

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)
AVA Documents
Document Sharing Hub

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

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

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

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*.

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.

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.

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)*.

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.

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.

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

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

INQ_PREDEFINED_TEXT_REPRESENTATION
 INQ_REFLECTANCE_FACILITIES
 INQ_REFLECTANCE_REPRESENTATION
 INQ_RENDERING_COLOUR_MODEL_FACILITIES
 INQ_TEXT_REPRESENTATION
 INQ_WS_STATE_TABLE_LENGTHS
 NON_UNIFORM_B_SPLINE_CURVE
 NON_UNIFORM_B_SPLINE_CURVE
 NON_UNIFORM_B_SPLINE_SURFACE
 NON_UNIFORM_B_SPLINE_SURFACE
 POLYLINE_SET
 QUADRILATERAL_MESH
 QUADRILATERAL_MESH
 SET_BACK_DATA_MAPPING_INDEX
 SET_BACK_DATA_MAPPING_METHOD
 SET_BACK_INTERIOR_COLOUR
 SET_BACK_INTERIOR_INDEX
 SET_BACK_INTERIOR_SHADING_METHOD
 SET_BACK_INTERIOR_STYLE
 SET_BACK_INTERIOR_STYLE_INDEX
 SET_BACK_REFLECTANCE_INDEX
 SET_BACK_REFLECTANCE_MODEL
 SET_BACK_REFLECTANCE_PROPERTIES
 SET_COLOUR_MAPPING_INDEX
 SET_COLOUR_MAPPING_REPRESENTATION
 SET_CURVE_APPROXIMATION_CRITERIA
 SET_DATA_MAPPING_INDEX
 SET_DATA_MAPPING_METHOD
 SET_DATA_MAPPING_REPRESENTATION
 SET_DEPTH_CUE_INDEX
 SET_DEPTH_CUE_REPRESENTATION
 SET_EDGE_COLOUR
 SET_EDGE_REPRESENTATION
 SET_FACET_CULLING_MODE
 SET_FACET_DISTINGUISHING_MODE
 SET_INTERIOR_COLOUR
 SET_INTERIOR_REPRESENTATION
 SET_INTERIOR_SHADING_METHOD
 SET_LIGHT_SOURCE_REPRESENTATION
 SET_LIGHT_SOURCE_STATE
 SET_OF_FILL_AREA_SETS
 SET_OF_FILL_AREA_SETS
 SET_PARAMETRIC_SURFACE_CHARACTERISTICS
 SET_PARAMETRIC_SURFACE_INDEX
 SET_PARAMETRIC_SURFACE_REPRESENTATION
 SET_PATTERN_REPRESENTATION
 SET_POLYLINE_COLOUR
 SET_POLYLINE_REPRESENTATION
 SET_POLYLINE_SHADING_METHOD
 SET_POLYMARKER_COLOUR
 SET_POLYMARKER_REPRESENTATION
 SET_REFLECTANCE_INDEX
 SET_REFLECTANCE_MODEL
 SET_REFLECTANCE_PROPERTIES
 SET_REFLECTANCE_REPRESENTATION
 SET_RENDERING_COLOUR_MODEL
 SET_SURFACE_APPROXIMATION_CRITERIA
 SET_TEXT_COLOUR
 SET_TEXT_REPRESENTATION
 TRIANGLE_SET
 TRIANGLE_SET
 TRIANGLE_STRIP
 TRIANGLE_STRIP

inquire predefined text representation plus
 inquire reflectance facilities
 inquire reflectance representation
 inquire rendering colour model facilities
 inquire text representation plus
 inquire workstation state table lengths plus
 non-uniform B-spline curve
 non-uniform B-spline curve with colour
 non-uniform B-spline surface
 non-uniform B-spline surface with data
 polyline set 3 with colour
 quadrilateral mesh 3 with data
 quadrilateral mesh with data
 set back data mapping index
 set back data mapping method
 set back interior colour
 set back interior index
 set back interior shading method
 set back interior style
 set back interior style index
 set back reflectance index
 set back reflectance model
 set back reflectance properties
 set colour mapping index
 set colour mapping representation
 set curve approximation criteria
 set data mapping index
 set data mapping method
 set data mapping representation
 set depth cue index
 set depth cue representation
 set edge colour
 set edge representation plus
 set facet culling mode
 set facet distinguishing mode
 set interior colour
 set interior representation plus
 set interior shading method
 set light source representation
 set light source state
 set of fill area sets 3 with data
 set of fill area sets with data
 set parametric surface characteristics
 set parametric surface index
 set parametric surface representation
 set pattern representation plus
 set polyline colour
 set polyline representation plus
 set polyline shading method
 set polymarker colour
 set polymarker representation plus
 set reflectance index
 set reflectance model
 set reflectance properties
 set reflectance representation
 set rendering colour model
 set surface approximation criteria
 set text colour
 set text representation plus
 triangle set 3 with data
 triangle set with data
 triangle strip 3 with data
 triangle strip with data

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.

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.

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.

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.

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.

5.14.6 Metafile function utilities

page 149: No changes.

5.15 Conformal variants

page 150: No changes.

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,

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.

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,

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,

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,

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,

```

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;

```

```
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;
```

```
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;
```

```
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;
```

```
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;
```

```
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;
```

```

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;

```

```

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_GEOMETRY_SPLINE_3;
  NURB_CURVE_LIMITS : SPC.RANGE_OF_MAGNITUDES;

when NON_UNIFORM_B_SPLINE_CURVE_WITH_COLOUR =>
  NURB_CURVE_COLOUR_GEOMETRY_SPLINE
    : MC.ACCESS_CURVE_GEOMETRY_SPLINE_3;
  NURB_CURVE_COLOUR_LIMITS : SPC.RANGE_OF_MAGNITUDES;
  NURB_CURVE_COLOUR_COLOURSPLINE
    : ACCESS_CURVE_COLOURSPLINE;

when NON_UNIFORM_B_SPLINE_SURFACE =>
  NURB_SURFACE_GEOMETRY_SPLINE
    : MC.ACCESS_SURFACE_GEOMETRY_SPLINE;
  NURB_SURFACE_TRIM_CURVES
    : ACCESS_LIST_OF_TRIMCURVE_LIST;

when NON_UNIFORM_B_SPLINE_SURFACE_WITH_DATA =>
  NURB_SURFACE_DATA_GEOMETRY_SPLINE
    : MC.ACCESS_SURFACE_GEOMETRY_SPLINE;
  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;

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;

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.

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.

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.

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.

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.

ARRAY_OF_EDGE_FLAG_PAIR

type ARRAY_OF_EDGE_FLAG_PAIR is
array (POSITIVE range $\langle \rangle$,
POSITIVE range $\langle \rangle$) 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.

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);
```

```

    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.

```

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.
```

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.

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);

```

```

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);

```

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;

```

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.

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;

```

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.