INTERNATIONAL STANDARD

ISO/IEC 9592-4

First edition
1992-09-01

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

Part 4:
Plus Lumière und Surfaces, PHIGS PLUS

Systèmes de traitement de l'information — Infographie — Interface de programmation du système graphique hiérarchisé (PHIGS) —

Partie 4: Plus Lumière und Surfaces, PHIGS PLUS

Reference number
ISO/IEC 9592-4:1992(E)

Contents

Page

Foreword

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

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

International Standard ISO/IEC 9592-4 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*.

ISO/IEC 9592 consists of the following parts, under the general title *Information processing systems — Computer graphics — Programmer's Hierarchical Interactive Graphics System (PHIGS)*:

— *Part 1: Functional description*

— *Part 2: Archive file format*

— *Part 3: Clear-text encoding of archive file*

— *Part 4: Plus Lumière und Surfaces, PHIGS PLUS*

Annex D forms an integral part of this part of ISO/IEC 9592. Annexes A, B, C, E and F are for information only.

Introduction

ISO/IEC 9592-1 provides a set of functions for the definition, display and modification of 2D or 3D graphical data. It does not provide support for simulating the effects of lighting, shading, and other properties that are important for the display of multi-dimensional data. This part of ISO/IEC 9592 specifies a basic set of such functionality for use in conjunction with the functionality defined in ISO/IEC 9592-1 and its amendment 1.

To provide this support, PHIGS PLUS defines

a) output primitives specified by rational and non-rational B-spline curves and surfaces;

b) output primitives containing both geometric and non-geometric data in their definition;

c) attributes that control the application of lighting and shading to both the new primitives and the primitives specified in ISO/IEC 9592-1;

d) a generalized mechanism for colour specification to allow non-indexed colour specification.

Information processing systems– Computer graphics– Programmer's Hierarchical Interactive Graphics System (PHIGS)– Part 4—Plus Lumière und Surfaces, PHIGS PLUS

1 Scope

This part of ISO/IEC 9592 specifies an additional set of functionality of the Programmer's Hierarchical Interactive Graphics System. This additional functionality is intended to satisfy basic application requirements in the areas of lighting and shading and defines additional primitives and functionality for controlling the rendering of 3D objects. It relies on the coexistence of the functions and functionality specified in ISO/IEC 9592-1, and is meant to extend that functionality in the above areas.

It is the intent of this part of ISO/IEC 9592 to be compatible with ISO/IEC 9592-1 and its Amendment 1. That is, in a standard conforming PHIGS PLUS implementation all functions defined in ISO/IEC 9592-1 and not altered by ISO/IEC 9592-4 shall perform as specified in ISO/IEC 9592-1, and all functions defined in ISO/IEC 9592-1 but altered in ISO/IEC 9592-4 shall perform as specified in ISO/IEC 9592-1 and ISO/IEC 9592-4, and, an application functionally conforming to ISO/IEC 9592-1 produces the same effect running on a standard conforming PHIGS PLUS implementation as it would produce running on a standard conforming PHIGS ISO/IEC 9592-1 implementation, excepting such differences among implementations as are allowed in ISO/IEC 9592-1. If PHIGS PLUS functions are used, they should only cause the extended effects specified in this part of ISO/IEC 9592.

This part of ISO/IEC 9592 defines a language independent extension to a graphics system for integration into a programming language. PHIGS PLUS is embedded in a language layer obeying the particular conventions of the language. Such language bindings are specified for ISO or ISO/IEC languages in ISO/IEC 9593.

2 Normative references

The following standards contain provisions which, through reference in this text, constitute provisions of this part of ISO/IEC 9592. 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 9592 are encouraged to investigate the possibility of applying the most recent editions of the standards indicated below. Members of IEC and ISO maintain registers of currently valid International Standards.

ISO/IEC 646:1991, *Information technology – ISO 7-bit coded character set for information exchange.*

ISO/IEC 7942:1985, *Information processing systems – Computer graphics – Graphical Kernel System (GKS) functional description.*

ISO/IEC 8632:1987, *Information processing systems – Computer graphics – Metafile for storage and transfer of picture description information.*

ISO/IEC 8805:1988, *Information processing systems – Computer graphics – Graphical Kernel System for Three Dimensions (GKS-3D) functional description.*

ISO/IEC 9592-1:1989, *Information processing systems – Computer graphics – Programmer's Hierarchical Interactive Graphics System (PHIGS) – Part 1: Functional description.*

ISO/IEC 9592-1:1989/Amd.1:1992, *Information processing systems – Computer graphics – Programmer's Hierarchical Interactive Graphics System (PHIGS) – Part 1: Functional description – Amendment 1.*

ISO/IEC 9592-2:1989, *Information processing systems – Computer graphics – Programmer's Hierarchical Interactive Graphics System (PHIGS) – Part 2: Archive file format.*

ISO/IEC 9592-2:1989/Amd.1:1992, *Information processing systems – Computer graphics – Programmer's Hierarchical Interactive Graphics System (PHIGS) – Part 2: Archive file format – Amendment 1.*

ISO/IEC 9592-3:1989, *Information processing systems – Computer graphics – Programmer's Hierarchical Interactive Graphics System (PHIGS) – Part 3: Clear text encoding of archive file.*

ISO/IEC 9592-3:1989/Amd.1:1992, *Information processing systems – Computer graphics – Programmer's Hierarchical Interactive Graphics System (PHIGS) – Part 3: Clear text encoding of archive file – Amendment 1.*

ISO/IEC 9593:1990, *Information processing systems – Computer graphics – Programmer's Hierarchical Interactive Graphics System (PHIGS) language bindings.*

3 Definitions

For the purpose of this part of ISO/IEC 9592 the following definitions apply. This part of ISO/IEC 9592 also makes use of the definitions in ISO/IEC 9592-1. (Terms used within definitions in this clause that are themselves defined in this clause are italicized.)

3.1 ambient light source: A *light source* that contributes to the *reflectance calculation* independently of the orientation or position of the area being illuminated or the location of the viewer's eye.

3.2 ambient reflection coefficient: The fraction of *ambient light* reflected from an area.

3.3 area primitive: Any of the output primitives: fill area, fill area set, cell array, *fill area set with data*, cell array PLUS, *set of fill area sets with data*, *triangle set* with data, *triangle strip* with data, *quadrilateral mesh* with data, non-uniform B-spline surface and non-uniform B-spline surface with data. In addition, some generalized drawing primitives may have this classification.

3.4 attenuation coefficient: A coefficient that determines the decrease in intensity of light as a function of the distance between a *light source* and an illuminated object.

3.5 back facing: A back-facing facet has a *facet normal* that, when transformed to NPC, has a negative Z component. See also *front facing*.

3.6 colour mapping: The conversion of direct colours in the *rendering pipeline* to other colours before they are displayed on the workstation.

3.7 colour spline: The parametric curve or surface in colour space (or homogeneous colour space) defining the colour distribution over an output primitive.

3.8 concentration exponent: A parameter of a *spot light* source that specifies the relative decrease of light as the angle of the light diverges from the centreline of the light source's *cone of influence*.

3.9 cone of influence: A conceptual cone that represents the influence of light from a *spot light* source. The cone of influence is defined by the light source's position, direction and *spread angle*.

3.10 data mapping: The conversion of application-specific data or colour to *intrinsic colour*.

3.11 depth cueing: An effect in which the colours of points on an output primitive are combined with a specified depth cue colour. The degree of combination is dependent on the depth (Z in normalized projection coordinates) of the points.

3.12 depth cue mode: A field in each entry of the *depth cue table* of the workstation state list that indicates whether or not *depth cueing* should be performed.

3.13 depth cue table: A table in the workstation state list that contains information used to control *depth cueing*.

3.14 diffuse reflection: An approximation of the light reflected equally in all directions from an area.

3.15 diffuse reflection coefficient: The fraction of light from non-*ambient light* sources that is diffusely reflecting from an area.

3.16 direct colour specification: A non-indexed method of specifying colour where the components of the colour, i.e., coordinates in colour space, are specified together with the colour model in which those components are expressed.

3.17 directional light source: A *light source* that contributes to the *reflectance calculation* dependent on the orientation of the area being illuminated but independent of the area's position.

3.18 edge visibility flag: An indicator that is part of the specification of some output primitives, such as *fill area set with data*, that controls whether an individual edge is visible.

3.19 eye point: A point in world coordinates that transforms to infinite positive Z in normalized projection coordinates. This point is used in the *reflectance calculation* for determining viewing-position-dependent effects of lighting.

3.20 facet: An interior segment of an *area primitive*. Each facet of an output primitive is defined by a subset of the primitive's set of vertices. The subset is dependent on the individual primitive type, and in the case of parametric surfaces, on the approximation of the surface. Facets have an orientation in NPC described as *back-facing* or *front-facing*.

3.21 facet culling: The process of removing *front-facing* or *back-facing facets* of area primitives.

3.22 facet data: *Intrinsic colour* data or a *normal vector* specified with an *area primitive*.

3.23 facet normal: A *normal vector* associated with a *facet* of an *area primitive*. Facet normals are used to determine the orientation of a facet and in some cases for determining the *reflectance normal*.

3.24 fill area set with data: An output primitive consisting of a set of coplanar polygons. It is similar to the fill area set output primitive defined in ISO/IEC 9592-1. The corresponding structure element may include other information such as colours or normals that are conditionally used to colour, light and shade the output primitive.

3.25 front facing: A front-facing facet has a *facet normal* that, when transformed to NPC, has a positive or zero Z component. See also *back facing*.

3.26 general colour: A data type that allows both the direct and indirect specification of colour. General colour specifies a colour type together with a type-dependent colour value. The colour type can either indicate a colour model, in which case the colour values are coordinates in the colour space corresponding to that model, or it can indicate that the colour is being specified indirectly, in which case the single colour value is an index into the workstation-dependent colour table.

3.27 geometry spline: The parametric curve or surface defining the geometry of a *parametric output primitive*.

3.28 indirect colour specification: A method of specifying colour via an index into a workstation dependent colour table.

3.29 intrinsic colour: The colour or colours of an output primitive that are independent of *lighting*, *depth cueing* and *colour mapping*.

3.30 intrinsic colour data: Colour or application-specific data associated with output primitives and specified in the output primitive's structure element. Intrinsic colour data, when specified, is conditionally used to determine the *intrinsic colour* of an output primitive.

3.31 isoparametric curve: A curve on a parametric surface produced by evaluating the surface over the range of one of its independent variables while holding its other independent variable constant.

3.32 knot vector: A non-decreasing sequence of real numbers that is part of the definition of non-uniform B-splines. This vector consists of values of the independent variables and is used in computing the B-spline basis polynomials.

3.33 light source: A simulated source of light.

3.34 light source direction: A unit vector that defines the orientation of oriented *light sources*.

3.35 light source state: A field in the traversal state list that selects which *light sources* in a workstation light source table are active.

3.36 lighting: See *reflectance calculation*.

3.37 normal vector: A unit length vector, typically indicating the orientation of a *facet* or object.

3.38 parameter range: The *parameter space* over which a parametric curve or surface is defined.

3.39 parameter range limits: Minimum and maximum parameter values, specified separately from any knot values, that limit the parameter range over which parametric curves are generated.

3.40 parametric output primitive: Output primitives defined as a mapping from a parameter space to modelling coordinates. Parametric output primitives defined in PHIGS PLUS are non-uniform B-spline curve, non-uniform B-spline curve with colour, non-uniform B-spline surface and non-uniform B-spline surface with data.

3.41 parameter space: The coordinate system of the independent variable(s) of parametric curves and surfaces. The parameter space is one-dimensional for curves and two-dimensional for surfaces.

3.42 polyline set with colour: An output primitive consisting of an unconnected set of polylines. The corresponding structure element may include colour information that is conditionally used to shade the primitive.

3.43 portion: A portion of an area primitive refers to one or more *facets* of the primitive that are distinguished as a group from its other facets by some property such as orientation or position relative to a *trimming loop*. The term can be applied to groups of explicitly defined facets as well as the facets conceptually used to approximate a non-uniform B-spline surface.

3.44 positional light source: A *light source* that contributes to the *reflectance calculation* dependent on the orientation and position of an area being illuminated relative to the light source.

3.45 quadrilateral mesh: An output primitive in which an array of quadrilaterals is specified by a two-dimensional array of vertices.

3.46 reflectance calculation: The computation of the effect of *light sources* on the colour of an area primitive's facets.

3.47 reflectance model: An aspect that selects the *reflectance calculation* and thereby specifies which lighting effects are to be displayed.

3.48 reflectance formulae: Formulae that model the light reflected by an *area primitive*.

3.49 reflectance normal: A vector used in the *reflectance calculation* and indicating the orientation of a primitive at a point on the primitive. The vector is conceptually perpendicular to the surface of an object being represented by an *area primitive*. It is derived from the *vertex normals* of the primitive, if specified, or the *facet normal*.

3.50 reflectance properties: An aspect of *area primitives* that indicates how a primitive reflects light.

3.51 rendering colour model: The colour model used for performing colour interpolation during *shading* and *depth cueing*.

3.52 rendering pipeline: A sequence of operations that performs *data mapping*, *lighting*, *shading*, *depth cueing*, and *colour mapping* of output primitives. Each of these operations is considered a stage in the rendering pipeline.

3.53 rigid-body transformation: A modeling transformation composed of at most translation, rotation, and scaling transformations, where translation transformations move every point of an object an equal distance in the same direction, the rotation transformations maintain relative angles, and the scaling transformations apply equal scaling in all coordinate dimensions.

3.54 set of fill area sets with data: An output primitive in which a number of possibly non-coplanar fill area sets are defined by indices into a single list of vertex data. The fill area sets are not required to form a closed or connected surface.

3.55 shading: The interpolation stage of the *rendering pipeline*.

3.56 specular colour: A *reflectance property* indicating the effect of a primitive on the colour of *specular reflections* from that primitive.

3.57 specular exponent: A non-negative number indicating the shininess of an area. The higher the specular exponent, the shinier the area. A specular exponent of 0 indicates a minimum relative degree of shininess.

3.58 specular reflection: An approximation of the unequal reflection of light in different directions from an *area primitive*, dependent on the relationship of the viewer to the primitive and the *light source*.

3.59 specular reflection coefficient: The fraction of non-*ambient light* contributing to *specular reflection*.

3.60 spot light source: A *light source* that contributes to the *reflectance calculation* dependent on the orientation and relative position of the area being illuminated. Light from such a source is restricted to a semi-infinite cone of influence and its intensity may decrease as it deviates from the centreline of this cone. (See *concentration exponent* and *spread angle*.)

3.61 spread angle: An angle that determines the shape of the *cone of influence* of a *spot light* source. Spread angle is the angle between the center of the cone of influence and the limit of the cone of influence measured at the position of the spot light source.

3.62 triangle set: An output primitive in which a number of possibly unrelated triangular facets are defined by indices into a single list of vertex data.

3.63 triangle strip: An output primitive comprised of a group of adjacent triangles formed by connecting a list of vertices such that the second and third vertices of each triangle are used as the first and second vertices of the next triangle.

3.64 trimming curve: A parametric curve in the parameter space of the surface to which it applies. Trimming curves are combined to form *trimming loops* which limit the *parameter range* over which a parametric surface is evaluated.

3.65 trimming loop: A sequence of connected and similarly oriented *trimming curves* that form a closed path. Trimming loops are used to limit the *parameter range* over which a parametric surface is evaluated.

3.66 vertex colour: A *general colour* associated with each vertex of some output primitives. This colour is conditionally used within the *rendering pipeline* to colour and shade the primitive.

3.67 vertex data: Geometric, *intrinsic colour data*, or *vertex normal* data specified at vertices of certain output primitives.

3.68 vertex normal: A *normal vector* optionally supplied with the *vertex data* of some area primitives.

3.69 with-data primitives: Any of the output primitives: *fill area set with data*, *set of fill area sets with data*, *triangle set* with data, *triangle strip* with data, *quadrilateral mesh* with data and non-uniform B-spline surface with data. In addition, some generalized drawing primitives may have this classification.

3.70 workstation light source. A *light source* entry in a workstation light source table.

4 The PHIGS PLUS system

4.1 About this part of ISO/IEC 9592

4.1.1 Specification and conformance

The set of functions known as PHIGS PLUS shall be as described in clauses 4, 5, and 6 of this part of ISO/IEC 9592 in addition to the functions described in clauses 4, 5, and 6 of ISO/IEC 9592-1. A conforming implementation of PHIGS PLUS shall be a conforming implementation of ISO/IEC 9592-1, as amended by amendments to that part, and in addition correctly implement all the functions described in clauses 4, 5, and 6 of this part of ISO/IEC 9592. Minimum support criteria for the additional functionality are specified in 4.10. A conforming implementation of PHIGS PLUS shall provide the additional information described in annex D as well as annex D of ISO/IEC 9592-1, and the minimum support specified in 4.10 as well as 4.14 of ISO/IEC 9592-1. In a conforming implementation all graphical capabilities that can be addressed by PHIGS PLUS shall be used only via PHIGS PLUS.

4.1.2 Registration

For certain parameters of the functions, PHIGS PLUS defines value ranges as being reserved for registration.[1] The meanings of such values and ranges will be defined in the Register. These procedures do not apply to values and value ranges defined as being workstation or implementation dependent; such values and ranges are not standardized.

4.1.3 Notational conventions

This part of ISO/IEC 9592 uses the following typographical conventions.

a) The names of primitive attributes appear in upper case.

b) Geometric aspects of primitives appear in upper case. Non-geometric aspects appear in lower case except when the corresponding individually specified attributes are being used, in which case they appear in upper case.

c) The names of entries in the PHIGS data structures, including those defined by PHIGS PLUS, appear in lower case surrounded by single quotation marks.

d) The names of structure elements appear in lower case surrounded by double quotation marks.

e) The names of PHIGS PLUS functions appear in upper case.

f) The values of an enumeration data type appear in upper case.

g) The names of data types appear in upper case.

1) For the purpose of this part of ISO/IEC 9592 and according to the rules for the designation and operation of registration authorities in the ISO/IEC Directives, the ISO and IEC Councils have designated the following as the registration authority: National Institute of Standards and Technology (National Computer Systems Laboratory), The Registration Authority for Graphical Items, A-266 Technology Building, Gaithersburg, MD 20899, USA

4.2 Overview and concepts

4.2.1 Overview

The Programmer's Hierarchical Interactive System (PHIGS) Plus Lumière Und Surfaces (PHIGS PLUS) provides a functional interface between an application program and a configuration of graphical input and output devices. The functional interface contains all of PHIGS, the basic functions for interactive graphics on a wide range of graphics equipment, and adds new facilities for the specification of curved lines and curved and facetted surfaces. Lighting and other effects such as depth modulation can be specified to allow realistic presentation of the geometry. There is direct support for the visualization of data that can be associated with the geometry.

The interface is at such a level of abstraction that hardware peculiarities are shielded from the application program. As a result, a simplified interface presenting the additional primitives is obtained.

Some PHIGS PLUS concepts are described as being workstation dependent. This means the implementation is allowed to specify the characteristics of these concepts on a workstation by workstation basis. All workstation dependent concepts are explicit in the standard. Other PHIGS concepts are described as being implementation dependent. This means that an implementation is allowed to determine the effect (rather than having the effect mandated by the standard). However such effects shall be the same on all workstations in a single implementation.

Annex D (taken with annex D of ISO/IEC 9592-1) describes the permitted differences between PHIGS PLUS implementations.

4.2.2 Concepts

PHIGS PLUS defines a set of output primitives whose definitions contain, in addition to the geometry of the primitive, information that further specifies certain characteristics of the primitive and influences its appearance on the display surface. These output primitives are called *with-data primitives*.[2] The data defined for these with-data primitives is used to colour and shade the primitive, and in the case of *area primitives*, to indicate the primitive's apparent orientation and control the visibility of its edges. Depending on the output primitive, the information that can be specified in addition to geometry is *intrinsic colour data* that indicates a colour distribution across the primitive, *normal vectors* that indicate the primitive's apparent orientation, and *edge visibility flags* that control the visibility of an area primitive's edges. This data is optionally specified and conditionally used when rendering the primitive. Some of the with-data primitives are the output primitive types defined in ISO/IEC 9592-1 extended to include the additional information. Some other with-data primitives offer data compaction and other advantages.

A colour is specified in PHIGS PLUS as a *general colour*, which consists of a colour type and a type-dependent colour value. The colour type can indicate any one of a number of colour models, in which case the corresponding colour value is interpreted as a value in that colour space. Alternatively, the colour type can indicate an indirect colour, in which case the colour value is an index into the workstation colour table. PHIGS PLUS thus supports the indirect colour specification of ISO/IEC 9592-1 as a subset of a more general colour specification mechanism. PHIGS PLUS defines general colour extensions to most areas of ISO/IEC 9592-1 where colours are specified.

PHIGS PLUS has the concept of a *rendering pipeline*. Conceptually, all output primitives pass through this pipeline as they are displayed. Operations are performed in the pipeline that affect the colour and appearance of each facet of a primitive. These operations include, for most primitives, the mapping of application-specific data to colour (*data mapping*), the calculation of reflectance effects (*lighting*), data interpolation (*shading*), and *depth cueing*. Output primitive attributes and any data attached to the primitives are conditionally used in these operations. The last stage of the rendering pipeline is a *colour mapping* that maps the resultant colours from the earlier stages into workstation dependent colours. The application may have control over which of these stages are active and what methods are used within each stage.

2) In this subclause, terms in the glossary are italicized when first used.

Data mapping in PHIGS PLUS provides a mechanism to convert application-specific data to colour within the rendering pipeline. Data such as temperature, pressure or other scalar or multi-dimensional information can be specified for most area primitives. Selected portions of this data may then be used during traversal to determine the *intrinsic colour* of the primitive, which is the colour or set of colours associated with a primitive before the effects of lighting are applied. The method to use to convert the data to intrinsic colour is under application control. More data can be specified than is used during any single rendering of a primitive.

Explicitly specified and implicitly specified aspects control the operations of the rendering pipeline. Explicitly specified aspects are *facet normals*, used to determine facet orientation, and *vertex normals* used to determine *reflectance normals* in *reflectance calculations*. When not supplied, facet normals are implicitly determined from the geometry of the facet.

PHIGS PLUS defines the *light sources* and suggests their interaction with output primitives. Several types of light sources are defined: *ambient*, *directional*, *positional* and *spot*. *Workstation light sources* are associated with workstations and are defined as entries in the 'light source table' in the workstation state list.[3] Active light sources are selected from this table by the 'current light source state' in the traversal state list.

Each *facet* of an area primitive has an orientation, either *front facing* or *back facing* as determined in normalized projection coordinates from the facet normal. PHIGS PLUS defines separate attributes for front and back facing facets of area primitives and provides controls for determining when they should be applied. PHIGS PLUS also defines an attribute to control whether individual facets are removed prior to viewing based on their orientation.

PHIGS PLUS defines non-uniform B-spline curve and surface output primitives. These output primitives are specified by a series of control points and parameter values that define non-uniform B-spline curves and surfaces in terms of the B-spline representation. This interface allows the exact specification of commonly required shapes including conics, conic sections such as circles, ellipses, parabolae, spheres, and spheroids without resorting to special cases. The curves and surfaces are represented by non-rational and rational splines using the mathematically robust B-spline representation.- Variants of these primitives are defined that allow specification of an intrinsic colour data distribution over the primitive.

PHIGS PLUS replaces some fields of some of the data structures defined in ISO/IEC 9592-1 and adds a number of fields to these data structures. Replacements and additions have been made to the PHIGS description table, traversal state list, workstation state list and workstation description table, and are listed in clause 6.

3) Light sources with their specification included directly in the CSS are not defined by this part of ISO/IEC 9592, but their inclusion as GSEs or GDPs is not prohibited.

4.3 PHIGS PLUS structure elements

4.3.1 Output primitive structure elements

The output primitive structure elements defined by this part of ISO/IEC 9592 are listed in table 1.

Table 1 — Output primitive structure elements

polyline set 3 with colour	triangle strip 3 with data
fill area set 3 with data	triangle strip with data
fill area set with data	quadrilateral mesh 3 with data
cell array 3 PLUS	quadrilateral mesh with data
set of fill area sets 3 with data	non-uniform B-spline curve 3
set of fill area sets with data	non-uniform B-spline curve 3 with colour
triangle set 3 with data	non-uniform B-spline surface 3
triangle set with data	non-uniform B-spline surface 3 with data

4.3.2 Attribute specification structure elements

The attribute specification structure elements defined by this part of PHIGS are listed in table 2.

Table 2 — Attribute specification structure elements

set polyline colour	set back reflectance properties
set polyline shading method	set interior shading method
set polymarker colour	set back interior shading method
set text colour	set reflectance model
set edge colour	set back reflectance model
set back interior style	set reflectance index
set back interior style index	set back reflectance index
set interior colour	set light source state
set back interior colour	set depth cue index
set back interior index	set rendering colour model
set facet distinguishing mode	set colour mapping index
set facet culling mode	set curve approximation criteria
set data mapping method	set surface approximation criteria
set back data mapping method	set parametric surface characteristics
set data mapping index	set parametric surface index
set back data mapping index	set individual ASF (additions)
set reflectance properties	

4.3.3 Structure element archive

This part of ISO/IEC 9592 defines output primitive and attribute specification structure elements. The archive file format for these structure elements is defined in ISO/IEC 9592-2 and its amendment 1. The clear text encoding of that format is defined in ISO/IEC 9592-3 and its amendment 1.

4.4 Output primitives

4.4.1 General

PHIGS PLUS augments the set of output primitives defined in ISO/IEC 9592-1 by defining additional output primitives and output primitive structure elements. Some of these new output primitives can have optional data associated with them. The optional data can be some combination of vertex normals, facet normals, edge visibility flags, and intrinsic colour data, as shown in figure 1. Depending on various attribute settings, PHIGS PLUS uses this information at various points in the rendering pipeline to light, shade and otherwise determine the appearance of the output primitive. If the application does not provide the optional information, PHIGS PLUS, under some conditions, computes portions of it and uses the computed information in the rendering pipeline.

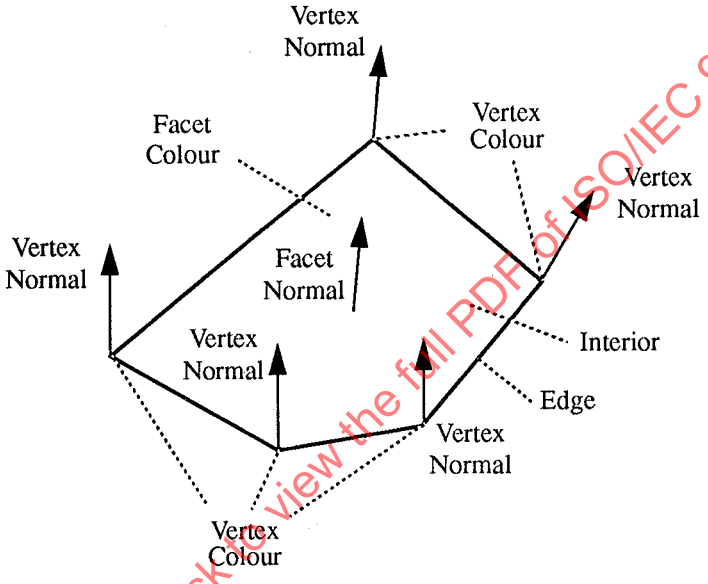


Figure 1 — Data that may be associated with a facet

When present, normal vectors should be of unit length. The specification of a non-unit length normal vector does not produce an error when the element is created. When primitives containing non-unit length normal vectors are interpreted during traversal, the effect of the non-unit length normals is workstation dependent.

4.4.2 Polyline set with colour

PHIGS PLUS generates a set of unconnected polylines. Each polyline is defined by vertex sequences. Colour information can be specified with each vertex as part of the primitive definition.

4.4.3 Fill area set with data

PHIGS PLUS generates a fill area set that can be empty, hollow, hatched, or filled with a colour distribution or pattern. The edges of a fill area set consist of the line segments comprising its defining polygons. The primitive specification may contain any combination of facet or vertex intrinsic colour data, vertex normals, a facet normal, and edge visibility flags. The interior of a fill area set with data primitive is the same as that defined in ISO/IEC 9592-1 for a fill area set primitive.

Ensuring that all the points of the fill area set are coplanar is the responsibility of the application. While slightly non-coplanar points (e.g. owing to numerical roundoff errors) should be mapped to a plane by the implementation, the effect of larger divergences is workstation dependent.

4.4.4 Cell array PLUS

PHIGS PLUS generates a cell array. The cell colours are specified with general colours.

4.4.5 Set of fill area set with data

PHIGS PLUS generates a group of fill area sets. The vertices of each fill area set are specified by lists of indices into a single list of vertices. This allows vertices and their associated data to be shared among the fill area sets. The specification may contain any combination of facet or vertex intrinsic colour data, vertex normals, facet normals, and edge visibility flags.

The fill area sets are not required to be coplanar. Each individual fill area set in the set of fill area sets, however, is expected to be planar. Ensuring that each fill area set is planar is the responsibility of the application. The effect produced by non-planar fill area sets is workstation dependent.

4.4.6 Triangle set with data

PHIGS PLUS generates a set of triangles (see figure 2). The vertices of each triangle are specified by three indices into a list of vertices. The set of triangles need not be coplanar. The primitive definition allows specification of any combination of facet or vertex intrinsic colour data, vertex normals, facet normals, and edge visibility flags.

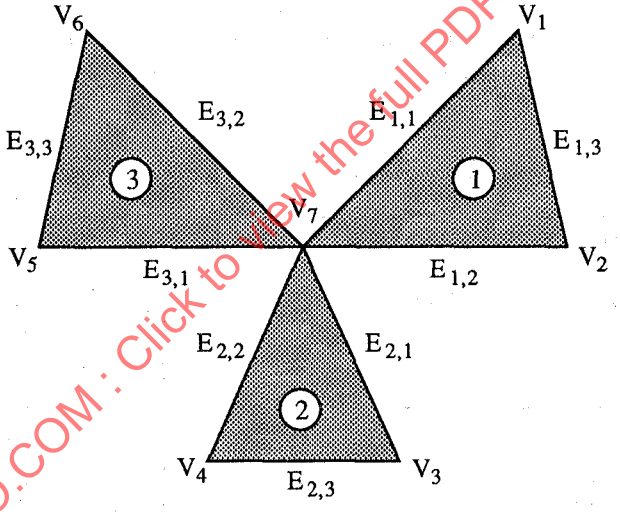


Figure 2 — A triangle set primitive (The vertex list is {V_1, V_2, V_3, V_4, V_5, V_6, V_7}. The list of vertex indices is {(1, 7, 2), (3, 7, 4), (5, 7, 6). The list of edge visibility flags is {$E_{1,1}$, $E_{1,2}$, $E_{1,3}$, $E_{2,1}$, $E_{2,2}$, $E_{2,3}$, $E_{3,1}$, $E_{3,2}$, $E_{3,3}$})

4.4.7 Triangle strip with data

PHIGS PLUS generates a set of $N-2$ triangles from a list of N vertices (see figure 3). Each triangle of the strip has as its vertices three successive vertices from the vertex list. The set of triangles need not be coplanar. The triangles are implicitly numbered by the ordering of the defining list of vertices. The first triangle in the strip is numbered 1. The edges of a triangle strip are the line segments forming the boundaries of all triangles in the strip. There is only one edge between any two adjacent triangles, hence there are $2(N-2) + 1$ edges. The primitive definition allows specification of any combination of facet or vertex intrinsic colour data, vertex normals, facet normals, and edge visibility flags.

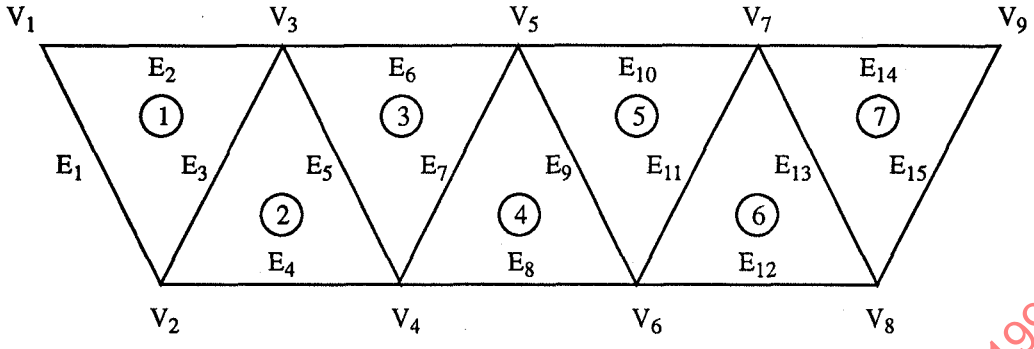


Figure 3 — A triangle strip primitive (Vertices, V_i, are identified by their position in the vertex list. Edges, E_i, are identified by their position in the optional list of edge visibility flags)

4.4.8 Quadrilateral mesh with data

PHIGS PLUS generates a set of $(M-1) \times (N-1)$ quadrilaterals from a two dimensional array of $M \times N$ vertices (see figure 4). The set of quadrilaterals need not be coplanar. The edges of a quadrilateral mesh are the line segments forming the boundaries of the individual quadrilaterals in the mesh. There is only one edge between any two adjacent quadrilaterals, hence there are $M(N-1) + N(M-1)$ edges. The primitive definition allows specification of any combination of facet or vertex intrinsic colour data, vertex normals, facet normals, and edge visibility flags. Quadrilaterals in the mesh can be non-planar. Non-planar quadrilaterals are rendered in a workstation dependent way.

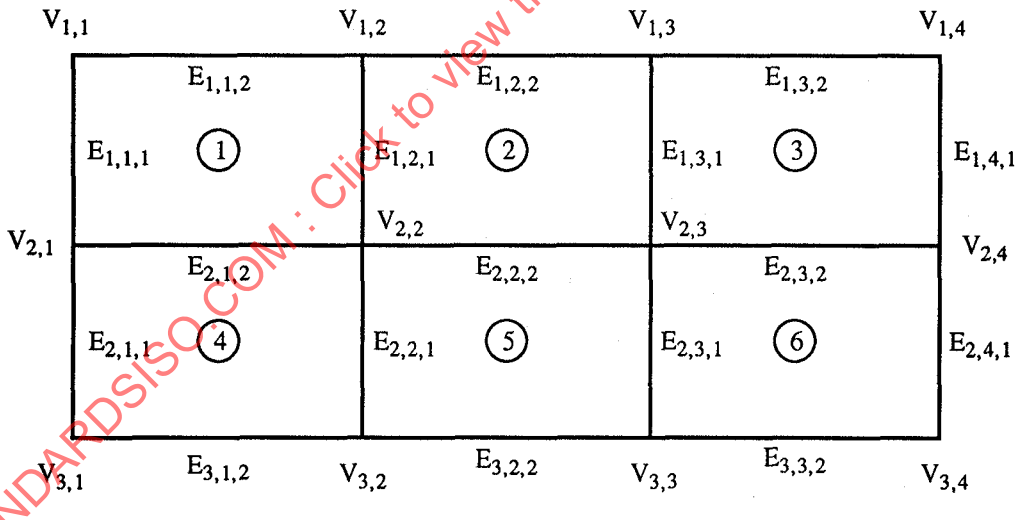


Figure 4 — A quadrilateral mesh primitive, M = 3, N = 4 (Vertices $V_{i,j}$ are identified by their position in the vertex array. Edges $E_{i,j,k}$ are identified by their position in the optional edge flag array)

4.4.9 Non-uniform B-spline curve

PHIGS PLUS generates a non-uniform B-spline curve of a specified order over a specified range of a single independent parameter. The curve is specified by its order, its rationality, a list of control points in modelling coordinates, a list of knots in the one-dimensional parameter space, and two parameter range limits. Either a rational or a non-rational B-spline curve can be specified. The extreme values of the knots specify the curve's parameter range. Parameter range

limits can be specified that limit the range over which the independent variable is evaluated and the B-spline curve is generated.

A B-spline curve is defined in PHIGS PLUS as a mapping from a bounded one-dimensional parameter space into a set of points that compose the curve:

$$C(t) = \sum_{i=1}^{n} B_{i,k}(t)\, P_i \tag{1}$$

where

$C(t)$ is the curve;

P_i are the n control points;

$B_{i,k}(t)$ is the i-th B-spline basis function of order k;

t is the parameter, $t_k \le t \le t_{n+1}$.

The B-spline basis functions are determined by the order k and a knot vector, $\{t_j\}, j = 1 \text{ to } n+k$, where the sequence t_j is non-decreasing; thus the number of knots is equal to the number of control points plus the curve order. The parameter range is defined by the knot values t_k and t_{n+1}, and defines the bounds of the parameter space domain over which the curve is defined. The B-spline basis functions are defined by the recursion formula:

$$B_{i,1}(t) = \begin{cases} 1, & t_i \le t \le t_{i+1} \\ 0, & \text{elsewhere} \end{cases} \tag{2a}$$

$$B_{i,k}(t) = \begin{cases} 0, & t < t_i \text{ or } t < t_{i+k} \\ \dfrac{(t - t_i)\, B_{i,k-1}(t)}{t_{i+k-1} - t_i} + \dfrac{(t_{i+k} - t)\, B_{i+1,k-1}(t)}{t_{i+k} - t_{i+1}}, & t_i \le t \le t_{i+k} \end{cases} \tag{2b}$$

The terms $\dfrac{(t - t_i)\, B_{i,k-1}(t)}{t_{i+k-1} - t_i}$ and $\dfrac{(t_{i+k} - t)\, B_{i+1,k-1}(t)}{t_{i+k} - t_{i+1}}$ may produce the indeterminate quantity $\dfrac{0}{0}$ when multiple identical knot values are specified. In these cases the terms are defined to be 0.

The curve is said to be uniform if there exists a real number, δ, such that $t_{j+1} - t_j = \delta$ for all j such that $k \le j \le n$; otherwise the curve is said to be non-uniform.

A non-rational non-uniform B-spline curve output primitive is generated by applying equation 1 to a set of modelling coordinate points, $P_i = (x_i, y_i, z_i)$.

A rational B-spline curve is defined in PHIGS PLUS as a mapping from a bounded one-dimensional parameter space into a set of homogeneous points:

$$C^h(t) = \sum_{i=1}^{n} B_{i,k}(t)\, P_i^h \tag{3}$$

where $P_i^h = (w_i q_i, w_i r_i, \ldots, w_i)$.

The projection of $C^h(t)$ into non-homogeneous coordinates is achieved by dividing all the curve's coordinates by the last coordinate, w. This definition of a B-spline curve applies to the non-uniform B-spline curve output primitive, any colour spline associated with that output primitive, and trimming curves associated with non-uniform B-spline surfaces.

A rational non-uniform B-spline curve output primitive is generated by applying equation (3) to a set of homogeneous modelling coordinate points, $P_i^h = (w_i x_i, w_i y_i, w_i z_i, w_i)$, and projecting the resulting points to three-dimensional modelling coordinates.

In both the rational and non-rational cases the curve generated by PHIGS PLUS is the result of mapping a portion of the curve's parameter range to modelling coordinates. The portion to map is indicated by parameter range limits specified with the structure element. The curve is mapped only within and at the parameter range limits, and this portion constitutes the geometry of the curve for the purposes of display, picking and incremental spatial search. The parameter range limits shall be within the specified parameter range of the curve.

Conceptually, a non-uniform B-spline curve is rendered by evaluating a number of sample points on or near the curve. The density of the sample points is controlled by the curve approximation criteria. If the sampling is higher than the density of points on the display surface, displaying the sample points themselves produces a complete rendering of the curve. Alternatively, adjacent sample points may be joined by straight lines that are then displayed.

4.4.10 Non-uniform B-spline curve with colour

PHIGS PLUS generates a non-uniform B-spline curve. The curve's definition contains an optional non-uniform B-spline curve that describes the colour distribution over the curve's geometry. The control points of this colour spline are colour coordinates in one of the available directly-specifiable colour models.

The control points of the colour spline are colour values of the appropriate dimension for the specified colour model. The parameter range of the colour spline shall wholly include the parameter range limits of the geometry spline, thus the colour for any point on the mapped portion of the curve can be determined by evaluating the colour spline at the corresponding parameter value.

The colour spline need not have the same order, rationality nor knot sequence as the geometry spline. The values of the colour spline control point coordinates can be outside the range normally associated with the colour value. (This is necessary to allow colour specification with rational B-splines, and to allow B-splines that interpolate a set of colours.) If the colour spline is rational, the dimension of the control points is one more than the dimension of the corresponding colour model coordinates. The colour spline is defined by equations (1) and (3).

The curve's colour is determined from the colour spline at the sample points used for the geometry spline. These colours are then considered to be vertex colours. The polyline shading method aspect indicates how these vertex colours are applied to the rendering of the curve.

4.4.11 Non-uniform B-spline surface

PHIGS PLUS generates a non-uniform B-spline surface of two specified orders over a specified range of two independent parameters. The surface is specified by its order, its rationality, an array of control points in modelling coordinates, and a list of knots for each of the independent parameters. Either a rational or a non-rational B-spline surface can be specified. The extreme values of the knots specify the parameter range of the surface. An optional set of trimming curves (defined by non-uniform B-spline curves) can be specified to limit the parameter range over which the independent variables are evaluated and the B-spline surface is generated. If trimming curves are specified the edges of this primitive consist of the points on the displayed surface corresponding to the trimming curves. If no trimming curves are specified the edges consist of curves of constant parameter value at the parameter range boundaries of the surface.

A B-spline surface is defined in PHIGS PLUS as a mapping from a bounded two-dimensional parameter space into a set of points that compose the surface:

$$S(u, v) = \sum_{i=1}^{n} \sum_{j=1}^{m} B_{i,k}(u)\, B_{j,l}(v)\, P_{i,j} \qquad (4)$$

where

$S(u, v)$ is the surface;

$P_{i,j}$ are an $n \times m$ array of control points;

$B_{i,k}(u)$ is the i-th B-spline basis function of order k, defined by the knot vector $\{u_p\}$, $p = 1$ to $n+k$;

$B_{j,l}(v)$ is the j-th B-spline basis function of order l, defined by the knot vector $\{v_q\}$, $q = 1$ to $m+l$;

u and v are the two parameters, $u_k \le u \le u_{n+1}$, $v_l \le v \le v_{m+1}$.

The B-spline basis functions are determined by the orders k and l and the respective knot vectors, each of which is a non-decreasing parameter sequence. The B-spline basis functions are defined by equation (2) in 4.4.9. The parameter range is defined by the knot values u_k, u_{n+1}, and v_l, v_{m+1}, which define the bounds of the parameter space domains over which the surface is defined. The surface is said to be uniform in the u parameter if there exists a real number, δ, such that $u_{p+1} - u_p = \delta$ for all p such that $k \le p \le n$; otherwise the surface is said to be non-uniform in that parameter. The surface is said to be uniform in the v parameter if a similar test holds for that parameter. The surface is said to be non-uniform if it is non-uniform in either parameter.

A non-rational non-uniform B-spline surface output primitive is generated by applying equation (4) to a set of modelling coordinate points, $P_{i,j} = (x_{i,j}, y_{i,j}, z_{i,j})$. A rational non-uniform B-spline surface output primitive is generated by applying equation 4 to a set of homogeneous modelling coordinate points, $P_{i,j}^h = (w_{i,j}x_{i,j}, w_{i,j}y_{i,j}, w_{i,j}z_{i,j}, w_{i,j})$, and projecting the resulting points to three-dimensional modelling coordinates by dividing the first three coordinates of each point by the point's homogeneous coordinate, w.

In both the rational and non-rational cases the surface generated by PHIGS PLUS is the result of mapping either all or portions of the surface's parameter range to modelling coordinates. If only portions are to be mapped then they are identified with a set of trimming loops (4.4.12). For the purposes of display, picking and incremental spatial search, only the portions of the surface actually mapped to modelling coordinates constitute the geometry of the primitive.

4.4.12 Surface trimming

The B-spline surface mapping is defined on a two-dimensional parameter space rectangle defined by the surface's parameter range. The mapping of the surface can be restricted to portions of that parameter range by specifying a set of trimming loops that define those portions. Trimming loops are composed of one or more trimming curves. Trimming curves are B-spline curves, as defined by equation 1 or 3, that map a one-dimensional parameter range to the two-dimensional parameter range of the surface. In the non-rational case, the P_i of equation 1 are two-dimensional control points in the parameter space of the surface, $P_i = (u_i, v_i)$. The trimming curve is the result of applying equation 1 to these points. In the rational case the P_i are homogeneous control points $P_i^h = (w_i u_i, w_i v_i, w_i)$. The trimming curve is the result of applying equation 3 to these points and projecting the computed points to two-dimensional parameter space.

Each trimming loop is composed of one or more trimming curves of order greater than or equal to two, connected to each other in a head to tail fashion. The head of a curve is the point corresponding to its minimum parameter range limit, the tail is the point corresponding to its maximum parameter range limit. A correctly specified trimming loop is explicitly closed; that is, the tail of each curve touches the head of the subsequent curve, and the tail of the last curve touches the head of the first curve. Correctly specified trimming loops do not intersect themselves or other trimming loops of the surface. Trimming curves only touch other trimming curves of the surface or themselves when they are adjacent in a trimming loop, and then only at their endpoints. Each trimming curve should be continuous. The implementation may assume continuity, non-intersection, head-to-tail connectivity, and closure of trimming loops. During traversal, trimming loops that violate these assumptions are treated in a workstation dependent manner.

Two rules are used to define the portion of the surface's parameter range to map (see figure 5):

a) **Odd Winding Rule**: A point is in the mapped portion of the surface's parameter range if any ray projected from that point to infinity has an odd number of intersections with the set of trimming loops.

b) **Curve Orientation Rule**: For any given trimming loop, the mapped portion of the surface is "to the left" of the directed loop and the unmapped portion is "to the right". The term "to the right" is defined as follows: Let $f(t) = (u, v)$ denote the trimming curve, and (du, dv) the tangent to the curve at point t oriented in the direction of increasing t; then the vector $(-dv, du)$ is said to point to the left at point t, and the vector $(dv, -du)$ is said to point to the right.

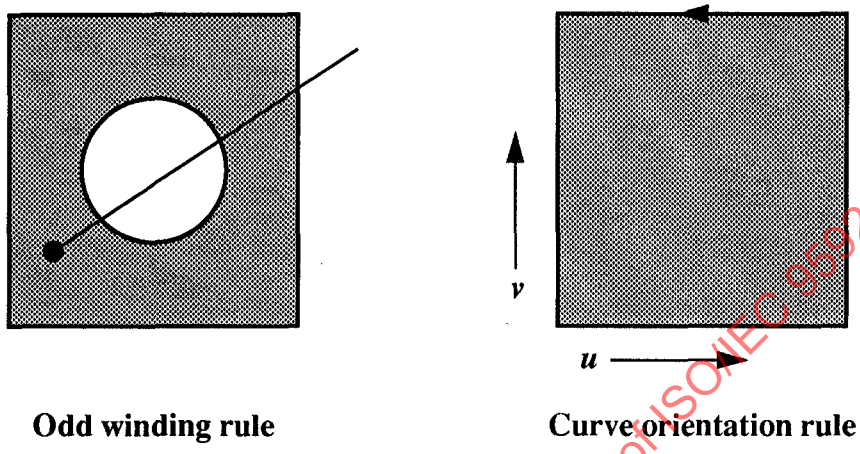


Figure 5 — Parameter space mapping rules for trimming curves (The mapped region is shaded)

The curve orientation rule requires that the curves be directed appropriately to define the mapped region. To be valid, the set of trimming loops shall satisfy both the above rules. If a set of trimming curves is specified that violate these rules, the resulting image is undefined and workstation dependent. See figure 6 for examples of valid trimming loop specifications. See figure 7 for examples of invalid trimming loop specifications.

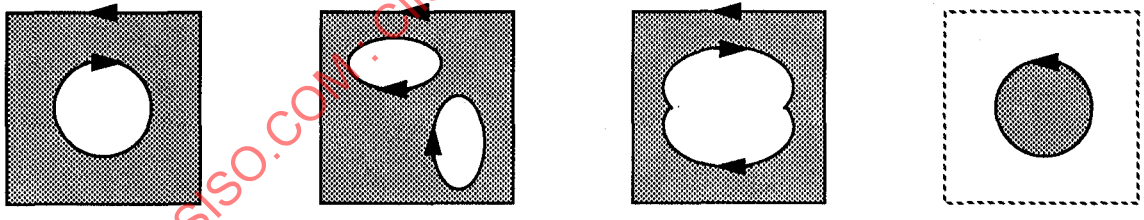


Figure 6 — Examples of valid trimming loops (Solid lines represent trimming loops, dashed lines represent the parameter range of the surface. Shading denotes the mapped region of the surface. Each arrowhead corresponds to a single trimming curve)

4.4.13 Non-uniform B-spline surface with data

PHIGS PLUS generates a non-uniform B-spline surface. The surface's definition may contain a number of associated non-uniform B-spline surfaces that, subject to data mapping, indicate the intrinsic colour of the surface. The control points of these associated surfaces are data or colour coordinates that describe the data or colour distribution over the primitive's geometry.

There may be a single color spline associated with the surface. The control points of the colour spline are colour values of the appropriate dimension for the specified colour type. When the colour spline is the source of intrinsic colour, the

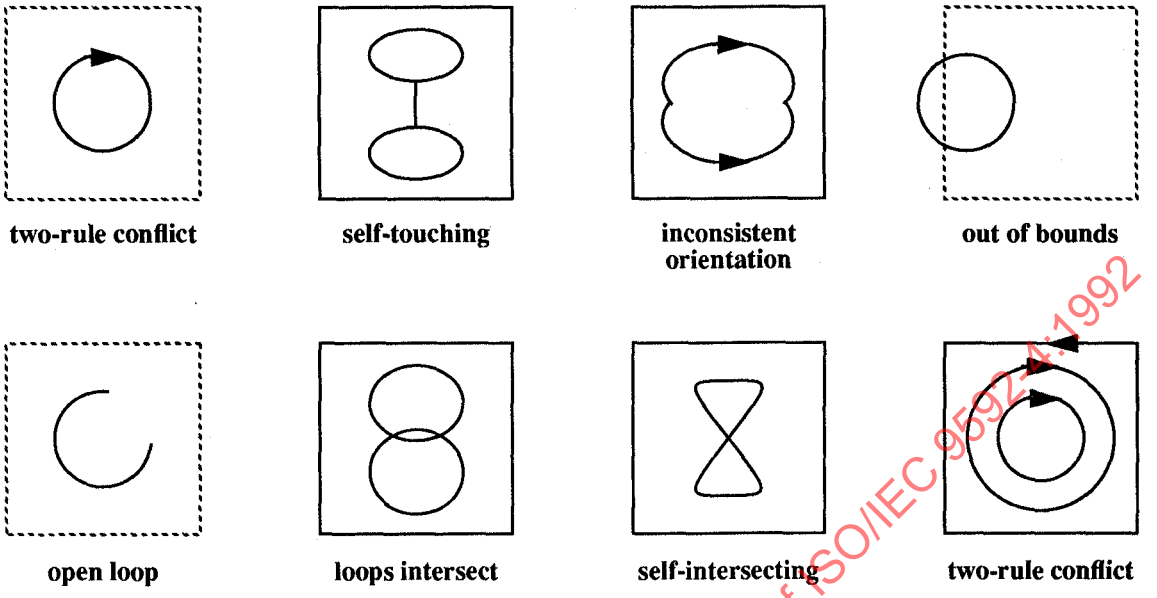


Figure 7 — Examples of invalid trimming loops (Solid lines represent trimming loops, dashed lines represent the parameter range of the surface. Each arrowhead corresponds to a single trimming curve.)

colour is determined from the colour spline at the sample points used for the geometry spline. These colours are then considered to be vertex colours of the corresponding facets.

The colour spline is a B-spline surface as defined by equation (4) in 4.4.11. It can be rational or non-rational. The colour spline need not have the same order, rationality or knot sequence as the geometry spline. The values of the control point coordinates can be outside the range normally associated with the colour type. (This is necessary to allow colour specification with rational B-splines, and to allow B-splines that interpolate a set of colours.) If the colour spline is rational the dimension of the control points is one more than the dimension of the corresponding colour model coordinates.

The parameter range of the colour spline shall wholly contain the parameter range of the geometry spline, or, if trimming loops are specified, the parameter range of the colour spline shall wholly contain the parameter range defined by the collection of trimming loops. Thus the colour for any point on the mapped geometry of the surface can be determined by evaluating the colour spline at the corresponding parameter value.

Application-specific data for data mapping can be associated with a non-uniform B-spline surface with data primitive by specifying an optional list of B-spline surfaces that define the data distribution over the surface geometry. The list of data splines corresponds to the list of real values associated with the other with-data area primitives, but there is only one list per primitive for non-uniform B-spline surfaces (as opposed to one list per facet or vertex for the other primitives). The data splines to use when mapping the data are indicated by the data selector in the data mapping method data record. When data mapping is the source of intrinsic colour, the data is determined from the data splines at the sample points used for the geometry spline. This data is then considered to be vertex data for the corresponding facets. Intrinsic colour is determined from this data according to the data mapping method and interior shading method aspects.

Data splines can have more than one dimension. Each data mapping method indicates how it uses the data values in multi-dimensional data splines. (The defined data mapping methods use only the first dimension.)

The data splines are B-spline surfaces as defined by equation (4) in 4.4.11. They can be rational or non-rational. The data splines need not have the same order, rationality nor knot sequence as the geometry spline. The parameter range of the data splines shall wholly contain the parameter range of the geometry spline, or, if trimming loops are specified, the parameter range of the data splines shall wholly contain the parameter range defined by the collection of trimming loops.

Thus the data values for any point on the mapped geometry of the surface can be determined by evaluating the data splines at the corresponding parameter value. Data splines can be specified in addition to an optional colour spline.

4.4.14 Area primitives and facets

Area primitives are fill area, fill area set, cell array, fill area set with data, cell array PLUS, set of fill area set with data, triangle set with data, triangle strip with data, quadrilateral mesh with data, non-uniform B-spline surface and non-uniform B-spline surface with data. For the purposes of specifying data and applying attributes area primitives are conceptually composed of facets. While B-spline surfaces are not inherently faceted, they are treated here as if faceted to describe their display and the application of interior attributes.

The definition of each primitive's facets are as follows:

a) the whole of a fill area or fill area set constitutes one facet;

b) each cell of a cell array or cell array PLUS constitutes one facet;

c) each fill area set of a set of fill area sets constitutes one facet;

d) each triangle of a triangle set or triangle strip constitutes one facet;

e) each quadrilateral of a quadrilateral mesh constitutes one facet;

f) Conceptually, a non-uniform B-spline surface is rendered by computing a number of sample points on or near the surface. The density of the sample points is controlled by the surface approximation criteria. If the sampling is higher than the density of points on the display surface, displaying the sample points themselves will produce a complete rendering of the surface. Alternatively, adjacent sample points can be joined to form polygons (usually triangles or quadrilaterals) that are then displayed as fill areas. The facets of the B-spline surface are the polygons or the sample points themselves (taken as areas). Depending on the approximation criteria, these facets can be smaller than the smallest displayable region on the workstation or as large as the surface itself.

   Vertex normals of facets are the normals to the surface at the corresponding sample points. The facet normal can be either a workstation dependent combination of the vertex normals, or the normal to the surface at some point on the facet. The point chosen in the latter case is workstation dependent.

4.4.15 Modelling clip

Modelling clip conceptually generates new vertices of output primitives at the intersection of the clipping planes and the primitive's boundary. If colour, data, or vertex normals are associated with the original vertices, then new colour, data, or vertex normals shall be interpolated at the new vertices. The interpolation method shall be such that if the geometry plus colour, geometry plus data, or geometry plus normal vectors are planar as defined by 4.6.4.2, then the interpolated values lie on the corresponding plane. If the set of vertices and associated data defining a facet are non-planar, then the results are workstation dependent.

4.5 Output primitive attributes

4.5.1 General colour specification

ISO/IEC 9592-1 defines only indirect colour specification. PHIGS PLUS defines direct colour specification and requires that both direct and indirect colour specification be supported.

Colour in PHIGS PLUS is specified through the concept of a general colour, which is capable of specifying colours both directly and indirectly. A general colour specifies a colour type and corresponding colour value. The colour type indicates either the colour model of the corresponding n-tuple of colour coordinates, or that the colour is being specified indirectly, in which case the single integer colour value is an index into the colour table of the workstation state list (see figure 8).

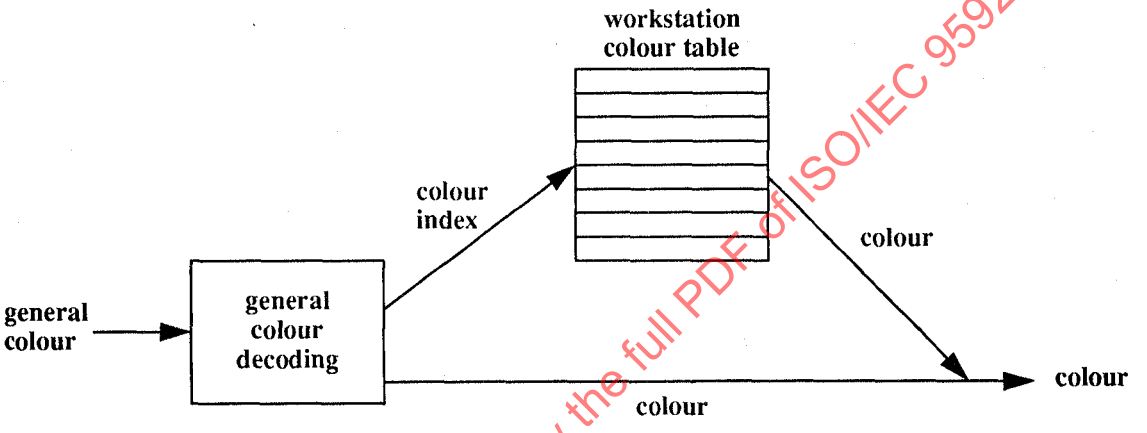


Figure 8 — A general colour holds either the index of an entry in the workstation colour table or a colour in a particular colour model

The defined colour types are listed in table 3. The result of resolving a general colour is a colour consisting (conceptually) of a colour model and the associated colour value.

Table 3 — Colour types

Value	Meaning
< 0	Implementation dependent.
0	INDIRECT (Indirectly specified colour)
1	RGB (Directly specified colour)
2	CIELUV (Directly specified colour)
3	HSV (Directly specified colour)
4	HLS (Directly specified colour)
≥ 5	Reserved for registration or future standardisation.[1]

1) Registered colour type values are drawn from the same set of values as colour model values.

As in ISO/IEC 9592-1, four colour models are defined: RGB, CIELUV, HSV, and HLS. See ISO/IEC 9592-1, annex I for a description of these colour models. The defined coordinate ranges are [0, 1] for all coordinates of the RGB and CIELUV

colour models, [0, 1] for the radius and height coordinates of the HSV and HLS colour models, and $(-\infty, +\infty)$ for the angle coordinate of the HSV, and HLS colour models. Structure elements may contain colour coordinates that are not within the defined range for the specified colour model. During traversal out of range colour coordinates are treated in a workstation dependent way.

If a non-supported colour type is encountered during traversal, colour type INDIRECT and a colour value of 1 is used. This is equivalent to the default colour behaviour specified in ISO/IEC 9592-1. If an undefined colour index is encountered during traversal a colour index of 1 is used.

To support the specification of general colour, the PHIGS traversal state list fields 'current XXX colour index', (where XXX is polyline, polymarker, text, interior, or edge) are replaced by the corresponding 'current XXX colour' fields in the traversal state list. For each of the "set XXX colour index" structure elements defined in ISO/IEC 9592-1, PHIGS PLUS defines a corresponding structure element, "set XXX colour". The ISO/IEC 9592-1 "set XXX colour index" structure elements are defined by PHIGS PLUS to set the 'current XXX colour' field in the traversal state list to colour type INDIRECT and the colour value to the specified colour index.

Each of the 'current XXX colour index ASF' fields in the traversal state list defined in ISO/IEC 9592-1 are replaced in PHIGS PLUS with the fields 'current XXX colour ASF'. The 'current XXX colour ASF' selects between bundled and individual colour values. If the 'current XXX colour ASF' is INDIVIDUAL the 'current XXX colour' is used. If the 'current XXX colour ASF' is BUNDLED the colour is taken from the entry in the extended XXX bundle table indicated by the 'current XXX index' (see figure 9). The ISO/IEC 9592-1 "set XXX colour index ASF" structure elements are defined by PHIGS PLUS to set the 'current XXX colour ASF' field in the traversal state list to the value specified in the structure element.

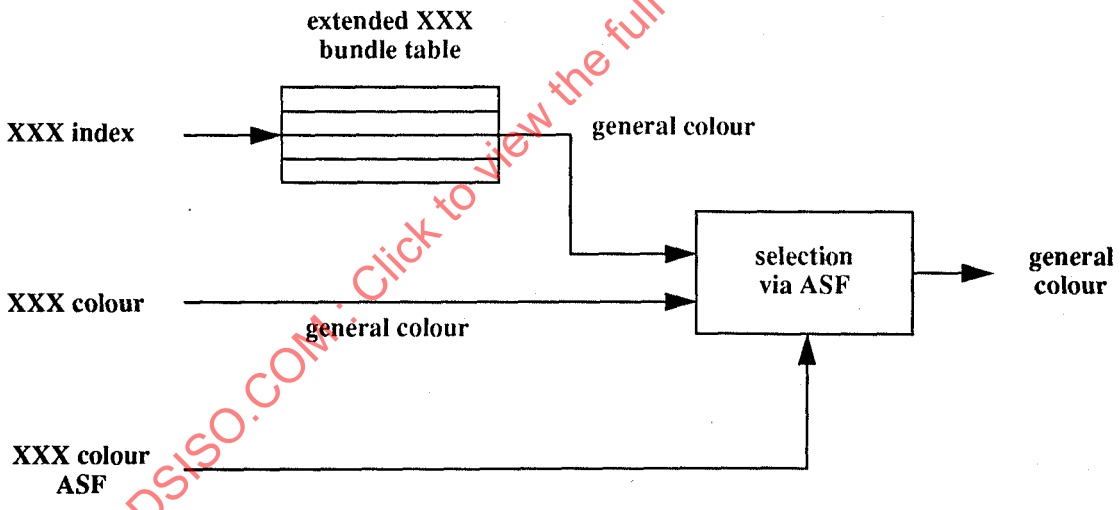


Figure 9 — Selection of the XXX COLOUR attribute

4.5.2 Extended workstation state and description tables

The 'colour index' field in the polyline, polymarker, text, interior, and edge bundles is replaced with a 'colour' field that enables these bundles to support general colour specification. The functions SET POLYLINE REPRESENTATION PLUS, SET POLYMARKER REPRESENTATION PLUS, SET TEXT REPRESENTATION PLUS, SET INTERIOR REPRESENTATION PLUS, and SET EDGE REPRESENTATION PLUS set these modified bundles. The functions, INQUIRE <*bundle type*> REPRESENTATION PLUS and INQUIRE PREDEFINED <*bundle type*> REPRESENTATION PLUS are the corresponding inquiry functions for these modified bundles. The ISO/IEC 9592-1 functions SET <POLYLINE | POLYMARKER | TEXT | INTERIOR | EDGE> REPRESENTATION are here defined to set the colour type of the bundle's 'colour' to INDIRECT and the value to the specified colour index.

The polyline bundle is extended to support bundled specification of the PHIGS PLUS polyline aspects. With the exception of the 'colour index' field, the extended polyline bundle contains both the fields specified in ISO/IEC 9592-1 and the additional fields specified in this part of ISO/IEC 9592. The ISO/IEC 9592-1 function SET POLYLINE REPRESENTATION sets only those fields specified in ISO/IEC 9592-1 and the 'colour' field. The additional fields are left unchanged.

The interior bundle is extended to support bundled specification of the PHIGS PLUS interior aspects. With the exception of the 'colour index' field, the extended interior bundle contains both the fields specified in ISO/IEC 9592-1 and the interior shading method field specified in this part of ISO/IEC 9592. The ISO/IEC 9592-1 function, SET INTERIOR REPRESENTATION, sets only those fields specified in ISO/IEC 9592-1 and the 'colour' field. The interior shading method field is left unchanged.

Pattern table entries are extended to support general specification of colour. The functions SET PATTERN REPRESENTATION PLUS, INQUIRE PATTERN REPRESENTATION PLUS, and INQUIRE PREDEFINED PATTERN REPRESENTATION PLUS are defined. The extended pattern table entry contains a 'colour type' field and an array of colour values. These replace the array of colour indices specified in ISO/IEC 9592-1. The ISO/IEC 9592-1 function SET PATTERN REPRESENTATION sets the 'colour type' to INDIRECT and the colour values to the specified colour indices.

Since the ISO/IEC 9592-1 inquiry functions cannot return a general colour specification, a new error is defined to be returned from INQUIRE POLYLINE REPRESENTATION, INQUIRE POLYMARKER REPRESENTATION, INQUIRE TEXT REPRESENTATION, INQUIRE INTERIOR REPRESENTATION, INQUIRE EDGE REPRESENTATION, and INQUIRE PATTERN REPRESENTATION when the requested entry contains a general colour specification and the colour type is not INDIRECT. The error is:

134 Ignoring function, the requested entry contains a general colour specification with colour type other than INDIRECT

Predefined entries shall contain colours of type INDIRECT.

Two bundle tables are defined to support bundled specification of reflectance aspects and parametric surface aspects. These tables are the reflectance bundle table and the parametric surface bundle table. The REFLECTANCE INDEX and BACK REFLECTANCE INDEX attributes indicate the entries in the reflectance bundle table to use when resolving bundled reflectance aspects for front and back facing facets, respectively, of area primitives. The PARAMETRIC SURFACE INDEX attribute indicates the bundle entry in the parametric surface bundle table to use when resolving bundled parametric surface attributes. Aspect source flags are defined for the aspects in these bundles.

The 'dynamic modification accepted for' entries in the workstation description table for the polyline bundle table, polymarker bundle table, text bundle table, interior bundle table, edge bundle table and pattern table apply to the tables as extended. Additional 'dynamic modification accepted for' entries in the workstation description table are defined for the reflectance bundle table, the parametric surface bundle table, and the data mapping, light source, depth cueing, and colour mapping tables. These entries are returned by the function INQUIRE DYNAMICS OF WORKSTATION ATTRIBUTES PLUS.

The enumeration values for the list of attributes used by generalized drawing primitives are extended to include the parametric surface and reflectance tables.

4.5.3 PHIGS PLUS attributes applied to PHIGS output primitives

4.5.3.1 General

Output primitive attributes defined in this part of PHIGS apply to the output primitives defined in ISO/IEC 9592-1 as listed in table 4.

Table 4 — PHIGS PLUS attributes applied to ISO/IEC 9592-1 primitives

POLYLINE	
POLYLINE COLOUR	POLYLINE COLOUR ASF
RENDERING COLOUR MODEL	
DEPTH CUE INDEX	
COLOUR MAPPING INDEX	
POLYMARKER	
POLYMARKER COLOUR	POLYMARKER COLOUR ASF
RENDERING COLOUR MODEL	
DEPTH CUE INDEX	
COLOUR MAPPING INDEX	
TEXT	
TEXT COLOUR	TEXT COLOUR ASF
RENDERING COLOUR MODEL	
DEPTH CUE INDEX	
COLOUR MAPPING INDEX	
ANNOTATION TEXT RELATIVE	
TEXT COLOUR	TEXT COLOUR ASF
RENDERING COLOUR MODEL	
DEPTH CUE INDEX	
COLOUR MAPPING INDEX	
FILL AREA and FILL AREA SET	
REFLECTANCE INDEX	
BACK INTERIOR INDEX	
BACK REFLECTANCE INDEX	
FACET DISTINGUISHING MODE	
FACET CULLING MODE	
INTERIOR COLOUR	INTERIOR COLOUR ASF
INTERIOR SHADING METHOD	INTERIOR SHADING METHOD ASF
REFLECTANCE PROPERTIES	REFLECTANCE PROPERTIES ASF
REFLECTANCE MODEL	REFLECTANCE MODEL ASF
BACK INTERIOR STYLE	BACK INTERIOR STYLE ASF
BACK INTERIOR STYLE INDEX	BACK INTERIOR STYLE INDEX ASF
BACK INTERIOR COLOUR	BACK INTERIOR COLOUR ASF
BACK INTERIOR SHADING METHOD	BACK INTERIOR SHADING METHOD ASF
BACK REFLECTANCE PROPERTIES	BACK REFLECTANCE PROPERTIES ASF
BACK REFLECTANCE MODEL	BACK REFLECTANCE MODEL ASF
LIGHT SOURCE STATE	
RENDERING COLOUR MODEL	
DEPTH CUE INDEX	
COLOUR MAPPING INDEX	

Table 4 — PHIGS PLUS attributes applied to ISO/IEC 9592-1 primitives (Continued)

CELL ARRAY	
REFLECTANCE INDEX	
BACK REFLECTANCE INDEX	
FACET DISTINGUISHING MODE	
FACET CULLING MODE	
REFLECTANCE PROPERTIES	REFLECTANCE PROPERTIES ASF
REFLECTANCE MODEL	REFLECTANCE MODEL ASF
BACK REFLECTANCE PROPERTIES	BACK REFLECTANCE PROPERTIES ASF
BACK REFLECTANCE MODEL	BACK REFLECTANCE MODEL ASF
LIGHT SOURCE STATE	
RENDERING COLOUR MODEL	
DEPTH CUE INDEX	
COLOUR MAPPING INDEX	
GENERALIZED DRAWING PRIMITIVE	
Zero or more of the attribute sets listed above.	

4.5.3.2 Polyline attributes

PHIGS PLUS defines the following additional aspects and attributes for polylines: the non-geometric aspect polyline colour, which is controlled by the POLYLINE INDEX or the individually specified attribute POLYLINE COLOUR; the individually specified attributes DEPTH CUE INDEX and COLOUR MAPPING INDEX. The source of the polyline colour is selected by the POLYLINE COLOUR ASF.

4.5.3.3 Polymarker attributes

PHIGS PLUS defines the following additional aspects and attributes for polymarkers: the non-geometric aspect polymarker colour; the individually specified attributes DEPTH CUE INDEX and COLOUR MAPPING INDEX. The polymarker colour aspect is controlled by the POLYMARKER INDEX or the individually specified attribute POLYMARKER COLOUR. The source of the polymarker colour is selected by the POLYMARKER COLOUR ASF.

4.5.3.4 Text attributes

PHIGS PLUS defines the following additional aspects and attributes for text: the non-geometric aspect text colour; the individually specified attributes DEPTH CUE INDEX and COLOUR MAPPING INDEX. The text colour aspect is controlled by the TEXT INDEX or the individually specified attribute TEXT COLOUR. The source of the text colour is selected by the TEXT COLOUR ASF.

4.5.3.5 Annotation text attributes

PHIGS PLUS defines the following additional aspects and attributes for annotation text: the non-geometric aspect text colour; the individually specified attributes DEPTH CUE INDEX and COLOUR MAPPING INDEX. The text colour aspect is controlled by the TEXT INDEX or the individually specified attribute TEXT COLOUR. The source of the text colour is selected by the TEXT COLOUR ASF.

4.5.3.6 Fill area attributes

PHIGS PLUS defines the following additional aspects and attributes for fill areas: the non-geometric aspects interior colour, interior shading method, reflectance properties, reflectance model, back interior style, back interior style index, back interior colour, back interior shading method, back reflectance properties, and back reflectance model; the individually specified attributes LIGHT SOURCE STATE, FACET DISTINGUISHING MODE, FACET CULLING MODE, RENDERING COLOUR MODEL, DEPTH CUE INDEX and COLOUR MAPPING INDEX. Aspects in the first group are controlled by the INTERIOR INDEX, REFLECTANCE INDEX, BACK INTERIOR INDEX, BACK REFLECTANCE INDEX, or the individually specified attributes INTERIOR COLOUR, INTERIOR SHADING METHOD, REFLECTANCE PROPERTIES, REFLECTANCE MODEL, BACK INTERIOR STYLE, BACK INTERIOR STYLE INDEX, BACK INTERIOR COLOUR, BACK INTERIOR SHADING METHOD, BACK REFLECTANCE PROPERTIES, and BACK REFLECTANCE MODEL. The source of the attributes for these aspects is selected by the corresponding aspect source flag attribute listed in table 4.

4.5.3.7 Fill area set attributes

Fill area sets are affected by the same PHIGS PLUS attributes that affect fill areas, plus the non-geometric aspects edge flag, edgetype, edgewidth scale factor, and edge colour, which are controlled by the EDGE INDEX or by the individually specified attributes EDGE FLAG, EDGETYPE, EDGEWIDTH SCALE FACTOR, and EDGE COLOUR. The source of these aspects is selected by the corresponding aspect source flag attribute. Fill area set edges are not affected by lighting or shading.

4.5.3.8 Cell array attributes

PHIGS PLUS defines the following additional aspects and attributes for cell arrays: the non-geometric aspects reflectance properties, reflectance model, back reflectance properties and back reflectance model; the individually specified attributes LIGHT SOURCE STATE, FACET DISTINGUISHING MODE, FACET CULLING MODE, RENDERING COLOUR MODEL, DEPTH CUE INDEX and COLOUR MAPPING INDEX. Aspects in the first group are controlled by the REFLECTANCE INDEX, BACK REFLECTANCE INDEX, or the individually specified attributes REFLECTANCE PROPERTIES, REFLECTANCE MODEL, BACK REFLECTANCE PROPERTIES and BACK REFLECTANCE MODEL. The source of the attributes for these aspects is selected by the corresponding aspect source flag attribute listed above. The interior colour, interior style, interior shading method, back interior style, back interior colour, and back interior shading method aspects and their corresponding aspect source flags do not apply to cell arrays.

4.5.3.9 Generalized drawing primitive attributes

Any of the PHIGS PLUS attributes listed in 4.5.4 may also apply to Generalized Drawing Primitives (GDPs). Whether bundle indices or associated individually specified attributes are used depends upon the values of the appropriate aspect source flags. The sets of attributes most appropriate for the specified GDP structure element are selected for the GDP as part of the definition of the GDP and are recorded in the workstation description table.

4.5.4 Attributes applied to PHIGS PLUS output primitives

4.5.4.1 General

The attributes defined in ISO/IEC 9592-1 and the attributes defined in this part of ISO/IEC 9592 that apply to each PHIGS PLUS output primitive are listed in table 5.

4.5.4.2 Polyline set with colour attributes

Polyline set with colour output primitives are affected by the same attributes as polylines. They are also affected by the RENDERING COLOUR MODEL and the polyline shading method aspect, which is controlled by the POLYLINE INDEX or the individually specified attribute POLYLINE SHADING METHOD. The source of the polyline shading method is selected by the POLYLINE COLOUR ASF. Each polyline of a set is treated as an individual polyline when applying the attributes.

4.5.4.3 Fill area set with data attributes

Fill area set with data output primitives are affected by the same attributes as fill area sets. Their display is also affected by the data mapping method and back data mapping method aspects.

4.5.4.4 Cell array PLUS attributes

Cell array PLUS output primitives are affected by the same attributes that affect cell arrays.

4.5.4.5 Set of fill area set with data attributes

Set of fill area set with data output primitives are affected by the same attributes that affect fill area set with data output primitives. Each fill area set of the set is rendered as a single fill area set with data. Interior styles HATCH and PATTERN apply to each fill area set in the set as if it were an individual fill area set.

4.5.4.6 Triangle set with data attributes

Triangle sets are affected by the same attributes that affect fill area set with data output primitives. Interior styles HATCH and PATTERN apply to the triangle set in a workstation dependent manner. For interior style HOLLOW the bounding polyline is drawn along boundaries created by clipping.

4.5.4.7 Triangle strip with data attributes

Triangle strips are affected by the same attributes that affect fill area set with data output primitives. Interior styles HATCH and PATTERN apply to the triangle strip in a workstation dependent manner. For interior style HOLLOW the bounding polyline is drawn along boundaries created by clipping.

4.5.4.8 Quadrilateral mesh with data attributes

Quadrilateral meshes are affected by the same attributes that affect fill area set with data output primitives. Interior styles HATCH and PATTERN apply to the quadrilateral mesh in a workstation dependent manner. For interior style HOLLOW the bounding polyline is drawn along boundaries created by clipping.

Table 5 — Attributes applied to PHIGS PLUS output primitives

POLYLINE SET WITH COLOUR	
POLYLINE INDEX	
LINETYPE	LINETYPE ASF
LINEWIDTH SCALE FACTOR	LINEWIDTH SCALE FACTOR ASF
POLYLINE COLOUR INDEX	POLYLINE COLOUR INDEX ASF
VIEW INDEX	
HLHSR IDENTIFIER	
PICK IDENTIFIER	
NAME SET	
POLYLINE COLOUR	POLYLINE COLOUR ASF
POLYLINE SHADING METHOD	POLYLINE SHADING METHOD ASF
RENDERING COLOUR MODEL	
DEPTH CUE INDEX	
COLOUR MAPPING INDEX	
FILL AREA SET WITH DATA **SET OF FILL AREA SETS WITH DATA** **TRIANGLE SET WITH DATA** **TRIANGLE STRIP WITH DATA** **QUADRILATERAL MESH WITH DATA**	
INTERIOR INDEX	
INTERIOR STYLE	INTERIOR STYLE ASF
INTERIOR STYLE INDEX	INTERIOR STYLE INDEX ASF
INTERIOR COLOUR INDEX	INTERIOR COLOUR INDEX ASF
EDGE INDEX	
EDGE FLAG	EDGE FLAG ASF
EDGETYPE	EDGETYPE ASF
EDGEWIDTH SCALE FACTOR	EDGEWIDTH SCALE FACTOR ASF
PATTERN SIZE	
PATTERN REFERENCE POINT	
PATTERN VECTORS	
VIEW INDEX	
HLHSR IDENTIFIER	
PICK IDENTIFIER	
NAME SET	
DATA MAPPING INDEX	
REFLECTANCE INDEX	
BACK INTERIOR INDEX	
BACK DATA MAPPING INDEX	

Table 5 — Attributes applied to PHIGS PLUS output primitives (Continued)

BACK REFLECTANCE INDEX	
FACET DISTINGUISHING MODE	
FACET CULLING MODE	
INTERIOR COLOUR	INTERIOR COLOUR ASF
INTERIOR SHADING METHOD	INTERIOR SHADING METHOD ASF
DATA MAPPING METHOD	DATA MAPPING METHOD ASF
REFLECTANCE PROPERTIES	REFLECTANCE PROPERTIES ASF
REFLECTANCE MODEL	REFLECTANCE MODEL ASF
BACK INTERIOR STYLE	BACK INTERIOR STYLE ASF
BACK INTERIOR STYLE INDEX	BACK INTERIOR STYLE INDEX ASF
BACK INTERIOR COLOUR	BACK INTERIOR COLOUR ASF
BACK INTERIOR SHADING METHOD	BACK INTERIOR SHADING METHOD ASF
BACK DATA MAPPING METHOD	BACK DATA MAPPING METHOD ASF
BACK REFLECTANCE PROPERTIES	BACK REFLECTANCE PROPERTIES ASF
BACK REFLECTANCE MODEL	BACK REFLECTANCE MODEL ASF
EDGE COLOUR	EDGE COLOUR ASF
LIGHT SOURCE STATE	
RENDERING COLOUR MODEL	
DEPTH CUE INDEX	
COLOUR MAPPING INDEX	
CELL ARRAY PLUS	
VIEW INDEX	
HLHSR IDENTIFIER	
PICK IDENTIFIER	
NAME SET	
REFLECTANCE INDEX	
BACK REFLECTANCE INDEX	
FACET DISTINGUISHING MODE	
FACET CULLING MODE	
REFLECTANCE PROPERTIES	REFLECTANCE PROPERTIES ASF
REFLECTANCE MODEL	REFLECTANCE MODEL ASF
BACK REFLECTANCE PROPERTIES	BACK REFLECTANCE PROPERTIES ASF
BACK REFLECTANCE MODEL	BACK REFLECTANCE MODEL ASF
LIGHT SOURCE STATE	
RENDERING COLOUR MODEL	
DEPTH CUE INDEX	
COLOUR MAPPING INDEX	
NON-UNIFORM B-SPLINE CURVE	
POLYLINE INDEX	

Table 5 — Attributes applied to PHIGS PLUS output primitives (Continued)

LINETYPE	LINETYPE ASF
LINEWIDTH SCALE FACTOR	LINEWIDTH SCALE FACTOR ASF
POLYLINE COLOUR INDEX	POLYLINE COLOUR INDEX ASF
VIEW INDEX	
HLHSR IDENTIFIER	
PICK IDENTIFIER	
NAME SET	
POLYLINE COLOUR	POLYLINE COLOUR ASF
CURVE APPROXIMATION CRITERIA	CURVE APPROXIMATION CRITERIA ASF
DEPTH CUE INDEX	
COLOUR MAPPING INDEX	
NON-UNIFORM B-SPLINE CURVE WITH COLOUR	
All those for Non-uniform B-spline curve, plus:	
POLYLINE SHADING METHOD	POLYLINE SHADING METHOD ASF
RENDERING COLOUR MODEL	
NON-UNIFORM B-SPLINE SURFACE	
INTERIOR INDEX	
INTERIOR STYLE	INTERIOR STYLE ASF
INTERIOR STYLE INDEX	INTERIOR STYLE INDEX ASF
INTERIOR COLOUR INDEX	INTERIOR COLOUR INDEX ASF
EDGE INDEX	
EDGE FLAG	EDGE FLAG ASF
EDGETYPE	EDGETYPE ASF
EDGEWIDTH SCALE FACTOR	EDGEWIDTH SCALE FACTOR ASF
PATTERN SIZE	
PATTERN REFERENCE POINT	
PATTERN VECTORS	
VIEW INDEX	
HLHSR IDENTIFIER	
PICK IDENTIFIER	
NAME SET	
REFLECTANCE INDEX	
PARAMETRIC SURFACE INDEX	
FACET DISTINGUISHING MODE	
FACET CULLING MODE	
INTERIOR COLOUR	INTERIOR COLOUR ASF
INTERIOR SHADING METHOD	INTERIOR SHADING METHOD ASF
REFLECTANCE PROPERTIES	REFLECTANCE PROPERTIES ASF
REFLECTANCE MODEL	REFLECTANCE MODEL ASF

Table 5 — Attributes applied to PHIGS PLUS output primitives (Continued)

BACK INTERIOR INDEX	
BACK REFLECTANCE INDEX	
BACK INTERIOR STYLE	BACK INTERIOR STYLE ASF
BACK INTERIOR STYLE INDEX	BACK INTERIOR STYLE INDEX ASF
BACK INTERIOR COLOUR	BACK INTERIOR COLOUR ASF
BACK INTERIOR SHADING METHOD	BACK INTERIOR SHADING METHOD ASF
BACK REFLECTANCE PROPERTIES	BACK REFLECTANCE PROPERTIES ASF
BACK REFLECTANCE MODEL	BACK REFLECTANCE MODEL ASF
EDGE COLOUR	EDGE COLOUR ASF
LIGHT SOURCE STATE	
SURFACE APPROXIMATION CRITERIA	SURFACE APPROXIMATION CRITERIA ASF
PARAMETRIC SURFACE CHARACTERISTICS	PARAMETRIC SURFACE CHARACTERISTICS ASF
RENDERING COLOUR MODEL	
DEPTH CUE INDEX	
COLOUR MAPPING INDEX	
NON-UNIFORM B-SPLINE SURFACE WITH DATA	
All those for NON-UNIFORM B-SPLINE SURFACE, plus:	
DATA MAPPING INDEX	
DATA MAPPING METHOD	DATA MAPPING METHOD ASF
BACK DATA MAPPING INDEX	
BACK DATA MAPPING METHOD	BACK DATA MAPPING METHOD ASF

4.5.4.9 Non-uniform B-spline curve attributes

Non-uniform B-spline curve output primitives are affected by the same attributes as polylines. They are also affected by the curve approximation criteria aspect. The curve approximation criteria aspect is controlled by the POLYLINE INDEX or the individually specified attribute CURVE APPROXIMATION CRITERIA. The source of the curve approximation criteria is selected by the CURVE APPROXIMATION CRITERIA ASF.

The curve approximation criteria indicates the minimum quality desired for the approximation of the curve's geometry. It provides an indication of the degree of approximation tolerable and the relative importance of image quality versus rendering speed. Specification of lower relative approximation quality is intended to produce relatively lower quality approximations at relatively higher speeds. Specification of higher relative approximation quality is intended to produce higher quality approximations at possibly lower speeds. Implementations may produce a higher quality approximation than that requested. Approximation of the curve shall not introduce any gaps, holes or other visible spatial discontinuities.

An approximation criteria consists of a type and a data record. The data record contains information specific to the type and is generally unique for each type. The defined curve approximation criteria types are

≤ 0 Implementation dependent.

1 WORKSTATION DEPENDENT: A workstation dependent approximation criteria is used. There are no data record contents for this type.

2 CONSTANT PARAMETRIC SUBDIVISION BETWEEN KNOTS: Sample points are at equal parameter increments between each pair of the curve's knots. The data record contains a single integer value that indicates the minimum number of sample points between each pair of knots. The value is interpreted as follows:

≤ 0: Sample points are at the parameter range limits and at the knots within the parameter range limits.

> 0: Sample points are at the parameter range limits and at the knots within the parameter range limits. In addition, sample points are at this number of positions between each pair of knots, but only if the position is within the parameter range limits.

3 CHORDAL SIZE in WC: Generate at least sufficient sample points such that the length in world coordinates of every line segment (chord) between the sample points is less than a specified real approximation value. The data record contains the single approximation value.

4 CHORDAL SIZE in NPC: Generate at least sufficient sample points such that the length in normalized projection coordinates of every line segment (chord) between the sample points is less than a specified real approximation value. The data record contains the single approximation value.

5 CHORDAL SIZE in DC: Generate at least sufficient sample points such that the length in device coordinates of every line segment (chord) between the sample points is less than a specified real approximation value. The data record contains the single approximation value.

6 CHORDAL DEVIATION in WC: Generate at least sufficient sample points such that the absolute value of the maximum deviation measured in world coordinates between the curve and line segments (chord) between sample points is less than a specified real approximation value. The data record contains the single approximation value.

7 CHORDAL DEVIATION in NPC: Generate at least sufficient sample points such that the absolute value of the maximum deviation measured in normalized projection coordinates between the curve and line segments (chord) between sample points is less than a specified real approximation value. The data record contains the single approximation value.

8 CHORDAL DEVIATION in DC: Generate at least sufficient sample points such that the absolute value of the maximum deviation measured in device coordinates between the curve and line segments (chord) between sample points is less than a specified real approximation value. The data record contains the single approximation value.

9 RELATIVE in WC: A single real approximation value between 0 and 1 indicates a relative quality of rendering to be maintained independent of scaling in world coordinates. Higher values specify a better relative quality. The data record contains the approximation value.

10 RELATIVE in NPC: A single real approximation value between 0 and 1 indicates a relative quality of rendering to be maintained independent of scaling in normalized projection coordinates. Higher values specify a better relative quality. The data record contains the approximation value.

11 RELATIVE in DC: A single real approximation value between 0 and 1 indicates a relative quality of rendering to be maintained independent of scaling in device coordinates. Higher values specify a better relative quality. The data record contains the approximation value.

≥ 12 Reserved for registration or future standardisation.

It is not intended that there be any correspondence between the approximations produced by different approximation criteria types. For example, the set of all possible approximations achievable by CHORDAL SIZE in WC can be disjoint from the set of all possible approximations achievable by RELATIVE in WC.

4.5.4.10 Non-uniform B-spline curve with colour attributes

Non-uniform B-spline curve with colour primitives are affected by the same attributes that affect non-uniform B-spline curve primitives. In addition, they are affected by the RENDERING COLOUR MODEL and the polyline shading method

aspect, which is controlled by the POLYLINE INDEX or the individually specified attribute POLYLINE SHADING METHOD. The source of the polyline shading method is selected by the POLYLINE COLOUR ASF.

4.5.4.11 Non-uniform B-spline surface attributes

With the exception of the data mapping method and back data mapping method aspects, non-uniform B-spline surfaces are affected by the same attributes that affect fill area set with data output primitives. In addition, they have the non-geometric aspects surface approximation criteria and parametric surface characteristics. These aspects are controlled by the PARAMETRIC SURFACE INDEX or the individually specified attributes SURFACE APPROXIMATION CRITERIA and PARAMETRIC SURFACE CHARACTERISTICS. The source of the surface approximation criteria is selected by the SURFACE APPROXIMATION CRITERIA ASF. The source of the parametric surface characteristics is selected by the PARAMETRIC SURFACE CHARACTERISTICS ASF.

Non-uniform B-spline surfaces can be approximated when displayed. Conceptually, the result of this approximation is a collection of facets. The surface approximation criteria indicates the minimum quality desired for the approximation of the surface's geometry. It provides an indication of the degree of approximation tolerable and the relative importance of image quality versus rendering speed. Specification of lower relative approximation quality is intended to produce relatively lower quality approximations at relatively higher speeds. Specification of higher relative approximation quality is intended to produce higher quality approximations at possibly lower speeds. Implementations may produce a higher quality approximation than that requested. Approximation of the surface shall not introduce any gaps, holes or other visible spatial discontinuities.

A surface approximation criteria consists of a type and a data record. The data record contains information specific to the type and is generally unique for each type. The defined surface approximation criteria types are

≤ 0 Implementation dependent.

1 WORKSTATION DEPENDENT: A workstation dependent approximation criteria is used. There are no data record contents for this type.

2 CONSTANT PARAMETRIC SUBDIVISION BETWEEN KNOTS: Sample points are at equal parameter increments between each pair of *u* knots and *v* knots. The data record contains two integer values that indicate the minimum number of sample points in the *u* and *v* parameter dimensions. Both values have the following interpretation:

≤ 0: Sample points are at the *u* and *v* knot values, and where lines of constant parameter across the surface at these knot values would intersect the approximated trimming loops.

> 0: Sample points are at the *u* and *v* knot values, at this number of positions between each pair of knot values in each parameter dimension, and where lines of constant parameter across the surface at these parameter values would intersect the approximated trimming loops.

3 CHORDAL SIZE in WC: Generate at least sufficient sample points such that the length in world coordinates of every line segment (chord) between the sample points for both the *u* and *v* parameter directions is less than the corresponding specified real approximation value. The data record contains two approximation values, one for each parameter direction.

4 CHORDAL SIZE in NPC: Generate at least sufficient sample points such that the length in normalized projection coordinates of every line segment (chord) between the sample points for both the *u* and *v* parameter directions is less than the corresponding specified real approximation value. The data record contains two approximation values, one for each parameter direction.

5 CHORDAL SIZE in DC: Generate at least sufficient sample points such that the length in device coordinates of every line segment (chord) between the sample points for both the *u* and *v* parameter directions is less than the corresponding specified real approximation value. The data record contains two approximation values, one for each parameter direction.

6 PLANAR DEVIATION in WC: Generate at least sufficient sample points such that the absolute value of the maximum deviation measured in world coordinates between the surface and the approximating facet is less than a specified real approximation value. The data record contains the approximation value.

7 PLANAR DEVIATION in NPC: Generate at least sufficient sample points such that the absolute value of the maximum deviation measured in normalized projection coordinates between the surface and the approximating facet is less than a specified real approximation value. The data record contains the approximation value.

8 PLANAR DEVIATION in DC: Generate at least sufficient sample points such that the absolute value of the maximum deviation measured in device coordinates between the surface and the approximating facet is less than a specified real approximation value. The data record contains the approximation value.

9 RELATIVE in WC: A single real approximation value between 0 and 1 indicates a relative quality of rendering to be maintained independent of scaling in world coordinates. Higher values specify a better relative quality. The data record contains the approximation value.

10 RELATIVE in NPC: A single real approximation value between 0 and 1 indicates a relative quality of rendering to be maintained independent of scaling in normalized projection coordinates. Higher values specify a better relative quality. The data record contains the approximation value.

11 RELATIVE in DC: A single real approximation value between 0 and 1 indicates a relative quality of rendering to be maintained independent of scaling in device coordinates. Higher values specify a better relative quality. The data record contains the approximation value.

≥ 12 Reserved for registration or future standardisation.

It is not intended that there be any correspondence between the approximations produced by different approximation criteria types. For example, the set of all possible approximations achievable by CHORDAL SIZE in WC can be disjoint from the set of all possible approximations achievable by RELATIVE in WC.

The parametric surface characteristics aspect further specifies the appearance of parametric surfaces beyond those aspects also associated with fill area sets. The aspect consists of a type and a data record. The data record contains information specific to the type and is generally unique for each type. The definition of each type indicates whether or not the application of that type to the surface distinguishes between front and back facing portions when the 'current facet distinguishing mode' is ON. If it does then the associated data record contains separate parameters for front and back facing portions. If it does not then only one set of parameters is defined and they apply to the surface as though the 'current facet distinguishing mode' were OFF.

The defined parametric surface characteristics types are

≤ 0 Implementation dependent.

1 NONE: No surface characteristics beyond those aspects defined for fill area sets are used. The data record is empty for this type.

2 WORKSTATION DEPENDENT: A workstation dependent representation that displays the shape of the surface is drawn. This method does not distinguish between front and back facing portions of the surface. The appearance of the representation is controlled by the appropriate set of output primitive attributes for the representation. It is workstation dependent how the representation interacts with any interior rendering indicated by the interior aspects. There are no data record contents for this type.

3 ISOPARAMETRIC CURVES: Isoparametric curves are drawn on the surface. The data record contains the number of curves to draw in each of the parameter dimensions and their placement. If the placement is UNIFORM OVER SURFACE the specified number of curves are evenly spaced within the parameter range of the surface; curves are also drawn at the limits of the parameter range. If the placement is UNIFORM BETWEEN KNOTS the specified number of curves are evenly spaced between each pair of knots; curves are also drawn at the knots. In both cases only the portions of isoparametric curves are drawn that are within the interior of the surface as defined by any trimming loops.

This method does not distinguish between front and back facing portions of the surface. The appearance of the isoparametric curves is controlled by the current polyline colour, linewidth scale factor, and linetype aspects. The approximation of the isoparametric curves is controlled by the surface approximation criteria. The isoparametric curves shall lie on the approximated surface. The isoparametric curves are drawn in addition to any interior rendering indicated by the interior style or back interior style. Isoparametric curves have higher visual priority than the primitive's filled or hollow interiors, but lower priority than the primitive's edges.

4 LEVEL CURVES in MC: Level curves are drawn on the surface. The curves correspond to the intersections of the surface and a finite set of planes perpendicular to a modelling coordinate direction vector. The positions of the planes are specified by a sequence of intersection points along an infinite line defined by a modelling coordinate origin point, P_0, and a direction vector, $\vec{V}$. as follows:

$$P_i = P_0 + t_i\vec{V}$$

The t_i are a sequence of parameters specifying the intersection points. They are in the range $-\infty < t_i < +\infty$. The P_i are the intersection points of the perpendicular planes with the infinite line. P_0 is a specified origin point in modelling coordinates, and $\vec{V}$ is the specified direction vector in modelling coordinates. The *i-th* plane is perpendicular to the direction vector, $\vec{V}$, and intersects the infinite line at point P_i. The data record consists of the origin point, P_0; the direction vector, $\vec{V}$; and the ordered list of parameters, t_i. Only the portions of the level curves are drawn that are within the interior of the surface as defined by any trimming loops.

This method does not distinguish between front and back facing portions of the surface. The appearance of the level curves is controlled by the current polyline colour, linewidth scale factor, and linetype aspects. The approximation of the level curves is controlled by the surface approximation criteria. The level curves shall lie on the approximated surface. The curves are drawn in addition to any interior rendering indicated by the interior style or back interior style. Level curves have higher visual priority than the primitive's filled or hollow interiors, but lower priority than the primitive's edges.

5 LEVEL CURVES in WC: Level curves are drawn on the surface. The curves correspond to the intersections of the surface and a finite set of planes perpendicular to a world coordinate direction vector. The positions of the planes are specified by a sequence of intersection points along an infinite line defined by a world coordinate origin point, P_0, and a direction vector, $\vec{V}$, as follows:

The t_i are a sequence of parameters specifying the intersection points. They are in the range $-\infty < t_i < +\infty$. The P_i are the intersection points of the perpendicular planes with the infinite line. P_0 is a specified origin point in world coordinates, and $\vec{V}$ is the specified direction vector in world coordinates. The *i-th* plane is perpendicular to the direction vector, $\vec{V}$, and intersects the infinite line at point P_i. The data record consists of the origin point, P_0; the direction vector, $\vec{V}$; and the ordered list of parameters, t_i. Only the portions of the level curves are drawn that are within the interior of the surface as defined by any trimming loops.

This method does not distinguish between front and back facing portions of the surface. The appearance of the level curves is controlled by the current polyline colour, linewidth scale factor, and linetype aspects. The approximation of the level curves is controlled by the surface approximation criteria. The level curves shall lie on the approximated surface. The curves are drawn in addition to any interior rendering indicated by the interior style or back interior style. Level curves have higher visual priority than the primitive's filled or hollow interiors, but lower priority than the primitive's edges.

≥ 6 Reserved for registration or future standardisation.

The effect of PATTERN and HATCH interior styles on parametric surfaces is workstation dependent.

Lighting and shading shall be applied to parametric surfaces. The interior shading method shall be applied to each facet.

Trimming curves, when specified, define the edges of a surface. If no trimming curves are specified, the edges consist of curves of constant parameter value at the parameter range boundaries of the surface. The edges lie on the approximated surface and are displayed using the edge attributes. The approximation of trimming curves is controlled by the trimming curve approximation criteria specified with each trimming curve. The trimming curve approximation criteria are analo-

gous to the curve approximation criteria, but the approximation types are restricted to the defined types 1 and 2 or appropriate implementation dependent types. The trimming curve definition contains an edge visibility flag. In conjunction with the EDGE FLAG the edge visibility flag controls the visibility of each trimming curve. When the EDGE FLAG is ON, the edge visibility flag determines whether the trimming curve is visible. When the EDGE FLAG is OFF, no trimming curves are displayed.

4.5.4.12 Non-uniform B-spline surface with data attributes

Non-uniform B-spline surface with data output primitives are affected by the same attributes as non-uniform B-spline surfaces. In addition, their display is affected by the data mapping method aspects.

4.5.4.13 Individual edge control for PHIGS PLUS area primitives

All area primitives defined in PHIGS PLUS except cell array PLUS can have edge visibility flags specified with them. If edge visibility flags are specified, each edge is displayed only if both the edge flag aspect and the corresponding edge visibility flag (or, where multiple facets share an edge, any of the corresponding edge visibility flags) in the output primitive structure element are set to ON. If the edge flag aspect is set to OFF, the edge visibility flags specified in the output primitive structure element are ignored. If edge visibility flags are not specified, then the edge flag aspect solely determines whether edges are displayed.

For all primitives that have edge visibility flags associated with them, the rendering of the edges in the presence of bounding polylines resulting from interior style HOLLOW has the same effect as for fill area set primitives.

4.5.4.14 Reflectance properties

Reflectance properties indicate how an area primitive models the reflection of light. All area primitives are affected by the reflectance properties aspect. Reflectance properties are specified by a type and a data record. The reflectance property types are as follows:

≤ 0 implementation dependent;

1 simple reflectance: The reflectance properties consist of an ambient reflection coefficient, diffuse reflection coefficient, specular reflection coefficient, specular colour and specular exponent. The ambient reflection coefficient controls the amount of ambient light reflected from the primitive. If the coefficient is zero, no ambient light is reflected. An increase in the coefficient increases the amount of ambient light reflected, up to a maximum of 1 for the coefficient. The diffuse reflection coefficient similarly controls the amount of non-ambient light reflected equally in all directions from a surface, resulting in a dull, matte appearance that is independent of the viewing direction. The specular reflection coefficient similarly controls the amount of non-ambient light contributing to reflections unequally from a surface, resulting in a shiny appearance.

In general there is a direction in which the specular reflected light is at a maximum, specified implicitly by a reflectance vector. The specular colour and specular exponent have significance if the specular reflection coefficient is non-zero. Whereas the intrinsic colour, which may vary across an output primitive, contributes to the ambient and diffuse reflections, the colour of the specular highlights is also contributed to by the specular colour aspect. The specular exponent controls the intensity of the specular reflection in directions other than the direction of the reflectance vector. Increasing the exponent decreases the intensity of the specular reflection away from the direction of the reflectance vector. The minimum value of the exponent is 0.

≥ 2 reserved for registration or future standardisation.

Reflectance properties is a non-geometric aspect and is both individually specified and bundled. Reflectance properties are in the entries in the reflectance bundle table in the workstation state list.

During structure traversal the 'current reflectance properties ASF' in the traversal state list determines the source of the reflectance properties. If it is BUNDLED then the reflectance properties specified in the reflectance bundle indicated by the 'current reflectance index' is used. If it is INDIVIDUAL then the 'current reflectance properties' in the traversal state

list is used. Similarly, the 'current back reflectance properties ASF' indicates the source of the back reflectance properties, either from the reflectance table bundle indicated by the 'current back reflectance properties index' or the 'current back reflectance properties' entry in the traversal state list.

During traversal, reflectance properties type 1 is used if the specified type is not available on the workstation. The data record values used in this case are the default values indicated in the PHIGS description table.

4.5.5 Implicitly specified attributes

4.5.5.1 General

Area primitives have aspects that are implicitly derived from output primitive structure element data, traversal state attributes, and workstation state tables. These implicit attributes are used in the PHIGS PLUS rendering pipeline.

4.5.5.2 Facet normal

The facet normal is used for determining the orientation of a facet of an area primitive. In some cases it is also used to determine reflectance normals. Facet normals can be supplied for the with-data area primitives. In the case of the other area primitives or where facet normals are not supplied for a with-data primitive, facet normals are computed for each facet using a method that depends on the output primitive type, as follows:

a) For a fill area or fill area set primitive a normal is computed by selecting three points: A, B, and C. Point A is the first point in the first list of vertices. Point B is the next point in that list that is not coincident with A. Point C is the next point in that list that is not colinear with A and B. The normal is the normalized cross product of the vector extending from A to B with the vector extending from A to C.

   If it is not possible to find three such points in the first list of vertices, then, in the case of fill area set, the rest of the lists are searched in order to select three appropriate points from a single list. If it is not possible to find three such points in any list, then the output primitive is degenerate, in which case the result is workstation dependent.

b) All cells of a cell array have the same normal. The normal for the three dimensional form of the cell array is the normalized cross product of the vector, $\vec{V}_1$, extending from point P to point Q with the vector, $\vec{V}_2$, extending from point P to point R (refer to ISO/IEC 9592-1 for the definition of P, Q and R):

$$\vec{N} = \frac{\vec{V}_1 \times \vec{V}_2}{|\vec{V}_1 \times \vec{V}_2|}$$

   The normal for the two dimensional form of the cell array is the normalized cross product of a vector along the side $(QX, PY) - (PX, PY)$ with the vector along the side $(PX, QY) - (PX, PY)$.

c) For a set of fill area set primitive the normal for each fill area set of the primitive is computed separately, using the procedure specified above for fill area sets.

d) For a triangle set primitive the normal for each triangle is computed as for a fill area.

e) For a triangle strip primitive the normal for each triangle is the normalized cross product of vectors along two of its sides. The vectors used are different for even numbered and odd numbered triangles. (The triangles are implicitly numbered by the ordering of the defining list of vertices. The first triangle in the strip is numbered 1.) Referring to figure 3, the normal for odd numbered triangles is defined by

$$\vec{N}_{i_{odd}} = \frac{(\vec{V}_{i+1} - \vec{V}_i) \times (\vec{V}_{i+2} - \vec{V}_i)}{|(\vec{V}_{i+1} - \vec{V}_i) \times (\vec{V}_{i+2} - \vec{V}_i)|}$$

where

$\vec{N}$ is the normal;

$\vec{P}_i$ are the vertex coordinates.

The normal for even numbered triangles is defined by

$$\vec{N}_{i_{even}} = \frac{(\vec{V}_{i+2} - \vec{V}_i) \times (\vec{V}_{i+1} - \vec{V}_i)}{\left|(\vec{V}_{i+2} - \vec{V}_i) \times (\vec{V}_{i+1} - \vec{V}_i)\right|}$$

f) For a quadrilateral mesh primitive the normal for each quadrilateral is the normalized cross product of its diagonals. Referring to figure 4, the normal is defined by

$$\vec{N}_{i,j} = \frac{(\vec{V}_{i+1,j+1} - \vec{V}_{i,j}) \times (\vec{V}_{i,j+1} - \vec{V}_{i+1,j})}{\left|(\vec{V}_{i+1,j+1} - \vec{V}_{i,j}) \times (\vec{V}_{i,j+1} - \vec{V}_{i+1,j})\right|}$$

where

$1 \le i \le M - 1$;

$1 \le j \le N - 1$;

$\vec{N}_{i,j}$ is the normal;

$\vec{V}_{i,j}$ are the vertex coordinates;

M and N are the dimensions of the two-dimensional array of vertex data ($M \times N$).

g) For a non-uniform B-spline surface primitive the normal is computed by normalizing the cross product of the tangent vector along the u direction, $\vec{T}_u$, (usually the partial derivative with respect to u), and the tangent vector along the v direction, $\vec{T}_v$, (usually the partial derivative with respect to v). The normal is computed as

$$\vec{N} = \frac{\vec{T}_u \times \vec{T}_v}{\left|\vec{T}_u \times \vec{T}_v\right|}$$

At positions on the surface where this cross product is not meaningful, such as where tangent vectors are colinear, the normal is computed in a workstation dependent way.

Notice that when the normal is computed, its sense is determined by the order of points in the lists that are part of the specification of the primitive. For fill area set, triangle set, triangle strip and quadrilateral mesh, the order of the primitive's vertices determines the sense of the computed normal. For the parametric surfaces, the sense is determined by the order of the surface's control points.

4.5.5.3 Facet orientation

PHIGS PLUS provides explicit controls to allow the appearance of an area primitive to be determined by the orientation of its facets. Individual facets can be culled from the primitive or be rendered with different attributes according to their orientation. The orientation of a facet is defined to be front facing if its facet normal has a positive or zero Z component when transformed to NPC, otherwise its orientation is back facing.

4.5.5.4 Reflectance normal

The reflectance normal vector defines the orientation of a facet at a certain point for the purpose of performing a reflectance calculation at that point. The determination of the reflectance normal is dependent on the presence of data optionally provided with an output primitive. If vertex normals are specified, then, depending on the interior shading

method, they are used or interpolated to define the reflectance normals at the reflectance calculation positions. If vertex normals are not specified then the facet normal is used.

If a facet is determined to be front facing, reflectance normals as described are used for subsequent reflectance calculations. If a facet is determined to be back facing, normals of the opposite sign are used for reflectance calculations.

4.5.5.5 Intrinsic colour

The intrinsic colour is the colour or colours associated with an output primitive prior to applying the reflectance calculation. The possible sources of intrinsic colour differ among different output primitives, and some output primitives have multiple possible sources. In the case of area primitives, the source of intrinsic colour is dependent on the interior style and the data mapping method. Table 6 lists the source of intrinsic colour for each output primitive. Where multiple sources are listed, they are listed in their decreasing order of default precedence.

Table 6 — Sources of intrinsic colour

Output Primitive	Sources of intrinsic colour
polyline	polyline colour aspect
polymarker	polymarker colour aspect
text	text colour aspect
annotation text	text colour aspect
fill area	pattern representation, interior colour aspect
fill area set	pattern representation, interior colour aspect
cell array	cell colours
generalized drawing primitive	any, according to the type of GDP
polyline set with colour	vertex colours, polyline colour aspect
fill area set with data	pattern representation, data mapping, interior colour aspect
cell array PLUS	cell colours
set of fill area sets with data	pattern representation, data mapping, interior colour aspect
triangle set with data	pattern representation, data mapping, interior colour aspect
triangle strip with data	pattern representation, data mapping, interior colour aspect
quadrilateral mesh with data	pattern representation, data mapping, interior colour aspect
non-uniform B-spline curve	polyline colour aspect
non-uniform B-spline curve with colour	colour spline, polyline colour aspect
non-uniform B-spline surface	pattern representation, interior colour aspect
non-uniform B-spline surface with data	pattern representation, data mapping, interior colour aspect

Where pattern representation is indicated in table 6, the pattern representation is the source of intrinsic colour only when the interior style is PATTERN. In this case the source of the intrinsic colour is the pattern representation indicated by the appropriate INTERIOR STYLE INDEX or BACK INTERIOR STYLE INDEX attribute. Intrinsic colour is derived from the pattern representation in a workstation dependent way.

Except in the case described in the preceding paragraph for interior style PATTERN, the source of intrinsic colour for the output primitives defined in ISO/IEC 9592-1 is the corresponding colour aspect for that primitive. For polyline set with colour primitives the source is the vertex colours, if specified, otherwise the source is the polyline colour aspect. For fill area set with data, set of fill area set with data, triangle set with data, triangle strip with data and quadrilateral mesh with data primitives the source of the intrinsic colour is determined by the data mapping method aspect. For cell array and cell

array PLUS primitives the intrinsic colour is the collection of the primitive's cell colours.[4] For non-uniform B-spline curve primitives the source of intrinsic colour is the polyline colour aspect. For non-uniform B-spline curve with colour primitives the source of intrinsic colour is the colour spline specified with the primitive, or the polyline colour aspect if no colour spline is specified. For non-uniform B-spline surface primitives the source of intrinsic colour is the interior colour aspect. For non-uniform B-spline surface with data primitives the source of intrinsic colour is determined by the data mapping method aspect if colour or data splines are specified; the intrinsic colour is the interior colour aspect if no colour or data splines are specified. The source of intrinsic colour for generalized drawing primitives can be any of the defined sources.

Since the interior shading method potentially influences the interpolation of colour and data, the intrinsic colour at each point of the output primitive is also dependent on the interior shading method.

4.5.6 Facet culling

Individual facets can be removed from an area primitive according to their orientation and the value of the FACET CULLING MODE attribute. A value of BACKFACING indicates that back-facing facets are removed from the primitive. A value of FRONTFACING indicates that front-facing facets are removed from the primitive. A value of NONE indicates that no facets are removed from the primitive. When a facet is removed, no portion of it is displayed, e.g., its interior, edges, or parametric surface characteristics.

4.5.7 Distinguishing facets by orientation

Different attributes can be applied to individual facets of an area primitive according to whether their orientation is front facing or back facing. The attribute FACET DISTINGUISHING MODE controls whether PHIGS PLUS uses the orientation of a facet to select which attributes to apply. When FACET DISTINGUISHING MODE has the value OFF, the same aspects are applied to both front-facing and back-facing facets. Many attributes have counterparts that are applied to back-facing facets when FACET DISTINGUISHING MODE has the value ON. The aspects, attributes, and ASFs that apply to back-facing facets are prefixed with BACK.

4.5.8 Hidden line and hidden surface removal

As in ISO/IEC 9592-1, implementation of any HLHSR capabilities is optional. If HLHSR capabilities are provided, however, an implementation shall not make methods available that interfere or conflict with PHIGS PLUS functionality.

4.5.9 Stability

The aspects of output primitives composed of planar convex facets (where planarity is in all of the geometry, colour, data, and vertex normals used—4.6.4.2—and convexity is in the geometry only) and undergoing only rigid transformations shall not appear to vary based only on the primitive's orientation in any coordinate space. Similarly, the aspects of parametric curve and parametric surface output primitives when those primitives undergo rigid transformations shall not appear to vary based only on the primitive's orientation in any coordinate space.

4) Colours of cell array and cell arrays PLUS primitives are, unlike other area primitives, substantial properties of the primitives themselves. Therefore, their intrinsic colour is taken solely from the cell array values, never being derived from any other aspects or attributes.

4.6 The PHIGS PLUS rendering pipeline

4.6.1 General

In PHIGS PLUS, colour values to be displayed on the display surface of a workstation are computed by the rendering pipeline (see figure 10). The conceptual rendering pipeline consists of a data mapping stage, a shading (interpolation) stage, a lighting stage, a depth cueing stage, and a colour mapping stage. The input to this pipeline includes the output primitive facets that remain after facet culling, and the primitive attributes as determined by the orientation of the primitive and the FACET DISTINGUISHING MODE attribute.

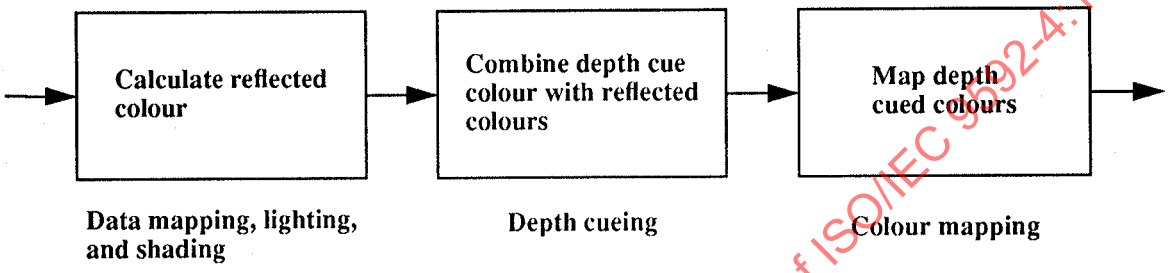


Figure 10 — The conceptual rendering pipeline

The rendering pipeline uses a primitive's attributes, orientation and position in world coordinates to determine the displayed colours for that primitive. The data mapping stage computes a primitive's intrinsic colour from data specified with the primitive. The lighting stage computes the effect of light sources on a primitive's appearance. Lighting in PHIGS PLUS is applied on a primitive by primitive basis; no interactions between objects such as shadows or reflections are defined. The shading stage interpolates data across a primitive. The depth cueing stage conditionally modifies the colours of the lit and shaded primitive by combining them with the depth cue colour on the basis of the primitive's Z component in NPC. The colour mapping stage maps the colours produced by the previous stages into other, possibly workstation dependent, colours before they are displayed.

4.6.1.1 Primitives affected by the rendering pipeline

Lighting and shading are applicable to only area primitives, with the exceptions that polyline primitives utilize shading for interpolation of colours, and cell arrays may be affected by lighting. It is acceptable not to apply lighting to cell array and cell array PLUS primitives, and shading does not apply to them. Markers, text and annotation text, as well as fill area set edges and edges of other area primitives are not processed by the lighting and shading stage. Data mapping applies to all with-data primitives. Depth cueing and colour mapping are applicable to all primitives.

The effect of lighting and shading self-coincident or self-intersecting facets of area primitives is workstation dependent.

4.6.1.2 The effect of the interior style on lighting and shading

Conceptually, lighting applies to HOLLOW, SOLID, and PATTERN interior styles; shading applies to HOLLOW, SOLID, PATTERN, and HATCH interior styles. It is acceptable for an implementation not to light and shade an area primitive when the interior style is PATTERN or HATCH. Lighting and shading are not applied when the interior style is EMPTY.

The effect of lighting and shading when the interior style is HOLLOW is the same as for SOLID, but only primitive boundaries are displayed. Points on the boundaries, including points in boundary segments created by clipping, have the same colour as they would if the interior style were SOLID. For interior style PATTERN the pattern is conceptually applied prior to lighting and shading, and the colours used in the reflectance calculation are taken from the pattern definition.

4.6.1.3 Aspects and attributes used in the rendering pipeline

The rendering pipeline makes use of the following aspects and attributes: polyline shading method, interior colour, interior style, interior style index, data mapping method, DATA MAPPING INDEX, reflectance properties, reflectance model, interior shading method, back interior colour, back interior style, back interior style index, back data mapping method, BACK DATA MAPPING INDEX, back reflectance properties, back reflectance model, back interior shading method, LIGHT SOURCE STATE, RENDERING COLOUR MODEL, DEPTH CUE INDEX, and COLOUR MAPPING INDEX. The rendering pipeline also makes use of the implicit eye point and any geometric and colour information specified with an output primitive.

The eye point is a physical parameter that is implicit in the PHIGS viewing matrices. The eye point is that point in WC that transforms to infinite positive Z in NPC (the homogeneous point (0, 0, 1, 0)). The eye information needed for the reflectance calculations can therefore be determined by transforming the NPC point (0,0,1,0) through the inverses of the view mapping and view orientation matrices, in that order. For perspective projections this operation will result in a location in world coordinates that corresponds to the projection reference point of the PHIGS view mapping utility function (if the PHIGS utility functions or comparable methods were used to derive the view mapping). For parallel projections this inverse transformation results in a direction vector on which the eye point lies at infinity. In the case of singular matrices there is either no determinate eye point or an infinite number of them. The implementation will choose an arbitrary location in the former case and one of the possible locations in the latter case.

4.6.2 Data mapping

Data mapping is a means of computing intrinsic colour from colour or other application-specific data associated with the following output primitives: fill area set with data, set of fill area sets with data, triangle set with data, triangle strip with data, quadrilateral mesh with data and non-uniform B-spline surface with data. The data mapping process converts one or more fields of data to a single colour at points on the output primitive where an intrinsic colour is needed. This colour is used as input to the reflectance calculation and subsequent stages of the rendering pipeline.

For all the output primitives listed in the previous paragraph except non-uniform B-spline surface with data, the data is specified as either a colour at each facet or vertex or a list of real values at each facet or vertex. If a list of real values is specified, more entries in the list can be specified than are used in any single traversal of the output primitive structure element. The values to use in a particular execution are specified separately in the data mapping method data record.

For non-uniform B-spline surface with data output primitives the data is in the form of a colour spline or a list of data splines. Each spline specifies the distribution of colour or data over the surface's geometry. If a list of data splines are specified, more entries in the list can be specified than are used in any single execution of the output primitive structure element. Which data splines to use in a particular execution are specified separately in the data mapping method data record. These data splines may have a greater dimensionality than that needed by a particular data mapping method (perhaps because the full dimensionality is used by some other data mapping method), in which case the description of the data mapping method shall describe the dimensions used. (The defined data mapping methods use only the first dimension of the data splines.)

The data mapping method aspect controls the mapping of data values to intrinsic colour. It specifies which data values are used and how they select the intrinsic colour. There are separate data mapping method aspects for front- and back-facing facets. The data mapping method aspect consists of a method indicator and a data record, the contents of which are method dependent.

Output primitives subject to data mapping may have more information specified with them than is used during any single execution of their structure element. For all but non-uniform B-spline surface with data output primitives this information can be a combination of vertex colours, vertex data, facet colours or facet data. For non-uniform B-spline surface output primitives both a colour spline and a list of data splines can be specified. Which of this information to use for a particular data mapping is indicated in the source selector list in the data mapping method data record. The source selector list is a list of flags indicating the precedence of the possible intrinsic colour sources. Each selector in the list may have one of the values listed in table 7.

Table 7 — Intrinsic colour source selectors

Selector value	Selector meaning
COLOUR_ASPECT	Use the appropriate colour aspect, e.g., interior colour, as the source of intrinsic colour.
FACET_COLOUR	Use facet colour as the source of intrinsic colour.
FACET_DATA	Use facet data as the source of intrinsic colour.
VERTEX_COLOUR	Use vertex colour as the source of intrinsic colour.
VERTEX_DATA	Use vertex data as the source of intrinsic colour.

The order of the flags in the selector list indicates the precedence of the corresponding data relative to the other sources. The first entry in the list has the highest precedence, the last entry has the lowest. When an output primitive subject to data mapping is encountered during traversal, the selector list is searched from first entry to last until a selector is found that corresponds to a source of intrinsic colour specified with the output primitive and applicable to the current data mapping method. If such a match occurs then the indicated source is used to determine the intrinsic colour of the primitive. If no match is found or the selector list is empty the interior colour aspect is used as the source of intrinsic colour. For non-uniform B-spline surface with data output primitives source selector flag values of FACET_COLOUR and VERTEX_COLOUR indicate that the surface colour spline is the source of intrinsic colour. Source selector flag values of FACET_DATA and VERTEX_DATA indicate that the data splines are the source of intrinsic colour.

Not all data mapping methods support all the above sources of intrinsic colour. Data mapping method COLOUR, for instance, uses only the appropriate colour aspect, vertex, or facet colour, so only those three source selector values are meaningful for that data mapping method. The description of each data mapping method indicates which sources are supported by that method. Intrinsic colour sources not meaningful to a particular data mapping method are ignored when that data mapping method is in effect.

The data mapping method is a non-geometric aspect and is both individually specified and bundled. Data mapping bundles are entries in the data mapping table in the workstation state list. They are set with the function SET DATA MAPPING REPRESENTATION. The front and back data mapping methods are set individually with the structure elements "set data mapping method" and "set back data mapping method", and are selected from the data mapping bundle table with the structure elements "set data mapping index" and "set back data mapping index". Each index indicates a bundle in the data mapping bundle table.

During traversal the 'current data mapping method ASF' in the traversal state list determines the source of the data mapping method. If it is BUNDLED then the data mapping method specified in the data mapping bundle indicated by the 'current data mapping index' is used. If it is INDIVIDUAL then the 'current data mapping method' in the traversal state list is used. Similarly, the 'current back data mapping method ASF' indicates the source of the back data mapping method, either from the data mapping bundle indicated by the 'current back data mapping index' or the 'current back data mapping method' entry in the traversal state list.

Data mapping method 1 is used during traversal if the specified data mapping method is not available on the workstation, or if the data mapping method or the data record contents are inconsistent with the data specified with a primitive. The data record values used in these cases are the default values indicated in the PHIGS description table. (Examples of inconsistencies include fewer data values specified than are needed and a data value index specifying a data item that does not exist in a list of data associated with the primitive.) The effect of this fallback behaviour shall be only on the display of the primitive associated with the inconsistency.

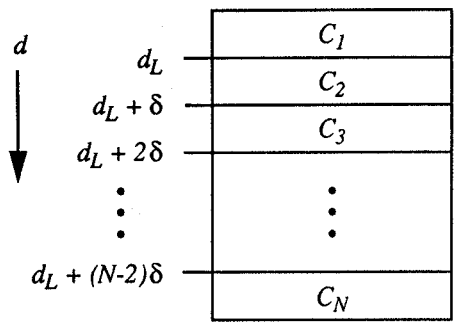
The defined data mapping methods are

≤ 0 Implementation dependent.

1 COLOUR: Any colours specified with the output primitive are used directly as the source of intrinsic colour. Only source selector values of COLOUR_ASPECT, FACET_COLOUR, and VERTEX_COLOUR are meaningful for this data mapping method. Other values can be specified but will be ignored. For non-uniform B-spline surface with

data output primitives, both FACET_COLOUR and VERTEX_COLOUR indicate that the surface's colour spline, if specified, is used as the source of intrinsic colour.

2 SINGLE VALUE, UNIFORM: A single data value is mapped to a colour selected from a specified list of N colours (see figure 11). The colour selected from this list is based on the relationship of the data value to a specified data range. Data values below the data range are assigned the first colour in the colour list. Data values above the data range are assigned the last colour in the colour list. Data values within the data range are assigned a colour by associating, uniformly, a portion of the range with each of the colours from the second to the penultimate colour in the colour list and selecting the colour associated with the portion of the range containing the data value. The colour-to-range association is defined in figure 11.



d = data value
d_L = lower range limit
d_H = upper range limit

$$\delta = \frac{d_H - d_L}{N-2}\ ,\ N > 2$$

Use colour C_1 when $d < d_L$

Use colour C_{n+2} when $d_L + n\delta \le d < d_L + (n+1)\delta$,
$0 \le n \le N-3$

Use colour C_N when $d \ge d_H$

Figure 11 — UNIFORM data mapping of data values to colours

The data value to map, subject to any interpolation indicated by the shading method aspect, is indicated by an index into the data lists. This index is specified in the data mapping method's data record. For non-uniform B-spline surface with data output primitives, the index indicates the single data spline in the list of data splines, if specified, that is used to determine the data values to map. If that data spline is multi-dimensional only the first coordinate value is used.

The data record for this method contains the intrinsic colour source selector list, the data value index, the two data range limits, and the list of colours. The list of colours must contain at least three colours, unless the lower range limit is equal to the upper range limit, in which case the list must contain at least two colours.

Only source selector values of FACET_DATA and VERTEX_DATA are meaningful for this data mapping method. Other values can be specified but will be ignored. For non-uniform B-spline surface with data output primitives both FACET_DATA and VERTEX_DATA indicate that the surface's data splines, if specified, are used as the source of intrinsic colour.

3 SINGLE VALUE, NON-UNIFORM: A single data value is mapped to a colour selected from a specified list of N colours (see figure 12). The colour selected is based on the relationship of the data value to a specified set of data ranges. The ranges are specified as a list of $N-1$ range boundaries in increasing order; N shall be ≥ 2. Each range boundary specifies the lower limit of the range above it, inclusive, and the upper limit of the range below it, non-inclusive, that is, range $i = [d_i,\ d_{i+1})$; thus the ranges are adjacent. The first range extends to negative infinity, the last range extends to positive infinity. Each range is associated with a colour in the colour list. Data values are assigned the colour associated with the range that contains them. Data values below the first range boundary are assigned the first colour in the colour list. Data values above the last range boundary are assigned the last colour in the colour list. The colour-to-range association is defined in figure 12. The number of colours in the list shall be one more than the number of boundaries.

The data value to map, subject to any interpolation indicated by the shading method aspect, is indicated by an index into the data lists. This index is specified in the data mapping method's data record. For non-uniform B-

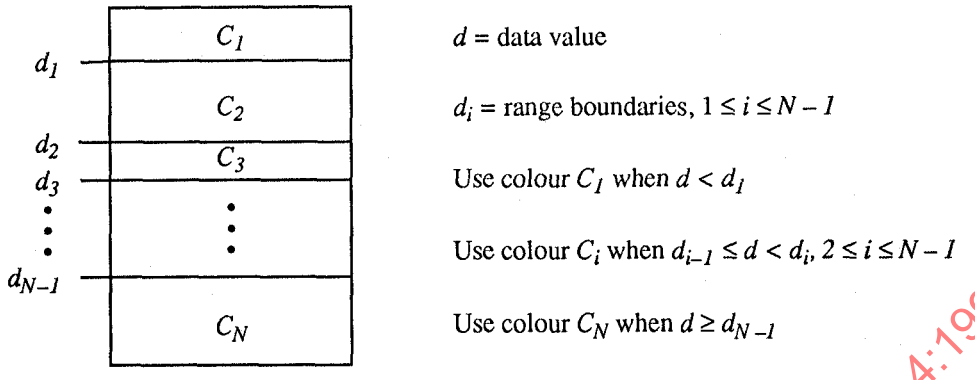


Figure 12 — NON-UNIFORM mapping of data values to colour

spline surface with data output primitives the index indicates the single data spline in the list of data splines, if specified, that is used to determine the data values to map. If that data spline is multi-dimensional only the first coordinate value is used.

The data record for this method contains the intrinsic colour source selector list, the data value index, the list of N colours, and the list of $N-1$ data range boundaries.

Only source selector values of FACET_DATA and VERTEX_DATA are meaningful for this data mapping method. Other values can be specified but will be ignored. For non-uniform B-spline surface with data output primitives, both FACET_DATA and VERTEX_DATA indicate that the surface's data splines, if specified, are used as the source of intrinsic colour.

4 BI-VALUE, UNIFORM: Two data values are mapped to a single colour selected from one of a specified set of colour lists, each list having possibly different lengths (see figure 13). The first data value, d_a, determines from which colour list to select the colour; the second data value, d_b, determines which colour to select from that list. The list and colour selected are based on the relationship of the data values to two specified data ranges, R_a for the d_a data values and R_b for the d_b data values. Values of d_a below the lower limit of R_a indicate that the first colour list is used. Values of d_a above or equal to the upper limit of R_a indicate that the last colour list is used. Values of d_a within R_a indicate the list to use by associating, uniformly, a portion of R_a with each of the colour lists from the second to the penultimate colour list and selecting the list associated with the portion of R_a containing the data value. The colour list to range association is defined in figure 13. Data value d_b is used to determine the colour in the selected list by applying the same mapping defined for data mapping method SINGLE VALUE, UNIFORM, using d_b as the data value, R_b as the range, and the colour list selected by d_a.

The two data values to map, subject to any interpolation indicated by the shading method aspect, are indicated by two indices into the data lists. The first index indicates d_a, the second index indicates d_b. These indices are specified in the data mapping method's data record. For non-uniform B-spline surface with data output primitives the indices indicate the two data splines in the list of data splines, if specified, that are used to determine the data values to map. If either of the two indicated data splines is multi-dimensional only the first coordinate value of each is used.

The data record for this method contains the intrinsic colour source selector list, the two data value indices, the data range limits for each of the two ranges, and the colour lists. The colour lists can be of differing lengths. There shall be at least three color lists specified, unless the limits of range Ra are equal, in which case there shall be at least two colour lists. Each list shall have at least three entries, unless the corresponding range limits, R_b, are equal, in which case there shall be at least two colours in the list.

Only source selector values of FACET_DATA and VERTEX_DATA are meaningful for this data mapping method. Other values can be specified but will be ignored. For non-uniform B-spline surface with data output primitives

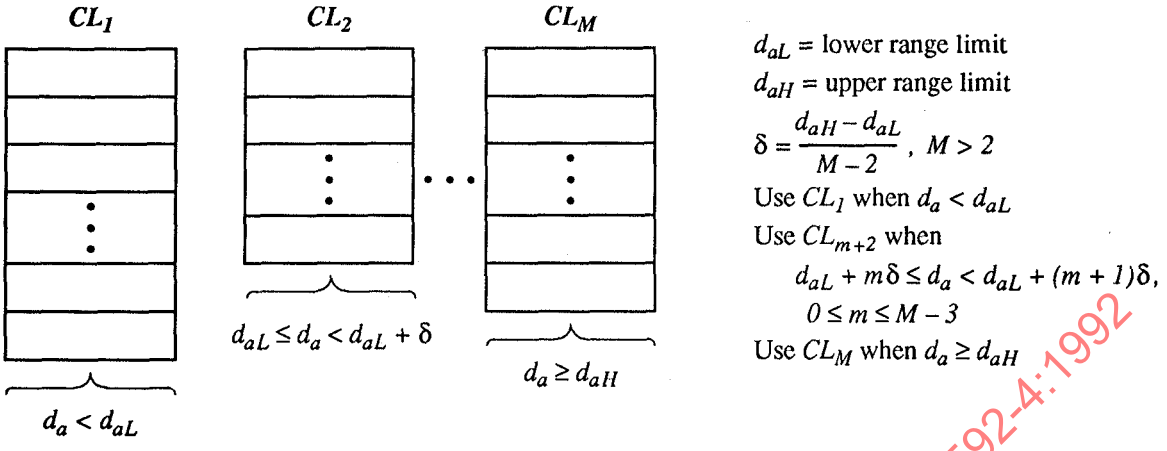


Figure 13 — UNIFORM mapping of data values to colour lists

both FACET_DATA and VERTEX_DATA indicate that the surface's data splines, if specified, are used as the source of intrinsic colour.

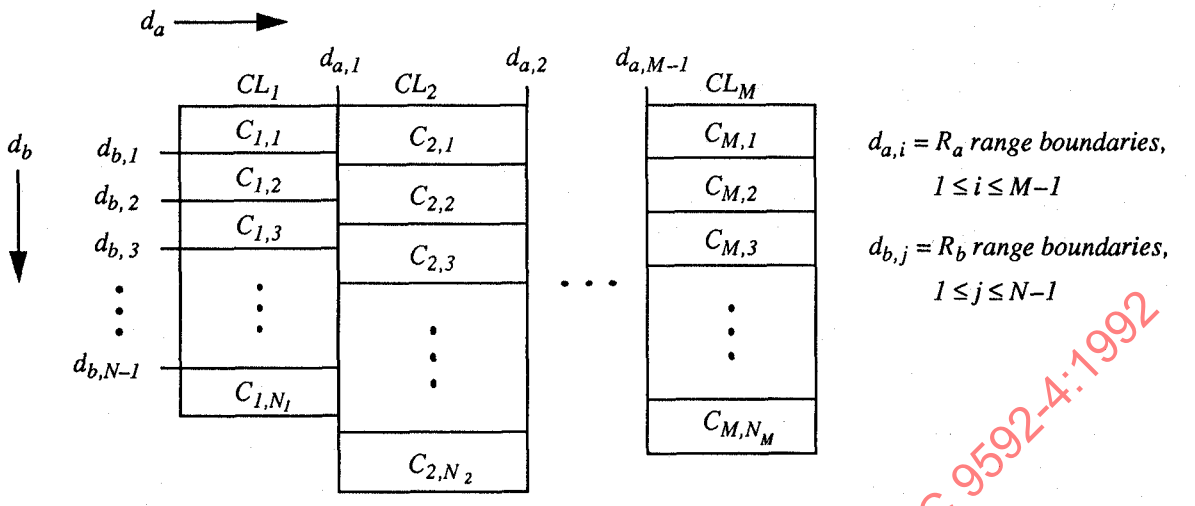
5 BI-VALUE, NON-UNIFORM: Two data values are mapped to a single colour selected from one entry in a set of colour lists (see figure 14). The colour selected is based on the relationship of a data value, d_a, to a specified set of data ranges, R_a, and the relationship of a data value, d_b, to one of a specified set of data ranges, $R_{b,i}$. There is one $R_{b,i}$ range set for each range in R_a. The ranges in each range set are specified as a list of range boundaries in increasing order. Each range boundary specifies the lower limit of the range above it, inclusive, and the upper limit of the range below it, non-inclusive, i.e., range $i = [d_i, d_{i+1})$, thus the ranges are adjacent. The first range of each set extends to negative infinity, the last range of each set extends to positive infinity. Each range in R_a is associated with a colour list, CL_i, and a range set, $R_{b,i}$. Each range in each $R_{b,i}$ is associated with a single entry in the corresponding colour list, CL_i. The data value d_a selects the $R_{b,i}$ range set and CL_i colour list associated with the range in R_a containing d_a. The data value d_b selects the colour in CL_i associated with the range in $R_{b,i}$ containing d_b. The colour-to-range association is defined in figure 11.

The two data values to map, subject to any interpolation indicated by the shading method aspect, are indicated by indices into the data lists. The first index indicates d_a, the second index indicates d_b. These indices are specified in the data mapping method's data record. For non-uniform B-spline surface with data output primitives the indices indicate the two data splines in the list of data splines, if specified, that are used to determine the data values to map. If either of the two indicated data splines is multi-dimensional only the first coordinate value of each is used.

The data record for this method contains the intrinsic colour source selector list, the two data value indices, the list of range boundaries for the R_a range, the lists of data range boundaries for the R_b ranges, and the colour lists. The R_a and R_b ranges must contain at least one boundary. There must be one more colour list than the number of range boundaries in range R_a. The colour lists can be of differing lengths. Each list shall contain one more colour than the number of range boundaries in the corresponding range, $R_{b,i}$.

Only source selector values of FACET_DATA and VERTEX_DATA are meaningful for this data mapping method. Other values can be specified but will be ignored. For non-uniform B-spline surface with data output primitives both FACET_DATA and VERTEX_DATA indicate that the surface's data splines, if specified, are used as the source of intrinsic colour.

≥ 6 Reserved for registration or future standardisation.



Colour $C_{i,j}$ is used, where j is the index of the j-*th* colour in the i-*th* colour list, and

$i = 1$ when $d_a < d_{a,1}$,
$i = M$ when $d_a \ge d_{a,M-1}$
$i = m$ when $d_{a,m-1} \le d_a \le d_{a,m}$, $2 \le m \le M-1$

$j = 1$ when $d_b < d_{b,1}$
$j = N$ when $d_b \ge d_{b,N-1}$
$j = n$ when $d_{b,n-1} \le d_b \le d_{b,n}$, $2 \le n \le N-1$

Figure 14 — BI-VALUE, NON-UNIFORM data mapping

4.6.3 Lighting

4.6.3.1 Reflectance calculation

The reflectance calculation is conceptually applied at one or more positions on the primitive being lit and produces a colour at such points. Input to the reflectance calculation conditionally includes the position on the primitive at which the reflectance calculation is performed, the reflectance normal and intrinsic colour at that position, the reflectance properties, the reflectance model, the set of active light sources, and potentially other information.

The reflectance model aspect selects the lighting model to use in the reflectance calculation. The defined models are

≤ 0 Implementation dependent.

1 No reflectance calculation performed; colour is not affected by lighting and shading.

2 Ambient reflectance effects computed.

3 Ambient and diffuse reflectance effects computed.

4 Ambient, diffuse, and specular reflectance effects computed.

≥ 5 Reserved for registration or future standardisation.

Reflectance model 1 is used during traversal if the specified reflectance model is not available.

It is recommended that the effect of the reflectance calculation be as if the calculations were performed in WC; however, this is not required. Suggested reflectance formulae are given in annex C.

4.6.3.2 Light sources

Light sources affect the display of area primitives. Their effect is determined by the reflectance calculation applied during the lighting and shading stage of the rendering pipeline.

All light sources have a light source type and a light source colour. They may also have additional information such as light source position, light source direction, concentration exponent, spread angle, and attenuation coefficients. Light source specifications have a light source type and a data record. The data record is specific to each light source type. The light source type is an integer that selects one of the following types:

≤ 0 Implementation dependent.

1 AMBIENT: Ambient light sources have a light source colour and affect area primitives independently of their orientation and position.

2 DIRECTIONAL: Directional light sources have a light source colour and light direction in world coordinates. Conceptually they are located at infinity. They affect an area primitive based on the primitive's orientation, but independently of its position. The light direction is indicated by a unit vector pointing in the direction the light travels.

3 POSITIONAL: Positional light sources have a light source colour, light source position in world coordinates, and attenuation coefficient. They affect an area primitive based on the primitive's position and orientation. The light position is given by a 3D point in world coordinates.

4 SPOT: Spot light sources have a light source colour, light source position in world coordinates, light direction in world coordinates, attenuation coefficient, concentration exponent, and spread angle. If a portion of an output primitive lies outside the cone of influence of a light source of this type, as determined by the light source position, light source direction and spread angle, the colour of that portion of the primitive is not affected by that light source. The light source position is given by a 3D point in world coordinates. The light direction is indicated by a unit vector pointing in the direction that light at the center of the light beam travels. The spread angle is given as the angle between the light direction vector (the center of the light beam) and the edge of the cone of influence.

≥ 5 Reserved for registration or future standardisation.

4.6.3.3 Workstation light sources

Workstation light source definitions are stored in the 'light source table' of the workstation state list. The table shall have at least two entries predefined. All entries in the 'light source table' can be set with the function SET LIGHT SOURCE REPRESENTATION.

The set of workstation light sources active during traversal is controlled by the "set light source state" structure element. The 'current light source state' in the traversal state list contains the list of active workstation light sources. Each entry in the list is an index into the 'light source table' of the workstation state list. Light sources are added to or removed from this list with the "set light source state" structure element. The 'current light source state' is saved prior to execution of each structure in a structure network and restored upon a return from each execution.

Entries in the 'current light source state' corresponding to undefined light source table entries are ignored. If the number of active non-ambient light sources indicated by the 'current light source state' exceeds the number supported by the workstation, it is workstation dependent which of the indicated light sources are used.

4.6.4 Shading

4.6.4.1 General

Shading in the rendering pipeline conditionally interpolates geometric information, intrinsic colour data, and reflectance calculation results across the primitive. The method of interpolation is controlled by the shading method, which indicates the information to be interpolated, if any.

4.6.4.2 Interpolation

Linear interpolation along line segments or across facets is the linear combination of the values associated with the vertices of the line segment or facet. The interpolation shall be continuous along the segment or across the facet.

When colour, data, or normals are present at vertices, then the values being interpolated are the combination of geometry plus colour, geometry plus data, and geometry plus normals. In this context, the normals are taken as the coordinate values of their end-point, assuming that their origin has been translated to the coordinate system origin. Each of the above combinations constitutes an n-dimensional space in which planarity is defined as follows:

a) Three non-colinear points in n-dimensional space define a plane.

b) A point is co-planar with three points that define a plane if its coordinate values can be uniquely expressed as a linear combination of the others.

Linear interpolation of n-dimensional values is equivalent to the separate linear interpolation of each of its coordinates. The effect of linear interpolation shall be as if it is performed in world coordinates, even though the interpolation may be carried out in another coordinate system.

Linear interpolation in PHIGS PLUS is defined only for line segments and triangular facets. Interpolation across facets with more than three vertices is implementation dependent. A recommended interpolation for such facets is to subdivide the facet into triangles and interpolate across the triangles.

4.6.4.3 Colour interpolation

Let the end-points of a line segment be P_1 and P_2, with colours C_1 and C_2. For any point P on the line segment, the interpolated colour, C, is defined by the formula:

$$C = \omega C_1 + \omega_2 C_2$$

where

$$\omega_1 = \frac{|P_2 - P|}{|P_2 - P_1|} \quad \text{and} \quad \omega_2 = \frac{|P_1 - P|}{|P_2 - P_1|}$$

and $|P - Q|$ denotes the Euclidean distance between the points P and Q.

Let the vertices of a triangle be P_1, P_2, and P_3, with colours C_1, C_2, and C_3. For any point P inside the triangle the interpolated colour C is defined by the formula:

$$C = \omega_1 C_1 + \omega_2 C_2 + \omega_3 C_3$$

where

$$\omega_1 = \frac{area\,(\overline{PP_2P_3})}{area\,(\overline{P_1P_2P_3})}, \quad \omega_2 = \frac{area\,(\overline{PP_1P_3})}{area\,(\overline{P_1P_2P_3})}, \quad \omega_3 = \frac{area\,(\overline{PP_1P_2})}{area\,(\overline{P_1P_2P_3})},$$

$\overline{PQR}$ denotes the triangle with vertices P, Q, and R, and *area(PQR)* denotes the area of the triangle formed by the vertices P, Q, and R.

4.6.4.4 Data interpolation

Let the end-points of a line segment be P_1 and P_2, with data values D_1 and D_2. For any point P on the line segment, the interpolated data value, D, is defined by the formula:

$$D = \omega_1 D_1 + \omega_2 D_2$$

where ω_1 and ω_2 are as given in 4.6.4.3.

Let the vertices of a triangle be P_1, P_2, and P_3, with data values D_1, D_2, and D_3. For any point P inside the triangle the interpolated data value, D, is defined by the formula:

$$D = \omega_1 D_1 + \omega_2 D_2 + \omega_3 D_3$$

where ω_1, ω_2, and ω_3 are as given in 4.6.4.3.

4.6.4.5 Normal-vector interpolation

It is recommended that interpolation of normal vectors be through the interpolation of their spherical coordinates with the same interpolation method as that used for colour and data; however, it is acceptable to interpolate normal vectors by interpolating their Cartesian coordinates with the same interpolation method as that used for colour and data. The resulting vector must be normalized before being used in the reflectance calculation.

4.6.4.6 Polyline shading

Shading of polylines is the interpolation along the line's segments of any vertex colours specified with the primitive. The line type attribute is applied after shading and acts like a masking operation. Each polyline of a polyline set is coloured independently of the others in the set.

The defined polyline shading methods are

≤ 0 Implementation dependent.

1 NONE: If vertex colours are specified then the *i*-th vertex colour is used to colour the *i*-th line segment. If vertex colours are not specified then the entire polyline is coloured in a single colour determined by the polyline colour aspect.

2 COLOUR: If vertex colours are specified then they are interpolated, in the rendering colour model, along the line segments. If vertex colours are not specified then the entire polyline is coloured in a single colour determined by the polyline colour aspect.

≥ 3 Reserved for registration or future standardisation.

4.6.4.7 Interior shading

The shading of area primitives is controlled by the interior shading method aspect, which indicates the information to be interpolated, if any. The interior shading method is selected by the INTERIOR SHADING METHOD attribute or the INTERIOR INDEX, depending on the INTERIOR SHADING METHOD ASF. The corresponding BACK attributes are used for back-facing facets. The defined interior shading methods are described below. The effects of each shading method for each interior style are described.

The defined interior shading methods are

≤ 0 Implementation dependent.

1 NONE: For interior style SOLID a reflectance calculation is performed for each facet of a primitive to produce a single reflected colour per facet (see figure 15). The point on the facet used in the reflectance calculation is workstation dependent. The colour used in the reflectance calculation is the primitive's intrinsic colour. When vertex colour or vertex data is the source of intrinsic colour, that information is interpolated at the point used in the reflectance calculation and mapped to a colour according to the method indicated by the data mapping method aspect. The resultant intrinsic colour is used in the reflectance calculation.

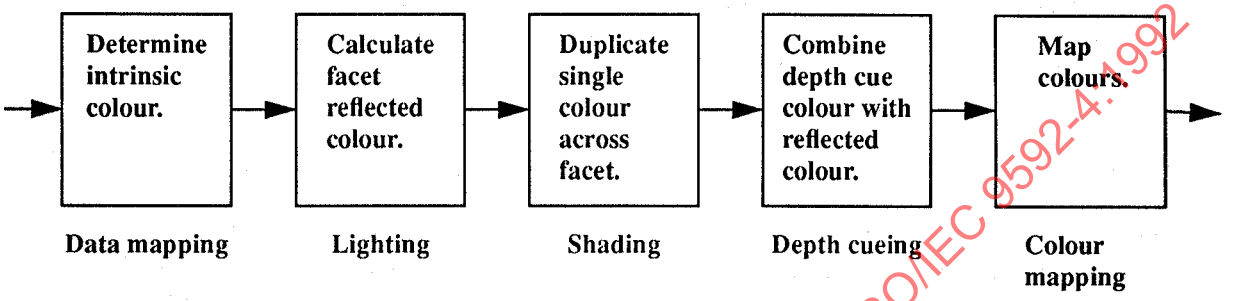


Figure 15 — A conceptual rendering pipeline for shading method 1 (NONE)

For interior style HATCH no interpolation is done and no reflectance calculation is performed.

For interior style PATTERN a single reflectance calculation is performed for each cell of the pattern after the pattern is projected onto the primitive. The positions on the primitive used in the reflectance calculation are workstation dependent.

2 COLOUR[5]: For interior style SOLID a reflectance calculation is performed at each vertex of a primitive, using the primitive's intrinsic colour and reflectance normal at that vertex. The resulting reflected colours are interpolated linearly across the primitive using the current RENDERING COLOUR MODEL (see figure 16).

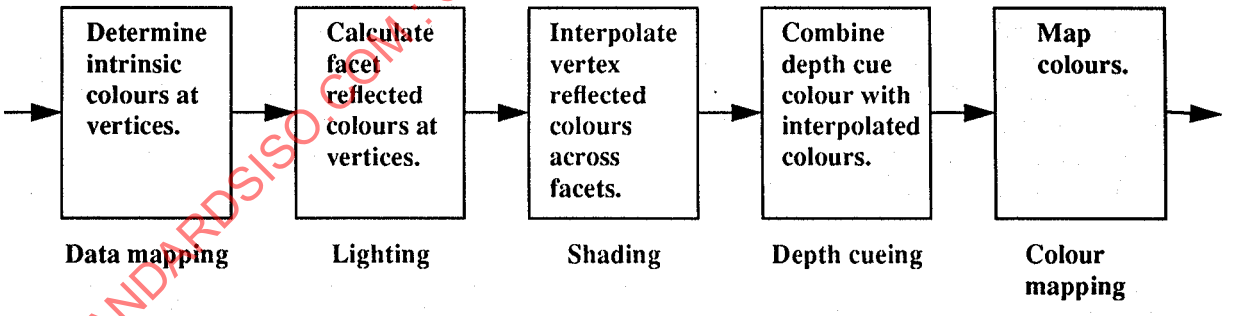


Figure 16 — A conceptual rendering pipeline for shading method 2 (COLOUR)

For interior style HATCH the intrinsic colours at the vertices of fill area and fill area set primitives are interpolated linearly across the primitive to determine the hatch colour at each position where the hatch is displayed. The application of interior style HATCH to the other area primitives is workstation dependent.

The effect for interior style PATTERN is the same as that described for interior shading method 1.

5) Traditionally known as Gouraud shading.

3 DATA: For interior style SOLID any reflectance calculation terms independent of the intrinsic colour may be computed at each vertex of the output primitive, using the reflectance normal at that vertex. (annex C lists the terms independent of intrinsic colour for the suggested reflectance formulae.) These terms are interpolated linearly across the primitive. A reflectance calculation is performed at each interpolated position to compute a reflected colour there, using the interpolated terms and the intrinsic colour at that position (see figure 17).

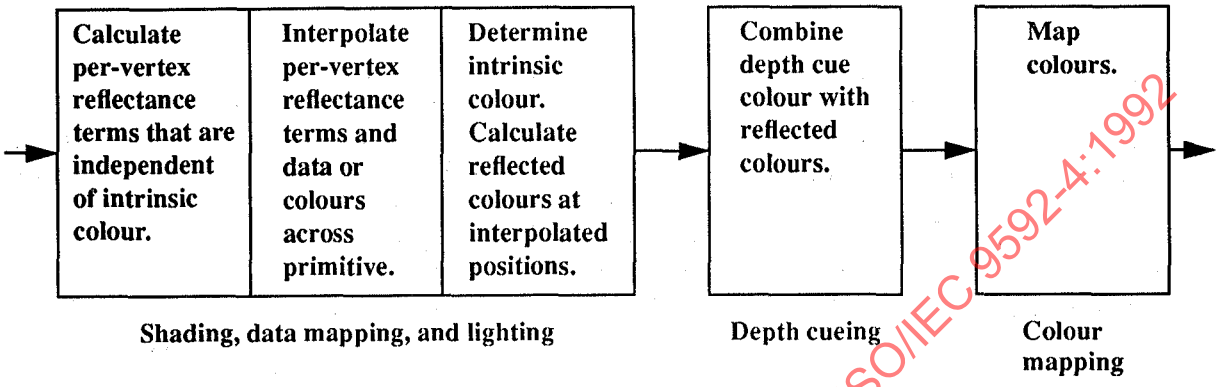


Figure 17 — A conceptual rendering pipeline for shading method 3 (DATA)

The intrinsic colour is determined as described in 4.5.5.5 and 4.6.2 at each position on the primitive where a reflectance calculation is performed. When vertex colour or vertex data is the source of intrinsic colour, that information is interpolated at the points on the primitive where a reflectance calculation is performed, and mapped to a colour according to the method indicated by the data mapping method.

The effect for interior style HATCH is the same as that described for shading method 2.

For interior style PATTERN the same procedure is used as for interior style SOLID, but the intrinsic colour at each position a reflectance calculation is performed is taken from the pattern cells after the pattern is projected onto the primitive.

4 DATA AND DOT: For interior style SOLID any dot products needed by the reflectance calculation are calculated from reflectance normals at a set of positions on the primitive. The dot products required for the calculation are dependent on the reflectance formulae used. (The dot products for the suggested reflectance formulae are listed in annex C.) These dot products are then interpolated linearly across the primitive. The reflectance calculation is performed at each interpolated position to produce a reflected colour at that position, using the interpolated dot products and the intrinsic colour at that position (see figure 18).

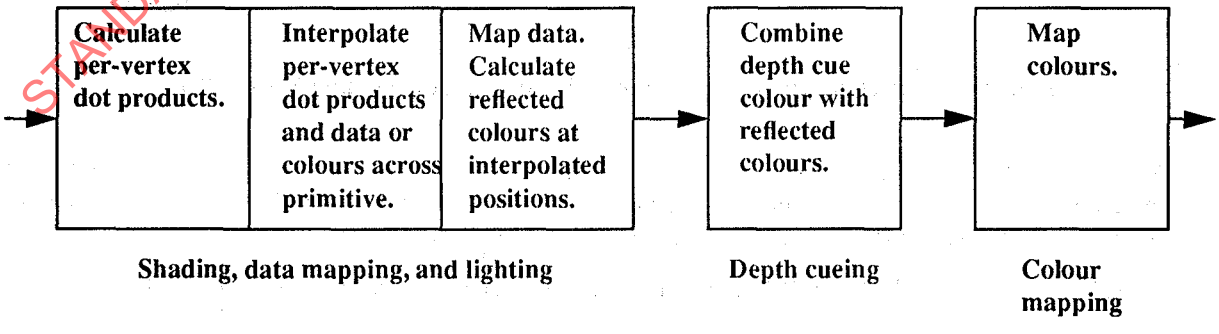


Figure 18 — A conceptual rendering pipeline for shading method 4 (DATA AND DOT)

The intrinsic colour is determined as described in 4.5.5.5 and 4.6.2 at each position on the primitive where a reflectance calculation is performed. When vertex colour or vertex data is the source of intrinsic colour, that information is interpolated at the points on the primitive where a reflectance calculation is performed, and mapped to a colour according to the method indicated by the data mapping method.

The effect for interior style HATCH is the same as that described for shading method 2.

For interior style PATTERN the same procedure is used as for interior style SOLID, but the intrinsic colour at each position a reflectance calculation is performed is taken from the pattern cells after the pattern is projected onto the primitive.

5 DATA AND NORMAL[6]: For interior style SOLID the reflectance normals are interpolated linearly across the primitive. A reflectance calculation is performed at each interpolated position to produce a reflected colour at that position, using the interpolated reflectance normal and the intrinsic colour at that position (see figure 19).

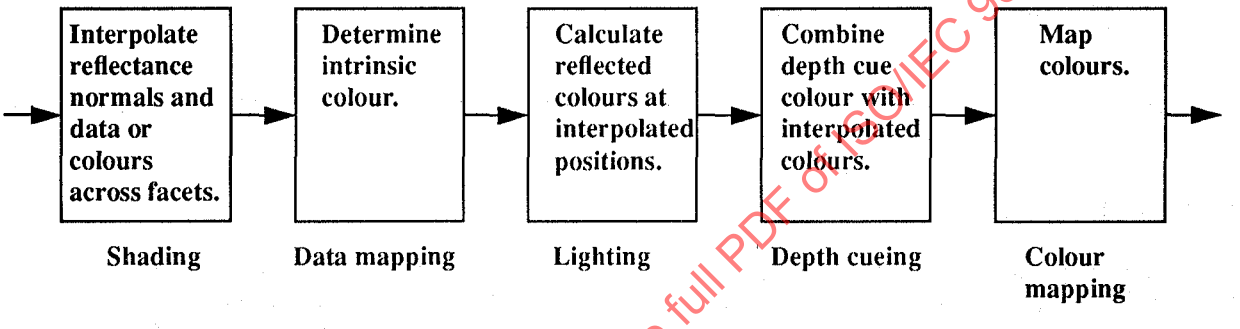


Figure 19 — A conceptual rendering pipeline for shading method 5 (DATA AND NORMAL)

The intrinsic colour is determined as described in 4.5.5.5 and 4.6.2 at each position on the primitive where a reflectance calculation is performed. When vertex colour or vertex data is the source of intrinsic colour, that information is interpolated at the points on the primitive where a reflectance calculation is performed, and mapped to a colour according to the method indicated by the data mapping method.

The effect for interior style HATCH is the same as that described for shading method 2.

For interior style PATTERN the same procedure is used as for interior style SOLID, but the intrinsic colour at each position a reflectance calculation is performed is taken from the pattern cells after the pattern is projected onto the primitive.

≥ 6 Reserved for registration or future standardisation.

Interior shading methods 2, 3, and 4 typically involve less computation than interior shading method 5, but may produce a lower quality image. This is most likely to be the case when the direction of vertex normals for a single facet vary substantially. Interior shading method 3 is included to interpolate data mapping data at a lower computational cost then methods 4 and 5.

4.6.5 The rendering colour model

The "set rendering colour model" structure element sets the colour model in which colour interpolation is performed during shading and depth cueing. The workstation dependent colour model is required; other models are optional. The effect of applying the reflectance calculation shall appear as though the calculation was performed in the rendering

6) Traditionally known as Phong shading.

colour model. The INQUIRE RENDERING COLOUR MODEL FACILITIES function returns the set of rendering colour models supported by a workstation. The defined rendering colour models are listed in table 8.

Table 8 — Rendering colour models

Value	Rendering colour model
< 0	Implementation dependent
0	WORKSTATION DEPENDENT
1	RGB
2	CIELUV
3	HSV
4	HLS
≥ 5	Reserved for registration or future standardisation

During traversal if the 'current rendering colour model' in the traversal state list is not supported, rendering colour model 0 is used.

4.6.6 Depth cueing

The depth cue stage modifies the colour of a primitive based on the Z coordinates of the primitive in NPC. The "set depth cue index" element selects an entry in the 'depth cue table' of the workstation state list. Each entry contains a depth cue mode, two depth cue reference planes, two depth cue scale factors and a depth cue colour (see figure 20).

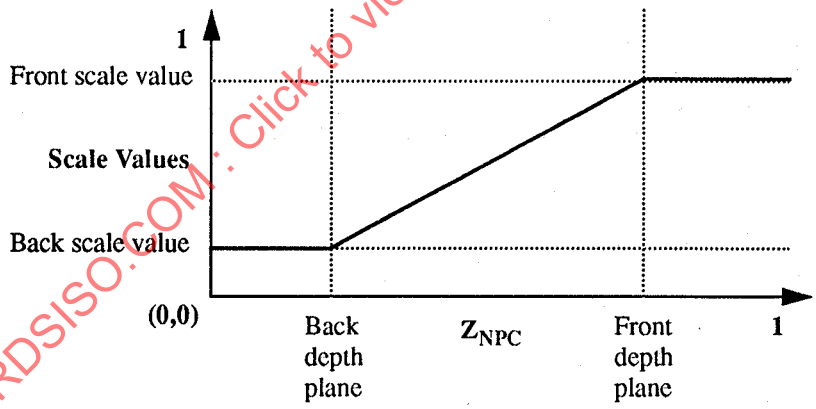


Figure 20 — Depth cue scaling

Depth cue mode is one of SUPPRESSED or ALLOWED, disabling or enabling the effect of depth cueing, respectively. A PHIGS implementation shall support at least depth cue mode SUPPRESSED. The depth cue reference planes are two Z values in NPC that define a back and a front reference plane for depth cueing. The depth cue scale factors are two values between 0 and 1. They correspond to the Z values of the depth cue reference planes and specify the portion of the primitive's colour to combine with a portion of the depth cue colour as a function of Z in NPC. The traversal default for the depth cue index is 1.

All entries in the depth cue table can be set with the function SET DEPTH CUE REPRESENTATION. Depth cue facilities can be inquired with the function INQUIRE DEPTH CUE FACILITIES. Entry 1 of the depth cue table is predefined to depth cue mode SUPPRESSED, other entries may also be predefined.

Depth cueing applies to all displayed portions of all primitives, including the edges of area primitives, their boundary when the interior style is HOLLOW, and the parametric surface characteristics (e.g., isoparametric curves) of parametric surfaces. The actual method used for depth cueing is workstation dependent. Suggested depth cueing formulae are given in annex E.

4.6.7 Colour mapping

The colour mapping stage converts the colours resulting from lighting, shading, and depth cueing to workstation dependent colours. The structure element SET COLOUR MAPPING INDEX specifies an index into the colour mapping table in the workstation state list. The traversal default for colour mapping index is 1. Each entry in the colour mapping table describes a method of mapping the colours produced by the depth cue stage into other colours. Colour mapping is the final stage of the rendering pipeline. The colours output from this stage are displayed as accurately as possible on the workstation's display surface.

Each entry in the colour mapping table contains a colour mapping method and a data record. The defined colour mapping methods and the description of the contents of their data records are:

≤ 0 Implementation dependent

1 TRUE: performs the identity mapping. The colours output are identical to those input from the depth cue stage. The data record contains no information.

2 PSEUDO: converts the colours to a single integer index. This index is used to select a colour in a list of colours specified in the data record. The data record contains the following entries:

- An integer specifying one of the colour models supported by the workstation to be used as the colour mapping colour model.
- An n-tuple of real numbers used for weighted averaging, called a weight vector, $W = (W_1, W_2, ..., W_n)$, where n is the number of dimensions in the specified colour mapping colour model.
- A list of colours in the specified colour model. Each entry in the list is specified by an n-tuple of real numbers corresponding to the coordinates of the specified colour model. An index value of 1 corresponds to the first entry of this list.

During traversal the mapping from a colour $(C_1, C_2, ..., C_m)$ in the current rendering colour model to an entry in the list of colours is achieved as follows:

- Convert $(C_1, C_2, ..., C_m)$ in the rendering colour model to the same colour $(C_1', C_2', ..., C_m')$ in the specified colour mapping colour model.
- Normalize the resulting colour components, if necessary, in a colour model dependent way so that each is in the range [0,1].
- Normalize the weight vector W by dividing it by the sum of its components, $(W_1 + W_2 + ... + W_n)$.
- Calculate $J = (R - 1)(W_1C_1' + W_2C_2' + ... + W_nC_n') + 1$, rounded to the nearest integer, where R is the number of colours in the list of colours given in the data record.
- Use J as an index to select one of the entries in the list of colours.
- The selected colour is output from the colour mapping stage and is displayed on the workstation as accurately as possible.

3 PSEUDO-N: convert the colours to a set of indices. These indices are used to select individual colour coordinates from lists of coordinates specified in the data record. This method is analogous to colour mapping method

PSEUDO, except that it specifies a separate list for each of the colour coordinates of the specified colour model. The output of this mapping is a colour composed of the n coordinates.

The data record contains the following entries:

— An integer specifying one of the colour models supported by the workstation to be used as the colour mapping colour model.

— n lists of colour coordinates, of length $R_1, R_2, \ldots, R_n$, where n is the number of dimensions in the specified colour mapping colour model. Entries in these lists are real numbers that represent coordinates in the respective dimensions of the specified colour mapping colour model. An index value of 1 corresponds to the first entry in each of these lists.

During traversal the mapping from a colour $(C_1, C_2, \ldots, C_m)$ in the current rendering colour model into n entries in the n lists is achieved as follows:

— Convert $(C_1, C_2, \ldots, C_m)$ in the rendering colour model to the same colour $(C_1', C_2', \ldots, C_m')$ in the specified colour mapping colour model.

— Normalize the resulting colour components, if necessary, in a colour model dependent way so that each is in the range [0,1].

— Calculate $J_1, J_2, \ldots, J_n$ such that $J_i = (R_i - 1)\, C_i + 1$, rounded to the nearest integer.

— Use $J_1, J_2, \ldots, J_n$ as indices to select colour components from the corresponding lists of colour coordinates.

— The n components define a colour in the specified colour mapping colour model. This colour is output from the colour mapping stage and displayed on the workstation as accurately as possible.

4 Reserved for future registration.

A PHIGS PLUS implementation shall support at least colour mapping method TRUE.

All entries in the colour mapping table can be set with the function SET COLOUR MAPPING REPRESENTATION. Entry 1 of the colour mapping table is predefined to colour mapping method 1 (TRUE). Other entries may also be predefined.

Colour mapping facilities can be inquired with the functions INQUIRE COLOUR MAPPING FACILITIES and INQUIRE COLOUR MAPPING METHOD FACILITIES. The function INQUIRE COLOUR MAPPING STATE returns the currently available colour mapping resources of an open workstation.

4.7 Workstations

PHIGS PLUS adds six categories of picture changes:

- data mapping representation modification
- reflectance representation modification
- parametric surface representation modification
- workstation light source representation modification
- depth cue representation modification
- colour mapping representation modification

For each category there is a "dynamic modification accepted for ..." entry in the workstation description table.

4.8 Graphical input

As in ISO/IEC 9592-1, picking occurs in the abstract 3D image created in NPC. The abstract image does not include portions of primitives that have been removed by facet culling.

Picking shall act as if performed on the geometric definition of the output primitives comprising the abstract image. Rendering effects, such as curve and surface approximation may be considered.

4.9 Limitations

4.9.1 General

PHIGS PLUS does not attempt to define correct behaviour for all possible combinations of parameters for the functionality it defines. PHIGS PLUS provides a wide range of functionality that is combined to achieve a broad range of effects. The quality of these effects is primarily dependent on the user's choice of data, careful use of the provided output primitives and attributes, and the computational cost the user is willing to sustain. PHIGS PLUS systems are expected to be capable of producing very high quality images and provide a range of computational-cost versus quality alternatives; but it is possible for a well implemented PHIGS PLUS system to produce poor quality images because poor choices were made by the user. Below are some specific areas known to require careful attention by the user.

4.9.2 Non-planar geometry and data

The effects produced by non-planar facet data are not defined. If the geometry data associated with a facet of an output primitive are not planar, as defined in 4.6.4.2, then the image produced from such information is workstation dependent. The workstation dependent behaviour need not be the same in all circumstances for a given workstation. Different combinations of non-planar data, geometry, coordinate transforms, and lighting specifications may result in different displayed images on the same workstation.

It is expected that if the deviation from planarity is small the workstation dependent behaviour should produce nearly the same effect as if the data were planar. This is not a requirement, however.

4.9.3 Relationship of shading method to geometry

The effects produced by the shading methods are heavily dependent on the granularity of an output primitive's geometry. A less computationally expensive shading method may be adequate for a primitive with relatively small facets, but may produce poor results for a primitive with larger facets. To achieve a high quality image the application may need to specify a more computationally expensive shading method or modify geometry so that a given shading method will produce the desired results.

4.9.4 Normal-vector interpolation

Interpolation of normal vectors may be carried out by interpolating the Cartesian coordinates of the vector rather than the spherical coordinates. This may lead to unexpected results if the interpolated normals are pointing in strongly diverging directions. An extreme example is the Cartesian linear interpolation of two nearly opposed vectors. The rate at which the normalized result sweeps the great circle containing the two end points of the vectors is extremely different between the spherical coordinate interpolation (constant rate) and the Cartesian coordinate interpolation.

4.9.5 Effects of transformations

Transformations affect the appearance of an image as follows: non-rigid transformations change the overall perceived shapes of objects, rigid transformations change the relative positions of objects and observer. Non-rigid transformations can occur in the modelling transform and in the viewing transform (the transform from world coordinates to normalized projection coordinates). Non-rigid modelling transformations are likely to produce anomalies in the lighting and shading of output primitives; PHIGS PLUS therefore leaves the effects of non-rigid transformations on lighting and shading undefined.

Modelling matrices that do not preserve angles are allowed in the PHIGS transformation pipeline. The application of these matrices, however, may invalidate the normals specified with primitives, and, since these normals are used in the lighting and shading calculations, can cause shading anomalies. For a primitive that does not have explicit normals, a PHIGS PLUS implementation may avoid this problem by computing the normals in world coordinates after applying the modelling transformation. The effect of non-angle-preserving modelling matrices on shading is workstation dependent.

The orientation of facets is determined by the Z-component of the facet normal in NPC. This normal can be supplied by the application and can be non-orthogonal to the plane of the facet. If FACET DISTINGUISHING MODE has the value ON and if different attributes are applied to front- and back-facing facets, then the appearance of a facet will change when the Z-component of such a supplied normal crosses 0 even though the eye point does not lie in the plane of the facet.

For some curve and surface approximation criteria, modeling or viewing transformations may lead to a change in the approximation of parametric curves and surfaces. In such a case coarse approximation criteria is likely to lead to geometric or shading instabilities as the parametric surface is transformed.

4.9.6 Approximation criteria and data splines

A poor choice of approximation criteria may cause under-sampling of colour or data splines associated with parametric curves and surfaces. Some approximation criteria types base their sampling frequency solely on variations in geometry and do not consider any associated colour or data. If the geometry is less widely varying than the colour or data, then these approximation criteria types may under-sample the colour or data splines, missing their variations. An example of this is applying curve approximation criteria type 6 (CHORDAL DEVIATION IN WC) to a relatively flat curve with a wavy colour distribution. Since CHORDAL DEVIATION IN WC approximation bases the sample frequency on the deviation of the approximate geometry from the exact geometry, it will compute a low sample frequency for the curve, maybe only two sample points if the curve is very flat. Since the colour is only determined from the colour spline at the sample points, this low frequency sampling will not capture the variation in the wavy colour spline.

Successfully capturing the information in colour and data splines requires that appropriate approximation criteria be used. Under-sampling can be avoided by specifying approximation criteria that samples the geometry at a high enough frequency to capture the variations in all the relevant data. An approximation type useful for this is one of the CHORDAL SIZE types. These give explicit control over the minimum sample size. To the limits of the workstation's capabilities they will sample all the splines to at least the desired frequency when given a sufficiently small approximation value.

4.10 Minimum support criteria

Capabilities are expressed in PHIGS PLUS by functions and ranges of parameters of those functions. It is recognized that not all implementations will be able to support all possible capabilities, so minimum support criteria are defined.

There are three different types of capability

- An explicitly defined and required capability. Every PHIGS PLUS implementation supports the capability.
- An explicitly defined and non-required capability. A PHIGS PLUS implementation may support the capability and, if it does, it is implemented according to the explicit definition in PHIGS PLUS.
- A conceptually defined and non-required capability. A PHIGS PLUS implementation may provide the capability. Its implementation follows general rules given by PHIGS PLUS.

ISO/IEC 9592-1 defines minimum support criteria in 4.14. All the minimum support criteria defined there is required by this part of ISO/IEC 9592 except where an explicit minimum requirement specified in this part is stricter.

The set of explicitly defined and required capabilities includes:

a) all explicitly defined and required capabilities of ISO/IEC 9592-1;
b) interior style SOLID;
c) directly specified colour;
d) colour types INDIRECT, RGB and CIELUV;
e) facet culling and facet distinguishing;
f) polyline shading methods 1 and 2;
g) interior shading methods 1 and 2;
h) data mapping method 1;
i) reflectance properties type 1;
j) reflectance models 1 to 4;
k) light source types 1 to 4 for workstation light sources;
l) parametric surface characteristics types 1, 2 and 3;
m) curve approximation criteria types 1 to 8;
n) surface approximation criteria types 1 to 8;
o) trimming curve approximation criteria types 1 and 2;
p) rendering colour model 0;
q) colour mapping type TRUE.
r) depth cue mode SUPPRESSED

The set of explicitly defined and non-required capabilities includes:

s) all explicitly defined and non-required capabilities of ISO/IEC 9592-1 that are not required by this part;
t) interior shading methods 3 to 5;
u) data mapping methods 2 to 5;
v) curve approximation criteria types 9, 10 and 11;
w) surface approximation criteria types 9, 10 and 11;
x) parametric surface characteristics types 4 and 5;

y) rendering colour models 1 to 4;

z) depth cue mode ALLOWED;

aa) colour mapping types PSEUDO and PSEUDO-N;

ab) the strict interpretation of the fields in the light source data records;

ac) The application of lighting to cell arrays.

The set of conceptually defined and non-required capabilities includes:

ad) all conceptually defined and non-required capabilities of ISO/IEC 9592-1 that are not required by this part;

ae) polyline shading methods beyond 2;

af) interior shading methods beyond 5;

ag) data mappings methods beyond 5;

ah) reflectance properties types beyond 1;

ai) reflectance models beyond 4;

aj) light source types beyond 4 for workstation light sources;

ak) curve approximation criteria types beyond 11;

al) surface approximation criteria types beyond 11;

am) parametric surface characteristics types beyond 5;

an) shading for HATCH interior style;

ao) lighting and shading for PATTERN interior style;

ap) rendering colour models beyond 4;

aq) depth cue effects.

ar) colour mapping types beyond 3.

Table 9 lists minimum quantities for various functionality defined in PHIGS PLUS.

Table 9 — Minimum required support

Item	Minimum
directly specifiable colour models	2
rendering colour models	1
polyline shading methods	2
interior styles[1]	2
data mapping methods[1]	1
interior shading methods[1]	4
reflectance model values[1]	4
maximum non-uniform B-spline curve order	6
curve approximation criteria types	8
maximum non-uniform B-spline surface order	6
surface approximation criteria types	8
trimming curve approximation criteria types	2
parametric surface characteristics types	3
maximum trimming curve order	4
light source types for workstation light sources	4
simultaneously active non-ambient light sources	1
colour mapping methods	1
predefined data mapping bundles	1
modifiable data mapping bundles	1
predefined reflectance bundles	1
modifiable reflectance bundles	20
predefined parametric surface bundles	1
modifiable parametric surface bundles	20
predefined depth cue table entries	1
modifiable depth cue table entries	6
predefined light source table entries	2
modifiable light source table entries	5
predefined colour mapping entries	1
modifiable colour mapping table entries	1

1) Available for both back and front aspects.

5 PHIGS PLUS Functional Specification

5.1 Notational Conventions

The heading of each function specifies:

a) The function's name;

b) The PHIGS operating states in which the function may be used. A quadruple representing each of the four PHIGS operating states is listed in the following form:

(<system state>,<workstation state>,<structure state>,<archive state>)

Each entry in the quadruple has two possible values, as described in ISO/IEC 9592-1, and in addition the value "*" indicating that that particular PHIGS operating state has no effect on the function.

The parameter lists indicate for each entry:

c) whether the entry is an input (In) or output (Out) parameter;

d) the name of the parameter;

e) for coordinate data, the coordinate system (TLC,MC,WC,NPC,DC) used in the function call (Coordinate systems are explained in ISO/IEC 9592-1.);

f) either the permitted values for enumeration type data, or any restriction on the value range for real and integer data, for example "> 0"; (The notation is explained in clause 6 of ISO/IEC 9592-1.)

g) the data type itself. PHIGS data types are used as well as new data types introduced by PHIGS PLUS; (The notation for PHIGS PLUS data types is explained in 6.2.)

h) whether the parameter is optional. Some PHIGS PLUS primitives take a variable set of parameters. Optional parameters or portions of parameters are bracketed with "{}". Bracketed entries may be nested.

All error states for a given function are identified.

5.2 Output primitive functions

POLYLINE SET 3 WITH COLOUR **(PHOP,*,STOP,*)**

Parameters:

{In	colour type		I}
In	vertex data	MC{,}	L{L(P3{,COLRV}))

Effect: This function fully specifies the polyline set with colour primitive. Depending on the 'edit mode', a "polyline set 3 with colour" element is inserted into the open structure after the 'element pointer' or replaces the element pointed at by the 'element pointer'. The 'element pointer' is then updated to point to this "polyline set 3 with colour" structure element. The values specified by the parameters are associated with the element.

The information specified with each vertex comprises the vertex coordinates and optionally the vertex colours. If vertex colours are specified, they are specified for all vertices. If vertex colours are specified, a colour type shall also be specified for the primitive.

When an element of this type is interpreted, a set of unconnected polylines is generated. One polyline is defined by each list of vertex data. The current values of the polyline set with colour attributes, as defined in the traversal state list, are bound to the primitive. The polyline set with colour attributes are defined in 4.5.4.2.

A "polyline set 3 with colour" element containing no vertex data lists will be placed in the open structure. When such an element is interpreted it will have no visual effect. A "polyline set 3 with colour" element containing one or more vertex data lists with less than two vertices will be placed in the open structure. When such an element is interpreted the lists with insufficient data will have no visual effect.

References: 4.4.2, 4.4.15, 4.5.1, 4.5.4, 4.5.5, 4.5.9, 4.6.1, 4.6.4.6, 4.9

Errors:

005 Ignoring function, function requires state (PHOP,,STOP,*)*

113 Ignoring function, the colour index value is less than zero

FILL AREA SET 3 WITH DATA **(PHOP,*,STOP,*)**

{In	colour type		I}
{In	facet data	{}{,MC}{,}	{COLRV}{,NORM}{,L(R)}}
{In	edge visibility flags	(OFF,ON)	L(L(E))}
In	vertex data	MC{,}{,MC}{,}	L(L(P3{,COLRV}{,NORM}{,L(R))})

Effect: This function fully specifies the three dimensional form of the fill area set with data primitive. Depending on the 'edit mode', a "fill area set 3 with data" element is inserted into the open structure after the 'element pointer' or replaces the element pointed at by the 'element pointer'. The 'element pointer' is then updated to point to this "fill area set 3 with data" structure element. The values specified by the parameters are associated with the element.

The facet data applies to the whole fill area set and optionally comprises facet colour, facet normal and facet application-specific data.

The information specified with each vertex comprises the vertex coordinates, and optionally the vertex colours, the vertex normals and the vertex application-specific data. If vertex colours are specified, they are specified for all vertices. The same applies to vertex normals and to vertex application-specific data.

If facet colour or vertex colours are specified, a colour type shall also be specified.

If vertex application-specific data are specified, all data lists shall contain the same number of entries.

When an element of this type is interpreted, a set of polygonal areas made up of a collection of subareas is generated. A subarea is defined by a single list of vertex data. Each subarea is implicitly closed. The current values of the fill area set with data attributes, as defined in the traversal state list, are bound to the primitive. The fill area set with data attributes are defined in 4.5.4.3.

A "fill area set 3 with data" element containing no lists of vertex data or one or more subareas with less than three vertices will be placed in the open structure. When such an element is encountered it will have no visual effect. A "fill area set 3 with data" element containing non-coplanar vertex coordinates is treated in an workstation dependent manner. The effect of non-unit length facet or vertex normals is workstation dependent.

Each edge visibility flag corresponds to an entry in the lists of vertex indices and controls the visibility of the edge from that vertex to its successor in the corresponding polygon boundary. The last entry of an edge visibility list controls the visibility of the edge between the last and first vertices of the corresponding polygon boundary. If edge visibility flags are specified an edge will only be displayed when the edge flag aspect is ON and the edge visibility flag is also ON.

References: 4.4.3, 4.4.14, 4.4.15, 4.5.1, 4.5.4, 4.5.5, 4.5.9, 4.6, 4.9

Errors:

005 Ignoring function, function requires state (PHOP,,STOP,*)*

113 Ignoring function, the colour index value is less than zero

513 Ignoring function, the number of edge visibility flags is inconsistent with the number of edges in the primitive

514 Ignoring function, the data lists do not all contain the same number of entries

FILL AREA SET WITH DATA **(PHOP,*,STOP,*)**

{In	colour type		I}
{In	facet data	{}{,MC}{,}	{COLRV}{,NORM}{,L(R)}}
{In	edge visibility flags	(OFF,ON)	L(L(E))}
In	vertex data	MC{,}{,MC}{,}	L(L(P2{,COLRV}{,NORM}{,L(R)}))

Effect: This function specifies the two dimensional shorthand form of the fill area set with data primitive. All vertices of this form of the primitive are assumed to have Z coordinate values of 0. Normal vectors for this form are three dimensional. Depending on the 'edit mode', a "fill area set with data" element is inserted into the open structure after the 'element pointer' or replaces the element pointed at by the 'element pointer'. The 'element pointer' is then updated to point to this "fill area set with data" structure element. The values specified by the parameters are associated with the element.

The facet data applies to the whole fill area set and optionally comprises facet colour, facet normal and facet application-specific data.

The information specified with each vertex comprises the vertex coordinates, and optionally the vertex colours, the vertex normals and the vertex application-specific data. If vertex colours are specified, they are specified for all vertices. The same applies to vertex normals and to vertex application-specific data.

If facet colour or vertex colours are specified, a colour type shall also be specified.

If vertex application-specific data are specified, all data lists shall contain the same number of entries.

When an element of this type is interpreted, a set of polygonal areas made up of a collection of subareas is generated. A subarea is defined by a single list of vertex data. Each subarea is implicitly closed. The current values of the fill area set with data attributes, as defined in the traversal state list, are bound to the primitive. The fill area set with data attributes are defined in 4.5.4.3.

A "fill area set with data" element containing no lists of vertex data or one or more subareas with less than three vertices will be placed in the open structure. When such an element is encountered it will have no visual effect. The effect of non-unit length facet or vertex normals is workstation dependent.

Each edge visibility flag corresponds to an entry in the lists of vertex indices and controls the visibility of the edge from that vertex to its successor in the corresponding polygon boundary. The last entry of an edge visibility list controls the visibility of the edge between the last and first vertices of the corresponding polygon boundary. If edge visibility flags are specified an edge will only be displayed when the edge flag aspect is ON and the edge visibility flag is also ON.

References: 4.4.3, 4.4.14, 4.4.15, 4.5.1, 4.5.4, 4.5.5, 4.5.9, 4.6, 4.9

Errors:

005 *Ignoring function, function requires state (PHOP,*,STOP,*)*

113 *Ignoring function, the colour index value is less than zero*

513 *Ignoring function, the number of edge visibility flags is inconsistent with the number of edges in the primitive*

514 *Ignoring function, the data lists do not all contain the same number of entries*

CELL ARRAY 3 PLUS **(PHOP,*,STOP,*)**

In	parallelogram P,Q,R	MC	3×P3
In	colour type		I
In	colour array		A(COLRV)

Effect: This function fully specifies the cell array PLUS primitive. Depending on the 'edit mode', a "cell array 3 PLUS" element is inserted into the open structure after the 'element pointer' or replaces the element pointed at by the

'element pointer'. The 'element pointer' is then updated to point to this "cell array 3 PLUS" structure element. The values specified by the parameters are associated with the element.

This form of the cell array primitive is based on the general colour type and allows direct specification of cell colours.

When an element of this type is interpreted, a cell array is generated using the element's cell parallelogram corners, the number of cells along each axis and the element's colour array. The current values of the cell array PLUS attributes, as defined in the traversal state list, are bound to the primitive. The cell array PLUS attributes are defined in 4.5.4.4.

The minimal simulation required is to draw the transformed boundaries of the element's cell parallelogram, using implementation dependent colour, linewidth and linetype.

A "cell array 3 PLUS" element containing an empty array of colours will be placed in the open structure. When such an element is encountered it will have no visual effect.

References: 4.4.4, 4.4.14, 4.5.1, 4.5.4, 4.5.5, 4.5.9, 4.6, 4.9

Errors:

005 *Ignoring function, function requires state (PHOP,*,STOP,*)*

117 *Ignoring function, one of the dimensions of the colour index array is less than zero*

113 *Ignoring function, the colour index value is less than zero*

SET OF FILL AREA SETS 3 WITH DATA **(PHOP,*,STOP,*)**

{In	colour type		I}
{In	facet data	{}{,MC}{,}	L({COLRV}{,NORM}{,L(R)})}
{In	edge visibility flags	(OFF,ON)	L(L(L(E)))}
In	vertex data	MC{,}{,MC}{,}	L(P3{,COLRV}{,NORM}{,L(R)})
In	vertex indices	(1...n)	L(L(L(I)))

Effect: This function fully specifies the three dimensional form of the set of fill area sets with data primitive. Depending on the 'edit mode', a "set of fill area sets 3 with data" element is inserted into the open structure after the 'element pointer' or replaces the element pointed at by the 'element pointer'. The 'element pointer' is then updated to point to this "set of fill area sets 3 with data" structure element. The values specified by the parameters are associated with the element.

The information supplied for each facet (each fill area set) optionally comprises the facet colour, facet normal and facet application-specific data. If facet colours are specified, they are required for all facets. The same applies to facet normals and to facet application-specific data.

The information supplied for each vertex comprises the vertex coordinates and optionally the vertex colour, vertex normal and the vertex application-specific data. If vertex colours are specified, they are required for all vertices. The same applies to vertex normals and to vertex application-specific data.

If facet colours or vertex colours are specified, a colour type shall also be specified.

If facet application-specific data are specified, all its data lists shall contain the same number of entries. If vertex application-specific data are specified, all its data lists shall contain the same number of entries.

When an element of this type is interpreted, a number of (possibly non-coplanar) fill area sets are generated. The data for each vertex is specified, just once, in the vertex data list. An index value of one selects the first vertex in the vertex data list, a value of two selects the second and so on. The vertices that define the polygons of each fill area set are selected via a list of lists of indices. Each list of indices determines a polygon boundary, which is implicitly closed to define an area. The current values of the set of fill area set with data attributes, as defined in the traversal state list, are bound to the primitive. The set of fill area set with data attributes are defined in 4.5.4.5.

A "set of fill area sets 3 with data" element containing a polygon boundary with less than three points is placed in the open structure. When such an element is interpreted the fill area set containing the insufficient data will have no visual effect. Non-planar fill area sets are treated in an workstation dependent manner. The effect of non-unit length facet or vertex normals is workstation dependent.

Each edge visibility flag corresponds to an entry in the lists of vertex indices and controls the visibility of the edge from that vertex to its successor in the corresponding polygon boundary. The last entry of an edge visibility list controls the visibility of the edge between the last and first vertices of the corresponding polygon boundary. If edge visibility flags are specified an edge will only be displayed when the edge flag aspect is ON and the edge visibility flag (or, where multiple facets share an edge, any of the edge visibility flags) for the corresponding edge is also set to ON.

References: 4.4.5, 4.4.14, 4.4.15, 4.5.1, 4.5.4, 4.5.5, 4.5.9, 4.6, 4.9

Errors:

005 *Ignoring function, function requires state (PHOP,*,STOP,*)*

113 *Ignoring function, the colour index value is less than zero*

504 *Ignoring function, one or more of the vertex indices is out of range*

513 *Ignoring function, the number of edge visibility flags is inconsistent with the number of edges in the primitive*

514 *Ignoring function, the data lists do not all contain the same number of entries*

516 *Ignoring function, the facet data is inconsistent with the number of facets in the primitive*

SET OF FILL AREA SETS WITH DATA **(PHOP,*,STOP,*)**

{In	colour type		I}
{In	facet data	{}{,MC}{,}	L({COLRV}{,NORM}{,L(R)})}
{In	edge visibility flags	(OFF,ON)	L(L(L(E)))}
In	vertex data	MC{,}{,MC}{,}	L(P2{,COLRV}{,NORM}{,L(R)})
In	vertex indices	(1...n)	L(L(L(I)))

Effect: This function specifies the two dimensional shorthand form of the set of fill area sets with data primitive. All vertices of the this form of the primitive are assumed to have Z coordinate values of 0. Normal vectors for this form are three dimensional. Depending on the 'edit mode', a "set of fill area sets with data" element is inserted into the open structure after the 'element pointer' or replaces the element pointed at by the 'element pointer'. The 'element pointer' is then updated to point to this "set of fill area sets with data" structure element. The values specified by the parameters are associated with the element.

The information supplied for each facet (each fill area set) optionally comprises the facet colour, facet normal and facet application-specific data. If facet colours are specified, they are required for all facets. The same applies to facet normals and to facet application-specific data.

The information supplied for each vertex comprises the vertex coordinates and optionally the vertex colour, vertex normal and the vertex application-specific data. If vertex colours are specified, they are required for all vertices. The same applies to vertex normals and to vertex application-specific data.

If facet colours or vertex colours are specified, a colour type shall also be specified.

If facet application-specific data are specified, all its data lists shall contain the same number of entries. If vertex application-specific data are specified, all its data lists shall contain the same number of entries.

When an element of this type is interpreted, a number of fill area sets are generated. The data for each vertex is specified, just once, in the vertex data list. An index value of one selects the first vertex in the vertex data list, a value of two selects the second and so on. The vertices that define the polygons of each fill area set are selected via a list of lists of indices. Each list of indices determines a polygon boundary, which is implicitly closed to

define an area. The current values of the set of fill area sets with data attributes, as defined in the traversal state list, are bound to the primitive. The set of fill area set with data attributes are defined in 4.5.4.5.

A "set of fill area sets with data" element containing a polygon boundary with less than three points is placed in the open structure. When such an element is interpreted the fill area set containing the insufficient data will have no visual effect. The effect of non-unit length facet or vertex normals is workstation dependent.

Each edge visibility flag corresponds to an entry in the lists of vertex indices and controls the visibility of the edge from that vertex to its successor in the corresponding polygon boundary. The last entry of an edge visibility list controls the visibility of the edge between the last and first vertices of the corresponding polygon boundary. If edge visibility flags are specified an edge will only be displayed when the edge flag aspect is ON and the edge visibility flag for (or, where multiple facets share an edge, any of the edge visibility flags) the corresponding edge is also set to ON.

References: 4.4.5, 4.4.14, 4.4.15, 4.5.1, 4.5.4, 4.5.5, 4.5.9, 4.6, 4.9

Errors:

005 *Ignoring function, function requires state (PHOP,*,STOP,*)*

113 *Ignoring function, the colour index value is less than zero*

504 *Ignoring function, one or more of the vertex indices is out of range*

513 *Ignoring function, the number of edge visibility flags is inconsistent with the number of edges in the primitive*

514 *Ignoring function, the data lists do not all contain the same number of entries*

516 *Ignoring function, the facet data is inconsistent with the number of facets in the primitive*

TRIANGLE SET 3 WITH DATA **(PHOP,*,STOP,*)**

{In	colour type		I}
{In	facet data	{}{,MC}{,}	L({COLRV}{,NORM}{,L(R)})}
{In	edge visibility flags	(OFF,ON)	L(3×E)}
In	vertex data	MC{,}{,MC}{,}	L(P3{,COLRV}{,NORM}{,L(R)})
In	vertex indices	(1...n)	L(3×I)

Effect: This function fully specifies the three dimensional form of the triangle set with data primitive. Depending on the 'edit mode', a "triangle set 3 with data" element is inserted into the open structure after the 'element pointer' or replaces the element pointed at by the 'element pointer'. The 'element pointer' is then updated to point to this "triangle set 3 with data" structure element. The values specified by the parameters are associated with the element.

The information supplied for each triangle of the triangle set optionally comprises facet colours, facet normals and facet application-specific data. If facet colours are specified, they are specified for all facets. The same applies to facet normals and to facet application-specific data.

The information specified with each vertex comprises the vertex coordinates, and optionally the vertex colours, the vertex normals and the vertex application-specific data. If vertex colours are specified, they are specified for all vertices. The same applies to vertex normals and to vertex application-specific data.

If facet colours or vertex colours are specified, a colour type shall also be specified.

If facet application-specific data are specified, all its data lists shall contain the same number of entries. If vertex application-specific data are specified, all its data lists shall contain the same number of entries.

When an element of this type is interpreted, a set of triangles is generated. The data for each vertex is specified, just once, in the vertex data list. The vertices in the vertex list may be used zero, one or more times. An index value of one selects the first vertex in the vertex data list, a value of two selects the second and so on. The

vertices that define the triangles are selected via a list of triples of indices. Each triple determines a triangle. The triangle set with data attributes are defined in 4.5.4.6.

A "triangle set 3 with data" element containing a list of vertex data with less than three elements or an empty list of vertex indices will be placed in the open structure. When such an element is interpreted it will have no visual effect. The effect of non-unit length facet or vertex normals is workstation dependent.

The list of edge visibility flags consists of a flag for each edge of each triangle in the set. If edge visibility flags are specified there shall be a 3-tuple of them for each specified triangle. The first flag in each 3-tuple corresponds to the edge between the first and second vertices of the triangle, the second flag corresponds to the edge between the second and third vertices of the triangle, the third flag corresponds to the edge between the third and first vertices of the triangle. If edge visibility flags are specified an edge will only be displayed when the edge flag aspect is ON and the edge visibility flag (or, where multiple facets share an edge, any of the edge visibility flags) for the corresponding edge is also ON.

References: 4.4.6, 4.4.14, 4.4.15, 4.5.1, 4.5.4, 4.5.5, 4.5.9, 4.6, 4.9

Errors:

005 *Ignoring function, function requires state (PHOP,*,STOP,*)*

113 *Ignoring function, the colour index value is less than zero*

504 *Ignoring function, one or more of the vertex indices is out of range*

513 *Ignoring function, the number of edge visibility flags is inconsistent with the number of edges in the primitive*

514 *Ignoring function, the data lists do not all contain the same number of entries*

516 *Ignoring function, the facet data is inconsistent with the number of facets in the primitive*

TRIANGLE SET WITH DATA **(PHOP,*,STOP,*)**

{In	colour type		I}
{In	facet data	{}{,MC}{,}	L({COLRV}{,NORM}{,L(R)})}
{In	edge visibility flags	(OFF,ON)	L(3×E)}
In	vertex data	MC{,}{,MC}{,}	L(P2{,COLRV}{,NORM}{,L(R)})
In	vertex indices	(1...n)	L(3×I)

Effect: This function specifies the two dimensional shorthand form of the triangle set with data primitive. All vertices of the this form of the primitive are assumed to have Z coordinate values of 0. Normal vectors for this form are three dimensional. Depending on the 'edit mode', a "triangle set with data" element is inserted into the open structure after the 'element pointer' or replaces the element pointed at by the 'element pointer'. The 'element pointer' is then updated to point to this "triangle set with data" structure element. The values specified by the parameters are associated with the element.

The information supplied for each triangle of the triangle set optionally comprises facet colours, facet normals and facet application-specific data. If facet colours are specified, they are specified for all facets. The same applies to facet normals and to facet application-specific data.

The information specified with each vertex comprises the vertex coordinates, and optionally the vertex colours, the vertex normals and the vertex application-specific data. If vertex colours are specified, they are specified for all vertices. The same applies to vertex normals and to vertex application-specific data.

If facet colours or vertex colours are specified, a colour type shall also be specified.

If facet application-specific data are specified, all its data lists shall contain the same number of entries. If vertex application-specific data are specified, all its data lists shall contain the same number of entries.

When an element of this type is interpreted, a set of triangles is generated. The data for each vertex is specified, just once, in the vertex data list. The vertices in the vertex list may be used zero, one or more times. An index

value of one selects the first vertex in the vertex data list, a value of two selects the second and so on. The vertices that define the triangles are selected via a list of triples of indices. Each triple determines a triangle. The triangle set with data attributes are defined in 4.5.4.6.

A "triangle set with data" element containing a list of vertex data with less than three elements or an empty list of vertex indices will be placed in the open structure. When such an element is interpreted it will have no visual effect. The effect of non-unit length facet or vertex normals is workstation dependent.

The list of edge visibility flags consists of a flag for each edge of each triangle in the set. If edge visibility flags are specified there shall be a 3-tuple of them for each specified triangle. The first flag in each 3-tuple corresponds to the edge between the first and second vertices of the triangle, the second flag corresponds to the edge between the second and third vertices of the triangle, the third flag corresponds to the edge between the third and first vertices of the triangle. If edge visibility flags are specified an edge will only be displayed when the edge flag aspect is ON and the edge visibility flag (or, where multiple facets share an edge, any of the edge visibility flags) for the corresponding edge is also ON.

References: 4.4.6, 4.4.14, 4.4.15, 4.5.1, 4.5.4, 4.5.5, 4.5.9, 4.6, 4.9

Errors:

005 *Ignoring function, function requires state (PHOP,*,STOP,*)*

113 *Ignoring function, the colour index value is less than zero*

504 *Ignoring function, one or more of the vertex indices is out of range*

513 *Ignoring function, the number of edge visibility flags is inconsistent with the number of edges in the primitive*

514 *Ignoring function, the data lists do not all contain the same number of entries*

516 *Ignoring function, the facet data is inconsistent with the number of facets in the primitive*

TRIANGLE STRIP 3 WITH DATA **(PHOP,*,STOP,*)**

{In	colour type		I}
{In	facet data	{}{,MC}{,}	L({COLRV}{,NORM}{,L(R)})}
{In	edge visibility flags	(OFF,ON)	L(E)}
In	vertex data	MC{,}{,MC}{,}	L(P3{,COLRV}{,NORM}{,L(R)})

Effect: This function fully specifies the three dimensional form of the triangle strip with data primitive. Depending on the 'edit mode', a "triangle strip 3 with data" element is inserted into the open structure after the 'element pointer' or replaces the element pointed at by the 'element pointer'. The 'element pointer' is then updated to point to this "triangle strip 3 with data" structure element. The values specified by the parameters are associated with the element.

The information supplied for each triangle of the triangle strip optionally comprises facet colours, facet normals and facet application-specific data. If facet colours are specified, they are specified for all triangles. The same applies to facet normals and to facet application-specific data.

The information specified with each vertex comprises the vertex coordinates, and optionally the vertex colours, the vertex normals and the vertex application-specific data. If vertex colours are specified, they are specified for all vertices. The same applies to vertex normals and to vertex application-specific data.

If facet colours or vertex colours are specified, a colour type shall also be specified.

If facet application-specific data are specified, all its data lists shall contain the same number of entries. If vertex application-specific data are specified, all its data lists shall contain the same number of entries.

When an element of this type is interpreted, a strip of triangles is generated. The strip is composed of $N-2$ triangles, where N is the number of vertices. The first triangle is formed from vertices 1, 2, and 3; the second triangle from vertices 2, 3 and 4; the $(N-2)$-nd triangle from vertices $N-2$, $N-1$ and N. The current values of

the triangle strip with data attributes, as defined in the traversal state list, are bound to the primitive. The triangle strip with data attributes are defined in 4.5.4.7.

A "triangle strip 3 with data" element containing a list of vertex data with less than three elements will be placed in the open structure. When such an element is interpreted it will have no visual effect. The effect of non-unit length facet or vertex normals is workstation dependent.

The list of edge visibility flags consists of a flag for each edge of the triangle strip. If edge visibility flags are specified there shall be $2(N-2)+1$ of them. Each flag, E_i, corresponds to the edge between vertices $trunc(\frac{i-1}{2})+1$ and $trunc(\frac{i}{2})+2$, where *trunc(t)* is the integer portion of the real number t. If edge visibility flags are specified an edge will only be displayed when the edge flag aspect is ON and the edge visibility flag is also ON.

References: 4.4.7, 4.4.14, 4.4.15, 4.5.1, 4.5.4, 4.5.5, 4.5.9, 4.6, 4.9

Errors:

005 *Ignoring function, function requires state (PHOP,*,STOP,*)*

113 *Ignoring function, the colour index value is less than zero*

513 *Ignoring function, the number of edge visibility flags is inconsistent with the number of edges in the primitive*

514 *Ignoring function, the data lists do not all contain the same number of entries*

516 *Ignoring function, the facet data is inconsistent with the number of facets in the primitive*

TRIANGLE STRIP WITH DATA **(PHOP,*,STOP,*)**

{In	colour type		I}
{In	facet data	{}{,MC}{,}	L({COLRV}{,NORM}{,L(R)})}
{In	edge visibility flags	(OFF,ON)	L(E)}
In	vertex data	MC{,}{,MC}{,}	L(P2{,COLRV}{,NORM}{,L(R)})

Effect: This function specifies the two dimensional X and Y shorthand form of the triangle strip with data primitive. All vertices of the two dimensional form of this primitive are assumed to have Z coordinate values of 0. Normal vectors for this form are three dimensional. Depending on the 'edit mode', a "triangle strip with data" element is inserted into the open structure after the 'element pointer' or replaces the element pointed at by the 'element pointer'. The 'element pointer' is then updated to point to this "triangle strip with data" structure element. The values specified by the parameters are associated with the element.

The information supplied for each triangle of the triangle strip optionally comprises facet colours, facet normals and facet application-specific data. If facet colours are specified, they are specified for all triangles. The same applies to facet normals and to facet application-specific data.

The information specified with each vertex comprises the vertex coordinates, and optionally the vertex colours, the vertex normals and the vertex application-specific data. If vertex colours are specified, they are specified for all vertices. The same applies to vertex normals and to vertex application-specific data.

If facet colours or vertex colours are specified, a colour type shall also be specified.

If facet application-specific data are specified, all its data lists shall contain the same number of entries. If vertex application-specific data are specified, all its data lists shall contain the same number of entries.

When an element of this type is interpreted, a strip of triangles is generated. The strip is composed of $N-2$ triangles, where N is the number of vertices. The first triangle is formed from vertices 1, 2, and 3; the second triangle from vertices 2, 3 and 4; the $(N-2)$-nd triangle from vertices $N-2$, $N-1$ and N. The current values of the triangle strip with data attributes, as defined in the traversal state list, are bound to the primitive. The triangle strip with data attributes are defined in 4.5.4.7.

If information per facet is present there will be $N-2$ pieces of information.

A "triangle strip with data" element containing a list of vertex data with less than three elements will be placed in the open structure. When such an element is interpreted it will have no visual effect. The effect of non-unit length facet or vertex normals is workstation dependent.

The list of edge visibility flags consists of a flag for each edge of the triangle strip. If edge visibility flags are specified there shall be $2(N-2)+1$ of them. Each flag, E_i, corresponds to the edge between vertices $trunc(\frac{i-1}{2})+1$ and $trunc(\frac{i}{2})+2$, where *trunc(t)* is the integer portion of the real number *t*. If edge visibility flags are specified an edge will only be displayed when the edge flag aspect is ON and the edge visibility flag is also ON.

References: 4.4.7, 4.4.14, 4.4.15, 4.5.1, 4.5.4, 4.5.5, 4.5.9, 4.6, 4.9

Errors:

005 *Ignoring function, function requires state (PHOP,*,STOP,*)*

113 *Ignoring function, the colour index value is less than zero*

513 *Ignoring function, the number of edge visibility flags is inconsistent with the number of edges in the primitive*

514 *Ignoring function, the data lists do not all contain the same number of entries*

516 *Ignoring function, the facet data is inconsistent with the number of facets in the primitive*

QUADRILATERAL MESH 3 WITH DATA **(PHOP,*,STOP,*)**

{In	colour type		I}
{In	facet data	{}{,MC}{,}	A({COLRV}{,NORM}{,L(R)})}
{In	edge visibility flags	(OFF,ON)	A(2×E)}
In	vertex data	MC{,}{,MC}{,}	A(P3{,COLRV}{,NORM}{,L(R)})

Effect: This function fully specifies the three dimensional form of the quadrilateral mesh with data primitive. Depending on the 'edit mode', a "quadrilateral mesh 3 with data" element is inserted into the open structure after the 'element pointer' or replaces the element pointed at by the 'element pointer'. The 'element pointer' is then updated to point to this "quadrilateral mesh 3 with data" structure element. The values specified by the parameters are associated with the element.

The information supplied for each quadrilateral of the quadrilateral mesh optionally comprises facet colours, facet normals and facet application-specific data. If facet colours are specified, they are specified for all quadrilaterals. The same applies to facet normals and to facet application-specific data.

The information specified with each vertex comprises the vertex coordinates, and optionally the vertex colours, the vertex normals and the vertex application-specific data. If vertex colours are specified, they are specified for all vertices. The same applies to vertex normals and to vertex application-specific data.

If facet colours or vertex colours are specified, a colour type shall also be specified.

If facet application-specific data are specified, all its data lists shall contain the same number of entries. If vertex application-specific data are specified, all its data lists shall contain the same number of entries.

When an element of this type is interpreted, a mesh of $(M-1)$ times $(N-1)$ connected quadrilaterals is generated, where the mesh is defined by a two-dimensional $M \times N$ array of vertex data. The data in each set of four neighbouring array elements defines a quadrilateral. The current values of the quadrilateral mesh with data attributes, as defined in the traversal state list, are bound to the primitive. The quadrilateral mesh with data attributes are defined in 4.5.4.8.

The set of quadrilaterals need not be coplanar. Individual quadrilaterals in the mesh may be non-planar, in which case they are rendered in a workstation dependent way.

A "quadrilateral mesh 3 with data" element containing less than two vertex data elements in either array dimension will be placed in the open structure. When such an element is interpreted it will have no visual effect. The effect of non-unit length facet or vertex normals is workstation dependent.

The array of edge visibility flags consists of a flag for each edge of the quadrilateral mesh. The edge visibility flags, if specified, are arranged in a three-dimensional $M \times N \times 2$ array. For i ranging from 1 to M, and j ranging from 1 to N, the $(i, j, 1)$ array element specifies the edge flag for the edge between the vertices i, j and $i+1$, j, and the $(i, j, 2)$ array element specifies the edge flag for the edge between the vertices i, j and $i, j+1$. The edge flag array elements $(M, j, 1)$, where $1 \leq j \leq N$, and $(i, N, 2)$, where $1 \leq i \leq M$, are not used. If edge visibility flags are specified an edge will only be displayed when the edge flag aspect is ON and the edge visibility flag is also ON.

References: 4.4.8, 4.4.14, 4.4.15, 4.5.1, 4.5.4, 4.5.5, 4.5.9, 4.6, 4.9

Errors:

005 *Ignoring function, function requires state (PHOP,*,STOP,*)*

113 *Ignoring function, the colour index value is less than zero*

513 *Ignoring function, the number of edge visibility flags is inconsistent with the number of edges in the primitive*

514 *Ignoring function, the data lists do not all contain the same number of entries*

516 *Ignoring function, the facet data is inconsistent with the number of facets in the primitive*

QUADRILATERAL MESH WITH DATA **(PHOP,*,STOP,*)**

{In	colour type		I}
{In	facet data	{}{,MC}{,}	A({COLRV}{,NORM}{,L(R)})}
{In	edge visibility flags	(OFF,ON)	A(2×E)}
In	vertex data	MC{,}{,MC}{,}	A(P2{,COLRV}{,NORM}{,L(R)})

Effect: This function specifies the two dimensional shorthand form of the quadrilateral mesh with data primitive. All vertices of the this form of the primitive are assumed to have Z coordinate values of 0. Normal vectors for this form are three dimensional. Depending on the 'edit mode', a "quadrilateral mesh with data" element is inserted into the open structure after the 'element pointer' or replaces the element pointed at by the 'element pointer'. The 'element pointer' is then updated to point to this "quadrilateral mesh with data" structure element. The values specified by the parameters are associated with the element.

The information supplied for each quadrilateral of the quadrilateral mesh optionally comprises facet colours, facet normals and facet application-specific data. If facet colours are specified, they are specified for all quadrilaterals. The same applies to facet normals and to facet application-specific data.

The information specified with each vertex comprises the vertex coordinates, and optionally the vertex colours, the vertex normals and the vertex application-specific data. If vertex colours are specified, they are specified for all vertices. The same applies to vertex normals and to vertex application-specific data.

If facet colours or vertex colours are specified, a colour type shall also be specified.

If facet application-specific data are specified, all its data lists shall contain the same number of entries. If vertex application-specific data are specified, all its data lists shall contain the same number of entries.

When an element of this type is interpreted, a mesh of $(M - 1)$ times $(N - 1)$ connected quadrilaterals is generated, where the mesh is defined by a two-dimensional $M \times N$ array of vertex data. The data in each set of four neighbouring array elements defines a quadrilateral. The current values of the quadrilateral mesh with data attributes, as defined in the traversal state list, are bound to the primitive. The quadrilateral mesh with data attributes are defined in 4.5.4.8.

A "quadrilateral mesh with data" element containing less than two vertex data elements in either array dimension will be placed in the open structure. When such an element is interpreted it will have no visual effect. The effect of non-unit length facet or vertex normals is workstation dependent.

The array of edge visibility flags consists of a flag for each edge of the quadrilateral mesh. The edge visibility flags, if specified, are arranged in a three-dimensional $M \times N \times 2$ array. For i ranging from 1 to M, and j ranging from 1 to N, the $(i, j, 1)$ array element specifies the edge flag for the edge between the vertices i, j and $i+1$, j, and the $(i, j, 2)$ array element specifies the edge flag for the edge between the vertices i, j and $i, j+1$. The edge flag array elements $(M, j, 1)$, where $1 \leq j \leq N$, and $(i, N, 2)$, where $1 \leq i \leq M$, are not used. If edge visibility flags are specified an edge will only be displayed when the edge flag aspect is ON and the edge visibility flag is also ON.

References: 4.4.8, 4.4.14, 4.4.15, 4.5.1, 4.5.4, 4.5.5, 4.5.9, 4.6, 4.9

Errors:

005 *Ignoring function, function requires state (PHOP,*,STOP,*)*

113 *Ignoring function, the colour index value is less than zero*

513 *Ignoring function, the number of edge visibility flags is inconsistent with the number of edges in the primitive*

514 *Ignoring function, the data lists do not all contain the same number of entries*

516 *Ignoring function, the facet data is inconsistent with the number of facets in the primitive*

NON-UNIFORM B-SPLINE CURVE 3 **(PHOP,*,STOP,*)**

In	spline order	(1...n)	I
In	knots		L(R)
In	parameter range limits (tmin, tmax)		2×R
In	rationality	(RATIONAL,NON-RATIONAL)	E
In	control points	MC, $w > 0$	L(P3H) | L(P3)

Effect: This function fully specifies the non-uniform B-spline curve 3 primitive. Depending on the 'edit mode', a "non-uniform B-spline curve 3" element is inserted into the open structure after the 'element pointer' or replaces the element pointed at by the 'element pointer'. The 'element pointer' is then updated to point to this "non-uniform B-spline curve 3" structure element. The values specified by the parameters are associated with the element.

The spline order shall be a positive integer. The knots shall form a non-decreasing sequence of real numbers.

The rationality parameter may have the value RATIONAL or NON-RATIONAL. When RATIONAL is specified the control points are specified as homogeneous modelling coordinates (data type P3H), with the restriction that the fourth coordinate be greater than zero. When NON-RATIONAL is specified the control points are non-homogeneous PHIGS modelling coordinates (data type P3).

The number of control points shall be at least as large as the order. The number of control points plus the spline order shall equal the number of knots.

The parameter range limits, *tmin* and *tmax*, specify over what range the B-spline curve is generated. *tmin* shall be less than *tmax*. *tmin* shall be greater than or equal to the *order*-th knot value. *tmax* shall be less than or equal to the $(k + 1 - order)$-th knot value (where k is the number of knots).

When an element of this type is interpreted, a non-uniform B-spline curve is generated for the parameter values between *tmin* and *tmax*. No curve is generated if the curve order is not supported. The current values of the non-uniform B-spline curve attributes, as defined in the traversal state list, are bound to the primitive. The non-uniform B-spline curve attributes are defined in 4.5.4.9.

References: 4.4.9, 4.5.4, 4.5.5, 4.5.9, 4.6, 4.8, 4.9

Errors:

005 *Ignoring function, function requires state (PHOP,*,STOP,*)*

500 *Ignoring function, the order of a spline is less than one*

501 *Ignoring function, a spline does not contain enough control points for its specified order*

502 *Ignoring function, the order of a spline is inconsistent with its number of knots and control points*

503 *Ignoring function, the knot sequence for a spline not non-decreasing*

506 *Ignoring function, the parameter range for a spline is inconsistent with its knots*

507 *Ignoring function, the fourth coordinate of a rational control point is less than or equal to zero*

NON-UNIFORM B-SPLINE CURVE 3 WITH COLOUR **(PHOP,*,STOP,*)**

In	spline order	(1...n)	I
In	knots		L(R)
In	parameter range limits (tmin, tmax)		2×R
In	rationality	(RATIONAL,NON-RATIONAL)	E
In	control points	MC, $w > 0$	L(P3H) | L(P3)
{In	colour spline		COLOURSPLINE}

Where COLOURSPLINE is:

spline order	(1...n)	I
knots		L(R)
rationality	(RATIONAL,NON-RATIONAL)	E
colour type	(-n...-1,1...n)	I
control points	$w > 0$	L(COLRVH) | L(COLRV)

Effect: This function fully specifies the non-uniform B-spline curve 3 with colour primitive. Depending on the 'edit mode', a "non-uniform B-spline curve 3 with colour" element is inserted into the open structure after the 'element pointer' or replaces the element pointed at by the 'element pointer'. The 'element pointer' is then updated to point to this "non-uniform B-spline curve 3 with colour" structure element. The values specified by the parameters are associated with the element.

The spline order shall be a positive integer. The knots shall form a non-decreasing sequence of real numbers.

The rationality parameter may have the value RATIONAL or NON-RATIONAL. When RATIONAL is specified the control points are specified as homogeneous modelling coordinates (data type P3H), with the restriction that the fourth coordinate be greater than zero. When NON-RATIONAL is specified the control points are non-homogeneous PHIGS modelling coordinates (data type P3).

The number of control points shall be at least as large as the order. The number of control points plus the spline order shall equal the number of knots.

The parameter range limits, *tmin* and *tmax*, specify over what range the B-spline curve is generated. *tmin* shall be less than *tmax*. *tmin* shall be greater than or equal to the *order*-th knot value. *tmax* shall be less than or equal to the $(k + 1 - order)$-th knot value (where k is the number of knots).

The colour spline rationality parameter may have the value RATIONAL or NON-RATIONAL. When RATIONAL is specified the dimension of the control points is one more than the dimension corresponding to the colour model, with the restriction that the homogeneous (last) coordinate be greater than zero. When NON-RATIONAL is specified the dimension of the control points is the same as the dimension corresponding to the colour model. The number of colour spline control points shall be at least as large as the colour spline order. The number of colour spline control points plus the colour spline order shall equal the number of colour spline knots.

The colour spline need not have the same order, knots, rationality, or control points as the geometry spline. The colour spline order shall be a positive integer and the colour spline knots shall form a non-decreasing sequence

of real numbers. The parameter range of the colour spline shall wholly include the parameter range limits of the geometry spline. The values of the colour spline control point coordinates may be outside of the range normally associated with the colour type. The colour type shall be other than INDIRECT.

When an element of this type is interpreted, a non-uniform B-spline curve with data is generated for the parameter values between *tmin* and *tmax*.

No curve is displayed if the curve order of the geometry or colour spline is not supported, or if the colour type is not available. The current values of the non-uniform B-spline curve with colour attributes, as defined in the traversal state list, are bound to the primitive. The non-uniform B-spline curve with colour attributes are defined in 4.5.4.10.

References: 4.4.10, 4.5.1, 4.5.4, 4.5.5, 4.5.9, 4.6, 4.8, 4.9

Errors:

005 *Ignoring function, function requires state (PHOP,*,STOP,*)*

500 *Ignoring function, the order of a spline is less than one*

501 *Ignoring function, a spline does not contain enough control points for its specified order*

502 *Ignoring function, the order of a spline is inconsistent with its number of knots and control points*

503 *Ignoring function, the knot sequence for a spline not non-decreasing*

506 *Ignoring function, the parameter range for a spline is inconsistent with its knots*

507 *Ignoring function, the fourth coordinate of a rational control point is less than or equal to zero*

517 *Ignoring function, the parameter range of the colour spline does not include the effective parameter range of the geometry spline*

NON-UNIFORM B-SPLINE SURFACE 3 **(PHOP,*,STOP,*)**

In	u spline order	(1...n)	I
In	v spline order	(1...n)	I
In	u knots		L(R)
In	v knots		L(R)
In	rationality	(RATIONAL,NON-RATIONAL)	E
In	control points	MC, $w > 0$	A(P3H) | A(P3)
{In	trimming loop definitions		L(L(TRIMCURVE))}

Where TRIMCURVE is:

trimming curve approximation criteria type		I
trimming curve approximation criteria data record		D
curve visibility flag	(OFF,ON)	E
curve order	(2...n)	I
curve knot vector		L(R)
curve parameter range limits (tmin, tmax)		2×R
rationality	(RATIONAL,NON-RATIONAL)	E
curve control points		L(P2H) | L(P2)

Effect: This function fully specifies the non-uniform B-spline surface 3 primitive. Depending on the 'edit mode', a "non-uniform B-spline surface 3" element is inserted into the open structure after the 'element pointer' or replaces the element pointed at by the 'element pointer'. The 'element pointer' is then updated to point to this "non-uniform B-spline surface 3" structure element. The values specified by the parameters are associated with the element.

The surface u and v orders shall be positive integers. The u knots and the v knots shall each form a non-decreasing sequence of real numbers.

The rationality parameter may have the enumerated value RATIONAL or NON-RATIONAL. When RATIONAL is specified the control points shall be specified as homogeneous modelling coordinates (data type P3H), with the restriction that the fourth coordinate be greater than zero. When NON-RATIONAL is specified the control points are non-homogeneous PHIGS modelling coordinates (data type P3).

The number of control points in each of the u and v directions shall be at least as large as the corresponding order. The number of control points in each direction plus the corresponding order shall equal the corresponding number of knots.

Trimming curve approximation criteria types are a subset of the curve approximation criteria types. Those applicable to trimming curves are:

≤ 0	implementation dependent
1	WORKSTATION DEPENDENT
2	CONSTANT PARAMETRIC SUBDIVISION BETWEEN KNOTS
3-11	not used
≥ 12	reserved for registration or future standardisation

See SET CURVE APPROXIMATION CRITERIA for the description of the data records.

If a specified trimming curve approximation criteria type is not available on a workstation, trimming curve approximation criteria type 1 is used.

When an element of this type is interpreted, a non-uniform B-spline surface is generated. No surface is generated if either of the surface orders or any of the trimming curve orders are not supported. The current values of the non-uniform B-spline surface attributes, as defined in the traversal state list, are bound to the primitive. The non-uniform B-spline surface attributes are defined in 4.5.4.11.

NOTE: Trimming curve approximation criteria types are registered in the ISO International Register of Graphical Items (4.1.2). When a trimming curve approximation criteria type has been approved by ISO/IEC, the trimming curve approximation criteria type value will be assigned by the Registration Authority.

References: 4.4.11, 4.4.12 4.4.14, 4.5.4, 4.5.5, 4.5.9, 4.6, 4.8, 4.9

Errors:

005 *Ignoring function, function requires state (PHOP,*,STOP,*)*

500 *Ignoring function, the order of a spline is less than one*

501 *Ignoring function, a spline does not contain enough control points for its specified order*

502 *Ignoring function, the order of a spline is inconsistent with its number of knots and control points*

503 *Ignoring function, the knot sequence for a spline not non-decreasing*

507 *Ignoring function, the fourth coordinate of a rational control point is less than or equal to zero*

508 *Ignoring function, a trimming curve's order is less than two*

138 *Ignoring function, one or more of the fields within the specified data record is in error*

NON-UNIFORM B-SPLINE SURFACE 3 WITH DATA **(PHOP,*,STOP,*)**

In	u spline order	(1...n)	I
In	v spline order	(1...n)	I
In	u knots		L(R)
In	v knots		L(R)
In	rationality	(RATIONAL,NON-RATIONAL)	E
In	control points	MC, $w > 0$	A(P3H) | A(P3)
{In	trimming loop definitions		L(L(TRIMCURVE))}
{In	colour spline		COLOURSPLINE}
{In	data splines		L(DATASPLINE)}

Where TRIMCURVE is:

trimming curve approximation criteria type		I
trimming curve approximation criteria data record		D
curve visibility flag	(OFF,ON)	E
curve order	(2...n)	I
curve knot vector		L(R)
curve parameter range limits	tmin, tmax	2×R
rationality	(RATIONAL,NON-RATIONAL)	E
curve control points		L(P2H) | L(P2)

Where COLOURSPLINE is:

u spline order	(1...n)	I
v spline order	(1...n)	I
u knots		L(R)
v knots		L(R)
rationality	(RATIONAL,NON-RATIONAL)	E
colour type	I ≠ 0	
control points	$w > 0$	A(COLRVH) | A(COLRV)

Where DATASPLINE is:

u spline order	(1...n)	I
v spline order	(1...n)	I
u knots		L(R)
v knots		L(R)
rationality	(RATIONAL,NON-RATIONAL)	E
control points	$w > 0$	A(DATAH) | A(DATA)

DATA = $(d_1, d_2, \ldots, d_n)$

DATAH = $(wd_1, wd_2, \ldots, wd_n, w)$

Effect: This function fully specifies the non-uniform B-spline surface 3 with data primitive. Depending on the 'edit mode', a "non-uniform B-spline surface 3 with data" element is inserted into the open structure after the 'element pointer' or replaces the element pointed at by the 'element pointer'. The 'element pointer' is then updated to point to this "non-uniform B-spline surface 3 with data" structure element. The values specified by the parameters are associated with the element.

The surface u and v orders shall be positive integers. The u knots and the v knots shall each form a non-decreasing sequence of real numbers.

The rationality parameter may have the enumerated value RATIONAL or NON-RATIONAL. When RATIONAL is specified the control points shall be specified as homogeneous modelling coordinates (data type P3H), with the restriction that the fourth coordinate be greater than zero. When NON-RATIONAL is specified the control points are non-homogeneous PHIGS modelling coordinates (data type P3).

The number of control points in each of the u and v directions shall be at least as large as the corresponding order. The number of control points in each direction plus the corresponding order shall equal the corresponding number of knots.

Trimming curve approximation criteria types are a subset of the curve approximation criteria types. Those applicable to trimming curves are:

≤ 0	implementation dependent
1	WORKSTATION DEPENDENT
2	CONSTANT PARAMETRIC SUBDIVISION BETWEEN KNOTS
3-11	not used
≥ 12	reserved for registration or future standardisation

See SET CURVE APPROXIMATION CRITERIA for the description of the data records.

If a specified trimming curve approximation criteria type is not available on a workstation, trimming curve approximation criteria type 1 is used.

The colour and data splines need not have the same order, knots, rationality, or control points as the geometry spline. The order of colour and data splines shall be a positive integer, and their knots shall form a non-decreasing sequence of real numbers. The parameter range of the colour and data splines shall wholly contain the parameter range of the geometry spline, or, if trimming loops are specified with the surface, the parameter range of the colour and data splines shall wholly contain the parameter range defined by the collection of trimming loops. The values of the colour spline control point coordinates may be outside of the range normally associated with the colour type. The colour type shall be other than INDIRECT.

When an element of this type is interpreted, a non-uniform B-spline surface with data is generated.

No surface is displayed if the orders of any of the geometry spline, colour spline, data spline or trimming curves is not supported, or if the colour model of the colour spline is not available. The current values of the non-uniform B-spline surface with data attributes, as defined in the traversal state list, are bound to the primitive. The non-uniform B-spline surface with data attributes are defined in 4.5.4.12.

NOTE: Trimming curve approximation criteria types are registered in the ISO International Register of Graphical Items (4.1.2). When a trimming curve approximation criteria type has been approved by ISO/IEC, the trimming curve approximation criteria type value will be assigned by the Registration Authority.

References: 4.4.13, 4.4.12, 4.4.14, 4.5.1, 4.5.4, 4.5.5, 4.5.9, 4.6, 4.8, 4.9

Errors:

005 *Ignoring function, function requires state (PHOP,*,STOP,*)*

500 *Ignoring function, the order of a spline is less than one*

501 *Ignoring function, a spline does not contain enough control points for its specified order*

502 *Ignoring function, the order of a spline is inconsistent with its number of knots and control points*

503 *Ignoring function, the knot sequence for a spline not non-decreasing*

507 *Ignoring function, the fourth coordinate of a rational control point is less than or equal to zero*

508 *Ignoring function, a trimming curve's order is less than two*

517 *Ignoring function, the parameter range of the colour spline does not include the effective parameter range of the geometry spline*

518 Ignoring function, the parameter range of the data spline does not include the effective parameter range of the geometry spline

138 Ignoring function, one or more of the fields within the specified data record is in error

5.3 Attribute specification functions

5.3.1 Bundled attribute selection

SET DATA MAPPING INDEX **(PHOP,*,STOP,*)**

In	data mapping index	(1...n)	I

Effect: Depending on the 'edit mode', a "set data mapping index" element is inserted into the open structure after the 'element pointer' or replaces the element pointed at by the 'element pointer'. The 'element pointer' is then updated to point to this "set data mapping index" structure element. The value specified by the parameter is associated with the element.

When an element of this type is interpreted, the 'current data mapping index' entry in the traversal state list is set to the value associated with the element. This value is used when displaying subsequent area primitives.

When an element of this type is interpreted, if the specified entry is not available on a workstation the entry corresponding to an index value of 1 is used.

References: 4.5.4, 4.6.1, 4.6.2

Errors:

005 Ignoring function, function requires state (PHOP,,STOP,*)*

100 Ignoring function, the bundle index value is less than one

SET REFLECTANCE INDEX **(PHOP,*,STOP,*)**

In	reflectance index	(1...n)	I

Effect: Depending on the 'edit mode', a "set reflectance index" element is inserted into the open structure after the 'element pointer' or replaces the element pointed at by the 'element pointer'. The 'element pointer' is then updated to point to this "set reflectance index" structure element. The value specified by the parameter is associated with the element.

When an element of this type is interpreted, the 'current reflectance index' entry in the traversal state list is set to the value associated with the element. This value is used when displaying subsequent area primitives.

When an element of this type is interpreted, if the specified entry is not available on a workstation, the entry corresponding to an index value of 1 is used.

References: 4.5.3, 4.5.4, 4.6.1

Errors:

005 Ignoring function, function requires state (PHOP,,STOP,*)*

100 Ignoring function, the bundle index value is less than one

SET BACK INTERIOR INDEX **(PHOP,*,STOP,*)**

In	back interior index	(1...n)	I

Effect: Depending on the 'edit mode', a "set back interior index" element is inserted into the open structure after the 'element pointer' or replaces the element pointed at by the 'element pointer'. The 'element pointer' is then

updated to point to this "set back interior index" structure element. The value specified by the parameter is associated with the element.

When an element of this type is interpreted, the 'current back interior index' entry in the traversal state list is set to the value associated with the element. This value is used when displaying subsequent back facing portions of area primitives.

When an element of this type is interpreted, if the specified entry is not available on a workstation, the entry corresponding to an index value of 1 is used.

References: 4.5.3, 4.5.4, 4.5.5.3, 4.6.1

Errors:

005 *Ignoring function, function requires state (PHOP,*,STOP,*)*

100 *Ignoring function, the bundle index value is less than one*

SET BACK DATA MAPPING INDEX **(PHOP,*,STOP,*)**

In	back data mapping index	(1...n)	I

Effect: Depending on the 'edit mode', a "set back data mapping index" element is inserted into the open structure after the 'element pointer' or replaces the element pointed at by the 'element pointer'. The 'element pointer' is then updated to point to this "set back data mapping index" structure element. The value specified by the parameter is associated with the element.

When an element of this type is interpreted, the 'current back data mapping index' entry in the traversal state list is set to the value associated with the element. This value is used when displaying subsequent back facing portions of area primitives.

When an element of this type is interpreted, if the specified entry is not available on a workstation, the entry corresponding to an index value of 1 is used.

References: 4.5.4, 4.5.5.3, 4.6.1, 4.6.2

Errors:

005 *Ignoring function, function requires state (PHOP,*,STOP,*)*

100 *Ignoring function, the bundle index value is less than one*

SET BACK REFLECTANCE INDEX **(PHOP,*,STOP,*)**

In	back reflectance index	(1...n)	I

Effect: Depending on the 'edit mode', a "set back reflectance index" element is inserted into the open structure after the 'element pointer' or replaces the element pointed at by the 'element pointer'. The 'element pointer' is then updated to point to this "set back reflectance index" structure element. The value specified by the parameter is associated with the element.

When an element of this type is interpreted, the 'current back reflectance index' entry in the traversal state list is set to the value associated with the element. This value is used when displaying subsequent back facing portions of area primitives.

When an element of this type is interpreted, if the specified entry is not available on a workstation, the entry corresponding to an index value of 1 is used.

References: 4.5.3, 4.5.4, 4.5.5.3, 4.6.1

Errors:

005 *Ignoring function, function requires state (PHOP,*,STOP,*)*

100 *Ignoring function, the bundle index value is less than one*

SET PARAMETRIC SURFACE INDEX **(PHOP,*,STOP,*)**

In parametric surface index (1...n) I

Effect: Depending on the 'edit mode', a "set parametric surface index" element is inserted into the open structure after the 'element pointer' or replaces the element pointed at by the 'element pointer'. The 'element pointer' is then updated to point to this "set parametric surface index" structure element. The value specified by the parameter is associated with the element.

When an element of this type is interpreted, the 'current parametric surface index' entry in the traversal state list is set to the value associated with the element. This value is used when displaying subsequent parametric surfaces.

When an element of this type is interpreted, if the specified entry is not available on a workstation, the entry corresponding to an index value of 1 is used.

References: 4.5.4

Errors:

005 *Ignoring function, function requires state (PHOP,*,STOP,*)*

100 *Ignoring function, the bundle index value is less than one*

5.3.2 Individual attribute selection

SET POLYLINE COLOUR **(PHOP,*,STOP,*)**

In polyline colour GCOLR

Effect: Depending on the 'edit mode', a "set polyline colour" element is inserted into the open structure after the 'element pointer' or replaces the element pointed at by the 'element pointer'. The 'element pointer' is then updated to point to this "set polyline colour" structure element. The value specified by the parameter is associated with the element.

When an element of this type is interpreted, the 'current polyline colour' entry in the traversal state list is set to the value associated with the element. This value is used when displaying subsequent output primitives when the 'current polyline colour ASF' in the traversal state list is INDIVIDUAL.

References: 4.5.1, 4.5.3, 4.5.4, 4.5.5.5, 4.6.4.6

Errors:

005 *Ignoring function, function requires state (PHOP,*,STOP,*)*

113 *Ignoring function, the colour index value is less than zero*

SET POLYLINE SHADING METHOD **(PHOP,*,STOP,*)**

In polyline shading method I

Effect: Depending on the 'edit mode', a "set polyline shading method" element is inserted into the open structure after the 'element pointer' or replaces the element pointed at by the 'element pointer'. The 'element pointer' is then updated to point to this "set polyline shading method" structure element. The value specified by the parameter is associated with the element.

When an element of this type is interpreted, the 'current polyline shading method' entry in the traversal state list is set to the value associated with the element. This value is used when displaying subsequent output primitives when the 'current polyline shading method ASF' in the traversal state list is INDIVIDUAL.

Polyline shading methods are as follows:

≤ 0	implementation dependent
1	NONE
2	COLOUR
≥ 3	reserved for registration or future standardisation

When an element of this type is interpreted, if the specified shading method is not available on a workstation, method 1 is used.

NOTE: Polyline shading methods are registered in the ISO International Register of Graphical Items (4.1.2). When a polyline shading method has been approved by ISO/IEC, the polyline shading method value will be assigned by the Registration Authority.

References: 4.5.4, 4.6.4.6, 4.6.5, 4.9

Errors:

005 Ignoring function, function requires state (PHOP,,STOP,*)*

SET POLYMARKER COLOUR **(PHOP,*,STOP,*)**

In polymarker colour GCOLR

Effect: Depending on the 'edit mode', a "set polymarker colour" element is inserted into the open structure after the 'element pointer' or replaces the element pointed at by the 'element pointer'. The 'element pointer' is then updated to point to this "set polymarker colour" structure element. The value specified by the parameter is associated with the element.

When an element of this type is interpreted, the 'current polymarker colour' entry in the traversal state list is set to the value associated with the element. This value is used when displaying subsequent POLYMARKER output primitives when the 'current polymarker colour ASF' in the traversal state list is INDIVIDUAL.

References: 4.5.1, 4.5.3, 4.5.4, 4.5.5.5

Errors:

005 Ignoring function, function requires state (PHOP,,STOP,*)*

113 Ignoring function, the colour index value is less than zero

SET TEXT COLOUR **(PHOP,*,STOP,*)**

In text colour GCOLR

Effect: Depending on the 'edit mode', a "set text colour" element is inserted into the open structure after the 'element pointer' or replaces the element pointed at by the 'element pointer'. The 'element pointer' is then updated to point to this "set text colour" structure element. The value specified by the parameter is associated with the element.

When an element of this type is interpreted, the 'current text colour' entry in the traversal state list is set to the value associated with the element. This value is used when displaying subsequent TEXT and ANNOTATION TEXT output primitives when the 'current text colour ASF' in the traversal state list is INDIVIDUAL.

References: 4.5.1, 4.5.3, 4.5.4, 4.5.5.5

Errors:

005 Ignoring function, function requires state (PHOP,,STOP,*)*

113 Ignoring function, the colour index value is less than zero

SET FACET DISTINGUISHING MODE **(PHOP,*,STOP,*)**

In facet distinguishing mode (OFF, ON) E

Effect: Depending on the 'edit mode', a "set facet distinguishing mode" element is inserted into the open structure after the 'element pointer' or replaces the element pointed at by the 'element pointer'. The 'element pointer' is then updated to point to this "set facet distinguishing mode" structure element. The value specified by the parameter is associated with the element.

When an element of this type is interpreted, the 'current facet distinguishing mode' entry in the traversal state list is set to the value associated with the element. The 'current facet distinguishing mode' is used to determine if the set of back attributes should be applied to back-facing portions of area primitives.

References: 4.5.3, 4.5.4, 4.5.5.3, 4.5.7, 4.6.1

Errors:

005 Ignoring function, function requires state (PHOP,,STOP,*)*

SET FACET CULLING MODE **(PHOP,*,STOP,*)**

In facet culling mode (NONE, BACKFACING, FRONTFACING) E

Effect: Depending on the 'edit mode', a "set facet culling mode" element is inserted into the open structure after the 'element pointer' or replaces the element pointed at by the 'element pointer'. The 'element pointer' is then updated to point to this "set facet culling mode" structure element. The value specified by the parameter is associated with the element.

When an element of this type is interpreted, the 'current facet culling mode' entry in the traversal state list is set to the value associated with the element. The 'current facet culling mode' is used to determine which, if any, portions of area primitives should be excluded from the abstract image based on their distinction as front or back facing.

References: 4.5.3, 4.5.4, 4.5.5.3, 4.5.6, 4.8

Errors:

005 Ignoring function, function requires state (PHOP,,STOP,*)*

SET INTERIOR COLOUR **(PHOP,*,STOP,*)**

In interior colour GCOLR

Effect: Depending on the 'edit mode', a "set interior colour" element is inserted into the open structure after the 'element pointer' or replaces the element pointed at by the 'element pointer'. The 'element pointer' is then updated to point to this "set interior colour" structure element. The value specified by the parameter is associated with the element.

When an element of this type is interpreted, the 'current interior colour' entry in the traversal state list is set to the value associated with the element. This value is used when displaying subsequent area primitives when the 'current interior colour ASF' in the traversal state list is INDIVIDUAL.

References: 4.5.1, 4.5.3, 4.5.4, 4.5.5.5, 4.6.2, 4.6.4

Errors:

005 Ignoring function, function requires state (PHOP,,STOP,*)*

113 Ignoring function, the colour index value is less than zero

SET INTERIOR SHADING METHOD **(PHOP,*,STOP,*)**

In interior shading method I

Effect: Depending on the 'edit mode', a "set interior shading method" element is inserted into the open structure after the 'element pointer' or replaces the element pointed at by the 'element pointer'. The 'element pointer' is then updated to point to this "set interior shading method" structure element. The value specified by the parameter is associated with the element.

When an element of this type is interpreted, the 'current interior shading method' entry in the traversal state list is set to the value associated with the element. This value is used when displaying subsequent area primitives when the 'current interior shading method ASF' in the traversal state list is INDIVIDUAL.

Interior shading methods are as follows:

≤ 0	implementation dependent
1	NONE
2	COLOUR
3	DATA
4	DATA and DOT
5	DATA and NORMAL
≥ 6	reserved for registration or future standardisation

When an element of this type is interpreted, if the specified interior shading method is not available on a workstation, interior shading method 1 is used.

NOTE: Interior shading methods are registered in the ISO International Register of Graphical Items (4.1.2). When an interior shading method has been approved by ISO/IEC, the interior shading method value will be assigned by the Registration Authority.

References: 4.5.3, 4.5.4, 4.6.1, 4.6.4, 4.6.5, 4.9

Errors:

005 Ignoring function, function requires state (PHOP,,STOP,*)*

SET DATA MAPPING METHOD **(PHOP,*,STOP,*)**

In	data mapping method	I
In	data mapping data record	D

Effect: Depending on the 'edit mode', a "set data mapping method" element is inserted into the open structure after the 'element pointer' or replaces the element pointed at by the 'element pointer'. The 'element pointer' is then updated to point to this "set data mapping method" structure element. The values specified by the parameters are associated with the element.

When an element of this type is interpreted, the 'current data mapping method' entry in the traversal state list is set to the values associated with the element. These values are used when displaying subsequent area primitives when the 'current data mapping method ASF' in the traversal state list is INDIVIDUAL.

The data mapping methods are as follows:

≤ 0	implementation dependent
1	COLOUR
2	SINGLE VALUE, UNIFORM
3	SINGLE VALUE, NON-UNIFORM
4	BI-VALUE, UNIFORM
5	BI-VALUE, NON-UNIFORM
≥ 6	reserved for registration or future standardisation

When an element of this type is interpreted, if the specified data mapping method is not available on a workstation, data mapping method 1 is used. If the specified data mapping method or data record is inconsistent with the data specified with a primitive, data mapping method 1 is used for that primitive. The data record values used in these cases are the default values indicated in the PHIGS description table.

NOTE: Data mapping methods are registered in the ISO International Register of Graphical Items (4.1.2). When a data mapping method has been approved by ISO/IEC, the data mapping method value will be assigned by the Registration Authority.

References: 4.5.1, 4.5.4, 4.5.5.5, 4.6.1, 4.6.2, 4.6.4

Data records:

Data mapping method = 1 (COLOUR)

1	list of source selectors	(COLOUR_ASPECT, VERTEX_COLOUR, VERTEX_DATA, FACET_COLOUR, FACET_DATA)	L(E)

Data mapping method = 2 (SINGLE VALUE, UNIFORM)

1	list of source selectors	(COLOUR_ASPECT, VERTEX_COLOUR, VERTEX_DATA, FACET_COLOUR, FACET_DATA)	L(E)
2	data value index	(1...n)	I
3	lower range limit		R
4	upper range limit		R
5	colour type		I
6	list of colour values		L(COLRV)

Data mapping method = 3 (SINGLE VALUE, NON-UNIFORM)

1	list of source selectors	(COLOUR_ASPECT, VERTEX_COLOUR, VERTEX_DATA, FACET_COLOUR, FACET_DATA)	L(E)
2	data value index	(1...n)	I
3	range boundaries		L(R)
4	colour type		I
5	list of colour values		L(COLRV)

Data mapping method = 4 (BI-VALUE, UNIFORM)

1	list of source selectors	(COLOUR_ASPECT, VERTEX_COLOUR, VERTEX_DATA, FACET_COLOUR, FACET_DATA)	L(E)
2	data value indices	(1...n)	2×I
3	lower limit of R_a range		R
4	upper limit of R_a range		R
5	lower limit of R_b range		R
6	upper limit of R_b range		R
7	colour type		I
8	list of colour value lists		L(L(COLRV))

Data mapping method = 5 (BI-VALUE, NON-UNIFORM)

1	list of source selectors	(COLOUR_ASPECT, VERTEX_COLOUR, VERTEX_DATA, FACET_COLOUR, FACET_DATA)	L(E)
2	data value indices	(1...n)	2×I
3	R_a range boundaries		L(R)
4	R_b range boundaries		L(L(R))
5	colour type		I
6	lists of colour value lists		L(L(COLRV))

Errors:

005 *Ignoring function, function requires state (PHOP,*,STOP,*)*

138 *Ignoring function, one or more of the fields within the specified data record is in error*

SET REFLECTANCE PROPERTIES **(PHOP,*,STOP,*)**

In	reflectance properties type	I
In	reflectance properties data record	D

Effect: Depending on the 'edit mode', a "set reflectance properties" element is inserted into the open structure after the 'element pointer' or replaces the element pointed at by the 'element pointer'. The 'element pointer' is then updated to point to this "set reflectance properties" structure element. The values specified by the parameters are associated with the element.

The reflectance property types are as follows:

≤ 0	implementation dependent
1	simple reflectance
≥ 2	reserved for registration or future standardisation

When an element of this type is interpreted, the 'current reflectance properties' entry in the traversal state list are set to the values associated with the element. These values are conditionally used when displaying front-facing portions of subsequent area primitives when the 'current reflectance properties ASF' in the traversal state list is INDIVIDUAL.

When an element of this type is interpreted, if the specified reflectance properties type is not available on a workstation, reflectance properties type 1 is used. The data record values used in this case are the default values indicated in the PHIGS description table.

NOTE: Reflectance property types are registered in the ISO International Register of Graphical Items (4.1.2). When a reflectance property type has been approved by ISO/IEC, the reflectance property value will be assigned by the Registration Authority.

References: 4.5.1, 4.5.3, 4.5.4, 4.6.1, 4.6.3

Data records:

Reflectance property type = 1 (simple reflectance)

1	ambient reflection coefficient	[0,1]	R
2	diffuse reflection coefficient	[0,1]	R
3	specular reflection coefficient	[0,1]	R
4	specular colour		GCOLR
5	specular exponent	≥ 0	R

Errors:

005 *Ignoring function, function requires state (PHOP,*,STOP,*)*

138 *Ignoring function, one or more of the fields within the specified data record is in error*

SET REFLECTANCE MODEL **(PHOP,*,STOP,*)**

In	reflectance model	I

Effect: Depending on the 'edit mode', a "set reflectance model" element is inserted into the open structure after the 'element pointer' or replaces the element pointed at by the 'element pointer'. The 'element pointer' is then updated to point to this "set reflectance model" structure element. The value specified by the parameter is associated with the element.

When an element of this type is interpreted, the 'current reflectance model' entry in the traversal state list is set to the value associated with the element. This value is used when displaying subsequent area primitives when the 'current reflectance model ASF' in the traversal state list is INDIVIDUAL.

The defined values for reflectance model are:

≤ 0	implementation dependent
1	no reflectance calculation performed
2	ambient reflectance effects produced
3	ambient and diffuse reflectance effects produced
4	ambient, diffuse, and specular reflectance effects produced
≥ 5	reserved for registration or future standardisation

When an element of this type is interpreted, if the specified value is not available on a workstation, a value of 1 is used.

NOTE: Reflectance models are registered in the ISO International Register of Graphical Items (4.1.2). When a reflectance model type has been approved by ISO/IEC, the reflectance model value will be assigned by the Registration Authority.

References: 4.5.3, 4.5.4, 4.6.1, 4.6.3, 4.6.5

Errors:

005 Ignoring function, function requires state (PHOP,,STOP,*)*

SET BACK INTERIOR STYLE **(PHOP,*,STOP,*)**

In back interior style (HOLLOW,SOLID,PATTERN,HATCH,EMPTY) E

Effect: Depending on the 'edit mode', a "set back interior style" element is inserted into the open structure after the 'element pointer' or replaces the element pointed at by the 'element pointer'. The 'element pointer' is then updated to point to this "set back interior style" structure element. The value specified by the parameter is associated with the element.

When an element of this type is interpreted, the 'current back interior style' entry in the traversal state list is set to the value associated with the element. This value is used when displaying back-facing facets of subsequent area primitives when the 'current back interior style ASF' in the traversal state list is INDIVIDUAL.

When an element of this type is interpreted, if the specified interior style is not available on a workstation, interior style HOLLOW is used.

References: 4.5.3, 4.5.4, 4.5.5.3, 4.6.1.2, 4.6.4

Errors:

005 Ignoring function, function requires state (PHOP,,STOP,*)*

SET BACK INTERIOR STYLE INDEX **(PHOP,*,STOP,*)**

In back interior style index I

Effect: Depending on the 'edit mode', a "set back interior style index" element is inserted into the open structure after the 'element pointer' or replaces the element pointed at by the 'element pointer'. The 'element pointer' is then updated to point to this "set back interior style index" structure element. The value specified by the parameter is associated with the element.

When an element of this type is interpreted, the 'current back interior style index' entry in the traversal state list is set to the value associated with the element. This value is used when displaying back-facing portions of

subsequent area primitives when the 'current back interior style index ASF' in the traversal state list is INDIVIDUAL.

For back interior styles HOLLOW, SOLID and EMPTY, the interior style index is not used. For interior style PATTERN, the back interior style index value is an index into a workstation's pattern table. For interior style HATCH, the back interior style index value determines which of a number of hatch styles is used: hatch styles greater than 0 are reserved for registration; hatch styles less than 1 are workstation dependent.

When an element of this type is interpreted, if the specified entry is not available on a workstation, the entry corresponding to an index value of 1 is used. If back interior style index 1 is not present on that workstation, the result is workstation dependent.

References: 4.5.3, 4.5.4, 4.5.5.3, 4.6.4

Errors:

005 Ignoring function, function requires state (PHOP,,STOP,*)*

SET BACK INTERIOR COLOUR **(PHOP,*,STOP,*)**

In back interior colour GCOLR

Depending on the 'edit mode', a "set back interior colour" element is inserted into the open structure after the 'element pointer' or replaces the element pointed at by the 'element pointer'. The 'element pointer' is then updated to point to this "set back interior colour" structure element. The value specified by the parameter is associated with the element.

When an element of this type is interpreted, the 'current back interior colour' entry in the traversal state list is set to the value associated with the element. This value is used when displaying subsequent back-facing portions of area primitives when the 'current back interior colour ASF' in the traversal state list is INDIVIDUAL.

References: 4.5.1, 4.5.3, 4.5.4, 4.5.5.3, 4.5.5.5, 4.6.4

Errors:

005 Ignoring function, function requires state (PHOP,,STOP,*)*

113 Ignoring function, the colour index value is less than zero

SET BACK INTERIOR SHADING METHOD **(PHOP,*,STOP,*)**

In back interior shading method I

Effect: Depending on the 'edit mode', a "set back interior shading method" element is inserted into the open structure after the 'element pointer' or replaces the element pointed at by the 'element pointer'. The 'element pointer' is then updated to point to this "set back interior shading method" structure element. The value specified by the parameter is associated with the element. See SET INTERIOR SHADING METHOD for the interior shading methods.

When an element of this type is interpreted, the 'current back interior shading method' entry in the traversal state list is set to the value associated with the element. This value is used when displaying back-facing portions of subsequent area primitives when the 'current back interior shading method ASF' in the traversal state list is INDIVIDUAL.

When an element of this type is interpreted, if the specified interior shading method is not available on a workstation, interior shading method 1 is used.

References: 4.5.3, 4.5.4, 4.5.5.3, 4.6.1, 4.6.4, 4.6.5, 4.9

Errors:

005 Ignoring function, function requires state (PHOP,,STOP,*)*

SET BACK DATA MAPPING METHOD **(PHOP,*,STOP,*)**

In	back data mapping method	I
In	back data mapping data record	D

Effect: Depending on the 'edit mode', a "set back data mapping method" element is inserted into the open structure after the 'element pointer' or replaces the element pointed at by the 'element pointer'. The 'element pointer' is then updated to point to this "set back data mapping method" structure element. The values specified by the parameters are associated with the element. See SET DATA MAPPING METHOD for the data mapping methods.

When an element of this type is interpreted, the 'current back data mapping method' entry in the traversal state list is set to the values associated with the element. These values are used when displaying back-facing portions of subsequent area primitives when the 'current back data mapping method ASF' in the traversal state list is INDIVIDUAL.

When an element of this type is interpreted, if the specified data mapping method is not available on a workstation, data mapping method 1 is used. If the specified data mapping method or data record is inconsistent with the data specified with a primitive, data mapping method 1 is used for that primitive. The data record values used in these cases are the default values indicated in the PHIGS description table.

References: 4.5.4, 4.5.5.3, 4.5.5.5, 4.6.1, 4.6.2, 4.6.4

Data records: See SET DATA MAPPING METHOD.

Errors:

005 *Ignoring function, function requires state (PHOP,*,STOP,*)*

138 *Ignoring function, one or more of the fields within the specified data record is in error*

SET BACK REFLECTANCE PROPERTIES **(PHOP,*,STOP,*)**

In	back reflectance properties type	I
In	back reflectance properties data record	D

Effect: Depending on the 'edit mode', a "set back reflectance properties" element is inserted into the open structure after the 'element pointer' or replaces the element pointed at by the 'element pointer'. The 'element pointer' is then updated to point to this "set back reflectance properties" structure element. The values specified by the parameters are associated with the element. See SET REFLECTANCE PROPERTIES for the reflectance property types.

When an element of this type is interpreted, the 'current back reflectance properties' entry in the traversal state list are set to the values associated with the element. These values are (possibly) used when displaying back-facing portions of subsequent area primitives when the 'current back reflectance properties ASF' in the traversal state list is INDIVIDUAL.

When an element of this type is interpreted, if the specified reflectance properties type is not available on a workstation, reflectance properties type 1 is used. The data record values used in this case are the default values indicated in the PHIGS description table.

References: 4.5.1, 4.5.3, 4.5.4, 4.5.5.3, 4.6.1, 4.6.3

Data records: See SET REFLECTANCE PROPERTIES.

Errors:

005 *Ignoring function, function requires state (PHOP,*,STOP,*)*

138 *Ignoring function, one or more of the fields within the specified data record is in error*

SET BACK REFLECTANCE MODEL **(PHOP,*,STOP,*)**

In	back reflectance model	I

Effect: Depending on the 'edit mode', a "set back reflectance model" element is inserted into the open structure after the 'element pointer' or replaces the element pointed at by the 'element pointer'. The 'element pointer' is then updated to point to this "set back reflectance model" structure element. The value specified by the parameter is associated with the element.

When an element of this type is interpreted, the 'current back reflectance model' entry in the traversal state list is set to the value associated with the element. This value is used when displaying back-facing portions of subsequent area primitives when the 'current back interior reflectance model ASF' in the traversal state list is INDIVIDUAL.

See SET REFLECTANCE MODEL for the defined values of reflectance model.

When an element of this type is interpreted, if the specified value is not available on a workstation, a value of 1 is used.

References: 4.5.3, 4.5.4, 4.5.5.3, 4.6.1, 4.6.3, 4.6.5

Errors:

005 Ignoring function, function requires state (PHOP,,STOP,*)*

SET LIGHT SOURCE STATE **(PHOP,*,STOP,*)**

In	light source activation list	(1...n)	L(I)
In	light source deactivation list	(1...n)	L(I)

Effect: Depending on the 'edit mode', a "set light source state" element is inserted into the open structure after the 'element pointer' or replaces the element pointed at by the 'element pointer'. The 'element pointer' is then updated to point to this "set light source state" structure element. The values specified by the parameters are associated with the element.

When an element of this type is interpreted, the 'current light source state' entry in the traversal state list is modified according to the two lists associated with the element: The entries in the activation list are added to the 'current light source state' if they are not already in the state; the entries in the deactivation list are removed from the 'current light source state' if they are in the state. The 'current light source state' is used to determine the set of active workstation light sources used when displaying subsequent output primitives. Each entry is an index of an active light source in a workstation's light source table.

When an element of this type is interpreted, only those workstation light source representations defined in the workstation's 'light source table' are potentially active. Entries in the 'current light source state' corresponding to undefined light source table entries are ignored. If the number of active non-ambient light sources indicated by the 'current light source state' exceeds the number supported by the workstation, it is workstation dependent which of the indicated light sources are used.

The same light source index cannot exist in both the activation and the deactivation lists.

References: 4.5.3, 4.5.4, 4.6.1, 4.6.3

Errors:

005 Ignoring function, function requires state (PHOP,,STOP,*)*

133 Ignoring function, one of the entries in the activation list or deactivation list is less than one

135 Ignoring function, the same entry exists in both the activation list and the deactivation list

SET EDGE COLOUR **(PHOP,*,STOP,*)**

In	edge colour	GCOLR

Effect: Depending on the 'edit mode', a "set edge colour" element is inserted into the open structure after the 'element pointer' or replaces the element pointed at by the 'element pointer'. The 'element pointer' is then updated to

point to this "set edge colour" structure element. The value specified by the parameter is associated with the element.

When an element of this type is interpreted, the 'current edge colour' entry in the traversal state list is set to the value associated with the element. This value is used when displaying subsequent output primitives when the 'current edge colour ASF' in the traversal state list is INDIVIDUAL.

References: 4.5.1, 4.5.3, 4.5.4, 4.5.5.5

Errors:

005 *Ignoring function, function requires state (PHOP,*,STOP,*)*

113 *Ignoring function, the colour index value is less than zero*

SET CURVE APPROXIMATION CRITERIA **(PHOP,*,STOP,*)**

In	approximation criteria type	I
In	approximation criteria data record	D

Effect: Depending on the 'edit mode', a "set curve approximation criteria" element is inserted into the open structure after the 'element pointer' or replaces the element pointed at by the 'element pointer'. The 'element pointer' is then updated to point to this "set curve approximation criteria" structure element. The values specified by the parameters are associated with the element.

When an element of this type is interpreted, the 'current curve approximation criteria' entry in the traversal state list is set to the values associated with the element. These values are used when displaying subsequent output primitives when the 'current curve approximation criteria ASF' in the traversal state list is INDIVIDUAL.

Curve approximation criteria types are as follows:

≤ 0	implementation dependent
1	WORKSTATION DEPENDENT
2	CONSTANT PARAMETRIC SUBDIVISION BETWEEN KNOTS
3	CHORDAL SIZE in WC
4	CHORDAL SIZE in NPC
5	CHORDAL SIZE in DC
6	CHORDAL DEVIATION in WC
7	CHORDAL DEVIATION in NPC
8	CHORDAL DEVIATION in DC
9	RELATIVE in WC
10	RELATIVE in NPC
11	RELATIVE in DC
≥ 12	reserved for registration or future standardisation

When an element of this type is interpreted, if the specified approximation type is not available on a workstation, type 1 is used. If the specified approximation value can not be achieved, the most accurate value available for the specified approximation type is used.

NOTE: Curve approximation criteria types are registered in the ISO International Register of Graphical Items (4.1.2). When a curve approximation criteria type has been approved by ISO/IEC, the curve approximation criteria type value will be assigned by the Registration Authority.

References: 4.4.9, 4.5.4, 4.5.4.9, 4.8

Data Record:

Curve approximation criteria type = 1 (WORKSTATION DEPENDENT)

(no data record contents)

Curve approximation criteria type = 2 (CONSTANT PARAMETRIC SUBDIVISION BETWEEN KNOTS)

1	count		I

Curve approximation criteria type = 3 (CHORDAL SIZE in WC)

1	approximation value		R

Curve approximation criteria type = 4 (CHORDAL SIZE in NPC)

1	approximation value		R

Curve approximation criteria type = 5 (CHORDAL SIZE in DC)

1	approximation value		R

Curve approximation criteria type = 6 (CHORDAL DEVIATION in WC)

1	approximation value		R

Curve approximation criteria type = 7 (CHORDAL DEVIATION in NPC)

1	approximation value		R

Curve approximation criteria type = 8 (CHORDAL DEVIATION in DC)

1	approximation value		R

Curve approximation criteria type = 9 (RELATIVE in WC)

1	approximation value	[0, 1]	R

Curve approximation criteria type = 10 (RELATIVE in NPC)

1	approximation value	[0, 1]	R

Curve approximation criteria type = 11 (RELATIVE in DC)

1	approximation value	[0, 1]	R

Errors:

005 *Ignoring function, function requires state (PHOP,*,STOP,*)*

138 *Ignoring function, one or more of the fields within the specified data record is in error*

SET SURFACE APPROXIMATION CRITERIA **(PHOP,*,STOP,*)**

In	approximation criteria type	I
In	approximation criteria data record	D

Effect: Depending on the 'edit mode', a "set surface approximation criteria" element is inserted into the open structure after the 'element pointer' or replaces the element pointed at by the 'element pointer'. The 'element pointer' is then updated to point to this "set surface approximation criteria" structure element. The values specified by the parameters are associated with the element.

When an element of this type is interpreted, the 'current surface approximation criteria' entry in the traversal state list is set to the values associated with the element. These values are used when displaying subsequent output primitives when the 'current surface approximation criteria ASF' in the traversal state list is INDIVIDUAL.

Surface approximation criteria types are as follows:

≤ 0	implementation dependent
1	WORKSTATION DEPENDENT
2	CONSTANT PARAMETRIC SUBDIVISION BETWEEN KNOTS
3	CHORDAL SIZE in WC
4	CHORDAL SIZE in NPC
5	CHORDAL SIZE in DC
6	PLANAR DEVIATION in WC
7	PLANAR DEVIATION in NPC
8	PLANAR DEVIATION in DC
9	RELATIVE in WC
10	RELATIVE in NPC
11	RELATIVE in DC
≥ 12	reserved for registration or future standardisation

When an element of this type is interpreted, if the specified approximation type is not available on a workstation, type 1 is used. If the specified approximation value can not be achieved, the most accurate value available for the specified approximation type is used.

NOTE: Surface approximation criteria types are registered in the ISO International Register of Graphical Items (4.1.2). When a surface approximation criteria type has been approved by ISO/IEC, the surface approximation criteria type value will be assigned by the Registration Authority.

References: 4.4.11, 4.4.14, 4.5.4, 4.8

Data Record:

Surface approximation criteria type = 1 (WORKSTATION DEPENDENT)

(no data record contents)

Surface approximation criteria type = 2 (CONSTANT PARAMETRIC SUBDIVISION BETWEEN KNOTS)

1	*u* count	I
2	*v* count	I

Surface approximation criteria type = 3 (CHORDAL SIZE in WC)

1	*u* approximation value	R
2	*v* approximation value	R

Surface approximation criteria type = 4 (CHORDAL SIZE in NPC)

1	*u* approximation value	R
2	*v* approximation value	R

Surface approximation criteria type = 5 (CHORDAL SIZE in DC)

1	*u* approximation value	R
2	*v* approximation value	R

Surface approximation criteria type = 6 (PLANAR DEVIATION in WC)

1	approximation value	R

Surface approximation criteria type = 7 (PLANAR DEVIATION in NPC)

1	approximation value	R

Surface approximation criteria type = 8 (PLANAR DEVIATION in DC)

1	approximation value		R

Surface approximation criteria type = 9 (RELATIVE in WC)

1	approximation value	[0, 1]	R

Surface approximation criteria type = 10 (RELATIVE in NPC)

1	approximation value	[0, 1]	R

Surface approximation criteria type = 11 (RELATIVE in DC)

1	approximation value	[0, 1]	R

Errors:

005 *Ignoring function, function requires state (PHOP,*,STOP,*)*

138 *Ignoring function, one or more of the fields within the specified data record is in error*

SET PARAMETRIC SURFACE CHARACTERISTICS **(PHOP,*,STOP,*)**

In	parametric surface characteristics type	I
In	parametric surface characteristics data record	D

Effect: Depending on the 'edit mode', a "set parametric surface characteristics" element is inserted into the open structure after the 'element pointer' or replaces the element pointed at by the 'element pointer'. The 'element pointer' is then updated to point to this "set parametric surface characteristics" structure element. The values specified by the parameters are associated with the element.

When an element of this type is interpreted, the 'current parametric surface characteristics' entry in the traversal state list is set to the values associated with the element. These values are used when displaying subsequent parametric surface output primitives.

Parametric surface characteristic types are as follows:

≤ 0	implementation dependent
1	NONE
2	WORKSTATION DEPENDENT
3	ISOPARAMETRIC CURVES
4	LEVEL CURVES in MC
5	LEVEL CURVES in WC
≥ 6	reserved for registration or future standardisation

When an element of this type is interpreted, if the specified parametric surface characteristic type is not available on a workstation, or the data record contents are invalid, type 1 is used and the data record contents are workstation dependent.

NOTE: Parametric surface characteristics types are registered in the ISO International Register of Graphical Items (4.1.2). When a parametric surface characteristics type has been approved by ISO/IEC, the parametric surface characteristics type value will be assigned by the Registration Authority.

References: 4.5.4

Data Record:

Parametric surface characteristics type = 1 (NONE)

(no data record contents)

Parametric surface characteristics type = 2 (WORKSTATION DEPENDENT)

(no data record contents)

Parametric surface characteristics type = 3 (ISOPARAMETRIC CURVES)

1	curve placement	(UNIFORM OVER SURFACE, UNIFORM BETWEEN KNOTS)	E
2	curve count in *u* direction		I
3	curve count in *v* direction		I

Parametric surface characteristics type = 4 (LEVEL CURVES in MC)

1	origin point	MC	P3
2	direction vector	MC	3×R
3	parameters		L(R)

Parametric surface characteristics type = 5 (LEVEL CURVES in WC)

1	origin point	WC	P3
2	direction vector	WC	3×R
3	parameters		L(R)

Errors:

005 *Ignoring function, function requires state (PHOP,*,STOP,*)*

138 *Ignoring function, one or more of the fields within the specified data record is in error*

SET RENDERING COLOUR MODEL **(PHOP,*,STOP,*)**

In	rendering colour model	I

Effect: Depending on the 'edit mode', a "set rendering colour model" element is inserted into the open structure after the 'element pointer' or replaces the element pointed at by the 'element pointer'. The 'element pointer' is then updated to point to this "set rendering colour model" structure element. The value specified by the parameter is associated with the element.

When an element of this type is interpreted, the 'current rendering colour model' entry in the traversal state list is set to the value associated with the element. This value is used when displaying subsequent output primitives.

Rendering colour model values are as follows

< 0	implementation dependent
0	WORKSTATION DEPENDENT
1	RGB
2	CIELUV
3	HSV
4	HLS
≥ 5	reserved for registration or future standardisation

When an element of this type is interpreted, if the specified colour model is not available on a workstation, rendering colour model 0 is used.

NOTE: Registered rendering colour model values are drawn from the same set of values as colour model values.

References: 4.5.3, 4.5.4, 4.6.5

Errors:

005 *Ignoring function, function requires state (PHOP,*,STOP,*)*

SET DEPTH CUE INDEX **(PHOP,*,STOP,*)**

In	depth cue index	(1...n)	I

Effect: Depending on the 'edit mode', a "set depth cue index" element is inserted into the open structure after the 'element pointer' or replaces the element pointed at by the 'element pointer'. The 'element pointer' is then updated to point to this "set depth cue index" structure element. The value specified by the parameter is associated with the element.

When an element of this type is interpreted, the 'current depth cue index' entry in the traversal state list is set to the value associated with the element. This value is used when displaying subsequent output primitives.

When an element of this type is interpreted, if the specified entry is not available on a workstation, the entry corresponding to an index value of 1 is used.

References: 4.5.3, 4.5.4, 4.6.1, 4.6.6

Errors:

005 Ignoring function, function requires state (PHOP,,STOP,*)*

120 Ignoring function, the depth cue index is less than one

SET COLOUR MAPPING INDEX **(PHOP,*,STOP,*)**

In	colour mapping index	(1...n)	I

Effect: Depending on the 'edit mode', a "set colour mapping index" element is inserted into the open structure after the 'element pointer' or replaces the element pointed at by the 'element pointer'. The 'element pointer' is then updated to point to this "set colour mapping index" structure element. The value specified by the parameter is associated with the element.

When an element of this type is interpreted, the 'current colour mapping index' entry in the traversal state list is set to the value associated with the element. This value is used when displaying subsequent output primitives.

When an element of this type is interpreted, if the specified entry is not available on a workstation, the entry corresponding to an index value of 1 is used.

References: 4.5.3, 4.5.4, 4.6.1, 4.6.7

Errors:

005 Ignoring function, function requires state (PHOP,,STOP,*)*

5.3.3 Aspect source flag setting

SET INDIVIDUAL ASF **(PHOP,*,STOP,*)**

In	attribute identifier	(see enumeration below)	E
In	attribute source	(BUNDLED,INDIVIDUAL)	E

Effect: Depending on the 'edit mode', a "set individual ASF" element is inserted into the open structure after the 'element pointer' or replaces the element pointed at by the 'element pointer'. The 'element pointer' is then updated to point to this "set individual ASF" structure element. The values specified by the parameters are associated with the element.

The following are the added PHIGS PLUS enumeration values for the PHIGS 'attribute identifier' enumeration type:

POLYLINE_COLOUR
POLYMARKER_COLOUR

TEXT_COLOUR
INTERIOR_COLOUR
EDGE_COLOUR
POLYLINE_SHADING_METHOD
INTERIOR_SHADING_METHOD
DATA_MAPPING_METHOD
REFLECTANCE_PROPERTIES
REFLECTANCE_MODEL
BACK_INTERIOR_STYLE
BACK_INTERIOR_STYLE_INDEX
BACK_INTERIOR_COLOUR
BACK_INTERIOR_SHADING_METHOD
BACK_DATA_MAPPING_METHOD
BACK_REFLECTANCE_PROPERTIES
BACK_REFLECTANCE_MODEL
CURVE_APPROXIMATION_CRITERIA
SURFACE_APPROXIMATION_CRITERIA
PARAMETRIC_SURFACE_CHARACTERISTICS

When an element of this type is interpreted, the Aspect Source Flag (ASF) in the PHIGS traversal state list, identified by the element's aspect identifier, is set to the value specified by the element's aspect source. These values are used by subsequent output primitives.

References: 4.5.1, 4.5.3, 4.5.4

Errors:

005 Ignoring function, function requires state (PHOP,,STOP,*)*

5.3.4 Workstation attribute table definition

SET POLYLINE REPRESENTATION PLUS **(PHOP,WSOP,*,*)**

In	workstation identifier		WI
In	polyline index	(1...n)	I
In	linetype		I
In	linewidth scale factor		R
In	polyline colour		GCOLR
In	polyline shading method		I
In	curve approximation criteria type		I
In	curve approximation criteria data record		D

Effect: In the polyline bundle table of the workstation state list of the specified workstation the given polyline index is associated with the specified parameters.

Linetypes are as follows:

≤ 0	implementation dependent
1	solid line
2	dashed line

3 dotted line

4 dashed-dotted line

≥ 5 reserved for registration or future standardisation

Linewidth Scale Factor: A scale factor applied to nominal linewidth. During structure traversal the linewidth scale factor is multiplied by the nominal linewidth on a workstation; the result is mapped by the workstation to the nearest available linewidth.

Polyline colour: Either an indirectly specified or directly specified colour. If the colour is indirectly specified, then, during structure traversal, if the specified colour index is not available, colour index 1 is used.

Polyline shading method: See SET POLYLINE SHADING METHOD for the defined polyline shading methods.

Curve approximation criteria: See SET CURVE APPROXIMATION CRITERIA for the defined curve approximation criteria types and the associated data records.

The polyline bundle table in the workstation state list has predefined entries taken from the workstation description table; a number of entries are predefined for every workstation of category OUTPUT or OUTIN. Any entry, including the predefined entries, may be set with this function.

During structure traversal the 'current polyline index' in the traversal state list is used to select an entry in the polyline bundle table of a workstation. If the entry corresponding to the 'current polyline index' is not defined, the entry corresponding to an index of 1 is used. Which aspects in the entry are used depends upon the setting of the corresponding aspect source flags.

References: 4.5.1, 4.5.2, 4.5.4.9, 4.6.4.6

Errors:

003 *Ignoring function, function requires state (PHOP,WSOP,*,*)*

054 *Ignoring function, the specified workstation is not open*

059 *Ignoring function, the specified workstation does not have output capability (i.e., the workstation category is neither OUTPUT, OUTIN, nor MO)*

100 *Ignoring function, the bundle index value is less than one*

103 *Ignoring function, setting this bundle table entry would exceed the maximum number of entries allowed in the workstation bundle table*

104 *Ignoring function, the specified linetype is not available on the specified workstation*

110 *Ignoring function, the specified colour model is not available on the workstation*

113 *Ignoring function, the colour index value is less than zero*

136 *Ignoring function, one of the components of the colour specification is out of range*

122 *Ignoring function, the specified polyline shading method is not available on the workstation*

127 *Ignoring function, the specified approximation criteria type is not available on the specified workstation*

138 *Ignoring function, one or more of the fields within the specified data record is in error*

SET POLYMARKER REPRESENTATION PLUS **(PHOP,WSOP,*,*)**

In	workstation identifier		WI
In	polymarker index	(1...n)	I
In	marker type		I
In	marker size scale factor		R
In	polymarker colour		GCOLR

Effect: In the polymarker bundle table of the workstation state list of the specified workstation the given polymarker index is associated with the specified parameters.

Marker type: Marker types are as follows:

≤ 0	implementation dependent
1	• dot
2	+ cross (plus sign)
3	* asterisk
4	O circle
5	× cross
≥ 6	reserved for registration or future standardisation

Marker size scale factor: A scale factor applied to the nominal marker size. During structure traversal the scale factor is multiplied by the nominal marker size on a workstation; the result is mapped by the workstation to the nearest available marker size.

Polymarker colour: Either an indirectly specified or directly specified colour. If the colour is indirectly specified, then, during structure traversal, if the specified colour index is not available, colour index 1 is used.

The polymarker bundle table in the workstation state list has predefined entries taken from the workstation description table; a number of entries are predefined for every workstation of category OUTPUT or OUTIN. Any entry, including the predefined entries, may be set with this function.

During structure traversal the 'current polymarker index' in the traversal state list is used to select an entry in the polymarker bundle table of a workstation. If the entry corresponding to the 'current polymarker index' is not defined, the entry corresponding to an index of 1 is used. Which aspects in the entry are used depends upon the setting of the corresponding aspect source flags.

References: 4.5.1, 4.5.2

Errors:

003 *Ignoring function, function requires state (PHOP,WSOP,*,*)*

054 *Ignoring function, the specified workstation is not open*

059 *Ignoring function, the specified workstation does not have output capability (i.e., the workstation category is neither OUTPUT, OUTIN, nor MO)*

100 *Ignoring function, the bundle index value is less than one*

103 *Ignoring function, setting this bundle table entry would exceed the maximum number of entries allowed in the workstation bundle table*

105 *Ignoring function, the specified marker type is not available on the specified workstation*

110 *Ignoring function, the specified colour model is not available on the workstation*

113 *Ignoring function, the colour index value is less than zero*

136 *Ignoring function, one of the components of the colour specification is out of range*

SET TEXT REPRESENTATION PLUS **(PHOP,WSOP,*,*)**

In	workstation identifier		WI
In	text index	(1...n)	I
In	text font		I
In	text precision	(STRING,CHAR,STROKE)	E
In	character expansion factor		R
In	character spacing		R
In	text colour		GCOLR

Effect: In the text bundle table of the workstation state list of the specified workstation the given text index is associated with the specified parameters.

Text font: A single text aspect; a particular text font can be available at some, but not necessarily all, precisions. The text font value is used to select a particular font on this workstation. Text font 1 contains a graphical representation of the characters defined in ISO/IEC 646 (or a character set including ISO/IEC 646). Text font 2 contains a graphical representation of the same characters that are visually distinguishable from text font 1. Text fonts greater than 2 are reserved for registration. Text fonts less than 1 are implementation dependent. During structure traversal if the text font is not available in the workstation, text font 1 is used on that workstation.

Text precision: The text precision value determines the fidelity with which the other aspects are used. The values of text precision, in order of increasing fidelity, are STRING, CHAR, and STROKE. During structure traversal if the specified text precision is not available on a workstation, the value STRING is used on that workstation.

Character expansion factor: Specifies the deviation of the width to height ratio of the characters from the ratio indicated by the font designer. During structure traversal only the magnitude of the character expansion factor is considered. If the result of taking the absolute value of the character expansion factor value is less than the smallest supported character expansion factor on a workstation, the smallest supported character expansion factor on that workstation is used. If the result is greater than the largest supported character expansion factor on that workstation, the largest supported character expansion factor on that workstation is used.

Character spacing: Specifies how much additional space is inserted between two adjacent character bodies. Character spacing is specified as a fraction of the font-nominal character height.

Text colour: Either an indirectly specified or directly specified colour. If the colour is specified indirectly, then, during structure traversal, if the specified colour index is not available, colour index 1 is used.

The text bundle table in the workstation state list has predefined entries taken from the workstation description table; a number of entries are predefined for every workstation of category OUTPUT or OUTIN. Any entry, including the predefined entries, may be set with this function.

During structure traversal the 'current text index' in the traversal state list is used to select an entry in the text bundle table of a workstation. If the entry corresponding to the 'current text index' is not defined, the entry corresponding to an index of 1 is used. Which aspects in the entry are used depends upon the setting of the corresponding aspect source flags.

References: 4.5.1, 4.5.2

Errors:

003 *Ignoring function, function requires state (PHOP,WSOP,*,*)*

054 *Ignoring function, the specified workstation is not open*

059 *Ignoring function, the specified workstation does not have output capability (i.e., the workstation category is neither OUTPUT, OUTIN, nor MO)*

100 *Ignoring function, the bundle index value is less than one*

103 *Ignoring function, setting this bundle table entry would exceed the maximum number of entries allowed in the workstation bundle table*

106 *Ignoring function, the specified font is not available for the requested text precision on the specified workstation*

110 *Ignoring function, the specified colour model is not available on the workstation*

113 *Ignoring function, the colour index value is less than zero*

136 *Ignoring function, one of the components of the colour specification is out of range*

SET INTERIOR REPRESENTATION PLUS **(PHOP,WSOP,*,*)**

In	workstation identifier		WI
In	interior index	(1...n)	I
In	interior style	(HOLLOW,SOLID,PATTERN,HATCH,EMPTY)	E
In	interior style index		I
In	interior colour		GCOLR
In	interior shading method		I

Effect: In the interior bundle table of the workstation state list of the specified workstation the given interior index is associated with the specified parameters.

In the following discussion only the front interior attribute names are used, although the descriptions apply to the back attributes as well.

Interior style: The interior style is used to determine in what style the interior of area should be filled. The possible values are HOLLOW, SOLID, PATTERN, HATCH and EMPTY. During structure traversal if the requested interior style is not available on a workstation, interior style HOLLOW is used on that workstation.

Interior style index: For interior styles HOLLOW, SOLID and EMPTY, the interior style index is not used. For interior style PATTERN, the interior style index is an index into the workstation's pattern table. For interior style HATCH, the interior style index determines which of a number of hatch styles is used: hatch styles greater than 0 are reserved for registration; hatch styles less than 1 are workstation dependent.

Interior shading method: See SET INTERIOR SHADING METHOD for the interior shading methods.

During structure traversal if the specified interior shading method is not available on a workstation, interior shading method 1 is used on that workstation.

Interior colour: Either an indirectly specified or directly specified colour. If the colour is specified indirectly, then, during structure traversal, if the specified colour index is not available, colour index 1 is used.

The interior bundle table in the workstation state list has predefined entries taken from the workstation description table; a number of entries are predefined for every workstation of category OUTPUT or OUTIN. Any entry, including the predefined entries, may be set with this function.

During structure traversal the 'current interior index' and 'current back interior index' in the traversal state list are used to select entries in the interior bundle table of a workstation. If the entry corresponding to the 'current interior index' or 'current back interior index' is not defined, the entry corresponding to an index of 1 is used. Which aspects in the entry are used depends upon the setting of the corresponding aspect source flags.

References: 4.5.1, 4.5.2, 4.6.1, 4.6.4.7

Errors:

003 *Ignoring function, function requires state (PHOP,WSOP,*,*)*

054 *Ignoring function, the specified workstation is not open*

059 *Ignoring function, the specified workstation does not have output capability (i.e., the workstation category is neither OUTPUT, OUTIN, nor MO)*

100 *Ignoring function, the bundle index value is less than one*

103 Ignoring function, setting this bundle table entry would exceed the maximum number of entries allowed in the workstation bundle table

108 Ignoring function, the specified interior style is not available on the workstation

112 Ignoring function, the pattern index value is less than one

110 Ignoring function, the specified colour model is not available on the workstation

113 Ignoring function, the colour index value is less than zero

123 Ignoring function, the specified interior shading method is not available on the workstation

136 Ignoring function, one of the components of the colour specification is out of range

SET EDGE REPRESENTATION PLUS **(PHOP,WSOP,*,*)**

In	workstation identifier		WI
In	edge index	(1...n)	I
In	edge flag	(OFF,ON)	E
In	edgetype		I
In	edgewidth scale factor		R
In	edge colour		GCOLR

Effect: In the edge bundle table of the workstation state list of the specified workstation the given edge index is associated with the specified parameters.

Edgetypes are as follows:

≤ 0	implementation dependent
1	solid line
2	dashed line
3	dotted line
4	dashed-dotted line
≥ 5	reserved for registration or future standardisation

Edgewidth Scale Factor: A scale factor applied to nominal edgewidth. During structure traversal the edgewidth scale factor is multiplied by the nominal edgewidth on a workstation; the result is mapped by the workstation to the nearest available edgewidth.

Edge colour: Either an indirectly specified or directly specified colour. If the colour is specified indirectly, then, during structure traversal, if the specified colour index is not available, colour index 1 is used.

The edge bundle table in the workstation state list has predefined entries taken from the workstation description table; a number of entries are predefined for every workstation of category OUTPUT or OUTIN. Any entry, including the predefined entries, may be set with this function.

During structure traversal the 'current edge index' in the traversal state list is used to select an entry in the edge bundle table of a workstation. If the entry corresponding to the 'current edge index' is not defined, the entry corresponding to an index of 1 is used. Which aspects in the entry are used depends upon the setting of the corresponding aspect source flags.

References: 4.5.1, 4.5.2

Errors:

003 Ignoring function, function requires state (PHOP,WSOP,,*)*

054 Ignoring function, the specified workstation is not open

059 *Ignoring function, the specified workstation does not have output capability (i.e., the workstation category is neither OUTPUT, OUTIN, nor MO)*

100 *Ignoring function, the bundle index value is less than one*

103 *Ignoring function, setting this bundle table entry would exceed the maximum number of entries allowed in the workstation bundle table*

110 *Ignoring function, the specified colour model is not available on the workstation*

113 *Ignoring function, the colour index value is less than zero*

136 *Ignoring function, one of the components of the colour specification is out of range*

107 *Ignoring function, the specified edgetype is not available on the specified workstation*

SET DATA MAPPING REPRESENTATION **(PHOP,WSOP,*,*)**

In	workstation identifier		WI
In	data mapping index	(1...n)	I
In	data mapping method		I
In	data mapping data record		D

Effect: In the data mapping bundle table of the workstation state list of the specified workstation the given data mapping index is associated with the specified parameters.

See SET DATA MAPPING METHOD for the defined data mapping methods and their associated data records.

The data mapping bundle table in the workstation state list has predefined entries taken from the workstation description table; a number of entries are predefined for every workstation of category OUTPUT or OUTIN. Any entry, including the predefined entries, may be set with this function.

During structure traversal the 'current data mapping index' and 'current back data mapping index' in the traversal state list are used to select entries in the data mapping bundle table of a workstation. If the entry corresponding to either of these indices is not defined, the entry corresponding to an index of 1 is used. The use of the entries in each data mapping bundle depends upon the setting of the corresponding aspect source flags.

References: 4.6.1, 4.6.2

Errors:

003 *Ignoring function, function requires state (PHOP,WSOP,*,*)*

054 *Ignoring function, the specified workstation is not open*

059 *Ignoring function, the specified workstation does not have output capability (i.e., the workstation category is neither OUTPUT, OUTIN, nor MO)*

100 *Ignoring function, the bundle index value is less than one*

103 *Ignoring function, setting this bundle table entry would exceed the maximum number of entries allowed in the workstation bundle table*

137 *Ignoring function, the specified data mapping method is not available on the specified workstation*

138 *Ignoring function, one or more of the fields within the specified data record is in error*

SET REFLECTANCE REPRESENTATION **(PHOP,WSOP,*,*)**

In	workstation identifier		WI
In	reflectance index	(1...n)	I
In	reflectance model		I
In	reflectance properties type		I
In	reflectance properties data record		D

Effect: In the reflectance bundle table of the workstation state list of the specified workstation the given reflectance index is associated with the specified parameters.

See SET REFLECTANCE MODEL for the defined reflectance models. See SET REFLECTANCE PROPERTIES for the defined reflectance property types and their associated data records.

The reflectance bundle table in the workstation state list has predefined entries taken from the workstation description table; a number of entries are predefined for every workstation of category OUTPUT or OUTIN. Any entry, including the predefined entries, may be set with this function.

During structure traversal the 'current reflectance index' and 'current back reflectance index' in the traversal state list are used to select entries in the reflectance bundle table of a workstation. If the entry corresponding to the 'current reflectance index' or 'current back reflectance index' is not defined, the entry corresponding to an index of 1 is used. Which aspects in the entry are used depends upon the setting of the corresponding aspect source flags.

References: 4.5.1, 4.5.2, 4.6.1, 4.6.3

Errors:

003 *Ignoring function, function requires state (PHOP,WSOP,*,*)*

054 *Ignoring function, the specified workstation is not open*

059 *Ignoring function, the specified workstation does not have output capability (i.e., the workstation category is neither OUTPUT, OUTIN, nor MO)*

100 *Ignoring function, the bundle index value is less than one*

103 *Ignoring function, setting this bundle table entry would exceed the maximum number of entries allowed in the workstation bundle table*

124 *Ignoring function, the specified reflectance model is not available on the workstation*

139 *Ignoring function, the specified reflectance property type is not available on the specified workstation*

110 *Ignoring function, the specified colour model is not available on the workstation*

113 *Ignoring function, the colour index value is less than zero*

136 *Ignoring function, one of the components of the colour specification is out of range*

138 *Ignoring function, one or more of the fields within the specified data record is in error*

SET PARAMETRIC SURFACE REPRESENTATION **(PHOP,WSOP,*,*)**

In	workstation identifier		WI
In	parametric surface index	(1...n)	I
In	surface approximation criteria type		I
In	surface approximation criteria data record		D
In	parametric surface characteristics type		I
In	parametric surface characteristics data record		D

Effect: In the parametric surface bundle table of the workstation state list of the specified workstation the given parametric surface index is associated with the specified parameters.

See SET SURFACE APPROXIMATION CRITERIA for the defined surface approximation criteria types and their associated data records. See SET PARAMETRIC SURFACE CHARACTERISTICS for the defined parametric surface characteristics types and their associated data records.

The parametric surface bundle table in the workstation state list has predefined entries taken from the workstation description table; a number of entries are predefined for every workstation of category OUTPUT or OUTIN. Any entry, including the predefined entries, may be set with this function.

During structure traversal the 'current parametric surface index' in the traversal state list is used to select an entry in the parametric surface bundle table of a workstation. If the entry corresponding to the 'current parametric surface index' is not defined, the entry corresponding to an index of 1 is used. Which aspects in the entry are used depends upon the setting of the corresponding aspect source flags.

References: 4.5.2, 4.5.4.11

Errors:

003 *Ignoring function, function requires state (PHOP,WSOP,*,*)*

054 *Ignoring function, the specified workstation is not open*

059 *Ignoring function, the specified workstation does not have output capability (i.e., the workstation category is neither OUTPUT, OUTIN, nor MO)*

100 *Ignoring function, the bundle index value is less than one*

103 *Ignoring function, setting this bundle table entry would exceed the maximum number of entries allowed in the workstation bundle table*

127 *Ignoring function, the specified approximation criteria type is not available on the specified workstation*

128 *Ignoring function, the specified parametric surface characteristics type is not available on the specified workstation*

138 *Ignoring function, one or more of the fields within the specified data record is in error*

SET PATTERN REPRESENTATION PLUS **(PHOP,WSOP,*,*)**

In	workstation identifier		WI
In	pattern index	(1...n)	I
In	colour type		I
In	pattern colour array		A(COLRV)

Effect: In the pattern table of the workstation state list of the specified workstation the given pattern index is associated with the specified parameters.

A grid of DX x DY cells (DX horizontal, DY vertical) is specified. If the colour type indicates indirectly specified colours, then colour is given individually for each cell by colour index, a pointer into the colour table of the workstation. Otherwise, the colour is given individually for each cell by a colour value in the indicated colour model.

If the workstation supports interior style PATTERN, the pattern table in the workstation state list has predefined entries taken from the workstation description table; a number are predefined for every workstation of category OUTPUT and OUTIN supporting interior style PATTERN. Any table entry (including the predefined entries) may be set with this function.

When area primitives are displayed during structure traversal, if the currently selected (either via the interior bundle or individually, depending upon the corresponding ASF) interior style is PATTERN, the 'current interior style index' and 'current back interior style index' in the traversal state list refer to an entry in the PHIGS PLUS pattern table. If the 'current interior style index' or 'current back interior style index' is not present in the pattern table, interior style index 1 is used. If interior style index 1 is not present (i.e. interior style PATTERN is not supported for this workstation), the result is workstation dependent.

During structure traversal if the colour type of a specified pattern table entry is indirect, and if a pattern colour index specified in the pattern colour array is not available on the workstation, colour index 1 is used.

References: 4.5.1, 4.5.2, 4.6.4

Errors:

003 *Ignoring function, function requires state (PHOP,WSOP,*,*)*

054 *Ignoring function, the specified workstation is not open*

059 *Ignoring function, the specified workstation does not have output capability (i.e., the workstation category is neither OUTPUT, OUTIN, nor MO)*

112 *Ignoring function, the pattern index value is less than one*

103 *Ignoring function, setting this bundle table entry would exceed the maximum number of entries allowed in the workstation bundle table*

116 *Ignoring function, one of the dimensions of pattern colour array is less than one*

110 *Ignoring function, the specified colour model is not available on the workstation*

113 *Ignoring function, the colour index value is less than zero*

136 *Ignoring function, one of the components of the colour specification is out of range*

SET LIGHT SOURCE REPRESENTATION **(PHOP,WSOP,*,*)**

In	workstation identifier		WI
In	light source index	(1...n)	I
In	light source type		I
In	light source data record		D

Effect: In the light source table of the workstation state list of the specified workstation the given light source index is associated with the specified parameters.

Light source types are:

≤ 0	implementation dependent
1	AMBIENT
2	DIRECTIONAL
3	POSITIONAL
4	SPOT
≥ 5	reserved for registration or future standardisation

The light source table in the workstation state list has predefined entries taken from the workstation description table. Some entries are predefined for every workstation of category OUTPUT or OUTIN. Any entry in the light source table may be set with this function.

During structure traversal the 'current light source state' in the traversal state list is used to select zero or more entries in the light source table of a workstation. Entries in the 'current light source state' that are not defined in a workstation's light source table are ignored.

During structure traversal if the colour type of any selected workstation light source representation is indirect, and the colour index is not defined on the workstation, colour index 1 is used for that light source.

NOTE: Light source types are registered in the ISO International Register of Graphical Items (4.1.2). When a light source type has been approved by ISO/IEC, the light source type value will be assigned by the Registration Authority.

References: 4.5.1, 4.6.1, 4.6.3

Data records:

Light source type = 1 (ambient light source)

1	light source colour		GCOLR

Light source type = 2 (directional light source)

1	light source colour		GCOLR
2	light source direction	WC	NORM

Light source type = 3 (positional light source)

1	light source colour		GCOLR
2	light source position	WC	P3
3	attenuation coefficients		2×R

Light source type = 4 (spot light source)

1	light source colour		GCOLR
2	light source position	WC	P3
3	light source direction	WC	NORM
4	concentration exponent		R
5	attenuation coefficients		2×R
6	spread angle	[0,π]	R

Errors:

003 *Ignoring function, function requires state (PHOP,WSOP,*,*)*

054 *Ignoring function, the specified workstation is not open*

059 *Ignoring function, the specified workstation does not have output capability (i.e., the workstation category is neither OUTPUT, OUTIN, nor MO)*

103 *Ignoring function, setting this bundle table entry would exceed the maximum number of entries allowed in the workstation bundle table*

110 *Ignoring function, the specified colour model is not available on the workstation*

113 *Ignoring function, the colour index value is less than zero*

136 *Ignoring function, one of the components of the colour specification is out of range*

129 *Ignoring function, the light source index is less than one*

131 *Ignoring function, the specified light source type is not available on the workstation*

132 *Ignoring function, the specified spot light spread angle is out of range*

138 *Ignoring function, one or more of the fields within the specified data record is in error*

SET DEPTH CUE REPRESENTATION **(PHOP,WSOP,*,*)**

In	workstation identifier		WI
In	depth cue index	(1...n)	I
In	depth cue mode	(SUPPRESSED,ALLOWED)	E
In	depth cue reference planes	NPC, DQMIN<DQMAX	2×R
In	depth cue scale factors	[0,1]	2×R
In	depth cue colour		GCOLR

Effect: In the depth cue table of the workstation state list of the specified workstation the given depth cue index is associated with the specified parameters.

The depth cue table in the workstation state list has predefined entries taken from the workstation description table. At least entry 1 is predefined for every workstation of category OUTPUT or OUTIN.

During structure traversal the 'current depth cue index' in the traversal state list is used to select an entry in the depth cue table of the workstation state list. If the entry corresponding to the 'current depth cue index' is not defined, the entry corresponding to an index of 1 is used.

During structure traversal if the depth cue colour type of the selected representation is indirect, and the colour index is not defined on the workstation, colour index 1 is used.

References: 4.5.1, 4.6.1, 4.6.6

Errors:

003 *Ignoring function, function requires state (PHOP,WSOP,*,*)*

054 *Ignoring function, the specified workstation is not open*

059 *Ignoring function, the specified workstation does not have output capability (i.e., the workstation category is neither OUTPUT, OUTIN, nor MO)*

103 *Ignoring function, setting this bundle table entry would exceed the maximum number of entries allowed in the workstation bundle table*

120 *Ignoring function, the depth cue index is less than one*

119 *Ignoring function, the specified depth cue mode is not available on the workstation*

130 *Ignoring function, invalid reference planes; DQMIN > DQMAX*

140 *Ignoring function, one of the depth cue scale factors is not in the required range*

110 *Ignoring function, the specified colour model is not available on the workstation*

113 *Ignoring function, the colour index value is less than zero*

136 *Ignoring function, one of the components of the colour specification is out of range*

SET COLOUR MAPPING REPRESENTATION **(PHOP,WSOP,*,*)**

In	workstation identifier		WI
In	colour mapping index	(1...n)	I
In	colour mapping method		I
In	colour mapping data record		D

Effect: In the colour mapping table of the workstation state list of the specified workstation the given colour mapping index is associated with the specified parameters.

Colour mapping methods are:

≤ 0	implementation dependent
1	TRUE
2	PSEUDO
3	PSEUDO-N
≥ 4	reserved for registration or future standardisation

NOTE: Colour mapping types are registered in the ISO International Register of Graphical Items (4.1.2). When a colour mapping type has been approved by ISO/IEC, the colour mapping type value will be assigned by the Registration Authority.

References: 4.6.1, 4.6.7

Data Record:

Colour mapping method = 1 (TRUE)

(no data record contents)

Colour mapping method = 2 (PSEUDO)

1	colour model	I
2	weight vector	L(R)
3	colours	L(COLRV)

(There must be as many values in the weight vector as there are coordinate values in the colour tuple of the specified colour model.)

Colour mapping method = 3 (PSEUDO-N)

1	colour model	I
2	colour coordinates	L(L(R))

(The number of lists must match the specified colour model.)

Errors:

003 *Ignoring function, function requires state (PHOP,WSOP,*,*)*

054 *Ignoring function, the specified workstation is not open*

059 *Ignoring function, the specified workstation does not have output capability (i.e., the workstation category is neither OUTPUT, OUTIN, nor MO)*

103 *Ignoring function, setting this bundle table entry would exceed the maximum number of entries allowed in the workstation bundle table*

121 *Ignoring function, the colour mapping index is less than one*

110 *Ignoring function, the specified colour model is not available on the workstation*

113 *Ignoring function, the colour index value is less than zero*

136 *Ignoring function, one of the components of the colour specification is out of range*

125 *Ignoring function, the total of the colour range fields in all the table entries is too large*

126 *Ignoring function, the specified colour mapping method is not available on the specified workstation*

138 *Ignoring function, one or more of the fields within the specified data record is in error*

5.4 Inquiry functions

5.4.1 Introduction

Inquiry functions return values from the various state lists. The data types of the values and the default values of the state list entries are summarized in 6.1 through 6.6. Errors detected by inquiry functions are reported through an error indicator parameter (see ISO/IEC 9592-1). The ERROR HANDLING function is not called. Some inquiry functions that retrieve values from the workstation state lists have an input parameter of type 'Enumeration' that can take the following values:

a) SET: the values returned are those specified by the application program.

b) REALIZED: the values returned are those used by the workstation when the specified values are mapped to the available values on the workstation.

Inquiries for predefined representations in the workstation description table have no such parameter, unlike the corresponding inquiries for the representations in the workstation state list. The values of predefined representations are available on the workstation. Thus all values returned from a predefined representation are such that, if used by an application program to set a representation, a subsequent inquiry for that representation in the workstation state list would return the same values whether SET or REALIZED was specified.

5.4.2 Inquiry functions for workstation state list

INQUIRE POLYLINE REPRESENTATION PLUS **(PHOP,WSOP,*,*)**

In	workstation identifier		WI
In	polyline index	(1...n)	I
In	type of returned values	(SET,REALIZED)	E
Out	error indicator		I
Out	linetype		I
Out	linewidth scale factor		R
Out	polyline colour		GCOLR
Out	polyline shading method		I
Out	curve approximation criteria type		I
Out	curve approximation criteria data record		D

Effect: If the inquired information is available, the error indicator is returned as 0 and values are returned in the output parameters. See SET CURVE APPROXIMATION CRITERIA for a description of the curve approximation criteria data record.

If the specified polyline index is not present in the polyline bundle table and the specified type of returned value is REALIZED, the representation for polyline index 1 is returned.

If the inquired information is not available, the values returned in the output parameters are implementation dependent and the error indicator is set to one of the following error numbers to indicate the reason for non-availability:

003 *Ignoring function, function requires state (PHOP,WSOP,*,*)*

054 *Ignoring function, the specified workstation is not open*

059 *Ignoring function, the specified workstation does not have output capability (i.e., the workstation category is neither OUTPUT, OUTIN, nor MO)*

100 *Ignoring function, the bundle index value is less than one*

101 *Ignoring function, the specified representation has not been defined*

References: 4.5.1, 4.5.2, 4.5.4.9, 4.6.4.6

Errors: *none*

INQUIRE POLYMARKER REPRESENTATION PLUS **(PHOP,WSOP,*,*)**

In	workstation identifier		WI
In	polymarker index	(1...n)	I
In	type of returned values	(SET,REALIZED)	E
Out	error indicator		I
Out	marker type		I
Out	marker size scale factor		R
Out	polymarker colour		GCOLR

Effect: If the inquired information is available, the error indicator is returned as 0 and values are returned in the output parameters.

If the specified polymarker index is not present in the polymarker bundle table and the specified type of returned value is REALIZED, the representation for polymarker index 1 is returned.

If the inquired information is not available, the values returned in the output parameters are implementation dependent and the error indicator is set to one of the following error numbers to indicate the reason for non-availability:

003 *Ignoring function, function requires state (PHOP,WSOP,*,*)*

054 *Ignoring function, the specified workstation is not open*

059 *Ignoring function, the specified workstation does not have output capability (i.e., the workstation category is neither OUTPUT, OUTIN, nor MO)*

100 *Ignoring function, the bundle index value is less than one*

101 *Ignoring function, the specified representation has not been defined*

References: 4.5.1, 4.5.2

Errors: *none*

INQUIRE TEXT REPRESENTATION PLUS **(PHOP,WSOP,*,*)**

In	workstation identifier		WI
In	text index	(1...n)	I
In	type of returned values	(SET,REALIZED)	E
Out	error indicator		I
Out	text font		I
Out	text precision	(STRING,CHAR,STROKE)	E
Out	character expansion factor		R
Out	character spacing		R
Out	text colour		GCOLR

Effect: If the inquired information is available, the error indicator is returned as 0 and values are returned in the output parameters.

If the specified text index is not present in the text bundle table and the specified type of returned value is REALIZED, the representation for text index 1 is returned.

If the inquired information is not available, the values returned in the output parameters are implementation dependent and the error indicator is set to one of the following error numbers to indicate the reason for non-availability:

003 *Ignoring function, function requires state (PHOP,WSOP,*,*)*

054 *Ignoring function, the specified workstation is not open*

059 *Ignoring function, the specified workstation does not have output capability (i.e., the workstation category is neither OUTPUT, OUTIN, nor MO)*

100 *Ignoring function, the bundle index value is less than one*

101 *Ignoring function, the specified representation has not been defined*

References: 4.5.1, 4.5.2

Errors: *none*

INQUIRE INTERIOR REPRESENTATION PLUS **(PHOP,WSOP,*,*)**

In	workstation identifier		WI
In	interior index	(1...n)	I
In	type of returned values	(SET,REALIZED)	E
Out	error indicator		I
Out	interior style	(HOLLOW,SOLID,PATTERN,HATCH,EMPTY)	E
Out	interior style index		I
Out	interior colour		GCOLR
Out	interior shading method		I

Effect: If the inquired information is available, the error indicator is returned as 0 and values are returned in the output parameters.

If the specified interior index is not present in the interior bundle table and the specified type of returned value is REALIZED, the representation for interior index 1 is returned.

If the inquired information is not available, the values returned in the output parameters are implementation dependent and the error indicator is set to one of the following error numbers to indicate the reason for non-availability:

003 *Ignoring function, function requires state (PHOP,WSOP,*,*)*

054 *Ignoring function, the specified workstation is not open*

059 *Ignoring function, the specified workstation does not have output capability (i.e., the workstation category is neither OUTPUT, OUTIN, nor MO)*

100 *Ignoring function, the bundle index value is less than one*

101 *Ignoring function, the specified representation has not been defined*

References: 4.5.1, 4.5.2, 4.6.1, 4.6.4.7

Errors: *none*

INQUIRE EDGE REPRESENTATION PLUS **(PHOP,WSOP,*,*)**

In	workstation identifier		WI
In	edge index	(1...n)	I
In	type of returned values	(SET,REALIZED)	E
Out	error indicator		I
Out	edge flag	(OFF,ON)	E
Out	edgetype		I
Out	edgewidth scale factor		R
Out	edge colour		GCOLR

Effect: If the inquired information is available, the error indicator is returned as 0 and values are returned in the output parameters.

If the specified edge index is not present in the edge bundle table and the specified type of returned value is REALIZED, the representation for edge index 1 is returned.

If the inquired information is not available, the values returned in the output parameters are implementation dependent and the error indicator is set to one of the following error numbers to indicate the reason for non-availability:

003 *Ignoring function, function requires state (PHOP,WSOP,*,*)*

054 *Ignoring function, the specified workstation is not open*

059 *Ignoring function, the specified workstation does not have output capability (i.e., the workstation category is neither OUTPUT, OUTIN, nor MO)*

100 *Ignoring function, the bundle index value is less than one*

101 *Ignoring function, the specified representation has not been defined*

References: 4.5.1, 4.5.2

Errors: *none*

INQUIRE LIST OF DATA MAPPING INDICES **(PHOP,WSOP,*,*)**

In	workstation identifier		WI
Out	error indicator		I
Out	list of defined data mapping indices	(1...n)	L(I)

Effect: If the inquired information is available, the error indicator is returned as 0 and values are returned in the output parameters.

If the inquired information is not available, the values returned in the output parameters are implementation dependent and the error indicator is set to one of the following error numbers to indicate the reason for non-availability:

003 *Ignoring function, function requires state (PHOP,WSOP,*,*)*

054 *Ignoring function, the specified workstation is not open*

059 *Ignoring function, the specified workstation does not have output capability (i.e., the workstation category is neither OUTPUT, OUTIN, nor MO)*

References: 4.6.2

Errors: *none*

INQUIRE DATA MAPPING REPRESENTATION **(PHOP,WSOP,*,*)**

In	workstation identifier		WI
In	data mapping index	(1...n)	I
In	type of returned values	(SET,REALIZED)	E
Out	error indicator		I
Out	data mapping method		I
Out	data mapping data record		D

Effect: If the inquired information is available, the error indicator is returned as 0 and values are returned in the output parameters. See SET DATA MAPPING METHOD for a description of the data mapping data record.

If the specified data mapping index is not present in the data mapping bundle table and the specified type of returned value is REALIZED, the representation for data mapping index 1 is returned.

If the inquired information is not available, the values returned in the output parameters are implementation dependent and the error indicator is set to one of the following error numbers to indicate the reason for non-availability:

003 *Ignoring function, function requires state (PHOP,WSOP,*,*)*

054 *Ignoring function, the specified workstation is not open*

059 *Ignoring function, the specified workstation does not have output capability (i.e., the workstation category is neither OUTPUT, OUTIN, nor MO)*

100 *Ignoring function, the bundle index value is less than one*

101 *Ignoring function, the specified representation has not been defined*

References: 4.6.1, 4.6.2

Errors: *none*

INQUIRE LIST OF REFLECTANCE INDICES **(PHOP,WSOP,*,*)**

In	workstation identifier		WI
Out	error indicator		I
Out	list of defined reflectance indices	(1...n)	L(I)

Effect: If the inquired information is available, the error indicator is returned as 0 and values are returned in the output parameters.

If the inquired information is not available, the values returned in the output parameters are implementation dependent and the error indicator is set to one of the following error numbers to indicate the reason for non-availability:

003 *Ignoring function, function requires state (PHOP,WSOP,*,*)*

054 *Ignoring function, the specified workstation is not open*

059 *Ignoring function, the specified workstation does not have output capability (i.e., the workstation category is neither OUTPUT, OUTIN, nor MO)*

References: 4.5.2

Errors: *none*

INQUIRE REFLECTANCE REPRESENTATION **(PHOP,WSOP,*,*)**

In	workstation identifier		WI
In	reflectance index	(1...n)	I
In	type of returned values	(SET,REALIZED)	E
Out	error indicator		I
Out	reflectance model		I
Out	reflectance properties type		I
Out	reflectance properties data record		D

Effect: If the inquired information is available, the error indicator is returned as 0 and values are returned in the output parameters. See SET REFLECTANCE PROPERTIES for a description of the reflectance properties data record.

If the specified reflectance index is not present in the reflectance bundle table and the specified type of returned value is REALIZED, the representation for reflectance index 1 is returned.

If the inquired information is not available, the values returned in the output parameters are implementation dependent and the error indicator is set to one of the following error numbers to indicate the reason for non-availability:

003 *Ignoring function, function requires state (PHOP,WSOP,*,*)*

054 *Ignoring function, the specified workstation is not open*

059 *Ignoring function, the specified workstation does not have output capability (i.e., the workstation category is neither OUTPUT, OUTIN, nor MO)*

100 *Ignoring function, the bundle index value is less than one*

101 *Ignoring function, the specified representation has not been defined*

References: 4.5.1, 4.5.2, 4.6.1, 4.6.3

Errors: *none*

INQUIRE LIST OF PARAMETRIC SURFACE INDICES **(PHOP,WSOP,*,*)**

In	workstation identifier		WI
Out	error indicator		I
Out	list of defined parametric surface indices	(1...n)	L(I)

Effect: If the inquired information is available, the error indicator is returned as 0 and values are returned in the output parameters.

If the inquired information is not available, the values returned in the output parameters are implementation dependent and the error indicator is set to one of the following error numbers to indicate the reason for non-availability:

003 *Ignoring function, function requires state (PHOP,WSOP,*,*)*

054 *Ignoring function, the specified workstation is not open*

059 *Ignoring function, the specified workstation does not have output capability (i.e., the workstation category is neither OUTPUT, OUTIN, nor MO)*

References: 4.5.2

Errors: *none*

INQUIRE PARAMETRIC SURFACE REPRESENTATION **(PHOP,WSOP,*,*)**

In	workstation identifier		WI
In	parametric surface bundle index	(1...n)	I
In	type of returned values	(SET,REALIZED)	E
Out	error indicator		I
Out	surface approximation criteria type		I
Out	surface approximation criteria data record		D
Out	parametric surface characteristics type		I
Out	parametric surface characteristics data record		D

Effect: If the inquired information is available, the error indicator is returned as 0 and values are returned in the output parameters. See SET SURFACE APPROXIMATION CRITERIA for a description of the surface approximation criteria data record. See SET PARAMETRIC SURFACE CHARACTERISTICS for a description of the parametric surface characteristics data record.

If the specified parametric surface index is not present in the parametric surface bundle table and the specified type of returned value is REALIZED, the representation for reflectance index 1 is returned.

If the inquired information is not available, the values returned in the output parameters are implementation dependent and the error indicator is set to one of the following error numbers to indicate the reason for non-availability:

003 *Ignoring function, function requires state (PHOP,WSOP,*,*)*

054 *Ignoring function, the specified workstation is not open*

059 *Ignoring function, the specified workstation does not have output capability (i.e., the workstation category is neither OUTPUT, OUTIN, nor MO)*

100 *Ignoring function, the bundle index value is less than one*

101 *Ignoring function, the specified representation has not been defined*

References: 4.5.2, 4.5.4.11

Errors: *none*

INQUIRE PATTERN REPRESENTATION PLUS **(PHOP,WSOP,*,*)**

In	workstation identifier		WI
In	pattern index	(1...n)	I
In	type of returned values	(SET,REALIZED)	E
Out	error indicator		I
Out	colour type		I
Out	pattern colour array		A(COLRV)

Effect: If the inquired information is available, the error indicator is returned as 0 and values are returned in the output parameters.

If the specified pattern index is not present in the pattern table and the specified type of returned value is REALIZED, the representation for pattern index 1 is returned. (Pattern index 1 is present if interior style PATTERN is supported on the workstation.)

If the inquired information is not available, the values returned in the output parameters are implementation dependent and the error indicator is set to one of the following error numbers to indicate the reason for non-availability:

003 *Ignoring function, function requires state (PHOP,WSOP,*,*)*

054 *Ignoring function, the specified workstation is not open*

059 *Ignoring function, the specified workstation does not have output capability (i.e., the workstation category is neither OUTPUT, OUTIN, nor MO)*

101 *Ignoring function, the specified representation has not been defined*

109 *Ignoring function, interior style PATTERN is not supported on the workstation*

112 *Ignoring function, the pattern index value is less than one*

References: 4.5.1, 4.5.2, 4.5.4.11

Errors: *none*

INQUIRE LIST OF LIGHT SOURCE INDICES **(PHOP,WSOP,*,*)**

In	workstation identifier		WI
Out	error indicator		I
Out	list of defined light source indices	(1...n)	L(I)

Effect: If the inquired information is available, the error indicator is returned as 0 and values are returned in the output parameters.

If the inquired information is not available, the values returned in the output parameters are implementation dependent and the error indicator is set to one of the following error numbers to indicate the reason for non-availability:

003 *Ignoring function, function requires state (PHOP,WSOP,*,*)*

054 *Ignoring function, the specified workstation is not open*

059 *Ignoring function, the specified workstation does not have output capability (i.e., the workstation category is neither OUTPUT, OUTIN, nor MO)*

References: 4.6.3.3

Errors: *none*

INQUIRE LIGHT SOURCE REPRESENTATION **(PHOP,WSOP,*,*)**

In	workstation identifier		WI
In	light source index	(1...n)	I
In	type of returned values	(SET,REALIZED)	E
Out	error indicator		I
Out	light source type		I
Out	light source data record		D

Effect: If the inquired information is available, the error indicator is returned as 0 and values are returned in the output parameters.

If the specified light source index is not present in the light source table and the specified type of returned value is REALIZED, the representation for light source index 1 is returned.

See SET LIGHT SOURCE REPRESENTATION for the defined data record contents.

If the inquired information is not available, the values returned in the output parameters are implementation dependent and the error indicator is set to one of the following error numbers to indicate the reason for non-availability:

003 *Ignoring function, function requires state (PHOP,WSOP,*,*)*

054 *Ignoring function, the specified workstation is not open*

059 *Ignoring function, the specified workstation does not have output capability (i.e., the workstation category is neither OUTPUT, OUTIN, nor MO)*

101 *Ignoring function, the specified representation has not been defined*

129 *Ignoring function, the light source index is less than one*

References: 4.5.1, 4.6.1, 4.6.3

Errors: *none*

INQUIRE LIST OF DEPTH CUE INDICES **(PHOP,WSOP,*,*)**

In	workstation identifier		WI
Out	error indicator		I
Out	list of defined depth cue indices	(1...n)	L(I)

Effect: If the inquired information is available, the error indicator is returned as 0 and values are returned in the output parameters.

If the inquired information is not available, the values returned in the output parameters are implementation dependent and the error indicator is set to one of the following error numbers to indicate the reason for non-availability:

003 *Ignoring function, function requires state (PHOP,WSOP,*,*)*

054 *Ignoring function, the specified workstation is not open*

059 *Ignoring function, the specified workstation does not have output capability (i.e., the workstation category is neither OUTPUT, OUTIN, nor MO)*

References: 4.6.6

Errors: *none*

INQUIRE DEPTH CUE REPRESENTATION **(PHOP,WSOP,*,*)**

In	workstation identifier		WI
In	depth cue index	(1...n)	I
In	type of returned values	(SET,REALIZED)	E
Out	error indicator		I
Out	depth cue mode	(SUPPRESSED,ALLOWED)	E
Out	depth cue reference planes	NPC	2×R
Out	depth cue scale factors	[0,1]	2×R
Out	depth cue colour		GCOLR

Effect: If the inquired information is available, the error indicator is returned as 0 and values are returned in the output parameters.

If the specified depth cue index is not present in the depth cue table and the specified type of returned value is REALIZED, the representation for depth cue index 1 is returned.

If the inquired information is not available, the values returned in the output parameters are implementation dependent and the error indicator is set to one of the following error numbers to indicate the reason for non-availability:

003 *Ignoring function, function requires state (PHOP,WSOP,*,*)*

054 *Ignoring function, the specified workstation is not open*

059 *Ignoring function, the specified workstation does not have output capability (i.e., the workstation category is neither OUTPUT, OUTIN, nor MO)*

101 *Ignoring function, the specified representation has not been defined*

120 *Ignoring function, the depth cue index is less than one*

References: 4.5.1, 4.6.1, 4.6.6

Errors: *none*

INQUIRE COLOUR MAPPING STATE **(PHOP,WSOP,*,*)**

In	workstation identifier	WI
In	colour mapping method	I
Out	error indicator	I
Out	colour mapping state data record	D

Effect: If the inquired information is available, the error indicator is returned as 0 and values are returned in the output parameters.

If the inquired information is not available, the values returned in the output parameters are implementation dependent and the error indicator is set to one of the following error numbers to indicate the reason for non-availability:

003 *Ignoring function, function requires state (PHOP,WSOP,*,*)*

054 *Ignoring function, the specified workstation is not open*

059 *Ignoring function, the specified workstation does not have output capability (i.e., the workstation category is neither OUTPUT, OUTIN, nor MO)*

126 *Ignoring function, the specified colour mapping method is not available on the specified workstation*

References: 4.6.7

Data Record:

Colour mapping method = 1 (TRUE)

1	number of true colours available	I

Colour mapping method = 2 (PSEUDO)

1	number of pseudo colour entries available	I

Colour mapping method = 3 (PSEUDO-N)

(no data record contents)

Errors: *none*

INQUIRE LIST OF COLOUR MAPPING INDICES **(PHOP,WSOP,*,*)**

In	workstation identifier		N
Out	error indicator		I
Out	list of defined colour mapping indices	(1...n)	L(I)

Effect: If the inquired information is available, the error indicator is returned as 0 and values are returned in the output parameters.

If the inquired information is not available, the values returned in the output parameters are implementation dependent and the error indicator is set to one of the following error numbers to indicate the reason for non-availability:

003 *Ignoring function, function requires state (PHOP,WSOP,*,*)*

054 *Ignoring function, the specified workstation is not open*

059 *Ignoring function, the specified workstation does not have output capability (i.e., the workstation category is neither OUTPUT, OUTIN, nor MO)*

References: 4.6.7

Errors: *none*

INQUIRE COLOUR MAPPING REPRESENTATION **(PHOP,WSOP,*,*)**

In	workstation identifier		WI
In	colour mapping index	(1...n)	I
In	type of returned values	(SET,REALIZED)	E
Out	error indicator		I
Out	colour mapping method		I
Out	colour mapping data record		D

Effect: If the inquired information is available, the error indicator is returned as 0 and values are returned in the output parameters.

See SET COLOUR MAPPING REPRESENTATION for the defined data record formats that may be returned in the data record parameter.

If the specified colour mapping index is not present in the colour mapping table and the specified type of returned value is REALIZED, the representation for colour mapping index 1 is returned.

If the inquired information is not available, the values returned in the output parameters are implementation dependent and the error indicator is set to one of the following error numbers to indicate the reason for non-availability:

003 *Ignoring function, function requires state (PHOP,WSOP,*,*)*

054 *Ignoring function, the specified workstation is not open*

059 *Ignoring function, the specified workstation does not have output capability (i.e., the workstation category is neither OUTPUT, OUTIN, nor MO)*

121 *Ignoring function, the colour mapping index is less than one*

101 *Ignoring function, the specified representation has not been defined*

References: 4.6.1, 4.6.7

Errors: *none*

5.4.3 Inquiry functions for workstation description table

INQUIRE DIRECT COLOUR MODEL FACILITIES **(PHOP,*,*,*)**

In	workstation type	W
Out	error indicator	I
Out	list of available directly specifiable colour models	L(I)

Effect: If the inquired information is available, the error indicator is returned as 0 and values are returned in the output parameters.

If the inquired information is not available, the values returned in the output parameters are implementation dependent and the error indicator is set to one of the following error numbers to indicate the reason for non-availability:

002 *Ignoring function, function requires state (PHOP,*,*,*)*

052 *Ignoring function, workstation type not recognized by the implementation*

051 *Ignoring function, this information is not yet available for this generic workstation type; open a workstation of this type and use the specific workstation type*

059 *Ignoring function, the specified workstation does not have output capability (i.e., the workstation category is neither OUTPUT, OUTIN, nor MO)*

062 *Ignoring function, this information is not available for this MO workstation type*

References: 4.5.1

Errors: *none*

INQUIRE RENDERING COLOUR MODEL FACILITIES **(PHOP,*,*,*)**

In	workstation type	W
Out	error indicator	I
Out	list of available rendering colour models	L(I)

Effect: If the inquired information is available, the error indicator is returned as 0 and values are returned in the output parameters.

If the inquired information is not available, the values returned in the output parameters are implementation dependent and the error indicator is set to one of the following error numbers to indicate the reason for non-availability:

002 *Ignoring function, function requires state (PHOP,*,*,*)*

052 *Ignoring function, workstation type not recognized by the implementation*

051 *Ignoring function, this information is not yet available for this generic workstation type; open a workstation of this type and use the specific workstation type*

059 *Ignoring function, the specified workstation does not have output capability (i.e., the workstation category is neither OUTPUT, OUTIN, nor MO)*

062 *Ignoring function, this information is not available for this MO workstation type*

References: 4.6.5

Errors: *none*

INQUIRE DYNAMICS OF WORKSTATION ATTRIBUTES PLUS **(PHOP,*,*,*)**

In	workstation type		W
Out	error indicator		I
Out	data mapping representation	(IRG,IMM,CBS)	E
Out	reflectance representation	(IRG,IMM,CBS)	E
Out	para. surface representation	(IRG,IMM,CBS)	E
Out	workstation light source representation	(IRG,IMM,CBS)	E
Out	depth cue representation	(IRG,IMM,CBS)	E
Out	colour mapping representation	(IRG,IMM,CBS)	E

Effect: If the inquired information is available, the error indicator is returned as 0 and values are returned in the output parameters.

IRG means that implicit regeneration is necessary; IMM means the action is performed immediately; CBS means that the change can be simulated.

If the inquired information is not available, the values returned in the output parameters are implementation dependent and the error indicator is set to one of the following error numbers to indicate the reason for non-availability:

002 *Ignoring function, function requires state (PHOP,*,*,*)*

052 *Ignoring function, workstation type not recognized by the implementation*

051 *Ignoring function, this information is not yet available for this generic workstation type; open a workstation of this type and use the specific workstation type*

059 *Ignoring function, the specified workstation does not have output capability (i.e., the workstation category is neither OUTPUT, OUTIN, nor MO)*

062 *Ignoring function, this information is not available for this MO workstation type*

References: 4.5.2, 4.7

Errors: *none*

INQUIRE POLYLINE FACILITIES PLUS **(PHOP,*,*,*)**

In	workstation type		W
Out	error indicator		I
Out	number of available linetypes	(-n...-4,4...n)	I
Out	list of available linetypes		L(I)
Out	number of available linewidths	(0...n)	I
Out	nominal linewidth	DC > 0	R
Out	range of linewidths(minimum,maximum)	DC > 0	2×R
Out	list of available shading methods	(-n...n)	L(I)
Out	number of predefined polyline indices	(5...n)	I

Effect: If the inquired information is available, the error indicator is returned as 0 and values are returned in the output parameters.

If the number of available linetypes is positive, the list of available linetypes contains all registered and implementation dependent linetypes supported. If the number of available linetypes is negative, the implementation dependent linetypes are derived directly from the linetype value and the list of available linetypes contains the number of registered linetypes; the number of which is indicated by the absolute value of the number of available linetypes.

If the number of available linewidths is returned as 0, the workstation supports a continuous range of linewidths.

If the inquired information is not available, the values returned in the output parameters are implementation dependent and the error indicator is set to one of the following error numbers to indicate the reason for non-availability:

002 *Ignoring function, function requires state (PHOP,*,*,*)*

052 *Ignoring function, workstation type not recognized by the implementation*

051 *Ignoring function, this information is not yet available for this generic workstation type; open a workstation of this type and use the specific workstation type*

059 *Ignoring function, the specified workstation does not have output capability (i.e., the workstation category is neither OUTPUT, OUTIN, nor MO)*

062 *Ignoring function, this information is not available for this MO workstation type*

References: 4.5.2, 4.6.4.6

Errors: *none*

INQUIRE PREDEFINED POLYLINE REPRESENTATION PLUS **(PHOP,*,*,*)**

In	workstation type		W
In	predefined polyline index	(1...n)	I
Out	error indicator		I
Out	linetype		I
Out	linewidth scale factor		R
Out	polyline colour		GCOLR
Out	polyline shading method		I
Out	curve approximation criteria type		I
Out	curve approximation criteria data record		D

Effect: If the inquired information is available, the error indicator is returned as 0 and values are returned in the output parameters. See SET CURVE APPROXIMATION CRITERIA for a description of the curve approximation criteria data record.

If the inquired information is not available, the values returned in the output parameters are implementation dependent and the error indicator is set to one of the following error numbers to indicate the reason for non-availability:

002 *Ignoring function, function requires state (PHOP,*,*,*)*

052 *Ignoring function, workstation type not recognized by the implementation*

051 *Ignoring function, this information is not yet available for this generic workstation type; open a workstation of this type and use the specific workstation type*

059 *Ignoring function, the specified workstation does not have output capability (i.e., the workstation category is neither OUTPUT, OUTIN, nor MO)*

062 *Ignoring function, this information is not available for this MO workstation type*

100 *Ignoring function, the bundle index value is less than one*

102 *Ignoring function, the specified representation has not been predefined on this workstation*

References: 4.5.1, 4.5.2, 4.5.4.9, 4.6.4.6

Errors: *none*

INQUIRE PREDEFINED POLYMARKER REPRESENTATION PLUS **(PHOP,*,*,*)**

In	workstation type		W
In	predefined polymarker index	(1...n)	I
Out	error indicator		I
Out	marker type		I
Out	marker size scale factor		R
Out	polymarker colour		GCOLR

Effect: If the inquired information is available, the error indicator is returned as 0 and values are returned in the output parameters.

If the inquired information is not available, the values returned in the output parameters are implementation dependent and the error indicator is set to one of the following error numbers to indicate the reason for non-availability:

002 *Ignoring function, function requires state (PHOP,*,*,*)*

052 *Ignoring function, workstation type not recognized by the implementation*

051 *Ignoring function, this information is not yet available for this generic workstation type; open a workstation of this type and use the specific workstation type*

059 *Ignoring function, the specified workstation does not have output capability (i.e., the workstation category is neither OUTPUT, OUTIN, nor MO)*

062 *Ignoring function, this information is not available for this MO workstation type*

100 *Ignoring function, the bundle index value is less than one*

102 *Ignoring function, the specified representation has not been predefined on this workstation*

References: 4.5.1, 4.5.2

Errors: *none*

INQUIRE PREDEFINED TEXT REPRESENTATION PLUS **(PHOP,*,*,*)**

In	workstation type		W
In	predefined text index	(1...n)	I
Out	error indicator		I
Out	text font		I
Out	text precision	(STRING,CHAR,STROKE)	I
Out	character expansion factor		R
Out	character spacing		R
Out	text colour		GCOLR

Effect: If the inquired information is available, the error indicator is returned as 0 and values are returned in the output parameters.

If the inquired information is not available, the values returned in the output parameters are implementation dependent and the error indicator is set to one of the following error numbers to indicate the reason for non-availability:

002 *Ignoring function, function requires state (PHOP,*,*,*)*

052 *Ignoring function, workstation type not recognized by the implementation*

051 *Ignoring function, this information is not yet available for this generic workstation type; open a workstation of this type and use the specific workstation type*

059 *Ignoring function, the specified workstation does not have output capability (i.e., the workstation category is neither OUTPUT, OUTIN, nor MO)*

062 *Ignoring function, this information is not available for this MO workstation type*

100 *Ignoring function, the bundle index value is less than one*

102 *Ignoring function, the specified representation has not been predefined on this workstation*

References: 4.5.1, 4.5.2

Errors: *none*

INQUIRE INTERIOR FACILITIES PLUS **(PHOP,*,*,*)**

In	workstation type			W
Out	error indicator			I
Out	list of available interior styles	(HOLLOW,SOLID,PATTERN,HATCH,EMPTY)		E
Out	number of available hatch styles			I
Out	list of available hatch styles			L(I)
Out	list of available shading methods			L(I)
Out	number of predefined interior indices		(5...n)	I

Effect: If the inquired information is available, the error indicator is returned as 0 and values are returned in the output parameters.

If the number of available hatch styles is positive, the list of available hatch styles contains all registered and implementation dependent hatch styles supported. If the number of available hatch styles is negative, the implementation dependent hatch styles are derived directly from the hatch styles value and the list of available hatch styles contains the number of registered hatch styles; the number of which is indicated by the absolute value of the number of available hatch styles.

If the inquired information is not available, the values returned in the output parameters are implementation dependent and the error indicator is set to one of the following error numbers to indicate the reason for non-availability:

002 *Ignoring function, function requires state (PHOP,*,*,*)*

052 *Ignoring function, workstation type not recognized by the implementation*

051 *Ignoring function, this information is not yet available for this generic workstation type; open a workstation of this type and use the specific workstation type*

059 *Ignoring function, the specified workstation does not have output capability (i.e., the workstation category is neither OUTPUT, OUTIN, nor MO)*

062 *Ignoring function, this information is not available for this MO workstation type*

References: 4.5.2, 4.6.4.7

Errors: *none*

INQUIRE PREDEFINED INTERIOR REPRESENTATION PLUS **(PHOP,*,*,*)**

In	workstation type		W
In	predefined interior index	(1...n)	I
Out	error indicator		I
Out	interior style	(HOLLOW,SOLID,PATTERN,HATCH,EMPTY)	E
Out	interior style index		I
Out	interior colour		GCOLR
Out	interior shading method		I

Effect: If the inquired information is available, the error indicator is returned as 0 and values are returned in the output parameters.

If the inquired information is not available, the values returned in the output parameters are implementation dependent and the error indicator is set to one of the following error numbers to indicate the reason for non-availability:

002 *Ignoring function, function requires state (PHOP,*,*,*)*

052 *Ignoring function, workstation type not recognized by the implementation*

051 *Ignoring function, this information is not yet available for this generic workstation type; open a workstation of this type and use the specific workstation type*

059 *Ignoring function, the specified workstation does not have output capability (i.e., the workstation category is neither OUTPUT, OUTIN, nor MO)*

062 *Ignoring function, this information is not available for this MO workstation type*

100 *Ignoring function, the bundle index value is less than one*

102 *Ignoring function, the specified representation has not been predefined on this workstation*

References: 4.5.1, 4.5.2, 4.6.4.7

Errors: *none*

INQUIRE PREDEFINED EDGE REPRESENTATION PLUS **(PHOP,*,*,*)**

In	workstation type		W
In	predefined edge index	(1...n)	I
Out	error indicator		I
Out	edge flag	(OFF,ON)	E
Out	edgetype		I
Out	edgewidth scale factor		R
Out	edge colour		GCOLR

Effect: If the inquired information is available, the error indicator is returned as 0 and values are returned in the output parameters.

If the inquired information is not available, the values returned in the output parameters are implementation dependent and the error indicator is set to one of the following error numbers to indicate the reason for non-availability:

002 *Ignoring function, function requires state (PHOP,*,*,*)*

052 *Ignoring function, workstation type not recognized by the implementation*

051 *Ignoring function, this information is not yet available for this generic workstation type; open a workstation of this type and use the specific workstation type*

059 *Ignoring function, the specified workstation does not have output capability (i.e., the workstation category is neither OUTPUT, OUTIN, nor MO)*

062 *Ignoring function, this information is not available for this MO workstation type*

100 *Ignoring function, the bundle index value is less than one*

102 *Ignoring function, the specified representation has not been predefined on this workstation*

References: 4.5.1, 4.5.2

Errors: *none*

INQUIRE DATA MAPPING FACILITIES **(PHOP,*,*,*)**

In	workstation type		W
Out	error indicator		I
Out	list of available data mapping methods		L(I)
Out	number of predefined data mapping indices	(1...n)	I

Effect: If the inquired information is available, the error indicator is returned as 0 and values are returned in the output parameters.

If the inquired information is not available, the values returned in the output parameters are implementation dependent and the error indicator is set to one of the following error numbers to indicate the reason for non-availability:

002 *Ignoring function, function requires state (PHOP,*,*,*)*

052 *Ignoring function, workstation type not recognized by the implementation*

051 *Ignoring function, this information is not yet available for this generic workstation type; open a workstation of this type and use the specific workstation type*

059 *Ignoring function, the specified workstation does not have output capability (i.e., the workstation category is neither OUTPUT, OUTIN, nor MO)*

062 *Ignoring function, this information is not available for this MO workstation type*

References: 4.6.2

Errors: *none*

INQUIRE PREDEFINED DATA MAPPING REPRESENTATION **(PHOP,*,*,*)**

In	workstation type		W
In	predefined data mapping index	(1...n)	I
Out	error indicator		I
Out	data mapping method		I
Out	data mapping data record		D

Effect: If the inquired information is available, the error indicator is returned as 0 and values are returned in the output parameters. See SET DATA MAPPING METHOD for a description of the data mapping data record.

If the inquired information is not available, the values returned in the output parameters are implementation dependent and the error indicator is set to one of the following error numbers to indicate the reason for non-availability:

002 *Ignoring function, function requires state (PHOP,*,*,*)*

052 *Ignoring function, workstation type not recognized by the implementation*

051 *Ignoring function, this information is not yet available for this generic workstation type; open a workstation of this type and use the specific workstation type*

059 *Ignoring function, the specified workstation does not have output capability (i.e., the workstation category is neither OUTPUT, OUTIN, nor MO)*

062 *Ignoring function, this information is not available for this MO workstation type*

100 *Ignoring function, the bundle index value is less than one*

102 *Ignoring function, the specified representation has not been predefined on this workstation*

References: 4.6.2

Errors: *none*

INQUIRE REFLECTANCE FACILITIES **(PHOP,*,*,*)**

In	workstation type		W
Out	error indicator		I
Out	list of available reflectance models		L(I)
Out	number of predefined reflectance indices	(1...n)	I

Effect: If the inquired information is available, the error indicator is returned as 0 and values are returned in the output parameters.

If the inquired information is not available, the values returned in the output parameters are implementation dependent and the error indicator is set to one of the following error numbers to indicate the reason for non-availability:

002 *Ignoring function, function requires state (PHOP,*,*,*)*

052 *Ignoring function, workstation type not recognized by the implementation*

051 *Ignoring function, this information is not yet available for this generic workstation type; open a workstation of this type and use the specific workstation type*

059 *Ignoring function, the specified workstation does not have output capability (i.e., the workstation category is neither OUTPUT, OUTIN, nor MO)*

062 *Ignoring function, this information is not available for this MO workstation type*

References: 4.5.2

Errors: *none*

INQUIRE PREDEFINED REFLECTANCE REPRESENTATION **(PHOP,*,*,*)**

In	workstation type		W
In	predefined reflectance index	(1...n)	I
Out	error indicator		I
Out	reflectance model		I
Out	reflectance properties type		I
Out	reflectance properties data record		D

Effect: If the inquired information is available, the error indicator is returned as 0 and values are returned in the output parameters. See SET REFLECTANCE PROPERTIES for a description of the reflectance properties data record.

If the inquired information is not available, the values returned in the output parameters are implementation dependent and the error indicator is set to one of the following error numbers to indicate the reason for non-availability:

002 *Ignoring function, function requires state (PHOP,*,*,*)*

052 *Ignoring function, workstation type not recognized by the implementation*

051 *Ignoring function, this information is not yet available for this generic workstation type; open a workstation of this type and use the specific workstation type*

059 *Ignoring function, the specified workstation does not have output capability (i.e., the workstation category is neither OUTPUT, OUTIN, nor MO)*

062 *Ignoring function, this information is not available for this MO workstation type*

100 *Ignoring function, the bundle index value is less than one*

102 *Ignoring function, the specified representation has not been predefined on this workstation*

References: 4.5.1, 4.5.2, 4.6.3

Errors: *none*

INQUIRE CURVE AND SURFACE FACILITIES **(PHOP,*,*,*)**

In	workstation type		W
Out	error indicator		I
Out	maximum non-uniform B-spline curve order supported	(6...n)	I
Out	maximum non-uniform B-spline surface order supported	(6...n)	I
Out	maximum trimming curve order supported	(4...n)	I
Out	list of available curve approximation criteria types		L(I)
Out	list of available surface approximation criteria types		L(I)
Out	list of available trimming curve approximation criteria types		L(I)
Out	list of available parametric surface characteristics types		L(I)
Out	number of predefined parametric surface indices	(1...n)	I

Effect: If the inquired information is available, the error indicator is returned as 0 and values are returned in the output parameters.

If the inquired information is not available, the values returned in the output parameters are implementation dependent and the error indicator is set to one of the following error numbers to indicate the reason for non-availability:

002 *Ignoring function, function requires state (PHOP,*,*,*)*

052 *Ignoring function, workstation type not recognized by the implementation*

051 *Ignoring function, this information is not yet available for this generic workstation type; open a workstation of this type and use the specific workstation type*

059 *Ignoring function, the specified workstation does not have output capability (i.e., the workstation category is neither OUTPUT, OUTIN, nor MO)*

062 *Ignoring function, this information is not available for this MO workstation type*

References: 4.4.9, 4.4.11, 4.4.12, 4.5.2, 4.5.4.9, 4.5.4.11

Errors: *none*

INQUIRE PREDEFINED PARAMETRIC SURFACE REPRESENTATION **(PHOP,*,*,*)**

In	workstation type		W
In	predefined parametric surface index	(1...n)	I
Out	error indicator		I
Out	surface approximation criteria type		I
Out	surface approximation criteria data record		D
Out	parametric surface characteristics type		I
Out	parametric surface characteristics data record		D

Effect: If the inquired information is available, the error indicator is returned as 0 and values are returned in the output parameters. See SET SURFACE APPROXIMATION CRITERIA for a description of the surface approximation criteria data record. See SET PARAMETRIC SURFACE CHARACTERISTICS for a description of the parametric surface characteristics data record.

If the inquired information is not available, the values returned in the output parameters are implementation dependent and the error indicator is set to one of the following error numbers to indicate the reason for non-availability:

002 *Ignoring function, function requires state (PHOP,*,*,*)*

052 *Ignoring function, workstation type not recognized by the implementation*

051 *Ignoring function, this information is not yet available for this generic workstation type; open a workstation of this type and use the specific workstation type*

059 *Ignoring function, the specified workstation does not have output capability (i.e., the workstation category is neither OUTPUT, OUTIN, nor MO)*

062 *Ignoring function, this information is not available for this MO workstation type*

100 *Ignoring function, the bundle index value is less than one*

102 *Ignoring function, the specified representation has not been predefined on this workstation*

References: 4.5.2, 4.5.4.11

Errors: *none*

INQUIRE PREDEFINED PATTERN REPRESENTATION PLUS **(PHOP,*,*,*)**

In	workstation type		W
In	predefined pattern index	(1...n)	I
Out	error indicator		I
Out	colour type		I
Out	pattern colour array		A(COLRV)

Effect: If the inquired information is available, the error indicator is returned as 0 and values are returned in the output parameters.

If the inquired information is not available, the values returned in the output parameters are implementation dependent and the error indicator is set to one of the following error numbers to indicate the reason for non-availability:

002 *Ignoring function, function requires state (PHOP,*,*,*)*

052 *Ignoring function, workstation type not recognized by the implementation*

051 *Ignoring function, this information is not yet available for this generic workstation type; open a workstation of this type and use the specific workstation type*

059 *Ignoring function, the specified workstation does not have output capability (i.e., the workstation category is neither OUTPUT, OUTIN, nor MO)*

062 *Ignoring function, this information is not available for this MO workstation type*

100 *Ignoring function, the bundle index value is less than one*

102 *Ignoring function, the specified representation has not been predefined on this workstation*

References: 4.5.1, 4.5.2

Errors: *none*

INQUIRE LIGHT SOURCE FACILITIES **(PHOP,*,*,*)**

In	workstation type		W
Out	error indicator		I
Out	list of available workstation light source types		L(I)
Out	maximum number of simultaneously active non-ambient light sources	(1...n)	I
Out	number of predefined light source indices	(2...n)	I

Effect: If the inquired information is available, the error indicator is returned as 0 and values are returned in the output parameters.

If the inquired information is not available, the values returned in the output parameters are implementation dependent and the error indicator is set to one of the following error numbers to indicate the reason for non-availability:

002 *Ignoring function, function requires state (PHOP,*,*,*)*

052 *Ignoring function, workstation type not recognized by the implementation*

051 *Ignoring function, this information is not yet available for this generic workstation type; open a workstation of this type and use the specific workstation type*

059 *Ignoring function, the specified workstation does not have output capability (i.e., the workstation category is neither OUTPUT, OUTIN, nor MO)*

062 *Ignoring function, this information is not available for this MO workstation type*

References: 4.6.3

Errors: *none*

INQUIRE PREDEFINED LIGHT SOURCE REPRESENTATION **(PHOP,*,*,*)**

In	workstation type		W
In	predefined light source index	(1...n)	I
Out	error indicator		I
Out	light source type		I
Out	light source data record		D

Effect: If the inquired information is available, the error indicator is returned as 0 and values are returned in the output parameters. See SET LIGHT SOURCE REPRESENTATION for a description of the light source data record.

If the inquired information is not available, the values returned in the output parameters are implementation dependent and the error indicator is set to one of the following error numbers to indicate the reason for non-availability:

002 *Ignoring function, function requires state (PHOP,*,*,*)*

052 *Ignoring function, workstation type not recognized by the implementation*

051 *Ignoring function, this information is not yet available for this generic workstation type; open a workstation of this type and use the specific workstation type*

059 *Ignoring function, the specified workstation does not have output capability (i.e., the workstation category is neither OUTPUT, OUTIN, nor MO)*

062 *Ignoring function, this information is not available for this MO workstation type*

102 *Ignoring function, the specified representation has not been predefined on this workstation*

129 *Ignoring function, the light source index is less than one*

References: 4.5.1, 4.6.3

Errors: *none*

INQUIRE DEPTH CUE FACILITIES **(PHOP,*,*,*)**

In	workstation type		W
Out	error indicator		I
Out	number of predefined depth cue indices	(2...n)	I
Out	list of available depth cue modes	(SUPPRESSED, ALLOWED)	L(E)

Effect: If the inquired information is available, the error indicator is returned as 0 and values are returned in the output parameters.

If the inquired information is not available, the values returned in the output parameters are implementation dependent and the error indicator is set to one of the following error numbers to indicate the reason for non-availability:

002 *Ignoring function, function requires state (PHOP,*,*,*)*

052 *Ignoring function, workstation type not recognized by the implementation*

051 *Ignoring function, this information is not yet available for this generic workstation type; open a workstation of this type and use the specific workstation type*

059 *Ignoring function, the specified workstation does not have output capability (i.e., the workstation category is neither OUTPUT, OUTIN, nor MO)*

062 *Ignoring function, this information is not available for this MO workstation type*

References: 4.6.6

Errors: *none*

INQUIRE PREDEFINED DEPTH CUE REPRESENTATION **(PHOP,*,*,*)**

In	workstation type		W
In	predefined depth cue index	(1...n)	I
Out	error indicator		I
Out	depth cue mode	(SUPPRESSED,ALLOWED)	E
Out	depth cue reference planes	NPC	2×R
Out	depth cue scaling	[0,1]	2×R
Out	depth cue colour		GCOLR

Effect: If the inquired information is available, the error indicator is returned as 0 and values are returned in the output parameters.

If the inquired information is not available, the values returned in the output parameters are implementation dependent and the error indicator is set to one of the following error numbers to indicate the reason for non-availability:

002 *Ignoring function, function requires state (PHOP,*,*,*)*

052 *Ignoring function, workstation type not recognized by the implementation*

051 *Ignoring function, this information is not yet available for this generic workstation type; open a workstation of this type and use the specific workstation type*

059 *Ignoring function, the specified workstation does not have output capability (i.e., the workstation category is neither OUTPUT, OUTIN, nor MO)*

062 *Ignoring function, this information is not available for this MO workstation type*

102 *Ignoring function, the specified representation has not been predefined on this workstation*

120 *Ignoring function, the depth cue index is less than one*

References: 4.5.1, 4.6.6

Errors: *none*

INQUIRE COLOUR MAPPING FACILITIES **(PHOP,*,*,*)**

In	workstation type		W
Out	error indicator		I
Out	list of available colour mapping methods		L(I)
Out	number of predefined colour mapping indices	(1...n)	I

Effect: If the inquired information is available, the error indicator is returned as 0 and values are returned in the output parameters.

If the inquired information is not available, the values returned in the output parameters are implementation dependent and the error indicator is set to one of the following error numbers to indicate the reason for non-availability:

002 *Ignoring function, function requires state (PHOP,*,*,*)*

052 *Ignoring function, workstation type not recognized by the implementation*

051 *Ignoring function, this information is not yet available for this generic workstation type; open a workstation of this type and use the specific workstation type*

059 *Ignoring function, the specified workstation does not have output capability (i.e., the workstation category is neither OUTPUT, OUTIN, nor MO)*

062 *Ignoring function, this information is not available for this MO workstation type*

References: 4.6.7

Errors: *none*

INQUIRE COLOUR MAPPING METHOD FACILITIES **(PHOP,*,*,*)**

In	workstation type	W
In	colour mapping method	I
Out	colour mapping method facilities data record	D

Effect: If the inquired information is available, the error indicator is returned as 0 and values are returned in the output parameters.

If the inquired information is not available, the values returned in the output parameters are implementation dependent and the error indicator is set to one of the following error numbers to indicate the reason for non-availability:

002 *Ignoring function, function requires state (PHOP,*,*,*)*

052 *Ignoring function, workstation type not recognized by the implementation*

051 *Ignoring function, this information is not yet available for this generic workstation type; open a workstation of this type and use the specific workstation type*

059 *Ignoring function, the specified workstation does not have output capability (i.e., the workstation category is neither OUTPUT, OUTIN, nor MO)*

062 *Ignoring function, this information is not available for this MO workstation type*

126 *Ignoring function, the specified colour mapping method is not available on the specified workstation*

References: 4.6.7

Data Records:

Colour mapping method = 1 (true colour)

1	number of available true colours	I

Colour mapping method = 2 (pseudo colour)

1	maximum number of pseudo colour entries	I

Colour mapping method = 3 (pseudo-N colour)

(no data record contents)

Errors: *none*

INQUIRE PREDEFINED COLOUR MAPPING REPRESENTATION **(PHOP,*,*,*)**

In	workstation type		W
In	predefined colour mapping index	(1...n)	I
Out	error indicator		I
Out	colour mapping method		I
Out	colour mapping data record		D

Effect: If the inquired information is available, the error indicator is returned as 0 and values are returned in the output parameters. See SET COLOUR MAPPING REPRESENTATION for a description of the colour mapping data record.

If the inquired information is not available, the values returned in the output parameters are implementation dependent and the error indicator is set to one of the following error numbers to indicate the reason for non-availability:

002 *Ignoring function, function requires state (PHOP,*,*,*)*

052 *Ignoring function, workstation type not recognized by the implementation*

051 *Ignoring function, this information is not yet available for this generic workstation type; open a workstation of this type and use the specific workstation type*

059 *Ignoring function, the specified workstation does not have output capability (i.e., the workstation category is neither OUTPUT, OUTIN, nor MO)*

062 *Ignoring function, this information is not available for this MO workstation type*

121 *Ignoring function, the colour mapping index is less than one*

102 *Ignoring function, the specified representation has not been predefined on this workstation*

References: 4.6.7

Errors: *none*

INQUIRE GENERALIZED DRAWING PRIMITIVE 3 **(PHOP,*,*,*)**

In	workstation type		W
In	GDP 3 identifier		G3
Out	error indicator		I
Out	list of sets of attributes used	(see below)	L(E)

Effect: If the inquired information is available, the error indicator is returned as 0 and values are returned in the output parameters.

For registered GDP 3 identifiers the list of sets of attributes used is defined in the ISO International Register of Graphical Items. For implementation dependent GDP 3 identifiers the list of sets of attributes used is workstation dependent.

The enumeration values for the list of attributes used is: POLYLINE, POLYMARKER, TEXT, INTERIOR, EDGE, REFLECTANCE, PARAMETRIC SURFACE.

If the inquired information is not available, the values returned in the output parameters are implementation dependent and the error indicator is set to one of the following error numbers to indicate the reason for non-availability:

002 *Ignoring function, function requires state (PHOP,*,*,*)*

052 *Ignoring function, workstation type not recognized by the implementation*

051 *Ignoring function, this information is not yet available for this generic workstation type; open a workstation of this type and use the specific workstation type*

059 *Ignoring function, the specified workstation does not have output capability (i.e., the workstation category is neither OUTPUT, OUTIN, nor MO)*

064 *Ignoring function, the specified workstation type is not able to generate the specified generalized drawing primitive*

062 *Ignoring function, this information is not available for this MO workstation type*

References: 4.5.3

Errors: *none*

INQUIRE GENERALIZED DRAWING PRIMITIVE **(PHOP,*,*,*)**

In	workstation type		W
In	GDP identifier		G2
Out	error indicator		I
Out	list of sets of attributes used	(see below)	L(E)

Effect: If the inquired information is available, the error indicator is returned as 0 and values are returned in the output parameters.

For registered GDP identifiers the list of sets of attributes used is defined in the ISO International Register of Graphical Items. For implementation dependent GDP identifiers the list of sets of attributes used is workstation dependent.

The enumeration values for the list of attributes used is: POLYLINE, POLYMARKER, TEXT, INTERIOR, EDGE, REFLECTANCE, PARAMETRIC SURFACE.

If the inquired information is not available, the values returned in the output parameters are implementation dependent and the error indicator is set to one of the following error numbers to indicate the reason for non-availability:

002 *Ignoring function, function requires state (PHOP,*,*,*)*

052 *Ignoring function, workstation type not recognized by the implementation*

051 *Ignoring function, this information is not yet available for this generic workstation type; open a workstation of this type and use the specific workstation type*

059 *Ignoring function, the specified workstation does not have output capability (i.e., the workstation category is neither OUTPUT, OUTIN, nor MO)*

064 *Ignoring function, the specified workstation type is not able to generate the specified generalized drawing primitive*

062 *Ignoring function, this information is not available for this MO workstation type*

References: 4.5.3

Errors: *none*

INQUIRE WORKSTATION STATE TABLE LENGTHS PLUS **(PHOP,*,*,*)**

In	workstation type		W
Out	error indicator		I
Out	maximum number of data mapping bundle table entries	(1...n)	I
Out	maximum number of reflectance bundle table entries	(20...n)	I
Out	maximum number of parametric surface bundle table entries	(20...n)	I
Out	maximum number of light source table entries	(5...n)	I
Out	maximum number of depth cue table entries	(6...n)	I
Out	maximum number of colour mapping table entries	(1...n)	I

Effect: If the inquired information is available, the error indicator is returned as 0 and values are returned in the output parameters.

If the inquired information is not available, the values returned in the output parameters are implementation dependent and the error indicator is set to one of the following error numbers to indicate the reason for non-availability:

002 *Ignoring function, function requires state (PHOP,*,*,*)*

052 *Ignoring function, workstation type not recognized by the implementation*

051 *Ignoring function, this information is not yet available for this generic workstation type; open a workstation of this type and use the specific workstation type*

059 *Ignoring function, the specified workstation does not have output capability (i.e., the workstation category is neither OUTPUT, OUTIN, nor MO)*

062 *Ignoring function, this information is not available for this MO workstation type*

References: 4.5.2, 4.6.3.3, 4.6.6, 4.6.7

Errors: *none*

5.4.4 Inquiry functions for structure content

INQUIRE CURRENT ELEMENT TYPE AND SIZE **(PHOP,*,STOP,*)**

Out	error indicator		I
Out	element type	(see below)	E
Out	element size	(0...n)	I

Effect: If the inquired information is available, the error indicator is returned as 0 and values are returned in the output parameters.

The element type of the structure element pointed to by the 'element pointer' is returned. If the 'element pointer' is currently 0, a NIL value is returned. If the element type has no data associated with it 0 is returned for the element size.

The added PHIGS PLUS element type enumeration values are:

POLYLINE_SET_3_WITH_COLOUR,

FILL_AREA_SET_3_WITH_DATA,

FILL_AREA_SET_WITH_DATA,

CELL_ARRAY_3_PLUS,
SET_OF_FILL_AREA_SET_3_WITH_DATA,
SET_OF_FILL_AREA_SET_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,
DATA_MAPPING_INDEX,
REFLECTANCE_INDEX,
BACK_INTERIOR_INDEX,
BACK_DATA_MAPPING_INDEX,
BACK_REFLECTANCE_INDEX,
PARAMETRIC_SURFACE_INDEX,
POLYLINE_COLOUR,
POLYLINE_SHADING_METHOD,
POLYMARKER_COLOUR,
TEXT_COLOUR,
FACET_DISTINGUISHING_MODE,
FACET_CULLING_MODE,
INTERIOR_COLOUR,
INTERIOR_SHADING_METHOD,
DATA_MAPPING_METHOD,
REFLECTANCE_PROPERTIES,
REFLECTANCE_MODEL,
BACK_INTERIOR_STYLE,
BACK_INTERIOR_STYLE_INDEX,
BACK_INTERIOR_COLOUR,
BACK_INTERIOR_SHADING_METHOD,
BACK_DATA_MAPPING_METHOD,
BACK_REFLECTANCE_PROPERTIES,
BACK_REFLECTANCE_MODEL,
LIGHT_SOURCE_STATE,
EDGE_COLOUR,
CURVE_APPROXIMATION_CRITERIA,
SURFACE_APPROXIMATION_CRITERIA,