_INDEX;
   PRIORITY            : in RELATIVE_PRIORITY);

procedure SET_WS_WINDOW
  (WS                  : in WS_ID;
   WINDOW_LIMITS       : in NPC.RECTANGULAR_REGION_3);

procedure SET_WS_WINDOW
  (WS                  : in WS_ID;
   WINDOW_LIMITS       : in NPC.RECTANGULAR_REGION_2);

procedure SET_WS_VIEWPORT
  (WS                  : in WS_ID;
   VIEWPORT_LIMITS     : in DC.RECTANGULAR_REGION_3);

procedure SET_WS_VIEWPORT
  (WS                  : in WS_ID;
   VIEWPORT_LIMITS     : in DC.RECTANGULAR_REGION_2);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
-- TRANSFORMATION UTILITY FUNCTIONS

procedure TRANSLATE
  (VECTOR           : in  MC.VECTOR_3;
   ERROR_INDICATOR  : out ERROR_NUMBER;
   MATRIX           : out TRANSFORMATION_MATRIX_3);

procedure TRANSLATE
  (VECTOR           : in  MC.VECTOR_2;
   ERROR_INDICATOR  : out ERROR_NUMBER;
   MATRIX           : out TRANSFORMATION_MATRIX_2);

procedure SCALE
  (FACTOR           : in  MC.VECTOR_3;
   ERROR_INDICATOR  : out ERROR_NUMBER;
   MATRIX           : out TRANSFORMATION_MATRIX_3);

procedure SCALE
  (FACTOR           : in  MC.VECTOR_2;
   ERROR_INDICATOR  : out ERROR_NUMBER;
   MATRIX           : out TRANSFORMATION_MATRIX_2);

procedure ROTATE_X
  (ANGLE_X          : in  ANGLE;
   ERROR_INDICATOR  : out ERROR_NUMBER;
   MATRIX           : out TRANSFORMATION_MATRIX_3);

procedure ROTATE_Y
  (ANGLE_Y          : in  ANGLE;
   ERROR_INDICATOR  : out ERROR_NUMBER;
   MATRIX           : out TRANSFORMATION_MATRIX_3);

procedure ROTATE_Z
  (ANGLE_Z          : in  ANGLE;
   ERROR_INDICATOR  : out ERROR_NUMBER;
   MATRIX           : out TRANSFORMATION_MATRIX_3);

procedure ROTATE
  (ANGLE_Z          : in  ANGLE;
   ERROR_INDICATOR  : out ERROR_NUMBER;
   MATRIX           : out TRANSFORMATION_MATRIX_2);

procedure COMPOSE_MATRIX
  (MATRIX_A         : in  TRANSFORMATION_MATRIX_3;
   MATRIX_B         : in  TRANSFORMATION_MATRIX_3;
   ERROR_INDICATOR  : out ERROR_NUMBER;
   COMPOSED_MATRIX  : out TRANSFORMATION_MATRIX_3);

procedure COMPOSE_MATRIX
  (MATRIX_A         : in  TRANSFORMATION_MATRIX_2;
   MATRIX_B         : in  TRANSFORMATION_MATRIX_2;
   ERROR_INDICATOR  : out ERROR_NUMBER;
   COMPOSED_MATRIX  : out TRANSFORMATION_MATRIX_2);

procedure TRANSFORM_POINT
  (POINT            : in  MC.POINT_3;
   MATRIX           : in  TRANSFORMATION_MATRIX_3;
   ERROR_INDICATOR  : out ERROR_NUMBER;
   TRANSFORMED_POINT : out MC.POINT_3);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
procedure TRANSFORM_POINT
  (POINT                : in  MC.POINT_2;
   MATRIX               : in  TRANSFORMATION_MATRIX_2;
   ERROR_INDICATOR      : out ERROR_NUMBER;
   TRANSFORMED_POINT    : out MC.POINT_2);

procedure BUILD_TRANSFORMATION_MATRIX
  (FIXED_POINT          : in  MC.POINT_3;
   SHIFT_VECTOR         : in  MC.VECTOR_3;
   ANGLE_X              : in  ANGLE;
   ANGLE_Y              : in  ANGLE;
   ANGLE_Z              : in  ANGLE;
   SCALE_FACTORS        : in  MC.VECTOR_3;
   ERROR_INDICATOR      : out ERROR_NUMBER;
   MATRIX               : out TRANSFORMATION_MATRIX_3);

procedure BUILD_TRANSFORMATION_MATRIX
  (FIXED_POINT          : in  MC.POINT_2;
   SHIFT_VECTOR         : in  MC.VECTOR_2;
   ANGLE_Z              : in  ANGLE;
   SCALE_FACTORS        : in  MC.VECTOR_2;
   ERROR_INDICATOR      : out ERROR_NUMBER;
   MATRIX               : out TRANSFORMATION_MATRIX_2);

procedure COMPOSE_TRANSFORMATION_MATRIX
  (MATRIX               : in  TRANSFORMATION_MATRIX_3;
   FIXED_POINT          : in  MC.POINT_3;
   SHIFT_VECTOR         : in  MC.VECTOR_3;
   ANGLE_X              : in  ANGLE;
   ANGLE_Y              : in  ANGLE;
   ANGLE_Z              : in  ANGLE;
   SCALE_FACTORS        : in  MC.VECTOR_3;
   ERROR_INDICATOR      : out ERROR_NUMBER;
   COMPOSED_MATRIX      : out TRANSFORMATION_MATRIX_3);

procedure COMPOSE_TRANSFORMATION_MATRIX
  (MATRIX               : in  TRANSFORMATION_MATRIX_2;
   FIXED_POINT          : in  MC.POINT_2;
   SHIFT_VECTOR         : in  MC.VECTOR_2;
   ANGLE_Z              : in  ANGLE;
   SCALE_FACTORS        : in  MC.VECTOR_2;
   ERROR_INDICATOR      : out ERROR_NUMBER;
   COMPOSED_MATRIX      : out TRANSFORMATION_MATRIX_2);

procedure EVALUATE_VIEW_ORIENTATION_MATRIX
  (REFERENCE_POINT      : in  WC.POINT_3;
   NORMAL_VECTOR        : in  WC.VECTOR_3;
   UP_VECTOR            : in  WC.VECTOR_3;
   ERROR_INDICATOR      : out ERROR_NUMBER;
   ORIENTATION_MATRIX   : out TRANSFORMATION_MATRIX_3);

procedure EVALUATE_VIEW_ORIENTATION_MATRIX
  (REFERENCE_POINT      : in  WC.POINT_2;
   UP_VECTOR            : in  WC.VECTOR_2;
   ERROR_INDICATOR      : out ERROR_NUMBER;
   ORIENTATION_MATRIX   : out TRANSFORMATION_MATRIX_2);

procedure EVALUATE_VIEW_MAPPING_MATRIX
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
   (WINDOW_LIMITS          : in  VRC.RECTANGULAR_REGION_2;
    VIEWPORT_LIMITS        : in  NPC.RECTANGULAR_REGION_3;
    TYPE_OF_PROJECTION     : in  PROJECTION_TYPE;
    REFERENCE_POINT        : in  VRC.POINT_3;
    VIEW_DISTANCE          : in  VRC_TYPE;
    BACK_DISTANCE          : in  VRC_TYPE;
    FRONT_DISTANCE         : in  VRC_TYPE;
    ERROR_INDICATOR        : out ERROR_NUMBER;
    MAPPING_MATRIX         : out TRANSFORMATION_MATRIX_3);

procedure EVALUATE_VIEW_MAPPING_MATRIX
   (WINDOW_LIMITS     : in  VRC.RECTANGULAR_REGION_2;
    VIEWPORT_LIMITS   : in  NPC.RECTANGULAR_REGION_2;
    ERROR_INDICATOR   : out ERROR_NUMBER;
    MAPPING_MATRIX    : out TRANSFORMATION_MATRIX_2);

-- STRUCTURE MANIPULATION FUNCTIONS

procedure OPEN_STRUCTURE
   (STRUCTURE_IDENTIFIER : in STRUCTURE_ID);

procedure CLOSE_STRUCTURE;

procedure EXECUTE_STRUCTURE
   (STRUCTURE_IDENTIFIER : in STRUCTURE_ID);


procedure LABEL
   (LABEL_IDENTIFIER : in LABEL_ID);

procedure APPLICATION_DATA
   (DATA : in APPLICATION_DATA_RECORD);

procedure SET_EDIT_MODE
   (MODE : in EDIT_MODE);

procedure COPY_ALL_ELEMENTS_FROM_STRUCTURE
   (STRUCTURE_IDENTIFIER : in STRUCTURE_ID);

procedure SET_ELEMENT_POINTER
   (POSITION : in ELEMENT_POSITION);

procedure OFFSET_ELEMENT_POINTER
   (OFFSET : in ELEMENT_POSITION);

procedure SET_ELEMENT_POINTER_AT_LABEL
   (LABEL_IDENTIFIER : in LABEL_ID);

procedure DELETE_ELEMENT;

procedure DELETE_ELEMENT_RANGE
   (POSITION_1 : in ELEMENT_POSITION;
    POSITION_2 : in ELEMENT_POSITION);

procedure DELETE_ELEMENTS_BETWEEN_LABELS
   (LABEL_IDENTIFIER_1 : in LABEL_ID;
    LABEL_IDENTIFIER_2 : in LABEL_ID);

procedure EMPTY_STRUCTURE
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
  (STRUCTURE_IDENTIFIER : in STRUCTURE_ID);

procedure DELETE_STRUCTURE
  (STRUCTURE_IDENTIFIER : in STRUCTURE_ID);

procedure DELETE_STRUCTURE_NETWORK
  (STRUCTURE_IDENTIFIER : in STRUCTURE_ID;
   HANDLING_FLAG        : in REFERENCE_HANDLING_FLAG);

procedure DELETE_ALL_STRUCTURES;

procedure CHANGE_STRUCTURE_IDENTIFIER
  (ORIGINAL_IDENTIFIER   : in STRUCTURE_ID;
   RESULTING_IDENTIFIER  : in STRUCTURE_ID);

procedure CHANGE_STRUCTURE_REFERENCES
  (ORIGINAL_IDENTIFIER   : in STRUCTURE_ID;
   RESULTING_IDENTIFIER  : in STRUCTURE_ID);

procedure CHANGE_STRUCTURE_IDENTIFIER_AND_REFERENCES
  (ORIGINAL_IDENTIFIER   : in STRUCTURE_ID;
   RESULTING_IDENTIFIER  : in STRUCTURE_ID);

procedure POST_STRUCTURE
  (WS                    : in WS_ID;
   STRUCTURE_IDENTIFIER  : in STRUCTURE_ID;
   PRIORITY              : in DISPLAY_PRIORITY);

procedure UNPOST_STRUCTURE
  (WS                    : in WS_ID;
   STRUCTURE_IDENTIFIER  : in STRUCTURE_ID);

procedure UNPOST_ALL_STRUCTURES
  (WS : in WS_ID);

-- STRUCTURE ARCHIVING FUNCTIONS

procedure OPEN_ARCHIVE_FILE
  (ARCHIVE_IDENTIFIER : in ARCHIVE_ID;
   ARCHIVE_FILE       : in FILE_ID);

procedure CLOSE_ARCHIVE_FILE
  (ARCHIVE_IDENTIFIER : in ARCHIVE_ID);

procedure ARCHIVE_STRUCTURES
  (ARCHIVE_IDENTIFIER     : in ARCHIVE_ID;
   STRUCTURE_IDENTIFIERS  : in STRUCTURE_IDS.LIST_OF);

procedure ARCHIVE_STRUCTURE_NETWORKS
  (ARCHIVE_IDENTIFIER     : in ARCHIVE_ID;
   STRUCTURE_IDENTIFIERS  : in STRUCTURE_IDS.LIST_OF);

procedure ARCHIVE_ALL_STRUCTURES
  (ARCHIVE_IDENTIFIER : in ARCHIVE_ID);

procedure SET_CONFLICT_RESOLUTION
  (ARCHIVAL_CONFLICT   : in CONFLICT_RESOLUTION;
   RETRIEVAL_CONFLICT  : in CONFLICT_RESOLUTION);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
procedure RETRIEVE_STRUCTURE_IDENTIFIERS
  (ARCHIVE_IDENTIFIER      : in ARCHIVE_ID;
   STRUCTURE_IDENTIFIERS   : out STRUCTURE_IDS.LIST_OF);

procedure RETRIEVE_STRUCTURES
  (ARCHIVE_IDENTIFIER      : in ARCHIVE_ID;
   STRUCTURE_IDENTIFIERS   : in STRUCTURE_IDS.LIST_OF);

procedure RETRIEVE_STRUCTURE_NETWORKS
  (ARCHIVE_IDENTIFIER      : in ARCHIVE_ID;
   STRUCTURE_IDENTIFIERS   : in STRUCTURE_IDS.LIST_OF);

procedure RETRIEVE_ALL_STRUCTURES
  (ARCHIVE_IDENTIFIER : in ARCHIVE_ID);

procedure RETRIEVE_PATHS_TO_ANCESTORS
  (ARCHIVE_IDENTIFIER      : in ARCHIVE_ID;
   STRUCTURE_IDENTIFIER    : in STRUCTURE_ID;
   ORDER                   : in PATH_ORDER;
   DEPTH                   : in PATH_DEPTH;
   LIST_OF_PATHS           : out REFERENCE_PATHS.LIST_OF);

procedure RETRIEVE_PATHS_TO_DESCENDANTS
  (ARCHIVE_IDENTIFIER      : in ARCHIVE_ID;
   STRUCTURE_IDENTIFIER    : in STRUCTURE_ID;
   ORDER                   : in PATH_ORDER;
   DEPTH                   : in PATH_DEPTH;
   LIST_OF_PATHS           : out REFERENCE_PATHS.LIST_OF);

procedure DELETE_STRUCTURES_FROM_ARCHIVE
  (ARCHIVE_IDENTIFIER      : in ARCHIVE_ID;
   STRUCTURE_IDENTIFIERS   : in STRUCTURE_IDS.LIST_OF);

procedure DELETE_STRUCTURE_NETWORKS_FROM_ARCHIVE
  (ARCHIVE_IDENTIFIER      : in ARCHIVE_ID;
   STRUCTURE_IDENTIFIERS   : in STRUCTURE_IDS.LIST_OF);

procedure DELETE_ALL_STRUCTURES_FROM_ARCHIVE
  (ARCHIVE_IDENTIFIER : in ARCHIVE_ID);

-- INPUT FUNCTIONS

procedure SET_PICK_IDENTIFIER
  (PICK_IDENTIFIER : in PICK_ID);

procedure SET_PICK_FILTER
  (WS              : in WS_ID;
   PICK_DEVICE     : in PICK_DEVICE_NUMBER;
   PICKABILITY     : in NAME_SET_FILTER);

procedure INITIALIZE_LOCATOR
  (WS                  : in WS_ID;
   DEVICE              : in LOCATOR_DEVICE_NUMBER;
   INITIAL_VIEW_IND    : in VIEW_INDEX;
   INITIAL_POSITION    : in WC.POINT_3;
   ECHO_VOLUME         : in DC.RECTANGULAR_REGION_3;
   DATA_RECORD         : in LOCATOR_DATA_RECORD);

procedure INITIALIZE_LOCATOR
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
  (WS                 : in WS_ID;
   DEVICE             : in LOCATOR_DEVICE_NUMBER;
   INITIAL_VIEW_IND   : in VIEW_INDEX;
   INITIAL_POSITION   : in WC.POINT_2;
   ECHO_AREA          : in DC.RECTANGULAR_REGION_2;
   DATA_RECORD        : in LOCATOR_DATA_RECORD);

procedure INITIALIZE_STROKE
  (WS                 : in WS_ID;
   DEVICE             : in STROKE_DEVICE_NUMBER;
   INITIAL_VIEW_IND   : in VIEW_INDEX;
   INITIAL_STROKE     : in WC.POINT_LIST_3;
   ECHO_VOLUME        : in DC.RECTANGULAR_REGION_3;
   DATA_RECORD        : in STROKE_DATA_RECORD);

procedure INITIALIZE_STROKE
  (WS                 : in WS_ID;
   DEVICE             : in STROKE_DEVICE_NUMBER;
   INITIAL_VIEW_IND   : in VIEW_INDEX;
   INITIAL_STROKE     : in WC.POINT_LIST_2;
   ECHO_AREA          : in DC.RECTANGULAR_REGION_2;
   DATA_RECORD        : in STROKE_DATA_RECORD);

procedure INITIALIZE_VALUATOR
  (WS                 : in WS_ID;
   DEVICE             : in VALUATOR_DEVICE_NUMBER;
   INITIAL_VALUE      : in VALUATOR_VALUE;
   ECHO_VOLUME        : in DC.RECTANGULAR_REGION_3;
   DATA_RECORD        : in VALUATOR_DATA_RECORD);

procedure INITIALIZE_VALUATOR
  (WS                 : in WS_ID;
   DEVICE             : in VALUATOR_DEVICE_NUMBER;
   INITIAL_VALUE      : in VALUATOR_VALUE;
   ECHO_AREA          : in DC.RECTANGULAR_REGION_2;
   DATA_RECORD        : in VALUATOR_DATA_RECORD);

procedure INITIALIZE_CHOICE
  (WS                 : in WS_ID;
   DEVICE             : in CHOICE_DEVICE_NUMBER;
   INITIAL_STATUS     : in CHOICE_STATUS;
   INITIAL_CHOICE     : in CHOICE_NUMBER;
   ECHO_VOLUME        : in DC.RECTANGULAR_REGION_3;
   DATA_RECORD        : in CHOICE_DATA_RECORD);

procedure INITIALIZE_CHOICE
  (WS                 : in WS_ID;
   DEVICE             : in CHOICE_DEVICE_NUMBER;
   INITIAL_STATUS     : in CHOICE_STATUS;
   INITIAL_CHOICE     : in CHOICE_NUMBER;
   ECHO_AREA          : in DC.RECTANGULAR_REGION_2;
   DATA_RECORD        : in CHOICE_DATA_RECORD);

procedure INITIALIZE_PICK
  (WS                 : in WS_ID;
   DEVICE             : in PICK_DEVICE_NUMBER;
   INITIAL_STATUS     : in PICK_STATUS;
   INITIAL_PATH       : in PICK_PATH;
   ECHO_VOLUME        : in DC.RECTANGULAR_REGION_3;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
    DATA_RECORD  : in PICK_DATA_RECORD;
    ORDER        : in PATH_ORDER);

 procedure INITIALIZE_PICK
   (WS             : in WS_ID;
    DEVICE         : in PICK_DEVICE_NUMBER;
    INITIAL_STATUS : in PICK_STATUS;
    INITIAL_PATH   : in PICK_PATH;
    ECHO_AREA      : in DC.RECTANGULAR_REGION_2;
    DATA_RECORD    : in PICK_DATA_RECORD;
    ORDER          : in PATH_ORDER);

 procedure INITIALIZE_STRING
   (WS             : in WS_ID;
    DEVICE         : in STRING_DEVICE_NUMBER;
    INITIAL_STRING : in PHIGS_STRING;
    ECHO_VOLUME    : in DC.RECTANGULAR_REGION_3;
    DATA_RECORD    : in STRING_DATA_RECORD);

 procedure INITIALIZE_STRING
   (WS             : in WS_ID;
    DEVICE         : in STRING_DEVICE_NUMBER;
    INITIAL_STRING : in PHIGS_STRING;
    ECHO_AREA      : in DC.RECTANGULAR_REGION_2;
    DATA_RECORD    : in STRING_DATA_RECORD);

 procedure SET_LOCATOR_MODE
   (WS     : in WS_ID;
    DEVICE : in LOCATOR_DEVICE_NUMBER;
    MODE   : in OPERATING_MODE;
    SWITCH : in ECHO_SWITCH);

 procedure SET_STROKE_MODE
   (WS     : in WS_ID;
    DEVICE : in STROKE_DEVICE_NUMBER;
    MODE   : in OPERATING_MODE;
    SWITCH : in ECHO_SWITCH);

 procedure SET_VALUATOR_MODE
   (WS     : in WS_ID;
    DEVICE : in VALUATOR_DEVICE_NUMBER;
    MODE   : in OPERATING_MODE;
    SWITCH : in ECHO_SWITCH);

 procedure SET_CHOICE_MODE
   (WS     : in WS_ID;
    DEVICE : in CHOICE_DEVICE_NUMBER;
    MODE   : in OPERATING_MODE;
    SWITCH : in ECHO_SWITCH);

 procedure SET_PICK_MODE
   (WS     : in WS_ID;
    DEVICE : in PICK_DEVICE_NUMBER;
    MODE   : in OPERATING_MODE;
    SWITCH : in ECHO_SWITCH);

 procedure SET_STRING_MODE
   (WS     : in WS_ID;
    DEVICE : in STRING_DEVICE_NUMBER;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
    MODE    : in OPERATING_MODE;
    SWTICH  : in ECHO_SWITCH);

procedure REQUEST_LOCATOR
  (WS         : in  WS_ID;
   DEVICE     : in  LOCATOR_DEVICE_NUMBER;
   STATUS     : out INPUT_STATUS;
   VIEW_IND   : out VIEW_INDEX;
   POSITION   : out WC.POINT_3);

procedure REQUEST_LOCATOR
  (WS         : in  WS_ID;
   DEVICE     : in  LOCATOR_DEVICE_NUMBER;
   STATUS     : out INPUT_STATUS;
   VIEW_IND   : out VIEW_INDEX;
   POSITION   : out WC.POINT_2);

procedure REQUEST_STROKE
  (WS                : in  WS_ID;
   DEVICE            : in  STROKE_DEVICE_NUMBER;
   STATUS            : out INPUT_STATUS;
   VIEW_IND          : out VIEW_INDEX;
   STROKE_POINTS     : out WC.ACCESS_POINT_LIST_3);

procedure REQUEST_STROKE
  (WS                : in  WS_ID;
   DEVICE            : in  STROKE_DEVICE_NUMBER;
   STATUS            : out INPUT_STATUS;
   VIEW_IND          : out VIEW_INDEX;
   STROKE_POINTS     : out WC.ACCESS_POINT_LIST_2);

procedure REQUEST_VALUATOR
  (WS       : in  WS_ID;
   DEVICE   : in  VALUATOR_DEVICE_NUMBER;
   STATUS   : out INPUT_STATUS;
   VALUE    : out VALUATOR_VALUE);

procedure REQUEST_CHOICE
  (WS       : in  WS_ID;
   DEVICE   : in  CHOICE_DEVICE_NUMBER;
   STATUS   : out CHOICE_REQUEST_STATUS;
   CHOICE   : out CHOICE_NUMBER);

procedure REQUEST_PICK
  (WS                  : in  WS_ID;
   DEVICE              : in  PICK_DEVICE_NUMBER;
   DEPTH_TO_RETURN     : in  PATH_DEPTH;
   STATUS              : out PICK_REQUEST_STATUS;
   PATH                : out PICK_PATH);

procedure REQUEST_STRING
  (WS              : in  WS_ID;
   DEVICE          : in  STRING_DEVICE_NUMBER;
   STATUS          : out INPUT_STATUS;
   CHAR_STRING     : out INPUT_STRING);

procedure SAMPLE_LOCATOR
  (WS         : in  WS_ID;
   DEVICE     : in  LOCATOR_DEVICE_NUMBER;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
    VIEW_IND  : out VIEW_INDEX;
    POSITION  : out WC.POINT_3);

procedure SAMPLE_LOCATOR
  (WS         : in  WS_ID;
   DEVICE     : in  LOCATOR_DEVICE_NUMBER;
   VIEW_IND   : out VIEW_INDEX;
   POSITION   : out WC.POINT_2);

procedure SAMPLE_STROKE
  (WS                : in  WS_ID;
   DEVICE            : in  STROKE_DEVICE_NUMBER;
   VIEW_IND          : out VIEW_INDEX;
   STROKE_POINTS     : out WC.ACCESS_POINT_LIST_3);

procedure SAMPLE_STROKE
  (WS                : in  WS_ID;
   DEVICE            : in  STROKE_DEVICE_NUMBER;
   VIEW_IND          : out VIEW_INDEX;
   STROKE_POINTS     : out WC.ACCESS_POINT_LIST_2);

procedure SAMPLE_VALUATOR
  (WS       : in  WS_ID;
   DEVICE   : in  VALUATOR_DEVICE_NUMBER;
   VALUE    : out VALUATOR_VALUE);

procedure SAMPLE_CHOICE
  (WS       : in  WS_ID;
   DEVICE   : in  CHOICE_DEVICE_NUMBER;
   STATUS   : out CHOICE_STATUS;
   CHOICE   : out CHOICE_NUMBER);

procedure SAMPLE_PICK
  (WS                : in  WS_ID;
   DEVICE            : in  PICK_DEVICE_NUMBER;
   DEPTH_TO_RETURN   : in  PATH_DEPTH;
   STATUS            : out PICK_STATUS;
   PATH              : out PICK_PATH);

procedure SAMPLE_STRING
  (WS               : in  WS_ID;
   DEVICE           : in  STRING_DEVICE_NUMBER;
   CHAR_STRING      : out INPUT_STRING);

procedure AWAIT_EVENT
  (TIMEOUT : in  DURATION;
   WS      : out WS_ID;
   DEVICE  : out EVENT_DEVICE_NUMBER);

procedure FLUSH_DEVICE_EVENTS
  (WS       : in WS_ID;
   DEVICE   : in EVENT_QUEUE_DEVICE_NUMBER);

procedure GET_LOCATOR
  (VIEW_IND : out VIEW_INDEX;
   POSITION  : out WC.POINT_3);

procedure GET_LOCATOR
  (VIEW_IND : out VIEW_INDEX;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
    POSITION   : out WC.POINT_2);

procedure GET_STROKE
  (VIEW_IND        : out VIEW_INDEX;
   STROKE_POINTS   : out WC.ACCESS_POINT_LIST_3);

procedure GET_STROKE
  (VIEW_IND        : out VIEW_INDEX;
   STROKE_POINTS   : out WC.ACCESS_POINT_LIST_2);

procedure GET_VALUATOR
  (VALUE : out VALUATOR_VALUE);

procedure GET_CHOICE
  (STATUS  : out CHOICE_STATUS;
   CHOICE  : out CHOICE_NUMBER);

procedure GET_PICK
  (DEPTH_TO_RETURN : in  PATH_DEPTH;
   STATUS          : out PICK_STATUS;
   PATH            : out PICK_PATH);

procedure GET_STRING
  (CHAR_STRING : out INPUT_STRING);

-- METAFILE FUNCTIONS

procedure WRITE_ITEM_TO_METAFILE
  (WS    : in WS_ID;
   ITEM  : in METAFILE_DATA_RECORD);

procedure GET_ITEM_TYPE_FROM_METAFILE
  (WS              : in  WS_ID;
   TYPE_OF_ITEM    : out METAFILE_ITEM_TYPE;
   LENGTH          : out METAFILE_ITEM_LENGTH);

procedure READ_ITEM_FROM_METAFILE
  (WS              : in  WS_ID;
   MAX_LENGTH      : in  METAFILE_ITEM_LENGTH;
   ITEM            : out METAFILE_DATA_RECORD);

procedure INTERPRET_ITEM
  (ITEM : in METAFILE_DATA_RECORD);

-- INQUIRY FUNCTIONS

procedure INQ_SYSTEM_STATE_VALUE
  (STATE_VALUE : out SYSTEM_STATE);

procedure INQ_WS_STATE_VALUE
  (STATE_VALUE : out WS_STATE);

procedure INQ_STRUCTURE_STATE_VALUE
  (STATE_VALUE : out STRUCTURE_STATE);

procedure INQ_ARCHIVE_STATE_VALUE
  (STATE_VALUE : out ARCHIVE_STATE);

procedure INQ_LIST_OF_AVAILABLE_WS_TYPES
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
   (ERROR_INDICATOR : out ERROR_NUMBER;
    LIST_OF_TYPES   : out WS_TYPES.LIST_OF);

 procedure INQ_PHIGS_FACILITIES
   (ERROR_INDICATOR                : out ERROR_NUMBER;
    MAX_SIMUL_OPEN_WS              : out PHIGS_POSITIVE;
    MAX_SIMUL_OPEN_ARCHIVES        : out PHIGS_POSITIVE;
    NUMBER_AVAIL_NAMES             : out PHIGS_POSITIVE;
    AVAIL_CHAR_SETS                : out CHAR_SETS.LIST_OF;
    MAX_ISS_NORMAL_FILTER_LIST     : out PHIGS_POSITIVE;
    MAX_ISS_INVERTED_FILTER_LIST   : out PHIGS_POSITIVE);

 procedure INQ_GSE_FACILITIES
   (ERROR_INDICATOR        : out ERROR_NUMBER;
    LIST_AVAIL_GSES        : out GSE_IDS.LIST_OF;
    LIST_WS_DEPENDENCIES   : out WS_DEPENDENCIES.LIST_OF);

 procedure INQ_MODELLING_CLIPPING_FACILITIES
   (ERROR_INDICATOR          : out ERROR_NUMBER;
    NUMBER_DISTINCT_PLANES   : out PHIGS_POSITIVE;
    LIST_OF_OPERATIONS       : out MODELLING_CLIP_OPERATION_TYPES.LIST_OF);

 procedure INQ_EDIT_MODE
   (ERROR_INDICATOR : out ERROR_NUMBER;
    MODE            : out EDIT_MODE);

 procedure INQ_SET_OF_OPEN_WS
   (ERROR_INDICATOR : out ERROR_NUMBER;
    OPEN_WS         : out WS_IDS.LIST_OF);

 procedure INQ_STRUCTURE_IDENTIFIERS
   (ERROR_INDICATOR    : out ERROR_NUMBER;
    LIST_OF_STRUCTURES : out STRUCTURE_IDS.LIST_OF);

 procedure INQ_ARCHIVE_FILES
   (ERROR_INDICATOR          : out ERROR_NUMBER;
    LIST_ARCHIVE_IDENTIFIERS : out ARCHIVE_IDS.LIST_OF;
    LIST_ARCHIVE_FILES       : out VARIABLE_FILE_IDS.LIST_OF);

 procedure INQ_CONFLICT_RESOLUTION
   (ERROR_INDICATOR     : out ERROR_NUMBER;
    ARCHIVAL_CONFLICT   : out CONFLICT_RESOLUTION;
    RETRIEVAL_CONFLICT  : out CONFLICT_RESOLUTION);

 procedure INQ_ALL_CONFLICTING_STRUCTURES
   (ARCHIVE_IDENTIFIER  : in  ARCHIVE_ID;
    ERROR_INDICATOR     : out ERROR_NUMBER;
    LIST_OF_STRUCTURES  : out STRUCTURE_IDS.LIST_OF);

 procedure INQ_CONFLICTING_STRUCTURES_IN_NETWORK
   (ARCHIVE_IDENTIFIER    : in  ARCHIVE_ID;
    STRUCTURE_IDENTIFIER  : in  STRUCTURE_ID;
    SOURCE                : in  STRUCTURE_NETWORK_SOURCE;
    ERROR_INDICATOR       : out ERROR_NUMBER;
    LIST_OF_STRUCTURES    : out STRUCTURE_IDS.LIST_OF);

 procedure INQ_MORE_SIMULTANEOUS_EVENTS
   (ERROR_INDICATOR : out ERROR_NUMBER;
    EVENTS          : out MORE_EVENTS);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
procedure INQ_WS_CONNECTION_AND_TYPE
  (WS                 : in  WS_ID;
   ERROR_INDICATOR    : out ERROR_NUMBER;
   CONNECTION         : out VARIABLE_CONNECTION_ID;
   TYPE_OF_WS         : out WS_TYPE);

procedure INQ_LIST_OF_VIEW_INDICES
  (WS                  : in  WS_ID;
   ERROR_INDICATOR     : out ERROR_NUMBER;
   DEFINED_VIEW_LIST   : out VIEW_INDICES.LIST_OF);

procedure INQ_VIEW_REPRESENTATION
  (WS                                 : in  WS_ID;
   VIEW_IND                           : in  VIEW_INDEX;
   ERROR_INDICATOR                    : out ERROR_NUMBER;
   UPDATE                             : out UPDATE_STATE;
   REQUESTED_ORIENTATION_MATRIX       : out TRANSFORMATION_MATRIX_3;
   CURRENT_ORIENTATION_MATRIX         : out TRANSFORMATION_MATRIX_3;
   REQUESTED_MAPPING_MATRIX           : out TRANSFORMATION_MATRIX_3;
   CURRENT_MAPPING_MATRIX             : out TRANSFORMATION_MATRIX_3;
   REQUESTED_CLIPPING_LIMITS          : out NPC.RECTANGULAR_REGION_3;
   CURRENT_CLIPPING_LIMITS            : out NPC.RECTANGULAR_REGION_3;
   REQUESTED_XY_CLIPPING              : out CLIPPING_INDICATOR;
   CURRENT_XY_CLIPPING                : out CLIPPING_INDICATOR;
   REQUESTED_BACK_CLIPPING            : out CLIPPING_INDICATOR;
   CURRENT_BACK_CLIPPING              : out CLIPPING_INDICATOR;
   REQUESTED_FRONT_CLIPPING           : out CLIPPING_INDICATOR;
   CURRENT_FRONT_CLIPPING             : out CLIPPING_INDICATOR);


procedure INQ_HLHSR_MODE
  (WS                 : in  WS_ID;
   ERROR_INDICATOR    : out ERROR_NUMBER;
   UPDATE             : out UPDATE_STATE;
   CURRENT_MODE       : out HLHSR_MODE;
   REQUESTED_MODE     : out HLHSR_MODE);

procedure INQ_POSTED_STRUCTURES
  (WS                   : in  WS_ID;
   ERROR_INDICATOR      : out ERROR_NUMBER;
   LIST_OF_STRUCTURES   : out POSTED_STRUCTURES.LIST_OF);

procedure INQ_DISPLAY_UPDATE_STATE
  (WS                 : in  WS_ID;
   ERROR_INDICATOR    : out ERROR_NUMBER;
   DEFERRAL           : out DEFERRAL_MODE;
   MODIFICATION       : out MODIFICATION_MODE;
   EMPTINESS          : out DISPLAY_SURFACE_EMPTY;
   VISUAL_STATE       : out VISUAL_REPRESENTATION_STATE);

procedure INQ_LIST_OF_POLYLINE_INDICES
  (WS                 : in  WS_ID;
   ERROR_INDICATOR    : out ERROR_NUMBER;
   INDICES            : out POLYLINE_INDICES.LIST_OF);

procedure INQ_POLYLINE_REPRESENTATION
  (WS                  : in  WS_ID;
   POLYLINE_IND        : in  POLYLINE_INDEX;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
    RETURNED_VALUES : in  RETURN_VALUE_TYPE;
    ERROR_INDICATOR : out ERROR_NUMBER;
    TYPE_OF_LINE    : out LINETYPE;
    LINEWIDTH_SF    : out LINEWIDTH;
    LINE_COLOUR     : out COLOUR_INDEX);

procedure INQ_LIST_OF_POLYMARKER_INDICES
  (WS               : in  WS_ID;
   ERROR_INDICATOR  : out ERROR_NUMBER;
   INDICES          : out POLYMARKER_INDICES.LIST_OF);

procedure INQ_POLYMARKER_REPRESENTATION
  (WS               : in  WS_ID;
   POLYMARKER_IND   : in  POLYMARKER_INDEX;
   RETURNED_VALUES  : in  RETURN_VALUE_TYPE;
   ERROR_INDICATOR  : out ERROR_NUMBER;
   TYPE_OF_MARKER   : out MARKER_TYPE;
   SIZE             : out MARKER_SIZE;
   MARKER_COLOUR    : out COLOUR_INDEX);

procedure INQ_LIST_OF_TEXT_INDICES
  (WS               : in  WS_ID;
   ERROR_INDICATOR  : out ERROR_NUMBER;
   INDICES          : out TEXT_INDICES.LIST_OF);

procedure INQ_TEXT_REPRESENTATION
  (WS               : in  WS_ID;
   TEXT_IND         : in  TEXT_INDEX;
   RETURNED_VALUES  : in  RETURN_VALUE_TYPE;
   ERROR_INDICATOR  : out ERROR_NUMBER;
   FONT             : out TEXT_FONT;
   PRECISION        : out TEXT_PRECISION;
   EXPANSION        : out CHAR_EXPANSION;
   SPACING          : out CHAR_SPACING;
   TEXT_COLOUR      : out COLOUR_INDEX);

procedure INQ_LIST_OF_INTERIOR_INDICES
  (WS               : in  WS_ID;
   ERROR_INDICATOR  : out ERROR_NUMBER;
   INDICES          : out INTERIOR_INDICES.LIST_OF);

procedure INQ_INTERIOR_REPRESENTATION
  (WS               : in  WS_ID;
   INTERIOR_IND     : in  INTERIOR_INDEX;
   RETURNED_VALUES  : in  RETURN_VALUE_TYPE;
   ERROR_INDICATOR  : out ERROR_NUMBER;
   STYLE_OF_INTERIOR: out INTERIOR_STYLE;
   STYLE_IND        : out STYLE_INDEX;
   INTERIOR_COLOUR  : out COLOUR_INDEX);

procedure INQ_LIST_OF_EDGE_INDICES
  (WS               : in  WS_ID;
   ERROR_INDICATOR  : out ERROR_NUMBER;
   INDICES          : out EDGE_INDICES.LIST_OF);

procedure INQ_EDGE_REPRESENTATION
  (WS               : in  WS_ID;
   EDGE_IND         : in  EDGE_INDEX;
   RETURNED_VALUES  : in  RETURN_VALUE_TYPE;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
    ERROR_INDICATOR   : out ERROR_NUMBER;
    FLAG              : out EDGE_FLAG;
    TYPE_OF_EDGE      : out EDGETYPE;
    EDGEWIDTH_SF      : out EDGEWIDTH;
    EDGE_COLOUR       : out COLOUR_INDEX);

procedure INQ_LIST_OF_PATTERN_INDICES
  (WS                 : in  WS_ID;
   ERROR_INDICATOR    : out ERROR_NUMBER;
   INDICES            : out PATTERN_INDICES.LIST_OF);

procedure INQ_PATTERN_REPRESENTATION
  (WS                 : in  WS_ID;
   PATTERN_IND        : in  PATTERN_INDEX;
   RETURNED_VALUES    : in  RETURN_VALUE_TYPE;
   ERROR_INDICATOR    : out ERROR_NUMBER;
   PATTERN            : out ACCESS_COLOUR_MATRIX);


procedure INQ_COLOUR_MODEL
  (WS                 : in  WS_ID;
   ERROR_INDICATOR    : out ERROR_NUMBER;
   MODEL              : out COLOUR_MODEL);

procedure INQ_LIST_OF_COLOUR_INDICES
  (WS                 : in  WS_ID;
   ERROR_INDICATOR    : out ERROR_NUMBER;
   INDICES            : out COLOUR_INDICES.LIST_OF);

procedure INQ_COLOUR_REPRESENTATION
  (WS                 : in  WS_ID;
   COLOUR_IND         : in  COLOUR_INDEX;
   RETURNED_VALUES    : in  RETURN_VALUE_TYPE;
   ERROR_INDICATOR    : out ERROR_NUMBER;
   COLOUR             : out COLOUR_REPRESENTATION);

procedure INQ_HIGHLIGHTING_FILTER
  (WS                 : in  WS_ID;
   ERROR_INDICATOR    : out ERROR_NUMBER;
   HIGHLIGHTING       : out NAME_SET_FILTER);

procedure INQ_INVISIBILITY_FILTER
  (WS                 : in  WS_ID;
   ERROR_INDICATOR    : out ERROR_NUMBER;
   INVISIBILITY       : out NAME_SET_FILTER);

procedure INQ_WS_TRANSFORMATION
  (WS                              : in  WS_ID;
   ERROR_INDICATOR                 : out ERROR_NUMBER;
   UPDATE                          : out UPDATE_STATE;
   REQUESTED_WINDOW_LIMITS         : out NPC.RECTANGULAR_REGION_3;
   CURRENT_WINDOW_LIMITS           : out NPC.RECTANGULAR_REGION_3;
   REQUESTED_VIEWPORT_LIMITS       : out DC.RECTANGULAR_REGION_3;
   CURRENT_VIEWPORT_LIMITS         : out DC.RECTANGULAR_REGION_3);

procedure INQ_WS_TRANSFORMATION
  (WS                              : in  WS_ID;
   ERROR_INDICATOR                 : out ERROR_NUMBER;
   UPDATE                          : out UPDATE_STATE;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

9593 PT*3 90 ■ 4851903 0575600 238 ■


```
    REQUESTED_WINDOW_LIMITS     : out NPC.RECTANGULAR_REGION_2;
    CURRENT_WINDOW_LIMITS       : out NPC.RECTANGULAR_REGION_2;
    REQUESTED_VIEWPORT_LIMITS   : out DC.RECTANGULAR_REGION_2;
    CURRENT_VIEWPORT_LIMITS     : out DC.RECTANGULAR_REGION_2);

procedure INQ_LOCATOR_DEVICE_STATE
  (WS                   : in  WS_ID;
   DEVICE               : in  LOCATOR_DEVICE_NUMBER;
   RETURNED_VALUES      : in  RETURN_VALUE_TYPE;
   ERROR_INDICATOR      : out ERROR_NUMBER;
   MODE                 : out OPERATING_MODE;
   SWITCH               : out ECHO_SWITCH;
   INITIAL_VIEW_IND     : out VIEW_INDEX;
   INITIAL_POSITION     : out WC.POINT_3;
   ECHO_VOLUME          : out DC.RECTANGULAR_REGION_3;
   DATA_RECORD          : out LOCATOR_DATA_RECORD);

procedure INQ_LOCATOR_DEVICE_STATE
  (WS                   : in  WS_ID;
   DEVICE               : in  LOCATOR_DEVICE_NUMBER;
   RETURNED_VALUES      : in  RETURN_VALUE_TYPE;
   ERROR_INDICATOR      : out ERROR_NUMBER;
   MODE                 : out OPERATING_MODE;
   SWITCH               : out ECHO_SWITCH;
   INITIAL_VIEW_IND     : out VIEW_INDEX;
   INITIAL_POSITION     : out WC.POINT_2;
   ECHO_AREA            : out DC.RECTANGULAR_REGION_2;
   DATA_RECORD          : out LOCATOR_DATA_RECORD);

procedure INQ_STROKE_DEVICE_STATE
  (WS                       : in  WS_ID;
   DEVICE                   : in  STROKE_DEVICE_NUMBER;
   RETURNED_VALUES          : in  RETURN_VALUE_TYPE;
   ERROR_INDICATOR          : out ERROR_NUMBER;
   MODE                     : out OPERATING_MODE;
   SWITCH                   : out ECHO_SWITCH;
   INITIAL_VIEW_IND         : out VIEW_INDEX;
   INITIAL_STROKE_POINTS    : out WC.ACCESS_POINT_LIST_3;
   ECHO_VOLUME              : out DC.RECTANGULAR_REGION_3;
   DATA_RECORD              : out STROKE_DATA_RECORD);

procedure INQ_STROKE_DEVICE_STATE
  (WS                       : in  WS_ID;
   DEVICE                   : in  STROKE_DEVICE_NUMBER;
   RETURNED_VALUES          : in  RETURN_VALUE_TYPE;
   ERROR_INDICATOR          : out ERROR_NUMBER;
   MODE                     : out OPERATING_MODE;
   SWITCH                   : out ECHO_SWITCH;
   INITIAL_VIEW_IND         : out VIEW_INDEX;
   INITIAL_STROKE_POINTS    : out WC.ACCESS_POINT_LIST_2;
   ECHO_AREA                : out DC.RECTANGULAR_REGION_2;
   DATA_RECORD              : out STROKE_DATA_RECORD);

procedure INQ_VALUATOR_DEVICE_STATE
  (WS                   : in  WS_ID;
   DEVICE               : in  VALUATOR_DEVICE_NUMBER;
   ERROR_INDICATOR      : out ERROR_NUMBER;
   MODE                 : out OPERATING_MODE;
   SWITCH               : out ECHO_SWITCH;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
    INITIAL_VALUE       : out VALUATOR_VALUE;
    ECHO_VOLUME         : out DC.RECTANGULAR_REGION_3;
    DATA_RECORD         : out VALUATOR_DATA_RECORD);

procedure INQ_VALUATOR_DEVICE_STATE
  (WS                   : in  WS_ID;
    DEVICE              : in  VALUATOR_DEVICE_NUMBER;
    ERROR_INDICATOR     : out ERROR_NUMBER;
    MODE                : out OPERATING_MODE;
    SWITCH              : out ECHO_SWITCH;
    INITIAL_VALUE       : out VALUATOR_VALUE;
    ECHO_AREA           : out DC.RECTANGULAR_REGION_2;
    DATA_RECORD         : out VALUATOR_DATA_RECORD);

procedure INQ_CHOICE_DEVICE_STATE
  (WS                   : in  WS_ID;
    DEVICE              : in  CHOICE_DEVICE_NUMBER;
    ERROR_INDICATOR     : out ERROR_NUMBER;
    MODE                : out OPERATING_MODE;
    SWITCH              : out ECHO_SWITCH;
    INITIAL_STATUS      : out CHOICE_STATUS;
    INITIAL_CHOICE      : out CHOICE_NUMBER;
    ECHO_VOLUME         : out DC.RECTANGULAR_REGION_3;
    DATA_RECORD         : out CHOICE_DATA_RECORD);

procedure INQ_CHOICE_DEVICE_STATE
  (WS                   : in  WS_ID;
    DEVICE              : in  CHOICE_DEVICE_NUMBER;
    ERROR_INDICATOR     : out ERROR_NUMBER;
    MODE                : out OPERATING_MODE;
    SWITCH              : out ECHO_SWITCH;
    INITIAL_STATUS      : out CHOICE_STATUS;
    INITIAL_CHOICE      : out CHOICE_NUMBER;
    ECHO_AREA           : out DC.RECTANGULAR_REGION_2;
    DATA_RECORD         : out CHOICE_DATA_RECORD);

procedure INQ_PICK_DEVICE_STATE
  (WS                   : in  WS_ID;
    DEVICE              : in  PICK_DEVICE_NUMBER;
    RETURNED_VALUES     : in  RETURN_VALUE_TYPE;
    ERROR_INDICATOR     : out ERROR_NUMBER;
    MODE                : out OPERATING_MODE;
    SWITCH              : out ECHO_SWITCH;
    PICKABILITY         : out NAME_SET_FILTER;
    INITIAL_STATUS      : out PICK_STATUS;
    INITIAL_PATH        : out PICK_PATH;
    ECHO_VOLUME         : out DC.RECTANGULAR_REGION_3;
    DATA_RECORD         : out PICK_DATA_RECORD;
    ORDER               : out PATH_ORDER);

procedure INQ_PICK_DEVICE_STATE
  (WS                   : in  WS_ID;
    DEVICE              : in  PICK_DEVICE_NUMBER;
    RETURNED_VALUES     : in  RETURN_VALUE_TYPE;
    ERROR_INDICATOR     : out ERROR_NUMBER;
    MODE                : out OPERATING_MODE;
    SWITCH              : out ECHO_SWITCH;
    PICKABILITY         : out NAME_SET_FILTER;
    INITIAL_STATUS      : out PICK_STATUS;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

9593 PT*3 90 ■ 4851903 0575602 000 ■


```
    INITIAL_PATH          : out PICK_PATH;
    ECHO_AREA             : out DC.RECTANGULAR_REGION_2;
    DATA_RECORD           : out PICK_DATA_RECORD;
    ORDER                 : out PATH_ORDER);

procedure INQ_STRING_DEVICE_STATE
   (WS                    : in  WS_ID;
    DEVICE                : in  STRING_DEVICE_NUMBER;
    ERROR_INDICATOR       : out ERROR_NUMBER;
    MODE                  : out OPERATING_MODE;
    SWITCH                : out ECHO_SWITCH;
    INITIAL_CHAR_STRING   : out INPUT_STRING;
    ECHO_VOLUME           : out DC.RECTANGULAR_REGION_3;
    DATA_RECORD           : out STRING_DATA_RECORD);

procedure INQ_STRING_DEVICE_STATE
   (WS                    : in  WS_ID;
    DEVICE                : in  STRING_DEVICE_NUMBER;
    ERROR_INDICATOR       : out ERROR_NUMBER;
    MODE                  : out OPERATING_MODE;
    SWITCH                : out ECHO_SWITCH;
    INITIAL_CHAR_STRING   : out INPUT_STRING;
    ECHO_AREA             : out DC.RECTANGULAR_REGION_2;
    DATA_RECORD           : out STRING_DATA_RECORD);

procedure INQ_WS_CATEGORY
   (TYPE_OF_WS           : in  WS_TYPE;
    ERROR_INDICATOR      : out ERROR_NUMBER;
    CATEGORY             : out WS_CATEGORY);

procedure INQ_DISPLAY_SPACE_SIZE
   (TYPE_OF_WS           : in  WS_TYPE;
    ERROR_INDICATOR      : out ERROR_NUMBER;
    UNITS                : out DC_UNITS;
    MAX_DC_UNIT_SIZE     : out DC.SIZE_3;
    MAX_RASTER_UNIT_SIZE : out RASTER_UNIT_SIZE_3);

procedure INQ_DISPLAY_SPACE_SIZE
   (TYPE_OF_WS           : in  WS_TYPE;
    ERROR_INDICATOR      : out ERROR_NUMBER;
    UNITS                : out DC_UNITS;
    MAX_DC_UNIT_SIZE     : out DC.SIZE_2;
    MAX_RASTER_UNIT_SIZE : out RASTER_UNIT_SIZE_2);

procedure INQ_HLHSR_FACILITIES
   (TYPE_OF_WS           : in  WS_TYPE;
    ERROR_INDICATOR      : out ERROR_NUMBER;
    LIST_OF_IDENTIFIERS  : out HLHSR_IDS.LIST_OF;
    LIST_OF_MODES        : out HLHSR_MODES.LIST_OF);

procedure INQ_VIEW_FACILITIES
   (TYPE_OF_WS           : in  WS_TYPE;
    ERROR_INDICATOR      : out ERROR_NUMBER;
    NUMBER_OF_INDICES    : out PHIGS_POSITIVE);

procedure INQ_PREDEFINED_VIEW_REPRESENTATION
   (TYPE_OF_WS           : in  WS_TYPE;
    VIEW_IND             : in  VIEW_INDEX;
    ERROR_INDICATOR      : out ERROR_NUMBER;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
    ORIENTATION_MATRIX  : out TRANSFORMATION_MATRIX_3;
    MAPPING_MATRIX      : out TRANSFORMATION_MATRIX_3;
    CLIPPING_LIMITS     : out NPC.RECTANGULAR_REGION_3;
    XY_CLIPPING         : out CLIPPING_INDICATOR;
    BACK_CLIPPING       : out CLIPPING_INDICATOR;
    FRONT_CLIPPING      : out CLIPPING_INDICATOR);

procedure INQ_WS_CLASSIFICATION
  (TYPE_OF_WS           : in  WS_TYPE;
   ERROR_INDICATOR      : out ERROR_NUMBER;
   CLASS                : out DISPLAY_CLASS);

procedure INQ_DYNAMICS_OF_WS_ATTRIBUTES
  (TYPE_OF_WS                    : in  WS_TYPE;
   ERROR_INDICATOR               : out ERROR_NUMBER;
   POLYLINE_REPRESENTATION       : out DYNAMIC_MODIFICATION;
   POLYMARKER_REPRESENTATION     : out DYNAMIC_MODIFICATION;
   TEXT_REPRESENTATION           : out DYNAMIC_MODIFICATION;
   INTERIOR_REPRESENTATION       : out DYNAMIC_MODIFICATION;
   EDGE_REPRESENTATION           : out DYNAMIC_MODIFICATION;
   PATTERN_REPRESENTATION        : out DYNAMIC_MODIFICATION;
   COLOUR_REPRESENTATION         : out DYNAMIC_MODIFICATION;
   VIEW_REPRESENTATION           : out DYNAMIC_MODIFICATION;
   WS_TRANSFORMATION             : out DYNAMIC_MODIFICATION;
   HIGHLIGHTING_FILTER           : out DYNAMIC_MODIFICATION;
   INVISIBILITY_FILTER           : out DYNAMIC_MODIFICATION;
   HLHSR_MODE_CHANGE             : out DYNAMIC_MODIFICATION);

procedure INQ_DEFAULT_DISPLAY_UPDATE_STATE
  (TYPE_OF_WS           : in  WS_TYPE;
   ERROR_INDICATOR      : out ERROR_NUMBER;
   DEFERRAL             : out DEFERRAL_MODE;
   MODIFICATION         : out MODIFICATION_MODE);

procedure INQ_POLYLINE_FACILITIES
  (TYPE_OF_WS           : in  WS_TYPE;
   ERROR_INDICATOR      : out ERROR_NUMBER;
   NUMBER_OF_TYPES      : out PHIGS_INTEGER;
   LIST_OF_TYPES        : out LINETYPES.LIST_OF;
   NUMBER_OF_WIDTHS     : out PHIGS_NATURAL;
   NOMINAL_WIDTH        : out DC.MAGNITUDE;
   RANGE_OF_WIDTHS      : out DC.RANGE_OF_MAGNITUDES;
   NUMBER_OF_INDICES    : out PHIGS_NATURAL);

procedure INQ_PREDEFINED_POLYLINE_REPRESENTATION
  (TYPE_OF_WS           : in  WS_TYPE;
   POLYLINE_IND         : in  POLYLINE_INDEX;
   ERROR_INDICATOR      : out ERROR_NUMBER;
   TYPE_OF_LINE         : out LINETYPE;
   LINEWIDTH_SF         : out LINEWIDTH;
   LINE_COLOUR          : out COLOUR_INDEX);

procedure INQ_POLYMARKER_FACILITIES
  (TYPE_OF_WS           : in  WS_TYPE;
   ERROR_INDICATOR      : out ERROR_NUMBER;
   NUMBER_OF_TYPES      : out PHIGS_INTEGER;
   LIST_OF_TYPES        : out MARKER_TYPES.LIST_OF;
   NUMBER_OF_SIZES      : out PHIGS_NATURAL;
   NOMINAL_SIZE         : out DC.MAGNITUDE;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
    RANGE_OF_SIZES       : out DC.RANGE_OF_MAGNITUDES;
    NUMBER_OF_INDICES    : out PHIGS_NATURAL);

procedure INQ_PREDEFINED_POLYMARKER_REPRESENTATION
  (TYPE_OF_WS            : in  WS_TYPE;
    POLYMARKER_IND       : in  POLYMARKER_INDEX;
    ERROR_INDICATOR      : out ERROR_NUMBER;
    TYPE_OF_MARKER       : out MARKER_TYPE;
    SIZE                 : out MARKER_SIZE;
    MARKER_COLOUR        : out COLOUR_INDEX);

procedure INQ_TEXT_FACILITIES
  (TYPE_OF_WS                      : in  WS_TYPE;
    ERROR_INDICATOR                : out ERROR_NUMBER;
    LIST_OF_FONT_PRECISION_PAIRS   : out TEXT_FONT_PRECISIONS.LIST_OF;
    NUMBER_OF_HEIGHTS              : out PHIGS_NATURAL;
    RANGE_OF_HEIGHTS               : out DC.RANGE_OF_MAGNITUDES;
    NUMBER_OF_EXPANSIONS           : out PHIGS_NATURAL;
    RANGE_OF_EXPANSIONS            : out DC.RANGE_OF_MAGNITUDES;
    NUMBER_OF_INDICES              : out PHIGS_NATURAL);

procedure INQ_PREDEFINED_TEXT_REPRESENTATION
  (TYPE_OF_WS            : in  WS_TYPE;
    TEXT_IND             : in  TEXT_INDEX;
    ERROR_INDICATOR      : out ERROR_NUMBER;
    FONT                 : out TEXT_FONT;
    PRECISION            : out TEXT_PRECISION;
    EXPANSION            : out CHAR_EXPANSION;
    SPACING              : out CHAR_SPACING;
    TEXT_COLOUR          : out COLOUR_INDEX);

procedure INQ_ANNOTATION_FACILITIES
  (TYPE_OF_WS                      : in  WS_TYPE;
    ERROR_INDICATOR                : out ERROR_NUMBER;
    LIST_OF_ANNOTATION_STYLES      : out ANNOTATION_STYLES.LIST_OF;
    NUMBER_OF_CHAR_HEIGHTS         : out PHIGS_NATURAL;
    RANGE_OF_CHAR_HEIGHTS          : out DC.RANGE_OF_MAGNITUDES);

procedure INQ_TEXT_EXTENT
  (TYPE_OF_WS                : in  WS_ID;
    FONT                     : in  TEXT_FONT;
    EXPANSION                : in  CHAR_EXPANSION;
    SPACING                  : in  CHAR_SPACING;
    HEIGHT                   : in  MC.MAGNITUDE;
    PATH                     : in  TEXT_PATH;
    ALIGNMENT                : in  TEXT_ALIGNMENT;
    CHAR_STRING              : in  PHIGS_STRING;
    ERROR_INDICATOR          : out ERROR_NUMBER;
    EXTENT_RECTANGLE         : out MC.RECTANGULAR_REGION_2;
    CONCATENATION_OFFSET     : out MC.VECTOR_2);

-- This procedure using MC coordinates is for use with text primitives.


procedure INQ_TEXT_EXTENT
  (TYPE_OF_WS                : in  WS_ID;
    FONT                     : in  TEXT_FONT;
    EXPANSION                : in  CHAR_EXPANSION;
    SPACING                  : in  CHAR_SPACING;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
    HEIGHT                      : in  NPC.MAGNITUDE;
    PATH                        : in  TEXT_PATH;
    ALIGNMENT                   : in  TEXT_ALIGNMENT;
    CHAR_STRING                 : in  PHIGS_STRING;
    ERROR_INDICATOR             : out ERROR_NUMBER;
    EXTENT_RECTANGLE            : out NPC.RECTANGULAR_REGION_2;
    CONCATENATION_OFFSET        : out NPC.VECTOR_2);

-- This procedure using NPC coordinates is for use with annotation text
-- primitives.


procedure INQ_INTERIOR_FACILITIES
  (TYPE_OF_WS                   : in  WS_TYPE;
   ERROR_INDICATOR              : out ERROR_NUMBER;
   LIST_OF_INTERIOR_STYLES      : out INTERIOR_STYLES.LIST_OF;
   NUMBER_OF_HATCH_STYLES       : out PHIGS_NATURAL;
   LIST_OF_HATCH_STYLES         : out HATCH_STYLES.LIST_OF;
   NUMBER_OF_INDICES            : out PHIGS_NATURAL);

procedure INQ_PREDEFINED_INTERIOR_REPRESENTATION
  (TYPE_OF_WS                   : in  WS_TYPE;
   INTERIOR_IND                 : in  INTERIOR_INDEX;
   ERROR_INDICATOR              : out ERROR_NUMBER;
   STYLE_OF_INTERIOR            : out INTERIOR_STYLE;
   STYLE_IND                    : out STYLE_INDEX;
   INTERIOR_COLOUR              : out COLOUR_INDEX);

procedure INQ_EDGE_FACILITIES
  (TYPE_OF_WS                   : in  WS_TYPE;
   ERROR_INDICATOR              : out ERROR_NUMBER;
   NUMBER_OF_TYPES              : out PHIGS_INTEGER;
   LIST_OF_TYPES                : out EDGETYPES.LIST_OF;
   NUMBER_OF_WIDTHS             : out PHIGS_NATURAL;
   NOMINAL_WIDTH                : out DC.MAGNITUDE;
   RANGE_OF_WIDTHS              : out DC.RANGE_OF_MAGNITUDES;
   NUMBER_OF_INDICES            : out PHIGS_NATURAL);

procedure INQ_PREDEFINED_EDGE_REPRESENTATION
  (TYPE_OF_WS                   : in  WS_TYPE;
   EDGE_IND                     : in  EDGE_INDEX;
   ERROR_INDICATOR              : out ERROR_NUMBER;
   FLAG                         : out EDGE_FLAG;
   TYPE_OF_EDGE                 : out EDGETYPE;
   EDGEWIDTH_SF                 : out EDGEWIDTH;
   EDGE_COLOUR                  : out COLOUR_INDEX);

procedure INQ_PATTERN_FACILITIES
  (TYPE_OF_WS                   : in  WS_TYPE;
   ERROR_INDICATOR              : out ERROR_NUMBER;
   NUMBER_OF_INDICES            : out PHIGS_NATURAL);

procedure INQ_PREDEFINED_PATTERN_REPRESENTATION
  (TYPE_OF_WS                   : in  WS_TYPE;
   PATTERN_IND                  : in  PATTERN_INDEX;
   ERROR_INDICATOR              : out ERROR_NUMBER;
   PATTERN                      : out ACCESS_COLOUR_MATRIX);

procedure INQ_COLOUR_MODEL_FACILITIES
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

9593 PT*3 90 ■ 4851903 0575606 756 ■


```
    (TYPE_OF_WS          : in  WS_TYPE;
     ERROR_INDICATOR     : out ERROR_NUMBER;
     LIST_OF_MODELS      : out COLOUR_MODELS.LIST_OF;
     DEFAULT_MODEL       : out COLOUR_MODEL);

  procedure INQ_COLOUR_FACILITIES
    (TYPE_OF_WS                 : in  WS_TYPE;
     ERROR_INDICATOR            : out ERROR_NUMBER;
     NUMBER_OF_COLOURS          : out PHIGS_NATURAL;
     AVAILABILITY_OF_COLOUR     : out COLOUR_AVAILABLE;
     NUMBER_OF_INDICES          : out PHIGS_NATURAL;
     PRIMARY_COLOURS            : out CHROMATICITY_COEFFICIENT_SET);

  procedure INQ_PREDEFINED_COLOUR_REPRESENTATION
    (TYPE_OF_WS          : in  WS_TYPE;
     INDEX               : in  COLOUR_INDEX;
     ERROR_INDICATOR     : out ERROR_NUMBER;
     COLOUR              : out COLOUR_REPRESENTATION);

  procedure INQ_LIST_OF_AVAILABLE_GDP
    (TYPE_OF_WS          : in  WS_TYPE;
     ERROR_INDICATOR     : out ERROR_NUMBER;
     LIST_OF_GDP         : out GDP_3_IDS.LIST_OF);

  procedure INQ_LIST_OF_AVAILABLE_GDP
    (TYPE_OF_WS          : in  WS_TYPE;
     ERROR_INDICATOR     : out ERROR_NUMBER;
     LIST_OF_GDP         : out GDP_IDS.LIST_OF);

  procedure INQ_GDP
    (TYPE_OF_WS                    : in  WS_TYPE;
     GDP                           : in  GDP_3_ID;
     ERROR_INDICATOR               : out ERROR_NUMBER;
     LIST_OF_ATTRIBUTE_SETS_USED   : out ATTRIBUTES_USED.LIST_OF);

  procedure INQ_GDP
    (TYPE_OF_WS                    : in  WS_TYPE;
     GDP                           : in  GDP_ID;
     ERROR_INDICATOR               : out ERROR_NUMBER;
     LIST_OF_ATTRIBUTE_SETS_USED   : out ATTRIBUTES_USED.LIST_OF);

  procedure INQ_LIST_OF_AVAILABLE_GSE
    (TYPE_OF_WS          : in  WS_TYPE;
     ERROR_INDICATOR     : out ERROR_NUMBER;
     LIST_OF_GSE         : out GSE_IDS.LIST_OF);

  procedure INQ_NUMBER_OF_DISPLAY_PRIORITIES_SUPPORTED
    (TYPE_OF_WS               : in  WS_TYPE;
     ERROR_INDICATOR          : out ERROR_NUMBER;
     NUMBER_OF_PRIORITIES     : out PHIGS_NATURAL);

  procedure INQ_WS_STATE_TABLE_LENGTHS
    (TYPE_OF_WS                  : in  WS_TYPE;
     ERROR_INDICATOR             : out ERROR_NUMBER;
     MAX_POLYLINE_ENTRIES        : out PHIGS_NATURAL;
     MAX_POLYMARKER_ENTRIES      : out PHIGS_NATURAL;
     MAX_TEXT_ENTRIES            : out PHIGS_NATURAL;
     MAX_INTERIOR_ENTRIES        : out PHIGS_NATURAL;
     MAX_EDGE_ENTRIES            : out PHIGS_NATURAL;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
   MAX_PATTERN_INDICES          : out PHIGS_NATURAL;
   MAX_COLOUR_INDICES           : out PHIGS_NATURAL;
   MAX_VIEW_INDICES             : out PHIGS_NATURAL);

procedure INQ_DYNAMICS_OF_STRUCTURES
  (TYPE_OF_WS                   : in  WS_TYPE;
   ERROR_INDICATOR              : out ERROR_NUMBER;
   CONTENT_MODIFICATION         : out DYNAMIC_MODIFICATION;
   POST                         : out DYNAMIC_MODIFICATION;
   UNPOST                       : out DYNAMIC_MODIFICATION;
   DELETE                       : out DYNAMIC_MODIFICATION;
   REFERENCE_MODIFICATION       : out DYNAMIC_MODIFICATION);

procedure INQ_NUMBER_OF_AVAILABLE_LOGICAL_INPUT_DEVICES
  (TYPE_OF_WS          : in  WS_TYPE;
   ERROR_INDICATOR     : out ERROR_NUMBER;
   LOCATOR             : out PHIGS_NATURAL;
   STROKE              : out PHIGS_NATURAL;
   VALUATOR            : out PHIGS_NATURAL;
   CHOICE              : out PHIGS_NATURAL;
   PICK                : out PHIGS_NATURAL;
   STRING              : out PHIGS_NATURAL);

procedure INQ_DEFAULT_LOCATOR_DEVICE_DATA
  (TYPE_OF_WS                   : in  WS_TYPE;
   DEVICE                       : in  LOCATOR_DEVICE_NUMBER;
   ERROR_INDICATOR              : out ERROR_NUMBER;
   INITIAL_POSITION             : out WC.POINT_3;
   LIST_OF_PROMPT_ECHO_TYPES    : out LOCATOR_PROMPT_ECHO_TYPES.LIST_OF;
   ECHO_VOLUME                  : out DC.RECTANGULAR_REGION_3;
   DATA_RECORD                  : out LOCATOR_DATA_RECORD);

procedure INQ_DEFAULT_LOCATOR_DEVICE_DATA
  (TYPE_OF_WS                   : in  WS_TYPE;
   DEVICE                       : in  LOCATOR_DEVICE_NUMBER;
   ERROR_INDICATOR              : out ERROR_NUMBER;
   INITIAL_POSITION             : out WC.POINT_2;
   LIST_OF_PROMPT_ECHO_TYPES    : out LOCATOR_PROMPT_ECHO_TYPES.LIST_OF;
   ECHO_AREA                    : out DC.RECTANGULAR_REGION_2;
   DATA_RECORD                  : out LOCATOR_DATA_RECORD);

procedure INQ_DEFAULT_STROKE_DEVICE_DATA
  (TYPE_OF_WS                   : in  WS_TYPE;
   DEVICE                       : in  STROKE_DEVICE_NUMBER;
   ERROR_INDICATOR              : out ERROR_NUMBER;
   MAX_BUFFER_SIZE              : out PHIGS_NATURAL;
   LIST_OF_PROMPT_ECHO_TYPES    : out STROKE_PROMPT_ECHO_TYPES.LIST_OF;
   ECHO_VOLUME                  : out DC.RECTANGULAR_REGION_3;
   DATA_RECORD                  : out STROKE_DATA_RECORD);

procedure INQ_DEFAULT_STROKE_DEVICE_DATA
  (TYPE_OF_WS                   : in  WS_TYPE;
   DEVICE                       : in  STROKE_DEVICE_NUMBER;
   ERROR_INDICATOR              : out ERROR_NUMBER;
   MAX_BUFFER_SIZE              : out PHIGS_NATURAL;
   LIST_OF_PROMPT_ECHO_TYPES    : out STROKE_PROMPT_ECHO_TYPES.LIST_OF;
   ECHO_AREA                    : out DC.RECTANGULAR_REGION_2;
   DATA_RECORD                  : out STROKE_DATA_RECORD);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

9593 PT*3 90 ■ 4851903 0575608 529 ■


```
procedure INQ_DEFAULT_VALUATOR_DEVICE_DATA
  (TYPE_OF_WS                    : in  WS_TYPE;
   DEVICE                        : in  VALUATOR_DEVICE_NUMBER;
   ERROR_INDICATOR               : out ERROR_NUMBER;
   INITIAL_VALUE                 : out VALUATOR_VALUE;
   LIST_OF_PROMPT_ECHO_TYPES     : out VALUATOR_PROMPT_ECHO_TYPES.LIST_OF;
   ECHO_VOLUME                   : out DC.RECTANGULAR_REGION_3;
   DATA_RECORD                   : out VALUATOR_DATA_RECORD);

procedure INQ_DEFAULT_VALUATOR_DEVICE_DATA
  (TYPE_OF_WS                    : in  WS_TYPE;
   DEVICE                        : in  VALUATOR_DEVICE_NUMBER;
   ERROR_INDICATOR               : out ERROR_NUMBER;
   INITIAL_VALUE                 : out VALUATOR_VALUE;
   LIST_OF_PROMPT_ECHO_TYPES     : out VALUATOR_PROMPT_ECHO_TYPES.LIST_OF;
   ECHO_AREA                     : out DC.RECTANGULAR_REGION_2;
   DATA_RECORD                   : out VALUATOR_DATA_RECORD);

procedure INQ_DEFAULT_CHOICE_DEVICE_DATA
  (TYPE_OF_WS                    : in  WS_TYPE;
   DEVICE                        : in  CHOICE_DEVICE_NUMBER;
   ERROR_INDICATOR               : out ERROR_NUMBER;
   MAX_CHOICES                   : out CHOICE_NUMBER;
   LIST_OF_PROMPT_ECHO_TYPES     : out CHOICE_PROMPT_ECHO_TYPES.LIST_OF;
   ECHO_VOLUME                   : out DC.RECTANGULAR_REGION_3;
   DATA_RECORD                   : out CHOICE_DATA_RECORD);

procedure INQ_DEFAULT_CHOICE_DEVICE_DATA
  (TYPE_OF_WS                    : in  WS_TYPE;
   DEVICE                        : in  CHOICE_DEVICE_NUMBER;
   ERROR_INDICATOR               : out ERROR_NUMBER;
   MAX_CHOICES                   : out CHOICE_NUMBER;
   LIST_OF_PROMPT_ECHO_TYPES     : out CHOICE_PROMPT_ECHO_TYPES.LIST_OF;
   ECHO_AREA                     : out DC.RECTANGULAR_REGION_2;
   DATA_RECORD                   : out CHOICE_DATA_RECORD);

procedure INQ_DEFAULT_PICK_DEVICE_DATA
  (TYPE_OF_WS                    : in  WS_TYPE;
   DEVICE                        : in  PICK_DEVICE_NUMBER;
   ERROR_INDICATOR               : out ERROR_NUMBER;
   LIST_OF_PROMPT_ECHO_TYPES     : out PICK_PROMPT_ECHO_TYPES.LIST_OF;
   ECHO_VOLUME                   : out DC.RECTANGULAR_REGION_3;
   DATA_RECORD                   : out PICK_DATA_RECORD);

procedure INQ_DEFAULT_PICK_DEVICE_DATA
  (TYPE_OF_WS                    : in  WS_TYPE;
   DEVICE                        : in  PICK_DEVICE_NUMBER;
   ERROR_INDICATOR               : out ERROR_NUMBER;
   LIST_OF_PROMPT_ECHO_TYPES     : out PICK_PROMPT_ECHO_TYPES.LIST_OF;
   ECHO_AREA                     : out DC.RECTANGULAR_REGION_2;
   DATA_RECORD                   : out PICK_DATA_RECORD);

procedure INQ_DEFAULT_STRING_DEVICE_DATA
  (TYPE_OF_WS                    : in  WS_TYPE;
   DEVICE                        : in  STRING_DEVICE_NUMBER;
   ERROR_INDICATOR               : out ERROR_NUMBER;
   MAX_STRING_BUFFER_SIZE        : out PHIGS_NATURAL;
   LIST_OF_PROMPT_ECHO_TYPES     : out STRING_PROMPT_ECHO_TYPES.LIST_OF;
   ECHO_VOLUME                   : out DC.RECTANGULAR_REGION_3;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
   DATA_RECORD                    : out STRING_DATA_RECORD);

procedure INQ_DEFAULT_STRING_DEVICE_DATA
  (TYPE_OF_WS                     : in  WS_TYPE;
   DEVICE                         : in  STRING_DEVICE_NUMBER;
   ERROR_INDICATOR                : out ERROR_NUMBER;
   MAX_STRING_BUFFER_SIZE         : out PHIGS_NATURAL;
   LIST_OF_PROMPT_ECHO_TYPES      : out STRING_PROMPT_ECHO_TYPES.LIST_OF;
   ECHO_AREA                      : out DC.RECTANGULAR_REGION_2;
   DATA_RECORD                    : out STRING_DATA_RECORD);

procedure INQ_SET_OF_WS_TO_WHICH_POSTED
  (STRUCTURE_IDENTIFIER : in  STRUCTURE_ID;
   ERROR_INDICATOR      : out ERROR_NUMBER;
   LIST_OF_WS           : out WS_IDS.LIST_OF);

procedure INQ_OPEN_STRUCTURE
  (ERROR_INDICATOR      : out ERROR_NUMBER;
   STATUS               : out OPEN_STRUCTURE_STATUS;
   STRUCTURE_IDENTIFIER : out STRUCTURE_ID);

procedure INQ_ELEMENT_POINTER
  (ERROR_INDICATOR : out ERROR_NUMBER;
   POSITION        : out RETURNED_ELEMENT_POSITION);

procedure INQ_CURRENT_ELEMENT_TYPE_AND_SIZE
  (ERROR_INDICATOR : out ERROR_NUMBER;
   ELEMENT_TYPE    : out STRUCTURE_ELEMENT_TYPE;
   SIZE            : out PHIGS_NATURAL);

procedure INQ_CURRENT_ELEMENT_CONTENT
  (ERROR_INDICATOR : out ERROR_NUMBER;
   ELEMENT_RECORD  : in out STRUCTURE_ELEMENT_RECORD);

procedure INQ_ELEMENT_TYPE_AND_SIZE
  (STRUCTURE_IDENTIFIER : in  STRUCTURE_ID;
   POSITION             : in  ELEMENT_POSITION;
   ERROR_INDICATOR      : out ERROR_NUMBER;
   ELEMENT_TYPE         : out STRUCTURE_ELEMENT_TYPE;
   SIZE                 : out PHIGS_NATURAL);

procedure INQ_ELEMENT_CONTENT
  (STRUCTURE_IDENTIFIER : in     STRUCTURE_ID;
   POSITION             : in     ELEMENT_POSITION;
   ERROR_INDICATOR      :    out ERROR_NUMBER;
   ELEMENT_RECORD       : in out STRUCTURE_ELEMENT_RECORD);

procedure INQ_STRUCTURE_STATUS
  (STRUCTURE_IDENTIFIER : in  STRUCTURE_ID;
   ERROR_INDICATOR      : out ERROR_NUMBER;
   STATUS               : out STRUCTURE_STATUS);

procedure INQ_PATHS_TO_ANCESTORS
  (STRUCTURE_IDENTIFIER : in  STRUCTURE_ID;
   ORDER                : in  PATH_ORDER;
   DEPTH                : in  PATH_DEPTH;
   ERROR_INDICATOR      : out ERROR_NUMBER;
   LIST_OF_PATHS        : out REFERENCE_PATHS.LIST_OF);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
procedure INQ_PATHS_TO_DESCENDANTS
  (STRUCTURE_IDENTIFIER : in  STRUCTURE_ID;
   ORDER                : in  PATH_ORDER;
   DEPTH                : in  PATH_DEPTH;
   ERROR_INDICATOR      : out ERROR_NUMBER;
   LIST_OF_PATHS        : out REFERENCE_PATHS.LIST_OF);

procedure ELEMENT_SEARCH
  (STRUCTURE_IDENTIFIER : in  STRUCTURE_ID;
   START_POSITION       : in  ELEMENT_POSITION;
   DIRECTION            : in  SEARCH_DIRECTION;
   INCLUSION_SET        : in  ELEMENT_TYPES.LIST_OF;
   EXCLUSION_SET        : in  ELEMENT_TYPES.LIST_OF;
   ERROR_INDICATOR      : out ERROR_NUMBER;
   SEARCH_STATUS        : out SEARCH_STATUS_INDICATOR;
   FOUND_POSITION       : out RETURNED_ELEMENT_POSITION);

procedure INCREMENTAL_SPATIAL_SEARCH
  (SEARCH_REFERENCE_POINT : in  WC.POINT_3;
   SEARCH_DISTANCE        : in  WC.MAGNITUDE;
   STARTING_PATH          : in  REFERENCE_PATH;
   MODELLING_CLIPPING     : in  CLIPPING_INDICATOR;
   SEARCH_CEILING_INDEX   : in  ELEMENT_POSITION;
   LIST_OF_NORMAL_FILTERS : in  NAME_SET_FILTERS.LIST_OF;
   LIST_OF_INVERTED_FILTERS : in  NAME_SET_FILTERS.LIST_OF;
   ERROR_INDICATOR        : out ERROR_NUMBER;
   FOUND_PATH             : out REFERENCE_PATH);

procedure INCREMENTAL_SPATIAL_SEARCH
  (SEARCH_REFERENCE_POINT : in  WC.POINT_2;
   SEARCH_DISTANCE        : in  WC.MAGNITUDE;
   STARTING_PATH          : in  REFERENCE_PATH;
   MODELLING_CLIPPING     : in  CLIPPING_INDICATOR;
   SEARCH_CEILING_INDEX   : in  ELEMENT_POSITION;
   LIST_OF_NORMAL_FILTERS : in  NAME_SET_FILTERS.LIST_OF;
   LIST_OF_INVERTED_FILTERS : in  NAME_SET_FILTERS.LIST_OF;
   ERROR_INDICATOR        : out ERROR_NUMBER;
   FOUND_PATH             : out REFERENCE_PATH);


procedure INQ_INPUT_QUEUE_OVERFLOW
  (ERROR_INDICATOR : out ERROR_NUMBER;
   WS              : out WS_ID;
   DEVICE          : out EVENT_QUEUE_DEVICE_NUMBER);

procedure INQ_ERROR_HANDLING_MODE
  (ERROR_INDICATOR : out ERROR_NUMBER;
   MODE            : out ERROR_HANDLING_MODE);

-- ERROR FUNCTIONS

procedure EMERGENCY_CLOSE_PHIGS;

procedure ERROR_HANDLING
  (ERROR_INDICATOR : in ERROR_NUMBER;
   SUBPROGRAM      : in SUBPROGRAM_NAME;
   ERROR_FILE      : in FILE_ID := PHIGS_STRING(DEFAULT_ERROR_FILE));

procedure ERROR_LOGGING
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
   (ERROR_INDICATOR : in ERROR_NUMBER;
    SUBPROGRAM      : in SUBPROGRAM_NAME;
    ERROR_FILE      : in FILE_ID := PHIGS_STRING(DEFAULT_ERROR_FILE));

procedure SET_ERROR_HANDLING_MODE
  (MODE : in ERROR_HANDLING_MODE);

-- Subprograms for Manipulating Input Data Records

-- Locator Data Record Operations

procedure BUILD_LOCATOR_DATA_RECORD
  (PROMPT_ECHO_TYPE : in  LOCATOR_PROMPT_ECHO_TYPE;
   DATA_RECORD      : out LOCATOR_DATA_RECORD);

procedure BUILD_LOCATOR_DATA_RECORD
  (PROMPT_ECHO_TYPE : in  LOCATOR_PROMPT_ECHO_TYPE;
   CONTENTS         : in  LINE_DATA;
   DATA_RECORD      : out LOCATOR_DATA_RECORD);

procedure BUILD_LOCATOR_DATA_RECORD
  (PROMPT_ECHO_TYPE : in  LOCATOR_PROMPT_ECHO_TYPE;
   CONTENTS         : in  INTERIOR_DATA;
   DATA_RECORD      : out LOCATOR_DATA_RECORD);

procedure BUILD_LOCATOR_DATA_RECORD
  (PROMPT_ECHO_TYPE : in  LOCATOR_PROMPT_ECHO_TYPE;
   INTERIOR_CONTENTS : in  INTERIOR_DATA;
   EDGE_CONTENTS     : in  EDGE_DATA;
   DATA_RECORD       : out LOCATOR_DATA_RECORD);

function LOCATOR_ATTRIBUTES_USED
          (DATA_RECORD : in LOCATOR_DATA_RECORD)
  return POLYLINE_FILL_AREA_CONTROL_FLAG;

function LINE_ATTRIBUTES (DATA_RECORD : in LOCATOR_DATA_RECORD)
  return LINE_DATA;

function INTERIOR_ATTRIBUTES (DATA_RECORD : in  LOCATOR_DATA_RECORD)
  return INTERIOR_DATA;

function EDGE_ATTRIBUTES (DATA_RECORD : in  LOCATOR_DATA_RECORD)
  return EDGE_DATA;

-- Stroke Data Record Operations

procedure BUILD_STROKE_DATA_RECORD
  (PROMPT_ECHO_TYPE : in  STROKE_PROMPT_ECHO_TYPE;
   BUFFER_SIZE      : in  PHIGS_POSITIVE;
   POSITION         : in  PHIGS_POSITIVE;
   INTERVAL         : in  WC.SIZE_2;
   TIME             : in  DURATION;
   DATA_RECORD      : out STROKE_DATA_RECORD);

procedure BUILD_STROKE_DATA_RECORD
  (PROMPT_ECHO_TYPE : in  STROKE_PROMPT_ECHO_TYPE;
   BUFFER_SIZE      : in  PHIGS_POSITIVE;
   POSITION         : in  PHIGS_POSITIVE;
   INTERVAL         : in  WC.SIZE_2;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

9593 PT*3 90 ■ 4851903 0575612 T5T ■


```
    TIME                : in  DURATION;
    CONTENTS            : in  MARKER_DATA;
    DATA_RECORD         : out STROKE_DATA_RECORD);

procedure BUILD_STROKE_DATA_RECORD
  (PROMPT_ECHO_TYPE : in  STROKE_PROMPT_ECHO_TYPE;
    BUFFER_SIZE       : in  PHIGS_POSITIVE;
    POSITION          : in  PHIGS_POSITIVE;
    INTERVAL          : in  WC.SIZE_2;
    TIME              : in  DURATION;
    CONTENTS          : in  LINE_DATA;
    DATA_RECORD       : out STROKE_DATA_RECORD);

function BUFFER_SIZE (DATA_RECORD : in STROKE_DATA_RECORD)
  return PHIGS_POSITIVE;

function POSITION (DATA_RECORD : in STROKE_DATA_RECORD)
  return PHIGS_POSITIVE;

function INTERVAL (DATA_RECORD : in STROKE_DATA_RECORD)
  return WC.SIZE_2;

function TIME (DATA_RECORD : in STROKE_DATA_RECORD)
  return DURATION;

function MARKER_ATTRIBUTES (DATA_RECORD : in STROKE_DATA_RECORD)
  return MARKER_DATA;

function LINE_ATTRIBUTES (DATA_RECORD : in STROKE_DATA_RECORD)
  return LINE_DATA;

-- Valuator Data Record Operations

procedure BUILD_VALUATOR_DATA_RECORD
  (PROMPT_ECHO_TYPE : in  VALUATOR_PROMPT_ECHO_TYPE;
    LOW_VALUE         : in  VALUATOR_VALUE;
    HIGH_VALUE        : in  VALUATOR_VALUE;
    DATA_RECORD       : out VALUATOR_DATA_RECORD);

function HIGH_VALUE (DATA_RECORD : in VALUATOR_DATA_RECORD)
  return VALUATOR_VALUE;

function LOW_VALUE (DATA_RECORD : in VALUATOR_DATA_RECORD)
  return VALUATOR_VALUE;

-- Choice Data Record Operations

procedure BUILD_CHOICE_DATA_RECORD
  (PROMPT_ECHO_TYPE : in  CHOICE_PROMPT_ECHO_TYPE;
    DATA_RECORD       : out CHOICE_DATA_RECORD);

procedure BUILD_CHOICE_DATA_RECORD
  (PROMPT_ECHO_TYPE : in  CHOICE_PROMPT_ECHO_TYPE;
    ARRAY_OF_PROMPTS  : in  CHOICE_PROMPTS.LIST_OF;
    DATA_RECORD       : out CHOICE_DATA_RECORD);

procedure BUILD_CHOICE_DATA_RECORD
  (PROMPT_ECHO_TYPE : in  CHOICE_PROMPT_ECHO_TYPE;
    ARRAY_OF_STRINGS  : in  CHOICE_PROMPT_STRING_LIST;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
    DATA_RECORD          : out CHOICE_DATA_RECORD);

procedure BUILD_CHOICE_DATA_RECORD
  (PROMPT_ECHO_TYPE      : in  CHOICE_PROMPT_ECHO_TYPE;
   STRUCTURE_IDENTIFIER  : in  STRUCTURE_ID;
   LIST_OF_PICK_IDS      : in  PICK_IDS.LIST_OF;
   DATA_RECORD           : out CHOICE_DATA_RECORD);

function ARRAY_OF_PROMPTS (DATA_RECORD : in CHOICE_DATA_RECORD)
  return CHOICE_PROMPTS.LIST_OF;

function ARRAY_OF_STRINGS (DATA_RECORD : in CHOICE_DATA_RECORD)
  return CHOICE_PROMPT_STRING_LIST;

function STRUCTURE_IDENTIFIER (DATA_RECORD : in CHOICE_DATA_RECORD)
  return STRUCTURE_ID;

function LIST_OF_PICK_IDS (DATA_RECORD : in CHOICE_DATA_RECORD)
  return PICK_IDS.LIST_OF;

-- Pick Data Record Operations.

procedure BUILD_PICK_DATA_RECORD
  (PROMPT_ECHO_TYPE : in  PICK_PROMPT_ECHO_TYPE;
   DATA_RECORD      : out PICK_DATA_RECORD);

-- String Data Record Operations

procedure BUILD_STRING_DATA_RECORD
  (PROMPT_ECHO_TYPE       : in  STRING_PROMPT_ECHO_TYPE;
   INPUT_BUFFER_SIZE      : in  PHIGS_NATURAL;
   INITIAL_CURSOR_POSITION : in  PHIGS_NATURAL;
   DATA_RECORD            : out STRING_DATA_RECORD);

function INPUT_BUFFER_SIZE (DATA_RECORD : in STRING_DATA_RECORD)
  return PHIGS_NATURAL;

function INITIAL_CURSOR_POSITION (DATA_RECORD : in STRING_DATA_RECORD)
  return PHIGS_NATURAL;

procedure DEALLOCATE
  (OBJECT : in out CHOICE_DATA_RECORD);

procedure DEALLOCATE
  (OBJECT : in out ESCAPE_DATA_RECORD);

procedure DEALLOCATE
  (OBJECT : in out GDP_3_RECORD);

procedure DEALLOCATE
  (OBJECT : in out GDP_RECORD);

procedure DEALLOCATE
  (OBJECT : in out GSE_RECORD);

procedure DEALLOCATE
  (OBJECT : in out LOCATOR_DATA_RECORD);

procedure DEALLOCATE
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
   (OBJECT : in out METAFILE_DATA_RECORD);

 procedure DEALLOCATE
   (OBJECT : in out NAME_SET)
   renames PHIGS_NAME_SET_FACILITY.DEALLOCATE;

 procedure DEALLOCATE
   (OBJECT : in out PICK_DATA_RECORD);

 procedure DEALLOCATE
   (OBJECT : in out STRING_DATA_RECORD);

 procedure DEALLOCATE
   (OBJECT : in out STROKE_DATA_RECORD);

 procedure DEALLOCATE
   (OBJECT : in out STRUCTURE_ELEMENT_RECORD);

 procedure DEALLOCATE
   (OBJECT : in out VALUATOR_DATA_RECORD);

 -- Subprograms for Manipulating Metafile Data Records

 procedure BUILD_NEW_METAFILE_DATA_RECORD
   (TYPE_OF_ITEM : in METAFILE_ITEM_TYPE;
    ITEM_DATA    : in PHIGS_STRING;
    ITEM         : out METAFILE_DATA_RECORD);

 function ITEM_DATA_RECORD_STRING
   (ITEM : in METAFILE_DATA_RECORD) return ACCESS_STRING;

 -- Subprograms for Manipulating Name Sets

 procedure BUILD_NAME_SET
   (MEMBERS : in  NAMES.LIST_OF;
    SET     : out NAME_SET)
   renames PHIGS_NAME_SET_FACILITY.BUILD_NAME_SET;

 procedure BUILD_NAME_SET_UNION
   (LEFT_SET  : in NAME_SET;
    RIGHT_SET : in NAME_SET;
    UNION     : out NAME_SET)
   renames PHIGS_NAME_SET_FACILITY.BUILD_NAME_SET_UNION;

   -- Constructs the logical union of two name sets.

 procedure BUILD_NAME_SET_INTERSECTION
   (LEFT_SET    : in NAME_SET;
    RIGHT_SET   : in NAME_SET;
    INTERSECTION : out NAME_SET)
   renames PHIGS_NAME_SET_FACILITY.BUILD_NAME_SET_INTERSECTION;

   -- Constructs the logical intersection of two name sets.

 procedure BUILD_NAME_SET_DIFFERENCE
   (LEFT_SET  : in NAME_SET;
    RIGHT_SET : in NAME_SET;
    DIFFERENCE : out NAME_SET)
   renames PHIGS_NAME_SET_FACILITY.BUILD_NAME_SET_DIFFERENCE;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
   -- Constructs the logical difference of two name sets.

procedure COPY_NAME_SET
  (ORIGINAL : in  NAME_SET;
   COPY     : out NAME_SET)
  renames PHIGS_NAME_SET_FACILITY.COPY_NAME_SET;

   -- Duplicates the provided name set.

function IS_EQUAL
  (LEFT_SET  : in NAME_SET;
   RIGHT_SET : in NAME_SET)
  return BOOLEAN
  renames PHIGS_NAME_SET_FACILITY.IS_EQUAL;

   -- Compares two name sets.

-- Constructs a name set from a list of classes.

procedure EXTRACT_NAMES
  (SET     : in  NAME_SET;
   MEMBERS : out NAMES.LIST_OF)
  renames PHIGS_NAME_SET_FACILITY.EXTRACT_NAMES;

-- Extracts the members from the provided name set.

procedure ADD_NAMES
  (MEMBERS : in     NAMES.LIST_OF;
   SET     : in out NAME_SET)
  renames PHIGS_NAME_SET_FACILITY.ADD_NAMES;

-- Adds members to the name set if not already there.

procedure REMOVE_NAMES
  (MEMBERS : in     NAMES.LIST_OF;
   SET     : in out NAME_SET)
  renames PHIGS_NAME_SET_FACILITY.REMOVE_NAMES;

-- Removes members from the name set if they are there.

function APPLY_FILTER
  (SET    : in NAME_SET;
   FILTER : in NAME_SET_FILTER)
  return NAME_SET_ACCEPTANCE
  renames PHIGS_NAME_SET_FACILITY.APPLY_FILTER;

-- Applies a filter to a name set and indicates whether the name
-- set passes through the filter.

function INQUIRE_MEMBERSHIP
  (MEMBER : in NAME;
   SET    : in NAME_SET)
  return NAME_SET_MEMBERSHIP
  renames PHIGS_NAME_SET_FACILITY.INQUIRE_MEMBERSHIP;

-- Returns an indication of whether the indicated member
-- is contained in the provided name set.

-- PRIVATE TYPE DEFINITIONS
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
private

 -- The following types define the specifications for the private
 -- types. Null records are used soley for the purposes of
 -- demonstrating compilability in this annex. An implementation
 -- would use some more appropriate declaration.

 type METAFILE_DATA_RECORD (TYPE_OF_ITEM : METAFILE_ITEM_TYPE := 0) is
   record
     null;
   end record;

 type CHOICE_DATA_RECORD
         (PROMPT_ECHO_TYPE : CHOICE_PROMPT_ECHO_TYPE := 1) is
   record
     null;
   end record;

 type LOCATOR_DATA_RECORD
         (PROMPT_ECHO_TYPE : LOCATOR_PROMPT_ECHO_TYPE := 1) is
   record
     null;
   end record;

 type PICK_DATA_RECORD
        (PROMPT_ECHO_TYPE : PICK_PROMPT_ECHO_TYPE := 1) is
   record
     null;
   end record;

 type STRING_DATA_RECORD
         (PROMPT_ECHO_TYPE : STRING_PROMPT_ECHO_TYPE := 1) is
   record
      null;
   end record;

 type STROKE_DATA_RECORD
         (PROMPT_ECHO_TYPE : STROKE_PROMPT_ECHO_TYPE := 1) is
   record
     null;
   end record;

 type VALUATOR_DATA_RECORD
         (PROMPT_ECHO_TYPE : VALUATOR_PROMPT_ECHO_TYPE := 1) is
   record
     null;
   end record;

end PHIGS;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
-- Minimal PHIGS_GDP Package

with PHIGS_TYPES;
use PHIGS_TYPES;
package PHIGS_GDP is


 type GDP_INTEGER_ARRAY  is array (SMALL_NATURAL range <>) of
                                  PHIGS_INTEGER;

 type GDP_FLOAT_ARRAY    is array (SMALL_NATURAL range <>) of
                                  GENERAL_FLOAT_PARAMETER;

 type GDP_STRING_ARRAY   is array (SMALL_NATURAL range <>) of
                                  PHIGS_STRING(1..80);

 type GDP_DATA_RECORD (NUM_OF_INTEGERS : SMALL_NATURAL := 0;
                       NUM_OF_REALS    : SMALL_NATURAL := 0;
                       NUM_OF_STRINGS  : SMALL_NATURAL := 0) is

   record
     INTEGER_ARRAY : GDP_INTEGER_ARRAY (1..NUM_OF_INTEGERS);
     REAL_ARRAY    : GDP_FLOAT_ARRAY   (1..NUM_OF_REALS);
     GDP_STRINGS   : GDP_STRING_ARRAY  (1..NUM_OF_STRINGS);
   end record;

 procedure GENERALIZED_GDP
   (GDP        : in GDP_3_ID;
    POINTS     : in MC.POINT_LIST_3;
    GDP_DATA   : in GDP_DATA_RECORD);

 procedure GENERALIZED_GDP
   (GDP        : in GDP_ID;
    POINTS     : in MC.POINT_LIST_2;
    GDP_DATA   : in GDP_DATA_RECORD);

end PHIGS_GDP;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
-- Minimal PHIGS_GSE Package

with PHIGS_TYPES;
use PHIGS_TYPES;
package PHIGS_GSE is


 type GSE_INTEGER_ARRAY is array (SMALL_NATURAL range <>) of
                                  PHIGS_INTEGER;

 type GSE_FLOAT_ARRAY   is array (SMALL_NATURAL range <>) of
                                  GENERAL_FLOAT_PARAMETER;

 type GSE_STRING_ARRAY  is array (SMALL_NATURAL range <>) of
                                  PHIGS_STRING(1..80);

 type GSE_DATA_RECORD (NUM_OF_INTEGERS : SMALL_NATURAL := 0;
                       NUM_OF_REALS    : SMALL_NATURAL := 0;
                       NUM_OF_STRINGS  : SMALL_NATURAL := 0) is

   record
     INTEGER_ARRAY : GSE_INTEGER_ARRAY (1..NUM_OF_INTEGERS);
     REAL_ARRAY    : GSE_FLOAT_ARRAY   (1..NUM_OF_REALS);
     GSE_STRINGS   : GSE_STRING_ARRAY  (1..NUM_OF_STRINGS);
   end record;

 procedure GENERALIZED_GSE
   (GSE       : in GSE_ID;
    GSE_DATA  : in GSE_DATA_RECORD);

end PHIGS_GSE;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
-- Minimal PHIGS_ESCAPE Package

with PHIGS_TYPES;
use PHIGS_TYPES;
package PHIGS_ESCAPE is


 type ESC_INTEGER_ARRAY  is array (SMALL_NATURAL range < >) of
                                  PHIGS_INTEGER;

 type ESC_FLOAT_ARRAY    is array (SMALL_NATURAL range < >) of
                                  GENERAL_FLOAT_PARAMETER;

 type ESC_STRING_ARRAY   is array (SMALL_NATURAL range < >) of
                                  PHIGS_STRING(1..80);

 type ESC_DATA_RECORD (NUM_OF_INTEGERS : SMALL_NATURAL := 0;
                       NUM_OF_REALS    : SMALL_NATURAL := 0;
                       NUM_OF_STRINGS  : SMALL_NATURAL := 0) is

   record
     INTEGER_ARRAY : ESC_INTEGER_ARRAY (1..NUM_OF_INTEGERS);
     REAL_ARRAY    : ESC_FLOAT_ARRAY   (1..NUM_OF_REALS);
     ESC_STRINGS   : ESC_STRING_ARRAY  (1..NUM_OF_STRINGS);
   end record;

 procedure GENERALIZED_ESC
   (ESC_NAME       : in ESCAPE_ID;
    ESC_DATA_IN    : in ESC_DATA_RECORD;
    ESC_DATA_OUT   : out ESC_DATA_RECORD);

end PHIGS_ESCAPE;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

9593 PT*3 90 ■ 4851903 0575620 026 ■

Annex B

(informative)

Cross Reference Listing of Implementation Defined Items

ITEM	SECTION
CHOICE_DATA_RECORD	4.2.3
ESCAPE_DATA_RECORD	4.2.2
GDP_3_RECORD	4.2.2
GDP_RECORD	4.2.2
GSE_RECORD	4.2.2
LOCATOR_DATA_RECORD	4.2.3
METAFILE_DATA_RECORD	4.2.3
NAME_SET	4.2.3
PICK_DATA_RECORD	4.2.3
STRING_DATA_RECORD	4.2.3
STROKE_DATA_RECORD	4.2.3
VALUATOR_DATA_RECORD	4.2.3
DEFAULT_ERROR_FILE	4.2.5
DEFAULT_LIST_SIZE	4.2.5
DEFAULT_MEMORY_UNITS	4.2.5
MIN_PHIGS_PHIGS_INTEGER	4.2.5
MAX_PHIGS_PHIGS_INTEGER	4.2.5
PHIGS_PRECISION	4.2.5
MAX_CHOICE_SMALL_PHIGS_NATURAL	4.2.5
MAX_COLOUR_MATRIX_SMALL_PHIGS_NATURAL	4.2.5
MAX_FILE_SMALL_PHIGS_NATURAL	4.2.5
MAX_INPUT_STRING_SMALL_PHIGS_NATURAL	4.2.5
MAX_SMALL_PHIGS_NATURAL	4.2.5
MAX_STRING_SMALL_PHIGS_NATURAL	4.2.5
MAX_ANNOTATION_STYLES_SUPPORTED	4.2.5
MAX_ARCHIVE_IDS_SUPPORTED	4.2.5
MAX_CHAR_SETS_SUPPORTED	4.2.5
MAX_CHOICE_PROMPTS_SUPPORTED	4.2.5
MAX_CHOICE_PROMPT_ECHO_TYPES_SUPPORTED	4.2.5
MAX_COLOUR_INDICES_SUPPORTED	4.2.5
MAX_COLOUR_MODELS_SUPPORTED	4.2.5
MAX_EDGE_INDICES_SUPPORTED	4.2.5
MAX_EDGETYPES_SUPPORTED	4.2.5
MAX_ESCAPE_IDS_SUPPORTED	4.2.5
MAX_FILE_IDS_SUPPORTED	4.2.5
MAX_GDP_3_IDS_SUPPORTED	4.2.5
MAX_GDP_IDS_SUPPORTED	4.2.5
MAX_GSE_IDS_SUPPORTED	4.2.5
MAX_HATCH_STYLES_SUPPORTED	4.2.5
MAX_HLHSR_IDS_SUPPORTED	4.2.5

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

9593 PT*3 90 ■ 4851903 0575621 T62 ■

MAX_HLHSR_MODES_SUPPORTED	4.2.5
MAX_INTERIOR_INDICES_SUPPORTED	4.2.5
MAX_LINETYPES_SUPPORTED	4.2.5
MAX_LOCATOR_PROMPT_ECHO_TYPES_SUPPORTED	4.2.5
MAX_MARKER_TYPES_SUPPORTED	4.2.5
MAX_MODELLING_CLIP_OPERATION_TYPES_SUPPORTED	4.2.5
MAX_MODELLING_CLIP_DISTINCT_PLANES_SUPPORTED	4.2.5
MAX_NAME_SET_CLASSES_SUPPORTED	4.2.5
MAX_NAME_SET_FILTER_LIST_LENGTH_SUPPORTED	4.2.5
MAX_OPEN_WS_SUPPORTED	4.2.5
MAX_PATH_DEPTH_SUPPORTED	4.2.5
MAX_PATTERN_INDICES_SUPPORTED	4.2.5
MAX_PICK_IDS_SUPPORTED	4.2.5
MAX_PICK_PROMPT_ECHO_TYPES_SUPPORTED	4.2.5
MAX_POLYLINE_INDICES_SUPPORTED	4.2.5
MAX_POLYMARKER_INDICES_SUPPORTED	4.2.5
MAX_POSTED_STRUCTURES_SUPPORTED	4.2.5
MAX_REFERENCE_PATHS_SUPPORTED	4.2.5
MAX_STRING_PROMPT_ECHO_TYPES_SUPPORTED	4.2.5
MAX_STROKE_PROMPT_ECHO_TYPES_SUPPORTED	4.2.5
MAX_STRUCTURE_IDS_SUPPORTED	4.2.5
MAX_TEXT_FONT_PRECISION_PAIRS_SUPPORTED	4.2.5
MAX_TEXT_INDICES_SUPPORTED	4.2.5
MAX_VALUATOR_PROMPT_ECHO_TYPES_SUPPORTED	4.2.5
MAX_VIEW_INDICES_SUPPORTED	4.2.5
MAX_WS_IDS_SUPPORTED	4.2.5
MAX_WS_TYPES_SUPPORTED	4.2.5
MAX_APPLICATION_DATA	4.2.5
MAX_COLOUR_COEFFICIENTS	4.2.5
MAX_MEMORY_UNITS	4.2.5
MAX_METAFILE_ITEM_LENGTH	4.2.5
MIN_METAFILE_ITEM_TYPE	4.2.5
MAX_METAFILE_ITEM_TYPE	4.2.5
MAX_NAME_SET_CLASSES_SUPPORTED	4.2.5

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

9593 PT*3 90 ■ 4851903 0575622 9T9 ■

Annex C

(informative)

Example Programs

C.1 Example Program 1: STAR

```
with PHIGS;
with PHIGS_TYPES;
use PHIGS;
use PHIGS_TYPES;

procedure STAR is

-- This program draws a yellow star on a blue background and writes
-- the title 'Star' in white under the star.

 -- Define the workstation variables and error logging file.

 MY_WS_ID              : constant WS_ID := 1;
 SOME_CONNECTION       : constant CONNECTION_ID := "UNIT_1";
 SOME_OUTPUT_TYPE      : constant WS_TYPE := 1;
 ERROR_FILE            : constant FILE_ID := "MY_ERROR_FILE";

 -- Define the points of the star.

 STAR_POINTS     : constant MC.POINT_LIST_2 :=
                     (( 0.951057,  0.309017 ),
                      (-0.951057,  0.309017 ),
                      ( 0.587785, -0.951057 ),
                      ( 0.0,       1.0      ),
                      (-0.587785, -0.951057 ));

 -- Define view parameters and other variables

 WINDOW                : VRC.RECTANGULAR_REGION_2 :=
                           (XMIN => -1.25, XMAX => 1.25,
                            YMIN => -1.25, YMAX => 1.25);

 VIEWPORT              : NPC.RECTANGULAR_REGION_2 :=
                           (XMIN => 0.0, XMAX => 1.0,
                            YMIN => 0.0, YMAX => 1.0);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
EI                      : ERROR_NUMBER := 0;
VIEW_MAPPING_MATRIX     : TRANSFORMATION_MATRIX_2;
IDENTITY                : constant TRANSFORMATION_MATRIX_2 :=
                                ((1.0,0.0,0.0),
                                 (0.0,1.0,0.0),
                                 (0.0,0.0,1.0));
DO_CLIP                 : CLIPPING_INDICATOR := CLIP;
STAR                    : STRUCTURE_ID := 1;
TEXT_POSITION           : MC.POINT_2 := (0.0,-1.0);

begin

-- Open PHIGS and a workstation

OPEN_PHIGS (ERROR_FILE);
OPEN_WS (MY_WS_ID,SOME_CONNECTION,SOME_OUTPUT_TYPE);

-- Center the window around the origin

EVALUATE_VIEW_MAPPING_MATRIX (WINDOW,
                              VIEWPORT,
                              EI,
                              VIEW_MAPPING_MATRIX);


SET_VIEW_REPRESENTATION (MY_WS_ID,
                         1,
                         IDENTITY,
                         VIEW_MAPPING_MATRIX,
                         VIEWPORT,
                         DO_CLIP);

-- Define the colours to use

SET_COLOUR_REPRESENTATION (MY_WS_ID,
                           COLOUR_IND => 0,
                           COLOUR => (RGB,0.0,0.0,1.0));

SET_COLOUR_REPRESENTATION (MY_WS_ID,
                           COLOUR_IND => 1,
                           COLOUR => (RGB,1.0,1.0,0.0));

SET_COLOUR_REPRESENTATION (MY_WS_ID,
                           COLOUR_IND => 2,
                           COLOUR => (RGB,1.0,1.0,1.0));

-- Open and post a structure

OPEN_STRUCTURE (STAR);
POST_STRUCTURE (MY_WS_ID,STAR,1.0);

-- Set interior attributes

SET_INTERIOR_STYLE (SOLID);
SET_INTERIOR_COLOUR_INDEX (1);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
-- Draw the star

FILL_AREA (STAR_POINTS);

-- Select large characters centered under the star

SET_CHAR_HEIGHT (HEIGHT = > 0.15);
SET_TEXT_ALIGNMENT (ALIGNMENT = > (CENTRE,HALF));
SET_TEXT_COLOUR_INDEX (TEXT_COLOUR = > 2);

-- Draw the title

TEXT (TEXT_POSITION, "STAR");

-- Close the structure

CLOSE_STRUCTURE;

-- Close the workstation and exit PHIGS

CLOSE_WS (MY_WS_ID);
CLOSE_PHIGS;

end STAR;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

9593 PT*3 90 ■ 4851903 0575625 608 ■

C.2 Example Program 2: IRON

```
with PHIGS;
with PHIGS_TYPES;
use PHIGS;
use PHIGS_TYPES;

procedure IRON is

-- This program draws a horizontal bar chart illustrating costs within
-- the iron industry. The user can select the data to be displayed
-- using a PHIGS choice device. The plot is adapted from 'Scientific
-- American' May 1984 page 39.

-- Choice device must support prompt and echo type 3.

 -- Define the workstation variables and error logging file.

 MY_WS_ID            : constant WS_ID := 1;
 SOME_CONNECTION     : constant CONNECTION_ID := "TTY";
 SOME_OUTIN_TYPE     : constant WS_TYPE := 2;
 ERROR_FILE          : constant FILE_ID := "MY_ERROR_FILE";

 -- Declare and initialize objects for choice device

 CHOICE_STRING_COUNT       : constant := 3;
 CHOICE_DEVICE             : constant CHOICE_DEVICE_NUMBER := 1;
 CHOICE_ERROR              : ERROR_NUMBER;
 VALUE_TYPE                : constant RETURN_VALUE_TYPE := SET;
 CHOICE_MODE               : OPERATING_MODE;
 CHOICE_ECHO_SWITCH        : ECHO_SWITCH;
 INITIAL_CHOICE            : CHOICE_NUMBER;
 INITIAL_STATUS            : CHOICE_STATUS;
 INITIAL_PROMPT_ECHO_TYPE  : CHOICE_PROMPT_ECHO_TYPE;
 CHOICE_INPUT_RECORD       : CHOICE_DATA_RECORD;
 CHOICE_ECHO_AREA          : DC.RECTANGULAR_REGION_2;
 CHOICE_STRINGS            : constant CHOICE_PROMPT_STRING_LIST :=
                                 (LENGTH => 3, LIST =>
                                  ((4,"U.S."),
                                   (9,"W.GERMANY"),
                                   (5,"JAPAN")));

 CHOICE_RECORD  : CHOICE_DATA_RECORD;
 CHOICE         : CHOICE_NUMBER;
 CHOICE_REQUEST : CHOICE_REQUEST_STATUS;
 STATUS         : CHOICE_STATUS;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

9593 PT*3 90 ■ 4851903 0575626 544 ■


```
-- Declare colour objects

COLOUR  : COLOUR_REPRESENTATION;
WHITE   : constant COLOUR_INDEX := 0;
BLACK   : constant COLOUR_INDEX := 1;
RED     : constant COLOUR_INDEX := 2;

-- Declare and initialize view mapping

WINDOW_LIMITS          : VRC.RECTANGULAR_REGION_2 :=
                           (XMIN => -115.0, XMAX => 160.0,
                            YMIN => -2.0,   YMAX => 14.0);

VIEWPORT_LIMITS        : NPC.RECTANGULAR_REGION_2 :=
                           (XMIN => 0.0, XMAX => 1.0,
                            YMIN => 0.0, YMAX => 1.0);

EI                     : ERROR_NUMBER := 0;
VIEW_MAPPING_MATRIX    : TRANSFORMATION_MATRIX_2;
IDENTITY               : constant TRANSFORMATION_MATRIX_2 :=
                                  ((1.0,0.0,0.0),
                                   (0.0,1.0,0.0),
                                   (0.0,0.0,1.0));

DO_CLIP                : CLIPPING_INDICATOR := CLIP;

-- Declare and initialize object for bar data to plot

MAX_DATA : constant := 6;
type IRON_DATA is array (1..MAX_DATA) of MC_TYPE;
US_DATA_1          : IRON_DATA := (69.0, 50.0, 15.0, 53.0, 57.0, 150.0);
US_DATA_2          : IRON_DATA := (72.0, 50.0, 103.0, 0.0, 0.0, 56.0);
GERMANY_DATA_1     : IRON_DATA := (65.0, 42.0, 3.0, 89.0, 52.0, 93.0);
GERMANY_DATA_2     : IRON_DATA := (70.0, 53.0, 102.0, 0.0, 0.0, 49.0);
JAPAN_DATA_1       : IRON_DATA := (65.0, 47.0, 2.0, 60.0, 52.0, 55.0);
JAPAN_DATA_2       : IRON_DATA := (70.0, 57.0, 105.0, 0.0, 0.0, 41.0);

-- Define structures

subtype COUNTRY_DATA is STRUCTURE_ID range 1..3;
CHART_BORDER : constant STRUCTURE_ID := 4;
US           : constant STRUCTURE_ID := 1;
GERMANY      : constant STRUCTURE_ID := 2;
JAPAN        : constant STRUCTURE_ID := 3;
CURRENT      : STRUCTURE_ID := 0;

procedure BARS (LENGTH   : MC_TYPE;
                POSITION : MC.POINT_2) is


 LEFT_HALF  : TEXT_ALIGNMENT := (LEFT,HALF);
 BAR_POINTS : MC.POINT_LIST_2 (1..4);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
begin

 if LENGTH = 0.0 then
  SET_TEXT_ALIGNMENT (LEFT_HALF);
  TEXT (POSITION, "0");
 else
  BAR_POINTS := ((X = > 0.0,          Y = > POSITION.Y + 0.4),
                 (X = > LENGTH,       Y = > POSITION.Y + 0.4),
                 (X = > LENGTH,       Y = > POSITION.Y - 0.4),
                 (X = > 0.0,          Y = > POSITION.Y - 0.4));
  FILL_AREA (BAR_POINTS);
 end if;

end BARS;

procedure TICKS (TICK_MARK_POSITION : in out MC.POINT_LIST_2) is

 TICK_MARK_LABEL_POSITION : MC.POINT_2;

begin

 for I in 1..4 loop
   POLYLINE (TICK_MARK_POSITION);
   TICK_MARK_POSITION(1).X := TICK_MARK_POSITION(1).X + 50.0;
   TICK_MARK_POSITION(2).X := TICK_MARK_POSITION(2).X + 50.0;
 end loop;

 -- Draw tick mark labels

 TICK_MARK_LABEL_POSITION.X := 0.0;
 TICK_MARK_LABEL_POSITION.Y := MC_TYPE(TICK_MARK_POSITION(1).Y);
 TEXT (TICK_MARK_LABEL_POSITION, "0");

 TICK_MARK_LABEL_POSITION.X := 50.0;
 TEXT (TICK_MARK_LABEL_POSITION, "50");

 TICK_MARK_LABEL_POSITION.X := 100.0;
 TEXT (TICK_MARK_LABEL_POSITION, "100");

 TICK_MARK_LABEL_POSITION.X := 150.0;
 TEXT (TICK_MARK_LABEL_POSITION, "150");

end TICKS;

procedure BORDER is

-- Draws the border surrounding the data

 LABEL_POSITION      : MC.POINT_2;
 TITLE_POSITION      : MC.POINT_2 := (37.5, -2.0);
 HEIGHT              : MC.MAGNITUDE := 0.5;
 LEFT_HALF           : TEXT_ALIGNMENT := (LEFT,HALF);
 CENTRE_BOTTOM       : TEXT_ALIGNMENT := (CENTRE,BOTTOM);
 CENTRE_CAP          : TEXT_ALIGNMENT := (CENTRE,CAP);
 ONLY_IF_NOT_EMPTY   : CONTROL_FLAG := CONDITIONALLY;

 BOX_POINTS          : constant MC.POINT_LIST_2 :=
                         ((0.0,0.0),(150.0,0.0),(150.0,12.0),
                          (0.0,12.0),(0.0,0.0));
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
type LABELS is array (1..6) of INPUT_STRING;
BAR_LABELS              : constant LABELS :=
                              ((5,"LABOR"),
                               (8,"IRON ORE"),
                               (12,"COKE OR COAL"),
                               (15,"PURCHASED SCRAP"),
                               (11,"OTHER COSTS"),
                               (12,"OTHER ENERGY"));

TOP_TICK_MARK_START : MC.POINT_LIST_2 (1..2) := ((0.0,12.0),
                                                 (0.0,11.9));


BOTTOM_TICK_MARK_START : MC.POINT_LIST_2 (1..2) := ((0.0,0.0),
                                                    (0.0,0.1));

begin

 -- Draw the box surrounding the chart area

 POLYLINE (BOX_POINTS);

 --  Draw the bar labels centered on the bar and flush left

 SET_TEXT_ALIGNMENT (LEFT_HALF);
 SET_CHAR_HEIGHT (HEIGHT);
 SET_TEXT_COLOUR_INDEX (BLACK);

 LABEL_POSITION.X := -99.0;
 for I in 1..6 loop
   LABEL_POSITION.Y := MC_TYPE(2.0 * (FLOAT(I)-1.0) + 1.2);
   TEXT (LABEL_POSITION, BAR_LABELS(I).CONTENTS);
 end loop;

 --  Draw the top and bottom tick marks (bottom in red)

 SET_TEXT_ALIGNMENT (CENTRE_BOTTOM);

 -- Call procedures to draw ticks

 TICKS (TOP_TICK_MARK_START);

 SET_TEXT_ALIGNMENT (CENTRE_CAP);
 SET_TEXT_COLOUR_INDEX (RED);
 TICKS (BOTTOM_TICK_MARK_START);

 -- Draw the title

 SET_TEXT_COLOUR_INDEX (BLACK);
 SET_TEXT_ALIGNMENT (CENTRE_BOTTOM);
 TEXT (TITLE_POSITION, "PRODUCTION_COST");

end border;

procedure DRAW (DATA1 : in out IRON_DATA;
                DATA2 : in out IRON_DATA) is
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
  FILL_INDEX : INTERIOR_INDEX := 1;
  POSITION   : MC.POINT_2;

 begin

  -- Use bundled ASF's except for colour

  for I in ASPECT loop
    case I is
      when INTERIOR_COLOUR | TEXT_COLOUR
         => SET_INDIVIDUAL_ASF (I,INDIVIDUAL);
      when others => SET_INDIVIDUAL_ASF (I,BUNDLED);
    end case;
  end loop;

  -- Draw the black bars

  SET_INTERIOR_COLOUR_INDEX (BLACK);
  SET_TEXT_COLOUR_INDEX (BLACK);
  SET_INTERIOR_INDEX (FILL_INDEX);
  for I in 1..6 loop
    POSITION.Y := MC_TYPE(2.0 * (FLOAT(I)-1.0) + 1.6);
    -- Call the procedure that draws the bars
    BARS (DATA1(I),POSITION);
  end loop;

  -- Draw the red bars

  SET_INTERIOR_COLOUR_INDEX (RED);
  SET_TEXT_COLOUR_INDEX (RED);
  FILL_INDEX := 2;
  SET_INTERIOR_INDEX (FILL_INDEX);
  for I in 1..6 loop
    POSITION.Y := MC_TYPE(2.0 * (FLOAT(I)-1.0) + 1.6);
    -- Call the procedure that draws the bars
    BARS (DATA2(I),POSITION);
  end loop;

 end DRAW;

begin

 -- Open PHIGS and a workstation

 OPEN_PHIGS (ERROR_FILE);
 OPEN_WS (MY_WS_ID,SOME_CONNECTION,SOME_OUTIN_TYPE);

 -- Define the view

 EVALUATE_VIEW_MAPPING_MATRIX (WINDOW_LIMITS,
                               VIEWPORT_LIMITS,
                               EI,
                               VIEW_MAPPING_MATRIX);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
SET_VIEW_REPRESENTATION (MY_WS_ID,
                         1,
                         IDENTITY,
                         VIEW_MAPPING_MATRIX,
                         VIEWPORT_LIMITS,
                         DO_CLIP);

-- Define the colours to use

SET_COLOUR_REPRESENTATION (MY_WS_ID,
                           COLOUR_IND = > WHITE,
                           COLOUR = > (RGB,1.0,1.0,1.0));

SET_COLOUR_REPRESENTATION (MY_WS_ID,
                           COLOUR_IND = > BLACK,
                           COLOUR = > (RGB,0.0,0.0,0.0));

SET_COLOUR_REPRESENTATION (MY_WS_ID,
                           COLOUR_IND = > RED,
                           COLOUR = > (RGB,1.0,0.0,0.0));

-- Create the border structure

OPEN_STRUCTURE (CHART_BORDER);
BORDER;
CLOSE_STRUCTURE;

-- Create US data

OPEN_STRUCTURE (US);
DRAW (US_DATA_1,US_DATA_2);
CLOSE_STRUCTURE;

-- Create Germany data

OPEN_STRUCTURE (GERMANY);
DRAW (GERMANY_DATA_1,GERMANY_DATA_2);
CLOSE_STRUCTURE;

-- Create Japan data

OPEN_STRUCTURE (JAPAN);
DRAW (JAPAN_DATA_1,JAPAN_DATA_2);
CLOSE_STRUCTURE;
-- Initialize the choice device

INQ_CHOICE_DEVICE_STATE (MY_WS_ID,
                         CHOICE_DEVICE,
                         CHOICE_ERROR,
                         CHOICE_MODE,
                         CHOICE_ECHO_SWITCH,
                         INITIAL_STATUS,
                         INITIAL_CHOICE,
                         CHOICE_ECHO_AREA,
                         CHOICE_RECORD);

BUILD_CHOICE_DATA_RECORD (STRING_PROMPT_CHOICE,
                          CHOICE_STRINGS,
                          CHOICE_RECORD);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

9593 PT*3 90 ■ 4851903 0575631 901 ■


```
INITIALIZE_CHOICE (MY_WS_ID,
                   CHOICE_DEVICE,
                   INITIAL_STATUS,
                   INITIAL_CHOICE,
                   CHOICE_ECHO_AREA,
                   CHOICE_RECORD);

-- Post border

POST_STRUCTURE (MY_WS_ID,CHART_BORDER,1.0);

-- Get the user's choice

loop
  REQUEST_CHOICE (MY_WS_ID,CHOICE_DEVICE,STATUS,CHOICE);
  if STATUS = OK then
    if CURRENT in COUNTRY_DATA then
      UNPOST_STRUCTURE (MY_WS_ID,CURRENT);
    end if;
    POST_STRUCTURE (MY_WS_ID,STRUCTURE_ID(CHOICE),0.5);
    CURRENT := STRUCTURE_ID(CHOICE);
  else
    exit;
  end if;
end loop;

-- Close workstation and shut down PHIGS

CLOSE_WS (MY_WS_ID);
CLOSE_PHIGS;

end IRON;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

C.3 Example Program 3: DYNASTAR

```
with PHIGS;
with PHIGS_TYPES;
use PHIGS;
use PHIGS_TYPES;

procedure DYNASTAR is

-- This program draws a yellow star on a blue background and writes
-- the title "Star" in green under the star.  It changes the viewing
-- parameters to zoom and pan on the star independent of the title.
-- It then demonstrates some of the dynamics of PHIGS by rolling the
-- star back and forth across the display by modifying the star's
-- modeling transformation.

 -- Define the workstation variables and error logging file.

 MY_WS_ID              : constant WS_ID := 1;
 SOME_CONNECTION       : constant CONNECTION_ID := "DISPLAY";
 SOME_OUTPUT_TYPE      : constant WS_TYPE := 2;
 ERROR_FILE            : constant FILE_ID := "MY_ERROR_FILE";

 -- Define the points of the star.

 STAR_POINTS  : constant MC.POINT_LIST_2 :=
                    (( 0.951057,  0.309017 ),
                     (-0.951057,  0.309017 ),
                     ( 0.587785, -0.951057 ),
                     ( 0.0,       1.0      ),
                     (-0.587785, -0.951057 ));

 -- Define transformations

 IDENTITY                    : constant TRANSFORMATION_MATRIX_2 :=
                                    ((1.0,0.0,0.0),
                                     (0.0,1.0,0.0),
                                     (0.0,0.0,1.0));

 SCALE_TRANSFORMATION  : TRANSFORMATION_MATRIX_2;
 ROLL_TRANSFORMATION   : TRANSFORMATION_MATRIX_2 := IDENTITY;
 EI                    : ERROR_NUMBER := 0;
 VIEW_MAPPING_MATRIX   : TRANSFORMATION_MATRIX_2;
 DO_CLIP               : CLIPPING_INDICATOR := CLIP;

 -- Define static and dynamic views

 STATIC_VIEW_WINDOW : constant VRC.RECTANGULAR_REGION_2 :=
                                  (XMIN = > -1.25,  XMAX = > 1.25,
                                   YMIN = > -1.25,  YMAX = > 1.25);
 DYNAMIC_VIEW_WINDOW : VRC.RECTANGULAR_REGION_2 := STATIC_VIEW_WINDOW;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
OUR_VIEWPORT_LIMITS : NPC.RECTANGULAR_REGION_2 :=
                              (XMIN => 0.0, XMAX => 1.0,
                               YMIN => 0.0, YMAX => 1.0);

STATIC     : constant VIEW_INDEX := 2;
DYNAMIC    : constant VIEW_INDEX := 1;

ZOOM_WINDOW : VRC_TYPE;

-- Define the colours to be used

BLUE       : constant COLOUR_INDEX := 0;
YELLOW     : constant COLOUR_INDEX := 1;
GREEN      : constant COLOUR_INDEX := 2;

begin

-- Open PHIGS and a workstation

OPEN_PHIGS (ERROR_FILE);
OPEN_WS (MY_WS_ID,SOME_CONNECTION,SOME_OUTPUT_TYPE);

-- Load the view table

EVALUATE_VIEW_MAPPING_MATRIX (STATIC_VIEW_WINDOW,
                              OUR_VIEWPORT_LIMITS,
                              EI,
                              VIEW_MAPPING_MATRIX);

SET_VIEW_REPRESENTATION (MY_WS_ID,
                         STATIC,
                         IDENTITY,
                         VIEW_MAPPING_MATRIX,
                         OUR_VIEWPORT_LIMITS,
                         DO_CLIP);

EVALUATE_VIEW_MAPPING_MATRIX (DYNAMIC_VIEW_WINDOW,
                              OUR_VIEWPORT_LIMITS,
                              EI,
                              VIEW_MAPPING_MATRIX);

SET_VIEW_REPRESENTATION (MY_WS_ID,
                         DYNAMIC,
                         IDENTITY,
                         VIEW_MAPPING_MATRIX,
                         OUR_VIEWPORT_LIMITS,
                         DO_CLIP);

-- Load the colour table

SET_COLOUR_REPRESENTATION (MY_WS_ID,BLUE,(RGB,0.0,0.0,1.0));
SET_COLOUR_REPRESENTATION (MY_WS_ID,YELLOW,(RGB,1.0,1.0,0.0));
SET_COLOUR_REPRESENTATION (MY_WS_ID,GREEN,(RGB,0.0,1.0,0.0));

-- Open picture structure

OPEN_STRUCTURE (1);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
-- Select the static view for the title

SET_VIEW_INDEX (STATIC);

-- Select large green characters centered under the star

SET_CHAR_HEIGHT (0.15);
SET_TEXT_ALIGNMENT ((HORIZONTAL => CENTRE,
                     VERTICAL   => HALF));

SET_TEXT_COLOUR_INDEX (GREEN);

-- Generate the title

TEXT ((0.0,-1.1),"Star");

-- Select the dynamic view

SET_VIEW_INDEX (DYNAMIC);

-- Choose solid yellow style for star

SET_INTERIOR_STYLE (SOLID);
SET_INTERIOR_COLOUR_INDEX (YELLOW);

-- Draw the star

FILL_AREA (STAR_POINTS);

-- Close the structure

CLOSE_STRUCTURE;

-- Set the deferral and update modes so that changes happen immediately

SET_DISPLAY_UPDATE_STATE (MY_WS_ID,ASAP,NIVE);

-- Post the structure to the open workstation

POST_STRUCTURE (MY_WS_ID,1,1.0);

-- Successively change the view mapping to zoom in

for I in 2..10 loop
 ZOOM_WINDOW := 1.25 / VRC_TYPE(I);
 EVALUATE_VIEW_MAPPING_MATRIX
    (WINDOW_LIMITS     => (-ZOOM_WINDOW,ZOOM_WINDOW,
                           -ZOOM_WINDOW,ZOOM_WINDOW),
     VIEWPORT_LIMITS   => OUR_VIEWPORT_LIMITS,
     ERROR_INDICATOR   => EI,
     MAPPING_MATRIX    => VIEW_MAPPING_MATRIX);

 SET_VIEW_REPRESENTATION (MY_WS_ID,
                          VIEW_IND             => DYNAMIC,
                          ORIENTATION_MATRIX   => IDENTITY,
                          MAPPING_MATRIX       => VIEW_MAPPING_MATRIX,
                          CLIPPING_LIMITS      => OUR_VIEWPORT_LIMITS,
                          XY_CLIPPING          => DO_CLIP);
end loop;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
-- and then out

for I in reverse 1..9 loop
 ZOOM_WINDOW := 1.25 / VRC_TYPE(I);
 EVALUATE_VIEW_MAPPING_MATRIX
    (WINDOW_LIMITS      => (-ZOOM_WINDOW,ZOOM_WINDOW,
                            -ZOOM_WINDOW,ZOOM_WINDOW),
     VIEWPORT_LIMITS    => OUR_VIEWPORT_LIMITS,
     ERROR_INDICATOR    => EI,
     MAPPING_MATRIX     => VIEW_MAPPING_MATRIX);

 SET_VIEW_REPRESENTATION (MY_WS_ID,
                          VIEW_IND              => DYNAMIC,
                          ORIENTATION_MATRIX    => IDENTITY,
                          MAPPING_MATRIX        => VIEW_MAPPING_MATRIX,
                          CLIPPING_LIMITS       => OUR_VIEWPORT_LIMITS,
                          XY_CLIPPING           => DO_CLIP);
end loop;

-- Successively change the view mapping to pan back and forth
-- across the display.

for I in 1..5 loop
 EVALUATE_VIEW_MAPPING_MATRIX
    (WINDOW_LIMITS      => (XMIN  => -1.25 - VRC_TYPE(I)*0.1,
                            XMAX =>  1.25 - VRC_TYPE(I)*0.1,
                            YMIN  => -1.25,
                            YMAX =>  1.25),
     VIEWPORT_LIMITS    => OUR_VIEWPORT_LIMITS,
     ERROR_INDICATOR    => EI,
     MAPPING_MATRIX     => VIEW_MAPPING_MATRIX);

 SET_VIEW_REPRESENTATION (MY_WS_ID,
                          VIEW_IND              => DYNAMIC,
                          ORIENTATION_MATRIX    => IDENTITY,
                          MAPPING_MATRIX        => VIEW_MAPPING_MATRIX,
                          CLIPPING_LIMITS       => OUR_VIEWPORT_LIMITS,
                          XY_CLIPPING           => DO_CLIP);
end loop;

for I in reverse 0..4 loop
 EVALUATE_VIEW_MAPPING_MATRIX
    (WINDOW_LIMITS      => (XMIN  => -1.25 - VRC_TYPE(I)*0.1,
                            XMAX =>  1.25 - VRC_TYPE(I)*0.1,
                            YMIN  => -1.25,
                            YMAX =>  1.25),
     VIEWPORT_LIMITS    => OUR_VIEWPORT_LIMITS,
     ERROR_INDICATOR    => EI,
     MAPPING_MATRIX     => VIEW_MAPPING_MATRIX);

 SET_VIEW_REPRESENTATION (MY_WS_ID,
                          VIEW_IND              => DYNAMIC,
                          ORIENTATION_MATRIX    => IDENTITY,
                          MAPPING_MATRIX        => VIEW_MAPPING_MATRIX,
                          CLIPPING_LIMITS       => OUR_VIEWPORT_LIMITS,
                          XY_CLIPPING           => DO_CLIP);
end loop;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0575636 493 ■


```
-- Successively change the modeling transformation to "roll"
-- the Star back and forth across the display surface.

-- Open the structure for editing

OPEN_STRUCTURE (1);

-- Identify the position at which the element pointer should be set

SET_ELEMENT_POINTER (9);

-- First scale the star down to make it more "rollable".

SCALE ((0.5,0.5),EI,SCALE_TRANSFORMATION);
SET_LOCAL_TRANSFORMATION (SCALE_TRANSFORMATION,REPLACE);

-- Insert the rolling transformation

SET_LOCAL_TRANSFORMATION (ROLL_TRANSFORMATION,POSTCONCATENATE);

-- Set up deferral mode so the deletes don't show

SET_DISPLAY_UPDATE_STATE (MY_WS_ID, WAIT, NIVE);

-- Loop to roll to the left

for I in 1..10 loop
 BUILD_TRANSFORMATION_MATRIX
    (FIXED_POINT          => (0.0,0.0),
     SHIFT_VECTOR         => (-MC_TYPE(I)*0.05,0.0),
     ANGLE_Z              => -0.5236*ANGLE(I),
     SCALE_FACTORS        => (1.0,1.0),
     ERROR_INDICATOR      => EI,
     MATRIX               => ROLL_TRANSFORMATION);
 SET_ELEMENT_POINTER (10);
 DELETE_ELEMENT;

 SET_LOCAL_TRANSFORMATION (ROLL_TRANSFORMATION,
                           POSTCONCATENATE);
 UPDATE_WS (MY_WS_ID,PERFORM);
end loop;

-- Roll back to the right

for I in reverse 0..9 loop
 BUILD_TRANSFORMATION_MATRIX
    (FIXED_POINT          => (0.0,0.0),
     SHIFT_VECTOR         => (-MC_TYPE(I)*0.05,0.0),
     ANGLE_Z              => 0.5236*ANGLE(I),
     SCALE_FACTORS        => (1.0,1.0),
     ERROR_INDICATOR      => EI,
     MATRIX               => ROLL_TRANSFORMATION);
 SET_ELEMENT_POINTER (10);
 DELETE_ELEMENT;

 SET_LOCAL_TRANSFORMATION (ROLL_TRANSFORMATION,
                           POSTCONCATENATE);
 UPDATE_WS (MY_WS_ID,PERFORM);
end loop;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
-- Now roll all the way to the right

for I in 1..10 loop
 BUILD_TRANSFORMATION_MATRIX
    (FIXED_POINT          => (0.0,0.0),
     SHIFT_VECTOR         => (MC_TYPE(I)*0.05,0.0),
     ANGLE_Z              => 0.5236*ANGLE(I),
     SCALE_FACTORS        => (1.0,1.0),
     ERROR_INDICATOR      => EI,
     MATRIX               => ROLL_TRANSFORMATION);
 SET_ELEMENT_POINTER (10);
 DELETE_ELEMENT;

 SET_LOCAL_TRANSFORMATION (ROLL_TRANSFORMATION,
                           POSTCONCATENATE);
 UPDATE_WS (MY_WS_ID,PERFORM);
end loop;

-- Now roll back to the left to where it all started

for I in reverse 0..9 loop
 BUILD_TRANSFORMATION_MATRIX
    (FIXED_POINT          => (0.0,0.0),
     SHIFT_VECTOR         => (MC_TYPE(I)*0.05,0.0),
     ANGLE_Z              => -0.5236*ANGLE(I),
     SCALE_FACTORS        => (1.0,1.0),
     ERROR_INDICATOR      => EI,
     MATRIX               => ROLL_TRANSFORMATION);
 SET_ELEMENT_POINTER (10);
 DELETE_ELEMENT;

 SET_LOCAL_TRANSFORMATION (ROLL_TRANSFORMATION,
                           POSTCONCATENATE);
 UPDATE_WS (MY_WS_ID,PERFORM);
end loop;

-- Close the workstation and PHIGS

CLOSE_WS (MY_WS_ID);
CLOSE_PHIGS;

end DYNASTAR;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0575638 266 ■

C.4 Example Program 4: TRANSFORM_POLYLINE

```
with UNCHECKED_DEALLOCATION;
with TEXT_IO;
with PHIGS;
with PHIGS_TYPES;
use TEXT_IO;
use PHIGS;
use PHIGS_TYPES;

procedure TRANSFORM_POLYLINE is

-- This program applies one or more transformations to a closed polyline
-- connecting a user-supplied set of points and then draws the result on
-- a plotter.

-- The implementation must support at least an OUTIN workstation and an
-- OUTPUT workstation.

-- This example assumes that all the required input devices are available
-- on the same OUTIN workstation.

 -- Define the workstation variables and error logging file.

 DISPLAY_WS_ID                : constant WS_ID := 1;
 HARDCOPY_WS_ID               : constant WS_ID := 2;
 SOME_DISPLAY_CONNECTION      : constant CONNECTION_ID := "UNIT_1";
 SOME_HARDCOPY_CONNECTION     : constant CONNECTION_ID := "UNIT_2";
 SOME_OUTIN_TYPE              : constant WS_TYPE := 1;
 SOME_OUTPUT_TYPE             : constant WS_TYPE := 2;
 ERROR_FILE                   : constant FILE_ID := "MY_ERROR_FILE";

 -- Define view parameters and other variables

 WINDOW                       : VRC.RECTANGULAR_REGION_2 :=
                                  (XMIN => -10.0, XMAX => 10.0,
                                   YMIN => -10.0, YMAX => 10.0);

 VIEWPORT                     : NPC.RECTANGULAR_REGION_2 :=
                                  (XMIN => 0.1, XMAX => 0.9,
                                   YMIN => 0.1, YMAX => 0.9);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
EI                          : ERROR_NUMBER := 0;
VIEW_MAPPING_MATRIX         : TRANSFORMATION_MATRIX_2;
IDENTITY                    : constant TRANSFORMATION_MATRIX_2 :=
                                    ((1.0,0.0,0.0),
                                     (0.0,1.0,0.0),
                                     (0.0,0.0,1.0));
DO_CLIP                     : CLIPPING_INDICATOR := CLIP;

ACCUMULATED_TRANSFORM       : TRANSFORMATION_MATRIX_2 := IDENTITY;
ORIGIN                      : constant MC.POINT_2 := (0.0, 0.0);

POLYLINE_OBJECT             : STRUCTURE_ID := 1;
POLYLINE_LABEL              : STRUCTURE_ID := 2;

VALID_POLYLINE              : INPUT_STATUS;


  function DEFINE_POLYLINE return INPUT_STATUS is

    INDATA_STATUS  : INPUT_STATUS;
    STROKE_VIEW    : VIEW_INDEX;
    INPUT_STROKE   : WC.ACCESS_POINT_LIST_2;
    CLOSED_STROKE  : MC.ACCESS_POINT_LIST_2;

    procedure DEALLOCATE is new UNCHECKED_DEALLOCATION (WC.POINT_LIST_2,
                                                        WC.ACCESS_POINT_LIST_2);

  begin

    -- Establish loop to get a valid stroke.  Loop will continue until a
    -- stroke in view 1 of more than one point is entered.

    loop
      PUT_LINE ("Stroke input needed for polyline ...");
      REQUEST_STROKE (DISPLAY_WS_ID,
                      DEVICE         => 1,
                      STATUS         => INDATA_STATUS,
                      VIEW_IND       => STROKE_VIEW,
                      STROKE_POINTS  => INPUT_STROKE);

      PUT ("Got ");
      PUT (POSITIVE'IMAGE(INPUT_STROKE'LAST));
      PUT_LINE (" points.");
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0575640 914 ■


```
    -- Check for valid stroke

    if INDATA_STATUS = OK then

     if (STROKE_VIEW = 1) and (INPUT_STROKE'LAST > = 2) then

       -- Close the stroke

       CLOSED_STROKE := new MC.POINT_LIST_2(1 .. INPUT_STROKE'last+1);
       for I in 1 .. INPUT_STROKE'last loop
         CLOSED_STROKE (I) := (MC_TYPE(INPUT_STROKE(I).X),
                               MC_TYPE(INPUT_STROKE(I).Y));
       end loop;
       CLOSED_STROKE (INPUT_STROKE'last+1) := CLOSED_STROKE (1);

       -- Insert polyline in open structure

       POLYLINE (CLOSED_STROKE.all);

       -- Free memory allocated for stroke

       DEALLOCATE (INPUT_STROKE);
       DEALLOCATE (CLOSED_STROKE);

       exit;
     end if;
    else
     exit;
    end if;

  end loop;

  return INDATA_STATUS;

end DEFINE_POLYLINE;


procedure APPLY_TRANSFORMATIONS is

  CHOICE_INDATA_STATUS     : CHOICE_REQUEST_STATUS;
  SELECTION                : CHOICE_NUMBER;
  TRANSLATE                : constant CHOICE_NUMBER := 1;
  ROTATE                   : constant CHOICE_NUMBER := 2;
  SCALE                    : constant CHOICE_NUMBER := 3;
  TERMINATION              : constant CHOICE_NUMBER := 4;

  STROKE_INDATA_STATUS     : INPUT_STATUS;
  STROKE_VIEW              : VIEW_INDEX;
  INPUT_STROKE             : WC.ACCESS_POINT_LIST_2;
  NO_TRANSLATION           : constant MC.VECTOR_2 := (0.0, 0.0);
  NO_SCALE                 : constant MC.VECTOR_2 := (1.0, 1.0);
  NO_ROTATION              : constant ANGLE := 0.0;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
VALUATOR_INDATA_STATUS : INPUT_STATUS;
VALUATOR_INPUT         : VALUATOR_VALUE;
ROTATION_ANGLE         : ANGLE;
TWO_PI                 : constant ANGLE := 6.2831853;

SCALE_AMOUNT           : MC.VECTOR_2;

package ANGLE_IO is new FLOAT_IO(ANGLE);
package MC_IO is new FLOAT_IO(MC_TYPE);
use ANGLE_IO, MC_IO;

begin

-- Apply repeated transformations to the structure.

-- Set the element pointer on the local modelling transformation element and
-- provide for repeated replacement of this element.

SET_ELEMENT_POINTER (1);
SET_EDIT_MODE (REPLACE);

-- Choose the transformation type

TRANSFORM: loop

   PUT_LINE ("Choice input needed ...");
   PUT_LINE ("     1 - translate");
   PUT_LINE ("     2 - rotate");
   PUT_LINE ("     3 - scale");
   PUT_LINE ("     4 - EXIT");

   REQUEST_CHOICE (DISPLAY_WS_ID,
                   DEVICE  => 1,
                   STATUS  => CHOICE_INDATA_STATUS,
                   CHOICE  => SELECTION);

   PUT ("Got choice = ");
   PUT_LINE (CHOICE_NUMBER'IMAGE(SELECTION));

   -- Provide for exit if selection not ok

   if CHOICE_INDATA_STATUS /= OK then
    SELECTION := 4;
   end if;

   -- Perform requested action

   case SELECTION is

     when TRANSLATE =>

       loop
         PUT_LINE ("Stroke input needed.  2 pts for translate ...");
         REQUEST_STROKE (DISPLAY_WS_ID,
                         DEVICE         => 1,
                         STATUS         => STROKE_INDATA_STATUS,
                         VIEW_IND       => STROKE_VIEW,
                         STROKE_POINTS  => INPUT_STROKE);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0575642 797 ■


```
        PUT ("Got ");
        PUT (POSITIVE'IMAGE(INPUT_STROKE'LAST));
        PUT_LINE (" points.");

        -- Exit if invalid stroke

        if STROKE_INDATA_STATUS /= OK then
          exit TRANSFORM;
        elsif (STROKE_VIEW = 1) and (INPUT_STROKE'LAST = 2) then
          COMPOSE_TRANSFORMATION_MATRIX
              (ACCUMULATED_TRANSFORM,
               ORIGIN,
               (MC_TYPE(INPUT_STROKE(2).X - INPUT_STROKE(1).X),
                MC_TYPE(INPUT_STROKE(2).Y - INPUT_STROKE(1).Y)),
               NO_ROTATION,
               NO_SCALE,
               EI,
               ACCUMULATED_TRANSFORM);
          exit;
        end if;
      end loop;

    when ROTATE = >

      PUT_LINE ("Valuator input needed for angle ...");
      REQUEST_VALUATOR (DISPLAY_WS_ID,
                        DEVICE  = > 1,
                        STATUS  = > VALUATOR_INDATA_STATUS,
                        VALUE   = > VALUATOR_INPUT);
      ROTATION_ANGLE := ANGLE(VALUATOR_INPUT) * TWO_PI;
      PUT ("Got angle: ");
      PUT (ROTATION_ANGLE, EXP = > 0); NEW_LINE;

      COMPOSE_TRANSFORMATION_MATRIX (ACCUMULATED_TRANSFORM,
                                     ORIGIN,
                                     NO_TRANSLATION,
                                     ROTATION_ANGLE,
                                     NO_SCALE,
                                     EI,
                                     ACCUMULATED_TRANSFORM);

    when SCALE = >

      loop
        PUT_LINE ("Valuator input needed for X scale factor ...");
        REQUEST_VALUATOR (DISPLAY_WS_ID,
                          DEVICE  = > 1,
                          STATUS  = > VALUATOR_INDATA_STATUS,
                          VALUE   = > VALUATOR_INPUT);
        SCALE_AMOUNT.X := MC_TYPE(VALUATOR_INPUT * 2.0);
        PUT ("Got X scale factor: ");
        PUT (SCALE_AMOUNT.X, EXP = > 0); NEW_LINE;

        exit when VALUATOR_INDATA_STATUS = OK;
      end loop;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
            loop
              PUT_LINE ("Valuator input needed for Y scale factor ...");
              REQUEST_VALUATOR (DISPLAY_WS_ID,
                                DEVICE  => 1,
                                STATUS  => VALUATOR_INDATA_STATUS,
                                VALUE   => VALUATOR_INPUT);
              SCALE_AMOUNT.Y := MC_TYPE(VALUATOR_INPUT * 2.0);
              PUT ("Got Y scale factor: ");
              PUT (SCALE_AMOUNT.Y, EXP => 0); NEW_LINE;
              exit when VALUATOR_INDATA_STATUS = OK;
            end loop;

            COMPOSE_TRANSFORMATION_MATRIX (ACCUMULATED_TRANSFORM,
                                           ORIGIN,
                                           NO_TRANSLATION,
                                           NO_ROTATION,
                                           SCALE_AMOUNT,
                                           EI,
                                           ACCUMULATED_TRANSFORM);

          when others =>

            null;

        end case;

        -- Reset the local transformation

        SET_LOCAL_TRANSFORMATION (ACCUMULATED_TRANSFORM,
        POSTCONCATENATE);
        UPDATE_WS (DISPLAY_WS_ID, PERFORM);

      end loop TRANSFORM;

      return;

    end APPLY_TRANSFORMATIONS;


begin

  -- Open PHIGS and a workstation

  OPEN_PHIGS (ERROR_FILE);
  OPEN_WS (DISPLAY_WS_ID, SOME_DISPLAY_CONNECTION, SOME_OUTIN_TYPE);

  -- Post the structure to the open workstation

  POST_STRUCTURE (DISPLAY_WS_ID, POLYLINE_OBJECT, 1.0);

  -- Define a 2D viewing transformations centered around the origin.

  EVALUATE_VIEW_MAPPING_MATRIX (WINDOW,
                                VIEWPORT,
                                EI,
                                VIEW_MAPPING_MATRIX);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
SET_VIEW_REPRESENTATION (DISPLAY_WS_ID,
                         1,
                         IDENTITY,
                         VIEW_MAPPING_MATRIX,
                         VIEWPORT,
                         DO_CLIP);

-- Set the display update state such that changes happen immediately

SET_DISPLAY_UPDATE_STATE (DISPLAY_WS_ID,
                          DEFERRAL      => ASAP,
                          MODIFICATION  => UWOR);

-- Open the structure which will contain the polyline

OPEN_STRUCTURE (POLYLINE_OBJECT);

-- Start with a local transformation

SET_LOCAL_TRANSFORMATION (ACCUMULATED_TRANSFORM, POSTCONCATENATE);

-- Use only bundled aspects

for I in ASPECT loop
  SET_INDIVIDUAL_ASF (I, BUNDLED);
end loop;

-- Use polyline bundle index 3

SET_POLYLINE_INDEX (3);

VALID_POLYLINE := DEFINE_POLYLINE;
if VALID_POLYLINE = OK then

  -- Create desired transformation

  APPLY_TRANSFORMATIONS;

  -- All done modifying so close structure

  CLOSE_STRUCTURE;

  -- Now plot the structure on a hardcopy device

  PUT_LINE ("Plotting on hardcopy device ...");
  OPEN_WS (HARDCOPY_WS_ID, SOME_HARDCOPY_CONNECTION, SOME_OUTPUT_TYPE);

  -- Use same view as for OUTIN workstation

  SET_VIEW_REPRESENTATION (DISPLAY_WS_ID,
                           1,
                           IDENTITY,
                           VIEW_MAPPING_MATRIX,
                           VIEWPORT,
                           DO_CLIP);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
  -- Set the hardcopy representations

  SET_COLOUR_MODEL (HARDCOPY_WS_ID, RGB);
  SET_COLOUR_REPRESENTATION (HARDCOPY_WS_ID,
                             COLOUR_IND  => 1,
                             COLOUR      => (RGB, 1.0, 0.0, 0.0));

  SET_POLYLINE_REPRESENTATION (HARDCOPY_WS_ID,
                               POLYLINE_IND  => 3,
                               TYPE_OF_LINE  => SOLID_LINE,
                               LINEWIDTH_SF  => 1.5,
                               LINE_COLOUR   => 1);

  SET_TEXT_REPRESENTATION (HARDCOPY_WS_ID,
                           TEXT_IND      => 2,
                           FONT          => 1,
                           PRECISION     => STROKE_PRECISION,
                           EXPANSION     => 1.0,
                           SPACING       => 0.0,
                           TEXT_COLOUR   => 1);

  -- Plot the transformed polyline

  POST_STRUCTURE (HARDCOPY_WS_ID, POLYLINE_OBJECT, 1.0);

  -- Label the transformed polyline with some text

  POST_STRUCTURE (HARDCOPY_WS_ID, POLYLINE_LABEL, 1.0);
  OPEN_STRUCTURE (POLYLINE_LABEL);
  SET_CHAR_HEIGHT (1.0);
  SET_TEXT_INDEX (2);
  SET_TEXT_ALIGNMENT ((HORIZONTAL => LEFT, VERTICAL => BASE));
  TEXT (ORIGIN, "This is a polyline.");
  CLOSE_STRUCTURE;
  UPDATE_WS (HARDCOPY_WS_ID, PERFORM);

 end if;

 -- Close down PHIGS

 CLOSE_WS (DISPLAY_WS_ID);
 CLOSE_WS (HARDCOPY_WS_ID);
 CLOSE_PHIGS;

 return;

end TRANSFORM_POLYLINE;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0575646 332 ■

C.5 Example Program 5: SHOW_LINETYPES

```
with TEXT_IO;
use TEXT_IO;
with PHIGS;
with PHIGS_TYPES;
use PHIGS;
use PHIGS_TYPES;

procedure SHOW_LINETYPES is

-- This program illustrates the available linetypes on a user-selected
-- workstation. It contains PHIGS initialization and shutdown modules
-- and demonstartes how to program modules which do not change any state
-- list entries.

 SELECTED_TYPE : WS_TYPE;
 EI            : ERROR_NUMBER;
 SELECTED_WS   : WS_ID := 1;


    procedure BEGIN_SESSION (SELECTED_TYPE :    out WS_TYPE;
                             EI                : in out ERROR_NUMBER) is

     ERROR_FILE          : constant FILE_ID := "MY_ERROR_FILE";
     AVAILABLE_TYPES     : WS_TYPES.LIST_OF;
     CURRENT_TYPE        : WS_TYPE;
     CURRENT_CATEGORY    : WS_CATEGORY;
     INPUT_FIELD         : STRING(1..80);
     LAST_CHAR           : NATURAL;
     CONNECTION          : CONNECTION_ID(1..80);
     CURRENT_STATE       : WS_STATE;

    begin

     -- Begin PHIGS session

     OPEN_PHIGS (ERROR_FILE);

     -- Get list of all available workstation types and print them

     INQ_LIST_OF_AVAILABLE_WS_TYPES (EI, AVAILABLE_TYPES);
     if EI /= 0 then
       return;
     end if;

     PUT_LINE (" The available OUTPUT and OUTIN workstation types are:");
     for I in 1 .. WS_TYPES.SIZE_OF_LIST (AVAILABLE_TYPES) loop
       CURRENT_TYPE := WS_TYPES.LIST_ELEMENT (I, AVAILABLE_TYPES);
       INQ_WS_CATEGORY (CURRENT_TYPE, EI, CURRENT_CATEGORY);
       if CURRENT_CATEGORY = OUTPUT or else CURRENT_CATEGORY = OUTIN then
```


--`,,`--`,,`,,`,`,,`---

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

9593 PT*3 90 ■ 4851903 0575647 279 ■


```
      PUT_LINE (WS_TYPE'IMAGE(CURRENT_TYPE));
    end if;
  end loop;

  -- Choose one workstation and open it

  PUT (" Please enter workstation type: ");
  GET_LINE (INPUT_FIELD, LAST_CHAR);
  CURRENT_TYPE := WS_TYPE'VALUE(INPUT_FIELD(1 .. LAST_CHAR));
  PUT (" Please enter connection identifier for the workstation: ");
  GET_LINE (INPUT_FIELD, LAST_CHAR);
  CONNECTION := CONNECTION_ID(INPUT_FIELD(1 .. LAST_CHAR));
  OPEN_WS (SELECTED_WS, CONNECTION, CURRENT_TYPE);

  -- Check operating state to ensure successful opening

  INQ_WS_STATE_VALUE (CURRENT_STATE);
  if CURRENT_STATE /= WSOP then
    EI := 3;
  else
    EI := 0;
  end if;

return;

end BEGIN_SESSION;


procedure END_SESSION is

begin

  CLOSE_WS (SELECTED_WS);
  CLOSE_PHIGS;

  return;

end END_SESSION;


procedure LINETYPE_DEMO (DEMO_WS_TYPE : in     WS_TYPE;
                         EI           : in out ERROR_NUMBER) is

-- Demonstrate all available linetypes.

-- This module assumes one workstation with workstation identifier specified
-- by global variable SELECTED_WS and workstation type DEMO_WS_TYPE to be
-- open.

-- The display surface is not cleared and the output stretches across the
-- current workstation viewport.

  UP_TO_DATE          : UPDATE_STATE;
  REQ_WINDOW_LIMITS   : NPC.RECTANGULAR_REGION_2;
  CUR_WINDOW_LIMITS   : NPC.RECTANGULAR_REGION_2;
  REQ_VIEWPORT_LIMITS : DC.RECTANGULAR_REGION_2;
  CUR_VIEWPORT_LIMITS : DC.RECTANGULAR_REGION_2;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
    NUMBER_LINETYPES      : PHIGS_INTEGER;
    LINETYPE_LIST         : LINETYPES.LIST_OF;
    NUMBER_WIDTHS         : PHIGS_INTEGER;
    NORMAL_WIDTH          : DC.MAGNITUDE;
    WIDTH_RANGE           : DC.RANGE_OF_MAGNITUDES;
    NUMBER_INDICES        : PHIGS_NATURAL;
    DEMO_LINE             : MC.POINT_LIST_2(1..2);
    DESIRED_HEIGHT        : NPC_TYPE;
    DISTANCE              : NPC_TYPE;
    CURRENT_LINETYPE      : LINETYPE;

  begin

    -- Get workstation transformation information

    INQ_WS_TRANSFORMATION (SELECTED_WS,
                           EI,
                           UP_TO_DATE,
                           REQ_WINDOW_LIMITS,
                           CUR_WINDOW_LIMITS,
                           REQ_VIEWPORT_LIMITS,
                           CUR_VIEWPORT_LIMITS);
    if EI /= 0 then
      return;
    end if;

    -- Get polyline facilities

    INQ_POLYLINE_FACILITIES (DEMO_WS_TYPE,
                             EI,
                             NUMBER_LINETYPES,
                             LINETYPE_LIST,
                             NUMBER_WIDTHS,
                             NORMAL_WIDTH,
                             WIDTH_RANGE,
                             NUMBER_INDICES);

    -- Prepare a structure for the linetype images

    POST_STRUCTURE (SELECTED_WS, 1, 1.0);
    OPEN_STRUCTURE (1);

    -- Set all ASF's to INDIVIDUAL

    for I in ASPECT loop
      SET_INDIVIDUAL_ASF (I, INDIVIDUAL);
    end loop;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
-- Set reasonable values for non-linetype attributes

SET_LINEWIDTH_SCALE_FACTOR (1.0);
SET_POLYLINE_COLOUR_INDEX (1);
SET_CHAR_UP_VECTOR ((0.0, 1.0));
SET_TEXT_PATH (RIGHT);
SET_TEXT_ALIGNMENT ((LEFT, HALF));
SET_TEXT_FONT (1);
SET_TEXT_PRECISION (CHAR_PRECISION);
SET_CHAR_EXPANSION_FACTOR (1.0);
SET_CHAR_SPACING (0.0);
SET_TEXT_COLOUR_INDEX (1);

-- Compute distance between lines

DISTANCE := (CUR_WINDOW_LIMITS.XMAX - CUR_WINDOW_LIMITS.XMIN)
                / NPC_TYPE(NUMBER_LINETYPES);

-- Set character height to half of the distance between lines, but not
-- more than 1/20th of the height of the current workstation viewport.

if DISTANCE/2.0 < (CUR_WINDOW_LIMITS.XMAX-CUR_WINDOW_LIMITS.XMIN)/20.0 then
  DESIRED_HEIGHT := DISTANCE / 2.0;
else
  DESIRED_HEIGHT := (CUR_WINDOW_LIMITS.XMAX-CUR_WINDOW_LIMITS.XMIN)/20.0;
end if;
SET_CHAR_HEIGHT (MC.MAGNITUDE(DESIRED_HEIGHT));

-- Lines stretch from the left bound to the middle of the current workstation
-- viewport.

DEMO_LINE(1).X := MC_TYPE(CUR_WINDOW_LIMITS.XMIN);
DEMO_LINE(2).X
    := MC_TYPE((CUR_WINDOW_LIMITS.XMIN+CUR_WINDOW_LIMITS.XMAX))/2.0;
DEMO_LINE(1).Y := MC_TYPE(CUR_WINDOW_LIMITS.YMAX - DISTANCE / 2.0);

-- Display each available linetype

for I in 1 .. POSITIVE(NUMBER_LINETYPES) loop
  CURRENT_LINETYPE := LINETYPES.LIST_ELEMENT(I, LINETYPE_LIST);
  SET_LINETYPE (CURRENT_LINETYPE);
  DEMO_LINE(2).Y := DEMO_LINE(1).Y;
  POLYLINE (DEMO_LINE);
  DEMO_LINE(1).Y := DEMO_LINE(1).Y - MC_TYPE(DISTANCE);

  -- Annotate linetype

  TEXT (DEMO_LINE(2), PHIGS_STRING(LINETYPE'IMAGE(CURRENT_LINETYPE)));

end loop;

EI := 0;

return;

end LINETYPE_DEMO;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*3 90 ■ 4851903 0575650 863 ■


```
begin
 -- Start up
 BEGIN_SESSION (SELECTED_TYPE, EI);
 -- Proceed if valid
 if EI /= 0 then
  -- Perform the demonstration of linetypes
  LINETYPE_DEMO (SELECTED_TYPE, EI);
 end if;
 -- Shut down
 END_SESSION;
 return;
end SHOW_LINETYPES;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

9593 PT*3 90 ■ 4851903 0575651 7TT ■

Annex D

(informative)

PHIGS Multi-Tasking

The binding of PHIGS functions as subprograms in an Ada package has a straight-forward implementation: PHIGS "state" data are declared as variables local to the package body, and they are directly accessed and updated by the bodies of the PHIGS subprograms. This approach will work when application programs use only sequential control structures. The problem is that concurrent calls on PHIGS subprograms may cause a state variable to be corrupted; e.g., by simultaneous attempts to write to it. This problem exists whether the concurrency is actual (with multiple processors) or simulated (via multiplexed execution on a single processor).

There is an implementation technique that overcomes this problem without changing the PHIGS interface as seen by the Ada application program. In short, the idea is to protect the package body data (i.e., the state variables) by localizing them to a task contained in the package body. For each subprogram that accesses the data, there will be a corresponding entry declared in the task. The same name can be used for the entry and the subprogram, taking advantage of Ada's overloading facility. The task body takes the form of a "monitor" -- i.e., a loop containing a selective wait with an accept branch for each entry. The accept statement performs the actual reading or writing of the state information as required by the corresponding PHIGS subprogram. The body of each PHIGS subprogram comprises simply a call of the identically named task entry. Thus, even if two tasks from a user application program concurrently call subprograms that update or access state variables, these will result in entry calls that are queued in first come / first served fashion. There is no danger of corrupting the state variables.

To illustrate this technique, the following example shows how a skeletal version of the PHIGS package might be written.

```
with PHIGS_TYPES;
use PHIGS_TYPES;
package PHIGS is


 procedure OPEN_PHIGS
   (ERROR_FILE              : in FILE_ID := DEFAULT_ERROR_FILE;
    AMOUNT_OF_MEMORY : in PHIGS_NATURAL := DEFAULT_MEMORY_UNITS);


 procedure OPEN_WS
   (WS              : in WS_ID;
    CONNECTION : in VARIABLE_CONNECTION_ID;
    TYPE_OF_WS  : in WS_TYPE);

 ...

end PHIGS;

-- Version for sequential application programs:

with ERROR_HANDLING;
package body PHIGS is
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
-- State variables:

CURRENT_OPERATING_STATE  : OPERATING_STATE       := PHCL;
CURRENT_WS_STATE         : WS_STATE             := WSCL;
SET_OF_OPEN_WORKSTATIONS : WS_IDS.LIST_OF       := WS_IDS.NULL_LIST;


procedure OPEN_PHIGS
  (ERROR_FILE          : in FILE_ID := DEFAULT_ERROR_FILE;
   AMOUNT_OF_MEMORY    : in PHIGS_NATURAL := DEFAULT_MEMORY_UNITS) is


begin
 ...

 if CURRENT_OPERATING_STATE /= PHCL then
  ERROR_HANDLING (1, "OPEN_PHIGS");
 else
  CURRENT_OPERATING_STATE := PHOP;
 end if;
 ...

end OPEN_PHIGS;


procedure OPEN_WS
  (WS              : in WS_ID;
   CONNECTION      : in VARIABLE_CONNECTION_ID;
   TYPE_OF_WS      : in WS_TYPE) is


begin
 ...

 if CURRENT_OPERATING_STATE /= PHOP then
  ERROR_HANDLING (8, "OPEN_WS");
 else
  CURRENT_WS_STATE := WSOP;
  WS_IDS.ADD_TO_LIST (WS, SET_OF_OPEN_WORKSTATIONS);
 end if;
 ...

end OPEN_WS;


procedure CLOSE_PHIGS is
begin
...

 if CURRENT_OPERATING_STATE /= PHOP then
  ERROR_HANDLING (2, "CLOSE_PHIGS");
 else
  ...
  CURRENT_OPERATING_STATE := PHCL;
 end if;
 ...

end CLOSE_PHIGS;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
 ...

end PHIGS;

-- Version for application programs that use tasking:

with ERROR_HANDLING;
package body PHIGS is


 task MONITOR is
  entry OPEN_PHIGS
    (ERROR_FILE              : in FILE_ID := DEFAULT_ERROR_FILE;
     AMOUNT_OF_MEMORY  : in PHIGS_NATURAL := DEFAULT_MEMORY_UNITS);


  entry OPEN_WS
    (WS              : in WS_ID;
     CONNECTION  : in VARIABLE_CONNECTION_ID;
     TYPE_OF_WS   : in WS_TYPE);

  entry CLOSE_PHIGS;
  ...

 end MONITOR;


 task body MONITOR is

 -- State Variables:

 CURRENT_OPERATING_STATE  : OPERATING_STATE      := PHCL;
 CURRENT_WS_STATE         : WS_STATE             := WSCL;
 SET_OF_OPEN_WORKSTATIONS : WS_IDS.LIST_OF       := WS_IDS.NULL_LIST;


 begin
  loop
   begin
    select
     accept OPEN_PHIGS
      (ERROR_FILE              : in FILE_ID := DEFAULT_ERROR_FILE;
       AMOUNT_OF_MEMORY  : in PHIGS_NATURAL := DEFAULT_MEMORY_UNITS) do


      ...

      if CURRENT_OPERATING_STATE /= PHCL then
       ERROR_HANDLING (1, "OPEN_PHIGS");
      else
       CURRENT_OPERATING_STATE := PHOP;
      end if;
      ...

     end OPEN_PHIGS;


    or
     accept OPEN_WS
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
        (WS              : in WS_ID;
         CONNECTION      : in VARIABLE_CONNECTION_ID;
         TYPE_OF_WS      : in WS_TYPE) do


        ...

        if CURRENT_OPERATING_STATE /= PHOP then
          ERROR_HANDLING (8, "OPEN_WS");
        else
          CURRENT_WS_STATE := WSOP;
          WS_IDS.ADD_TO_LIST (WS, SET_OF_OPEN_WORKSTATIONS);
        end if;
        ...

      end OPEN_WS;


    or
      accept CLOSE_PHIGS do
        ...

        if CURRENT_OPERATING_STATE /= PHOP then
          ERROR_HANDLING (2, "CLOSE_PHIGS");
        else
          ...

          CURRENT_OPERATING_STATE := PHCL;
        end if;
        ...

      end CLOSE_PHIGS;
      ...

    or
      terminate;

    end select;
  exception
    when others => null;
    end;
 end loop;
end MONITOR;


procedure OPEN_PHIGS
  (ERROR_FILE              : in FILE_ID := DEFAULT_ERROR_FILE;
   AMOUNT_OF_MEMORY        : in PHIGS_NATURAL := DEFAULT_MEMORY_UNITS) is
begin
  MONITOR.OPEN_PHIGS (ERROR_FILE, AMOUNT_OF_MEMORY);
end OPEN_PHIGS;


procedure OPEN_WS
  (WS              : in WS_ID;
   CONNECTION      : in VARIABLE_CONNECTION_ID;
   TYPE_OF_WS      : in WS_TYPE) is
begin
  MONITOR.OPEN_WS (WS, CONNECTION, TYPE_OF_WS);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
end OPEN_WS;

procedure CLOSE_PHIGS is
begin
  MONITOR.CLOSE_PHIGS;
end CLOSE_PHIGS;
...

end PHIGS;
```

Some comments on the interactions with exception handling in the case where the ERROR_HANDLING procedure raises PHIGS_ERROR: note that in both versions of the package body, the ERROR_HANDLING procedure is called. Suppose that the application program calls OPEN_PHIGS when PHIGS is already open. In the sequential version, calling ERROR_HANDLING from the body of the OPEN_PHIGS procedure will cause PHIGS_ERROR to be propagated back to the application program that called OPEN_PHIGS. In the tasking version, the same effect is achieved in the following fashion. During execution of the MONITOR task's accept statement for the OPEN_PHIGS entry, ERROR_HANDLING will be called and PHIGS_ERROR will be raised. By Ada semantics, since this exception is not handled by a local handler in the accept, it is propagated both (1) to the point following the accept, and (2) to the point of the entry call. For (1), it is handled in the block enclosing the select statement; thus the MONITOR task can continue the next iteration of the loop without disruption. For (2), note that this point is in the body of the OPEN_PHIGS procedure. Since there is no exception handler here, PHIGS_ERROR is propagated (as desired) back to the call in the application program.

The MONITOR task terminates through selection of the **terminate** alternative, when no further communication with this task from other tasks is possible.

There are a number of variations on the technique discussed and illustrated above that a PHIGS implementor may consider in the interest of increasing the potential parallelism in PHIGS application programs that use tasking. With the method just outlined, the entire set of state variables is protected by one task. If the state information can be partitioned into independent sets, with one monitor per set, then an application program task that reads/writes a variable in one set can do so concurrently with a task that is reading/writing a variable in another set. The PHIGS implementor should consider the tradeoffs.

Another variation on the protection of state variables is to distinguish between PHIGS functions that simply read data values and those that write them. The technique outlined above treats readers and writers in the same way; thus it prohibits two tasks in an application program from simultaneously reading state information. It is possible for the PHIGS implementor to program the monitor task in such a way that concurrent reading by application program tasks is permitted, but that neither concurrent writing nor concurrent reading and writing can occur. There are many different approaches to this problem, depending on whether the implementor takes into account such factors as:

- the possibility of an application task being aborted, and
- the possibility of an application task being "starved" for service because of the way that the entry accepts are programmed.

An implementation of PHIGS that supports multi-tasking Ada programs can still be used for programs that are purely sequential, though efficiency may be impaired. An implementor may wish to provide two bodies for the PHIGS package -- one for use with sequential applications, and the other for tasking. The practicality of this scheme depends on the sophistication of the Ada library manager and perhaps also the implementor.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

9593 PT*3 90 ■ 4851903 0575656 281 ■

Annex E

(informative)

Index

A
abbreviations
  ARCL 16
  AROP 16
  ASAP 16
  ASF 8, 16
  ASTI 16
  BNIG 16
  BNIL 16
  CBS 16
  CHAR 8, 16
  CSS 16
  DC 16
  ESC 8
  GDP 8, 16
  GSE 8, 16
  HLHSR 8, 16
  ID 16
  IDS 16
  IMM 16
  in procedure names 8
  in type definitions 16
  IND 16
  INQ 8
  IRG 16
  MAX 16
  MC 16
  MI 16
  MIN 16
  MISC 16
  MO 16
  NIVE 16
  NPC 16
  PHCL 16
  PHIGS 8, 10, 16
  PHOP 16
  PT 16
  REF 16
  SF 16
  STCL 16
  STOP 16
  U 8, 10
  UQUM 16
  UWOR 16
  VC 16
  WC 16
  WS 8, 10, 16
  WSCL 16
  WSOP 16
ACCESS_COLOUR_MATRIX 16, 194
ACCESS_HALF_SPACE_LIST_2 141, 162
ACCESS_HALF_SPACE_LIST_3 141, 162
ACCESS_LIST_OF_POINT_LIST_2 140, 161
ACCESS_LIST_OF_POINT_LIST_3 140, 161
ACCESS_POINT_LIST_2 140, 161
ACCESS_POINT_LIST_3 139, 161
ACCESS_STRING 17, 194
Ada 2, 3, 7
Ada program library 7
ADD_NAMES 146, 157, 238
ADD_NAMES_TO_SET 82, 206
ADD_TO_LIST 143, 153, 155
ANGLE 17, 164
ANNOTATION_STYLE 17, 164
ANNOTATION_STYLES 17, 164
ANNOTATION_TEXT_RELATIVE 75, 204
application program 7
APPLICATION_DATA 92, 211
APPLICATION_DATA_RANGE 17, 164
APPLICATION_DATA_RECORD 17, 164
APPLY_FILTER 146, 157, 238
archive file 7
ARCHIVE_ALL_STRUCTURES 96, 212
ARCHIVE_ID 18, 164
ARCHIVE_IDS 18, 165
ARCHIVE_STATE 18, 165
ARCHIVE_STRUCTURE_NETWORKS 96, 212
ARCHIVE_STRUCTURES 96, 212
ARRAY_OF_PROMPTS 137, 236
ARRAY_OF_STRINGS 138, 236
ASF 18, 165
ASPECT 18, 165
ATTRIBUTES_USED 19, 166
ATTRIBUTES_USED_TYPE 19, 165
AWAIT_EVENT 105, 217

B
BASE_TYPES 159
BUFFER_SIZE 136, 235
BUILD_CHOICE_DATA_RECORD 137, 235, 236
BUILD_LOCATOR_DATA_RECORD 134, 135, 234
BUILD_NAME_SET 145, 156, 237
BUILD_NAME_SET_DIFFERENCE 145, 157, 237
BUILD_NAME_SET_INTERSECTION 145, 157, 237
BUILD_NAME_SET_UNION 145, 157, 237
BUILD_NEW_METAFILE_DATA_RECORD 149, 237
BUILD_PICK_DATA_RECORD 138, 236
BUILD_STRING_DATA_RECORD 138, 236
BUILD_STROKE_DATA_RECORD 135, 136, 234, 235
BUILD_TRANSFORMATION_MATRIX 89, 210
BUILD_VALUATOR_DATA_RECORD 136, 235

C
CELL_ARRAY 76, 204
CHANGE_STRUCTURE_IDENTIFIER 94, 212
CHANGE_STRUCTURE_IDENTIFIER_AND_REFERENCES 95, 212
CHANGE_STRUCTURE_REFERENCES 94, 212
CHAR_EXPANSION 19, 166

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

9593 PT*3 90 ■ 4851903 0575657 118 ■

CHAR_SET 20, 166
CHAR_SETS 20, 166
CHAR_SPACING 20, 166
CHOICE_DATA_RECORD 66, 202, 239, 243
CHOICE_DEVICE_NUMBER 20, 167
CHOICE_NUMBER 20, 167
CHOICE_PROMPT 20, 167
CHOICE_PROMPT_ECHO_TYPE 21, 166
CHOICE_PROMPT_ECHO_TYPES 21, 166
CHOICE_PROMPT_STRING 21, 167
CHOICE_PROMPT_STRING_ARRAY 21, 167
CHOICE_PROMPT_STRING_LIST 21, 167
CHOICE_PROMPTS 20, 167
CHOICE_REQUEST_STATUS 22, 168
CHOICE_SMALL_NATURAL 22, 160
CHOICE_STATUS 22, 168
CHROMATICITY_COEFFICIENT 22, 170
CHROMATICITY_COEFFICIENT_SET 22, 170
CIELUV 69, 169
CLIPPING_INDICATOR 23, 168
CLOSE_ARCHIVE_FILE 95, 212
CLOSE_PHIGS 73, 203
CLOSE_STRUCTURE 91, 211
CLOSE_WORKSTATION 73
CLOSE_WS 203
COLOUR_AVAILABLE 23, 168
COLOUR_COEFFICIENT 23, 169
COLOUR_COEFFICIENT_ARRAY 23, 169
COLOUR_COEFFICIENTS 23, 169
COLOUR_COMPONENTS 24, 168
COLOUR_INDEX 24, 168
COLOUR_INDICES 24, 168
COLOUR_MATRIX 24, 168
COLOUR_MATRIX_SMALL_NATURAL 24, 160
COLOUR_MODEL 24, 169
COLOUR_MODELS 24, 169
COLOUR_REPRESENTATION 25, 169
COMPOSE_MATRIX 88, 209
COMPOSE_TRANSFORMATION_MATRIX 89, 90, 210
COMPOSITION_TYPE 25, 170
Configuration 69, 151
CONFLICT_RESOLUTION 26, 171
Conformance 3
CONNECTION_ID 26, 171
CONTROL_FLAG 26, 171
COORDINATE_COMPONENT_TYPE 139, 161
COPY_ALL_ELEMENTS_FROM_STRUCTURE 93, 211
COPY_NAME_SET 145, 157, 238
CROSS_HAIR_LOCATOR 69, 201

D
DASHED_DOTTED_EDGE 68, 201
DASHED_DOTTED_LINE 68, 200
DASHED_EDGE 68, 200
DASHED_LINE 68, 200
DC 26, 164
DC_TYPE 26, 163
DC_UNITS 26, 171
DEALLOCATE
  CHOICE_DATA_RECORD 147, 236
  ESCAPE_DATA_RECORD 147, 236
  GDP_3_RECORD 147, 236
  GDP_RECORD 148, 236
  GSE_RECORD 148, 236
  LOCATOR_DATA_RECORD 148, 236
  METAFILE_DATA_RECORD 148, 236
  NAME_SET 146, 148, 158, 237
  PICK_DATA_RECORD 148, 237
  STRING_DATA_RECORD 148, 237
  STROKE_DATA_RECORD 148, 237
  STRUCTURE_ELEMENT_RECORD 149, 237
  VALUATOR_DATA_RECORD 149, 237
DEFAULT_CHOICE 69, 201
DEFAULT_ERROR_FILE 70, 151, 243
DEFAULT_LIST_SIZE 70, 151, 243
DEFAULT_LOCATOR 69, 201
DEFAULT_MEMORY_UNITS 70, 151, 243
DEFAULT_PICK 69, 201
DEFAULT_STROKE 69, 201
DEFAULT_VALUATOR 69, 201
DEFERRAL_MODE 26, 171
DELETE_ALL_STRUCTURES 94, 212
DELETE_ALL_STRUCTURES_FROM_ARCHIVE 97, 213
DELETE_ELEMENT 93, 211
DELETE_ELEMENT_RANGE 93, 211
DELETE_ELEMENTS_BETWEEN_LABELS 94, 211
DELETE_FROM_LIST 143, 154, 155
DELETE_STRUCTURE 94, 212
DELETE_STRUCTURE_NETWORK 94, 212
DELETE_STRUCTURE_NETWORKS_FROM_ARCHIVE 97, 213
DELETE_STRUCTURES_FROM_ARCHIVE 97, 213
dependencies 4
DEVICE_NUMBER 27, 167
DIGITAL_LOCATOR 69, 201
DIGITAL_STROKE 69, 201
DIGITAL_VALUATOR 69, 201
DISPLAY_CLASS 27, 171
DISPLAY_PRIORITY 27, 189
DISPLAY_SURFACE_EMPTY 27, 171
DOT_MARKER 68, 200
DOTTED_EDGE 68, 201
DOTTED_LINE 68, 200
DYNAMIC_MODIFICATION 27, 172

E
ECHO_SWITCH 28, 172
EDGE_ATTRIBUTES 135, 234
EDGE_DATA 28, 181
EDGE_FLAG 28, 180
EDGE_INDEX 28, 180
EDGE_INDICES 28, 180
EDGETYPE 29, 180
EDGETYPES 29, 181
EDGEWIDTH 29, 181
EDIT_MODE 29, 172
ELEMENT_POSITION 29, 172
ELEMENT_SEARCH 131, 233
ELEMENT_TYPE 29, 142, 153, 172
ELEMENT_TYPES 31, 174
EMERGENCY_CLOSE_PHIGS 132, 233
EMPTY_STRUCTURE 94, 211
error codes 71
  precluded 72
error file 7
error handling 4
ERROR_HANDLING 4, 132, 233
ERROR_HANDLING_MODE 31, 174
ERROR_LOGGING 132, 233
ERROR_NUMBER 31, 174
ESC 134
ESC_DATA_RECORD 133, 242
ESC_FLOAT_ARRAY 133, 242
ESC_INTEGER_ARRAY 133, 242
ESC_STRING_ARRAY 133, 242
ESCAPE 133
ESCAPE_DATA_RECORD 32, 174, 243
ESCAPE_ID 32, 174
ESCAPE_IDS 32, 174
EVALUATE_VIEW_MAPPING_MATRIX 90, 91, 210, 211
EVALUATE_VIEW_ORIENTATION_MATRIX 90, 210
EVENT_DEVICE_NUMBER 32, 193

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

9593 PT*3 90 ■ 4851903 0575658 054 ■

Index

EVENT_QUEUE_DEVICE_NUMBER 33, 193
example programs
  DYNASTAR 255
  IRON 248
  SHOW_LINETYPES 269
  STAR 245
  TRANSFORM_POLYLINE 261
EXECUTE_STRUCTURE 91, 211
EXTRACT_NAMES 146, 157, 238

F
FILE_ID 33, 175
FILE_SMALL_NATURAL 34, 160
FILL_AREA 75, 204
FILL_AREA_SET 75, 76, 204
FLUSH_DEVICE_EVENTS 105, 217
functions 3

G
GDP 76, 77
GDP_3_ID 34, 175
GDP_3_IDS 34, 175
GDP_3_RECORD 34, 175, 243
GDP_DATA_RECORD 77, 240
GDP_FLOAT_ARRAY 76, 240
GDP_ID 34, 176
GDP_IDS 34, 176
GDP_INTEGER_ARRAY 76, 240
GDP_RECORD 35, 176, 243
GDP_STRING_ARRAY 76, 240
GENERAL_FLOAT_PARAMETER 35, 164
GENERALIZED_ESC 133, 242
GENERALIZED_GDP 77, 240
GENERALIZED_GSE 92, 241
GET_CHOICE 105, 218
GET_ITEM_TYPE_FROM_METAFILE 106, 218
GET_LOCATOR 105, 217
GET_PICK 106, 218
GET_STRING 106, 218
GET_STROKE 105, 147, 218
GET_VALUATOR 105, 218
GRAPHICAL_VALUATOR 69, 201
GROUP_HIGHLIGHT_PICK 69, 201
GSE 92
GSE_DATA_RECORD 92, 241
GSE_FLOAT_ARRAY 92, 241
GSE_ID 35, 176
GSE_IDS 35, 176
GSE_INTEGER_ARRAY 92, 241
GSE_RECORD 35, 176, 243
GSE_STRING_ARRAY 92, 241

H
HALF_SPACE_2 140, 162
HALF_SPACE_3 140, 162
HALF_SPACE_LIST_2 141, 162
HALF_SPACE_LIST_3 141, 162
HATCH_STYLE 36, 177
HATCH_STYLES 36, 177
HIGH_VALUE 137, 235
HLHSR_ID 36, 177
HLHSR_IDS 36, 177
HLHSR_MODE 36, 177
HLHSR_MODES 37, 177
HLS 69, 169
HORIZONTAL_ALIGNMENT 37, 177
HSV 69, 169

I
INCREMENTAL_SPATIAL_SEARCH 131, 233
INITIAL_CURSOR_POSITION 138, 236
INITIALIZE_CHOICE 99, 214
INITIALIZE_LOCATOR 98, 213
INITIALIZE_PICK 100, 214, 215
INITIALIZE_STRING 100, 215
INITIALIZE_STROKE 98, 214
INITIALIZE_VALUATOR 99, 214
INPUT_BUFFER_SIZE 138, 236
INPUT_CLASS 37, 178
INPUT_QUEUE_CLASS 37, 178
INPUT_STATUS 38, 178
INPUT_STRING 38, 178
INPUT_STRING_SMALL_NATURAL 38, 160
INQ_ALL_CONFLICTING_STRUCTURES 109, 219
INQ_ANNOTATION_FACILITIES 121, 227
INQ_ARCHIVE_FILES 108, 219
INQ_ARCHIVE_STATE_VALUE 107, 218
INQ_CHOICE_DEVICE_STATE 116, 117, 224
INQ_COLOUR_FACILITIES 124, 229
INQ_COLOUR_MODEL 113, 222
INQ_COLOUR_MODEL_FACILITIES 124, 228
INQ_COLOUR_REPRESENTATION 113, 222
INQ_CONFLICT_RESOLUTION 109, 219
INQ_CONFLICTING_STRUCTURES_IN_NETWORK 109, 219
INQ_CURRENT_ELEMENT_CONTENT 130, 232
INQ_CURRENT_ELEMENT_TYPE_AND_SIZE 129, 232
INQ_DEFAULT_CHOICE_DEVICE_DATA 127, 128, 231
INQ_DEFAULT_DISPLAY_UPDATE_STATE 120, 226
INQ_DEFAULT_LOCATOR_DEVICE_DATA 126, 230
INQ_DEFAULT_PICK_DEVICE_DATA 128, 231
INQ_DEFAULT_STRING_DEVICE_DATA 128, 129, 231, 232
INQ_DEFAULT_STROKE_DEVICE_DATA 126, 127, 230
INQ_DEFAULT_VALUATOR_DEVICE_DATA 127, 231
INQ_DISPLAY_SPACE_SIZE 118, 225
INQ_DISPLAY_UPDATE_STATE 110, 220
INQ_DYNAMICS_OF_STRUCTURES 125, 230
INQ_DYNAMICS_OF_WS_ATTRIBUTES 119, 226
INQ_EDGE_FACILITIES 123, 228
INQ_EDGE_REPRESENTATION 112, 221
INQ_EDIT_MODE 108, 219
INQ_ELEMENT_CONTENT 130, 232
INQ_ELEMENT_POINTER 129, 232
INQ_ELEMENT_TYPE_AND_SIZE 130, 232
INQ_ERROR_HANDLING_MODE 132, 233
INQ_GDP 125, 229
INQ_GSE_FACILITIES 108, 219
INQ_HIGHLIGHTING_FILTER 114, 222
INQ_HLHSR_FACILITIES 119, 225
INQ_HLHSR_MODE 110, 220
INQ_INPUT_QUEUE_OVERFLOW 131, 233
INQ_INTERIOR_FACILITIES 122, 228
INQ_INTERIOR_REPRESENTATION 112, 221
INQ_INVISIBILITY_FILTER 114, 222
INQ_LIST_OF_AVAILABLE_GDP 124, 229
INQ_LIST_OF_AVAILABLE_GSE 125, 229
INQ_LIST_OF_AVAILABLE_WS_TYPES 107, 218
INQ_LIST_OF_COLOUR_INDICES 113, 222
INQ_LIST_OF_EDGE_INDICES 112, 221
INQ_LIST_OF_INTERIOR_INDICES 112, 221
INQ_LIST_OF_PATTERN_INDICES 113, 222
INQ_LIST_OF_POLYLINE_INDICES 111, 220
INQ_LIST_OF_POLYMARKER_INDICES 111, 221
INQ_LIST_OF_TEXT_INDICES 111, 221
INQ_LIST_OF_VIEW_INDICES 109, 220
INQ_LOCATOR_DEVICE_STATE 114, 115, 223
INQ_MODELLING_CLIPPING_FACILITIES 108, 219
INQ_MORE_SIMULTANEOUS_EVENTS 109, 219
INQ_NUMBER_OF_AVAILABLE_LOGICAL_INPUT_DEVICES 126, 230

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

9593 PT*3 90 ■ 4851903 0575659 T90 ■

INQ_NUMBER_OF_DISPLAY_PRIORITIES_SUPPORTED 125, 229
INQ_OPEN_STRUCTURE 129, 232
INQ_PATHS_TO_ANCESTORS 130, 232
INQ_PATHS_TO_DESCENDANTS 131, 233
INQ_PATTERN_FACILITIES 123, 228
INQ_PATTERN_REPRESENTATION 113, 147, 222
INQ_PHIGS_FACILITIES 107, 219
INQ_PICK_DEVICE_STATE 117, 224
INQ_POLYLINE_FACILITIES 120, 226
INQ_POLYLINE_REPRESENTATION 111, 220
INQ_POLYMARKER_FACILITIES 120, 226
INQ_POLYMARKER_REPRESENTATION 111, 221
INQ_POSTED_STRUCTURES 110, 220
INQ_PREDEFINED_COLOUR_REPRESENTATION 124, 229
INQ_PREDEFINED_EDGE_REPRESENTATION 123, 228
INQ_PREDEFINED_INTERIOR_REPRESENTATION 123, 228
INQ_PREDEFINED_PATTERN_REPRESENTATION 124, 147, 228
INQ_PREDEFINED_POLYLINE_REPRESENTATION 226
INQ_PREDEFINED_POLYMARKER_REPRESENTATION 121, 227
INQ_PREDEFINED_TEXT_REPRESENTATION 121, 227
INQ_PREDEFINED_VIEW_REPRESENTATION 119, 225
INQ_SET_OF_OPEN_WS 108, 219
INQ_SET_OF_WS_TO_WHICH_POSTED 129, 232
INQ_STRING_DEVICE_STATE 117, 118, 225
INQ_STROKE_DEVICE_STATE 115, 147, 223
INQ_STRUCTURE_IDENTIFIERS 108, 219
INQ_STRUCTURE_STATE_VALUE 107, 218
INQ_STRUCTURE_STATUS 130, 232
INQ_SYSTEM_STATE_VALUE 107, 218
INQ_TEXT_EXTENT
  Annotation Text 122, 227
  Text 122, 227
INQ_TEXT_FACILITIES 121, 227
INQ_TEXT_REPRESENTATION 112, 221
INQ_VALUATOR_DEVICE_STATE 116, 223, 224
INQ_VIEW_FACILITIES 119, 225
INQ_VIEW_REPRESENTATION 110, 220
INQ_WS_CATEGORY 118, 225
INQ_WS_CLASSIFICATION 119, 226
INQ_WS_CONNECTION_AND_TYPE 109, 220
INQ_WS_STATE_TABLE_LENGTHS 125, 229
INQ_WS_STATE_VALUE 107, 218
INQ_WS_TRANSFORMATION 114, 222
INQUIRE_MEMBERSHIP 146, 158, 238
INTENSITY 38, 169
INTERIOR_ATTRIBUTES 135, 234
INTERIOR_DATA 38, 179
INTERIOR_INDEX 39, 178
INTERIOR_INDICES 39, 178
INTERIOR_STYLE 39, 178
INTERIOR_STYLES 39, 179
INTERPRET_ITEM 107, 218
INTERSECT_VOLUME 69, 201
INTERVAL 136, 235
INVALID_VALUES_INDICATOR 40, 179
IS_EQUAL 146, 157, 238
IS_IN_LIST 142, 153, 155
ITEM_DATA_RECORD_STRING 149, 237

L
LABEL 91, 211
LABEL_ID 40, 179
LEAD_LINE_ANNOTATION 69, 201
LINE_ATTRIBUTES 135, 136, 234, 235
LINE_DATA 40, 180
LINE_STROKE 69, 201
LINETYPE 40, 180
LINETYPES 40, 180
LINEWIDTH 40, 180
LIST 142, 153, 155
LIST_ELEMENT 142, 153, 155
LIST_OF 142, 143, 153, 154
LIST_OF_PICK_IDS 138, 236
LIST_OF_POINT_LIST_2 140, 161
LIST_OF_POINT_LIST_3 140, 161
LIST_SIZE 142, 153
LIST_VALUES 142, 153
LOCATOR_ATTRIBUTES_USED 135, 234
LOCATOR_DATA_RECORD 67, 202, 239, 243
LOCATOR_DEVICE_NUMBER 41, 181
LOCATOR_PROMPT_ECHO_TYPE 41, 181
LOCATOR_PROMPT_ECHO_TYPES 41, 181
LOW_VALUE 137, 235

M
MAGNITUDE 141, 162
MAGNITUDE_BASE_TYPE 141, 162
MARKER_ATTRIBUTES 136, 235
MARKER_DATA 41, 182
MARKER_SIZE 41, 182
MARKER_STROKE 69, 201
MARKER_TYPE 42, 182
MARKER_TYPES 42, 182
MAX_ANNOTATION_STYLES_SUPPORTED 70, 151, 243
MAX_APPLICATION_DATA 70, 151, 244
MAX_ARCHIVE_IDS_SUPPORTED 70, 151, 243
MAX_CHAR_SETS_SUPPORTED 70, 151, 243
MAX_CHOICE_PROMPT_ECHO_TYPES_SUPPORTED 70, 151, 243
MAX_CHOICE_PROMPTS_SUPPORTED 70, 151, 243
MAX_CHOICE_SMALL_NATURAL 70, 151
MAX_CHOICE_SMALL_PHIGS_NATURAL 243
MAX_COLOUR_COEFFICIENTS 71, 151, 244
MAX_COLOUR_INDICES_SUPPORTED 70, 151, 243
MAX_COLOUR_MATRIX_SMALL_NATURAL 70, 151
MAX_COLOUR_MATRIX_SMALL_PHIGS_NATURAL 243
MAX_COLOUR_MODELS_SUPPORTED 70, 151, 243
MAX_EDGE_INDICES_SUPPORTED 70, 151, 243
MAX_EDGETYPES_SUPPORTED 70, 151, 243
MAX_ESCAPE_IDS_SUPPORTED 70, 151, 243
MAX_FILE_IDS_SUPPORTED 70, 151, 243
MAX_FILE_SMALL_NATURAL 70, 151
MAX_FILE_SMALL_PHIGS_NATURAL 243
MAX_GDP_3_IDS_SUPPORTED 71, 151, 243
MAX_GDP_IDS_SUPPORTED 71, 151, 243
MAX_GSE_IDS_SUPPORTED 71, 151, 243
MAX_HATCH_STYLES_SUPPORTED 71, 151, 243
MAX_HLHSR_IDS_SUPPORTED 71, 151, 243
MAX_HLHSR_MODES_SUPPORTED 71, 151, 244
MAX_INPUT_STRING_SMALL_NATURAL 70, 152
MAX_INPUT_STRING_SMALL_PHIGS_NATURAL 243
MAX_INTEGER_VALUE 70, 152, 159
MAX_INTERIOR_INDICES_SUPPORTED 71, 152, 244
MAX_LINETYPES_SUPPORTED 71, 152, 244
MAX_LIST_SIZE 70, 142, 153
MAX_LOCATOR_PROMPT_ECHO_TYPES_SUPPORTED 71, 152, 244
MAX_MARKER_TYPES_SUPPORTED 71, 152, 244
MAX_MEMORY_UNITS 71, 152, 244
MAX_METAFILE_ITEM_LENGTH 71, 152, 244
MAX_METAFILE_ITEM_TYPE 71, 152, 244
MAX_MODELLING_CLIP_DISTINCT_PLANES_SUPPORTED 71, 152, 244
MAX_MODELLING_CLIP_OPERATION_TYPES_SUPPORTED 71, 152, 244
MAX_NAME_SET_CLASSES_SUPPORTED 244

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

9593 PT*3 90 ■ 4851903 0575660 702 ■

MAX_NAME_SET_FILTER_LIST_LENGTH_SUPPORTED 71, 152, 244
MAX_NAME_SUPPORTED 71, 152, 156
MAX_OPEN_WS_SUPPORTED 71, 152, 244
MAX_PATH_DEPTH_SUPPORTED 71, 152, 244
MAX_PATTERN_INDICES_SUPPORTED 71, 152, 244
MAX_PHIGS_PHIGS_INTEGER 243
MAX_PICK_IDS_SUPPORTED 71, 152, 244
MAX_PICK_PROMPT_ECHO_TYPES_SUPPORTED 71, 152, 244
MAX_POLYLINE_INDICES_SUPPORTED 71, 152, 244
MAX_POLYMARKER_INDICES_SUPPORTED 71, 152, 244
MAX_POSTED_STRUCTURES_SUPPORTED 71, 152, 244
MAX_REFERENCE_PATHS_SUPPORTED 71, 152, 244
MAX_SMALL_NATURAL 70, 152
MAX_SMALL_PHIGS_NATURAL 243
MAX_STRING_PROMPT_ECHO_TYPES_SUPPORTED 71, 152, 244
MAX_STRING_SMALL_NATURAL 70, 152
MAX_STRING_SMALL_PHIGS_NATURAL 243
MAX_STROKE_PROMPT_ECHO_TYPES_SUPPORTED 71, 152, 244
MAX_STRUCTURE_IDS_SUPPORTED 71, 152, 244
MAX_TEXT_FONT_PRECISION_PAIRS_SUPPORTED 71, 152, 244
MAX_TEXT_INDICES_SUPPORTED 71, 152, 244
MAX_VALUATOR_PROMPT_ECHO_TYPES_SUPPORTED 71, 152, 244
MAX_VIEW_INDICES_SUPPORTED 71, 152, 244
MAX_WS_IDS_SUPPORTED 71, 152, 244
MAX_WS_TYPES_SUPPORTED 71, 152, 244
MC 42, 163
MC_TYPE 42, 163
MEMORY_UNITS 42, 182
MESSAGE 74, 203
metafile 7
METAFILE_DATA_RECORD 67, 202, 239, 243
METAFILE_ITEM_LENGTH 42, 183
METAFILE_ITEM_TYPE 43, 182
MIN_INTEGER_VALUE 70, 152, 159
MIN_METAFILE_ITEM_TYPE 71, 152, 244
MIN_PHIGS_PHIGS_INTEGER 243
minimal program 6
MODELLING_CLIP_OPERATION_TYPE 43, 183
MODELLING_CLIP_OPERATION_TYPES 43, 183
MODIFICATION_MODE 43, 183
MORE_EVENTS 43, 183
multi-tasking 3, 6, 274

N
NAME 44, 144, 156, 183
NAME_SET 44, 67, 144, 147, 156, 158, 183, 243
NAME_SET_ACCEPTANCE 44, 144, 156, 183
NAME_SET_FILTER 44, 144, 156, 184
NAME_SET_FILTERS 44, 184
NAME_SET_MEMBERSHIP 45, 145, 156, 184
NAMES 44, 144, 156, 184
NPC 45, 163
NPC_TYPE 45, 163
NULL_LIST 142, 153, 154

O
OFF_ON 45, 164
OFFSET_ELEMENT_POINTER 93, 211
OPEN_ARCHIVE_FILE 95, 212
OPEN_PHIGS 73, 203
OPEN_STRUCTURE 91, 211
OPEN_STRUCTURE_STATUS 45, 184
OPEN_WORKSTATION 73
OPEN_WS 203
OPERATING_MODE 45, 184

P
PATH_DEPTH 46, 184
PATH_ORDER 46, 184
PATTERN_INDEX 46, 185
PATTERN_INDICES 46, 185
PHIGS 2
  data types 4
  package 6, 202
PHIGS_CONFIGURATION 7, 69, 70, 151
PHIGS_COORDINATE_SYSTEM 7, 139, 161
PHIGS_ERROR 134, 144, 200
PHIGS_ESCAPE 133, 242
PHIGS_GDP 240
PHIGS_GSE 92, 241
PHIGS_INTEGER 46, 159
PHIGS_LIST_UTILITIES 7, 141, 142, 153, 154
PHIGS_NAME_SET_FACILITY 7, 144, 156
PHIGS_NATURAL 47, 159
PHIGS_POSITIVE 47, 159
PHIGS_PRECISION 70, 152, 243
PHIGS_STRING 47, 159
PHIGS_TYPES 6, 159
PHIGS_UESC ... 134
PHIGS_UGDP ... 77
PHIGS_UGSE ... 92
PICK_DATA_RECORD 67, 202, 239, 243
PICK_DEVICE_NUMBER 47, 185
PICK_ID 47, 185
PICK_IDS 47, 185
PICK_PATH 47, 186
PICK_PATH_ARRAY 48, 186
PICK_PATH_ENTRY 48, 186
PICK_PROMPT_ECHO_TYPE 48, 185
PICK_PROMPT_ECHO_TYPES 48, 185
PICK_REQUEST_STATUS 48, 185
PICK_STATUS 49, 186
PLUS_MARKER 68, 200
POINT_2 139, 161
POINT_3 139, 161
POINT_LIST_2 139, 161
POINT_LIST_3 139, 161
POLYLINE 74, 203
POLYLINE_FILL_AREA_CONTROL_FLAG 49, 186
POLYLINE_INDEX 49, 179
POLYLINE_INDICES 49, 179
POLYMARKER 74, 203
POLYMARKER_INDEX 49, 181
POLYMARKER_INDICES 49, 182
POSITION 136, 235
POSITIVE_VIEW_INDEX 50
POST_STRUCTURE 95, 212
POSTED_STRUCTURE 50, 190
POSTED_STRUCTURES 50, 190
PROJECTION_TYPE 50, 187
PROMPT_ECHO_CHOICE 69, 201

R
RANGE_OF_MAGNITUDES 141, 162
RASTER_UNIT_SIZE_2 51, 187
RASTER_UNIT_SIZE_3 51, 187
RASTER_UNITS 50, 187
READ_ITEM_FROM_METAFILE 106, 218
RECTANGLE_LOCATOR 69, 201
RECTANGULAR_REGION_2 140, 162
RECTANGULAR_REGION_3 140, 161
REDRAW_ALL_STRUCTURES 73, 203
REFERENCE_HANDLING_FLAG 51, 187
REFERENCE_PATH 51, 188
REFERENCE_PATH_ARRAY 51, 188
REFERENCE_PATH_ENTRY 52, 187

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

9593 PT*3 90 ■ 4851903 0575661 649 ■

REFERENCE_PATHS 52, 188
registered item 7
RELATIVE_PRIORITY 52, 188
REMOVE_NAMES 146, 157, 238
REMOVE_NAMES_FROM_SET 82, 206
REPLACE_VOLUME 69, 201
REPRESENTATION 120
REQUEST_CHOICE 102, 216
REQUEST_LOCATOR 101, 102, 216
REQUEST_PICK 103, 216
REQUEST_STRING 103, 216
REQUEST_STROKE 102, 147, 216
REQUEST_VALUATOR 102, 216
RESTORE_MODELLING_CLIPPING_VOLUME 85, 208
RETRIEVE_ALL_STRUCTURES 97, 213
RETRIEVE_PATHS_TO_ANCESTORS 97, 213
RETRIEVE_PATHS_TO_DESCENDANTS 97, 213
RETRIEVE_STRUCTURE_IDENTIFIERS 96, 213
RETRIEVE_STRUCTURE_NETWORKS 96, 213
RETRIEVE_STRUCTURES 96, 213
RETURN_VALUE_TYPE 52, 188
RETURNED_ELEMENT_POSITION 52, 172
RGB 69, 169
ROTATE 88, 209
ROTATE_X 88, 209
ROTATE_Y 88, 209
ROTATE_Z 88, 209
RUBBER_BAND_LINE_LOCATOR 69, 201

S
SAMPLE_CHOICE 104, 217
SAMPLE_LOCATOR 103, 216, 217
SAMPLE_PICK 104, 217
SAMPLE_STRING 104, 217
SAMPLE_STROKE 103, 104, 147, 217
SAMPLE_VALUATOR 104, 217
SCALE 87, 209
SCALE_FACTOR 52, 159
SEARCH_DIRECTION 53, 188
SEARCH_STATUS_INDICATOR 53, 188
SET_ANNOTATION_STYLE 80, 205
SET_ANNOTATION_TEXT_ALIGNMENT 80, 205
SET_ANNOTATION_TEXT_CHAR_HEIGHT 80, 205
SET_ANNOTATION_TEXT_CHAR_UP_VECTOR 80, 205
SET_ANNOTATION_TEXT_PATH 80, 205
SET_CHAR_EXPANSION_FACTOR 79, 205
SET_CHAR_HEIGHT 79, 205
SET_CHAR_SPACING 79, 205
SET_CHAR_UP_VECTOR 80, 205
SET_CHOICE_MODE 101, 215
SET_COLOUR_MODEL 84, 207
SET_COLOUR_REPRESENTATION 83, 207
SET_CONFLICT_RESOLUTION 96, 212
SET_DISPLAY_UPDATE_STATE 74, 203
SET_EDGE_COLOUR_INDEX 81, 206
SET_EDGE_FLAG 81, 206
SET_EDGE_INDEX 78, 204
SET_EDGE_REPRESENTATION 83, 207
SET_EDGETYPE 81, 206
SET_EDGEWIDTH_SCALE_FACTOR 81, 206
SET_EDIT_MODE 93, 211
SET_ELEMENT_POINTER 93, 211
SET_ELEMENT_POINTER_AT_LABEL 93, 211
SET_ERROR_HANDLING_MODE 132, 234
SET_GLOBAL_TRANSFORMATION 85, 207, 208
SET_HIGHLIGHTING_FILTER 84, 207
SET_HLHSR_IDENTIFIER 84, 207
SET_HLHSR_MODE 84, 207
SET_INDIVIDUAL_ASF 82, 206
SET_INTERIOR_COLOUR_INDEX 81, 206
SET_INTERIOR_INDEX 78, 204
SET_INTERIOR_REPRESENTATION 83, 207
SET_INTERIOR_STYLE 81, 205
SET_INTERIOR_STYLE_INDEX 81, 206
SET_INVISIBILITY_FILTER 84, 207
SET_LINETYPE 78, 204
SET_LINEWIDTH_SCALE_FACTOR 78, 204
SET_LOCAL_TRANSFORMATION 84, 85, 207
SET_LOCATOR_MODE 100, 215
SET_MARKER_SIZE_SCALE_FACTOR 79, 205
SET_MARKER_TYPE 79, 205
SET_MODELLING_CLIPPING_INDICATOR 85, 208
SET_MODELLING_CLIPPING_VOLUME 85, 208
SET_PATTERN_REFERENCE_POINT 82, 206
SET_PATTERN_REFERENCE_POINT_AND_VECTORS 82, 206
SET_PATTERN_REPRESENTATION 83, 207
SET_PATTERN_SIZE 81, 206
SET_PICK_FILTER 98, 213
SET_PICK_IDENTIFIER 98, 213
SET_PICK_MODE 101, 215
SET_POLYLINE_COLOUR_INDEX 78, 205
SET_POLYLINE_INDEX 78, 204
SET_POLYLINE_REPRESENTATION 82, 206
SET_POLYMARKER_COLOUR_INDEX 79, 205
SET_POLYMARKER_INDEX 78, 204
SET_POLYMARKER_REPRESENTATION 82, 206
SET_STRING_MODE 101, 215
SET_STROKE_MODE 101, 215
SET_TEXT_ALIGNMENT 80, 205
SET_TEXT_COLOUR_INDEX 79, 205
SET_TEXT_FONT 79, 205
SET_TEXT_INDEX 78, 204
SET_TEXT_PATH 80, 205
SET_TEXT_PRECISION 79, 205
SET_TEXT_REPRESENTATION 83, 206
SET_VALUATOR_MODE 101, 215
SET_VIEW_INDEX 86, 208
SET_VIEW_REPRESENTATION 86, 208
SET_VIEW_TRANSFORMATION_INPUT_PRIORITY 86, 208
SET_WS_VIEWPORT 87, 208
SET_WS_WINDOW 86, 208
SIZE_2 141, 162
SIZE_3 141, 162
SIZE_OF_LIST 142, 153, 154
SMALL_NATURAL 53, 160
SOLID_EDGE 68, 200
SOLID_LINE 68, 200
STAR_MARKER 68, 200
STRING_DATA_RECORD 67, 202, 239, 243
STRING_DEVICE_NUMBER 53, 188
STRING_INPUT_CHOICE 69, 201
STRING_PROMPT_CHOICE 69, 201
STRING_PROMPT_ECHO_TYPE 53, 189
STRING_PROMPT_ECHO_TYPES 53, 189
STRING_SMALL_NATURAL 53, 160
STROKE_DATA_RECORD 68, 202, 239, 243
STROKE_DEVICE_NUMBER 54, 189
STROKE_PROMPT_ECHO_TYPE 54, 189
STROKE_PROMPT_ECHO_TYPES 54, 189
STRUCTURE_CHOICE 69, 201
STRUCTURE_ELEMENT_RECORD 54, 196
STRUCTURE_ELEMENT_TYPE 59, 189
STRUCTURE_HIGHLIGHT_PICK 69, 201
STRUCTURE_ID 59, 186
STRUCTURE_IDENTIFIER 138, 236
STRUCTURE_IDS 59, 186
STRUCTURE_NETWORK_SOURCE 59, 189
STRUCTURE_STATE 59, 189
STRUCTURE_STATUS 60, 190
STYLE_INDEX 60, 177

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

9593 PT*3 90 ■ 4851903 0575662 585 ■

Index

SUBPROGRAM_NAME 60, 190
SYSTEM_STATE 60, 190

T
TEXT 75, 203, 204
TEXT_ALIGNMENT 60, 191
TEXT_FONT 60, 191
TEXT_FONT_PRECISION 61, 191
TEXT_FONT_PRECISIONS 61, 191
TEXT_INDEX 61, 191
TEXT_INDICES 61, 191
TEXT_PATH 61, 192
TEXT_PRECISION 61, 191
TIME 136, 235
TRACKING_CROSS_LOCATOR 69, 201
TRANSFORM_POINT 88, 89, 209, 210
TRANSFORMATION_MATRIX_2 62, 192
TRANSFORMATION_MATRIX_3 62, 192
TRANSLATE 87, 209

U
UNCONNECTED_ANNOTATION 68, 201
UNPOST_ALL_STRUCTURES 95, 212
UNPOST_STRUCTURE 95, 212
UPDATE_REGENERATION_FLAG 62, 192
UPDATE_STATE 62, 192
UPDATE_WORKSTATION 73
UPDATE_WS 203

V
VALUATOR_DATA_RECORD 68, 202, 239, 243
VALUATOR_DEVICE_NUMBER 62, 192
VALUATOR_PROMPT_ECHO_TYPE 62, 192
VALUATOR_PROMPT_ECHO_TYPES 63, 192
VALUATOR_VALUE 63, 193
VARIABLE_CONNECTION_ID 63, 194
VARIABLE_FILE_ID 63, 175
VARIABLE_FILE_IDS 63, 175
VECTOR_2 140, 161
VECTOR_3 140, 161
VECTOR_PAIR_2 140, 161
VECTOR_PAIR_3 140, 161
VERTICAL_ALIGNMENT 64, 190
VIEW_INDEX 64, 194
VIEW_INDICES 64, 194
VISUAL_REPRESENTATION_STATE 64, 194
VRC 64, 163
VRC_TYPE 65, 163

W
WC 65, 163
WC_TYPE 65, 163
WRITE_ITEM_TO_METAFILE 106, 218
WS_CATEGORY 65, 195
WS_DEPENDENCIES 65, 195
WS_DEPENDENCY 65, 195
WS_ID 65, 195
WS_IDS 66, 195
WS_STATE 66, 195
WS_TYPE 66, 195
WS_TYPES 66, 195

X
X_MARKER 68, 200

Z
ZERO_MARKER 68, 200

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

UDC 681.3.01 : 681.3.072.2(084) : 800.92

Descriptors : data processing, information interchange, graphic data processing, programming (computers), computer programs, programming languages, Ada.

Price based on 285 pages

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale