INTERNATIONAL STANDARD

ISO/IEC 8613-10

First edition
1991-06-01

AMENDMENT 4
1992-11-15

Information processing – Text and office systems – Office Document Architecture (ODA) and interchange format –

Part 10:
Formal specifications

AMENDMENT 4: Formal specification of the geometric graphics content architectures

Traitement de l'information – Bureautique – Architecture des documents de bureau (ODA) et format d'échange –

Partie 10: Spécifications formelles

AMENDEMENT 4: Spécifications formelles des architectures de contenus graphiques géométriques

Reference number
ISO/IEC 8613-10:1991/Amd. 4:1992 (E)

Foreword

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

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

Amendment 4 to International Standard ISO/IEC 8613-10:1991 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*.

ISO/IEC 8613 consists of the following parts, under the general title *Information processing – Text and office systems – Office Document Architecture (ODA) and interchange format :*

- *Part 1: Introduction and general principles*
- *Part 2: Document structures*
- *Part 4: Document profile*
- *Part 5: Office Document Interchange Format (ODIF)*
- *Part 6: Character content architectures*
- *Part 7: Raster graphics content architectures*
- *Part 8: Geometric graphics content architectures*
- *Part 10: Formal specifications*

Annex E forms an integral part of ISO/IEC 8613-10.

International Organization for Standardization
Case postale 56 • CH-1211 Genève 20 • Switzerland
Printed in Switzerland

Information processing – Text and office systems – Office Document Architecture (ODA) and interchange format –

Part 10:
Formal specifications

AMENDMENT 4: Formal specification of the geometric graphics content architectures

Insert a new annex E as follows:

Annex E

(normative)

Formal specification of the geometric graphics content architectures

E.1 Introduction

This annex gives a formal specification of the geometric graphics content architectures as described in ISO 8613-8. This annex is composed of 5 clauses:

Clause E.1 provides a general introduction, including a list of all definitions which are given in E.2, E.3 and E.4.

Clause E.2 provides the interface to the document profile and its formal specification.

Clauses E.3 and E.4 provide the interface to the document architecture by giving a formal specification of geometric graphics presentation attributes and content portion attributes that apply to geometric graphics content portions.

Clause E.5 is an index to the terms (definitions, operators, attribute names) used in E.2, E.3 and E.4.

At any time a clause number is specified in the semi-formal descriptions this refers to a clause number in ISO 8613-8.

What follows is the outline of the formula which specifies the geometric graphics content architectures. The dots indicate formal text fragments which have been left out for the sake of readability. The full formula can be obtained by replacing each line (apart from the *and*) with the definition which is referenced by the superscript of the predicate symbol or operator symbol, respectively. The variables used in the definition of the predicate have to be replaced by those appearing in the outline (if they are different). NOTE: A definition is a formula, hence it may never yield an undefined result, whatever value has been inserted for the variable.

... IsProfileDefaultableGeometricGraphicsContentArchitectureAttribute$^{8.1}(att)$ ...
and ... IsProfileGeometricGraphicsCodingSpecification$^{8.2}(v)$ ...
and ... IsProfileGeometricGraphicsPresentationFeature$^{8.3}(v)$ ...
and ... SatisfiesGeometricGraphicsContentArchitectureConstraints$^{8.4}(prof, doby)$ ...
and ... IsGeometricGraphicsContentPortionDescription$^{8.5}(cont)$ ...
and ... IsGeometricGraphicsContentPresentationAttribute$^{8.6}(att)$ ...
and ... IsGeometricGraphicsContentCodingAttribute$^{8.7}(att)$ ...
and ... IsGeometricGraphicsContentPortionAttributeSet$^{8.8}(as)$ ...
and ... IsGeometricGraphicsEncodingAnnouncerValue$^{8.9}(v)$ ...
and ... IsDirectColourValuePair$^{8.10}(v)$ ...
and ... IsLineRenditionValue$^{8.11}(v)$ ...
and ... IsSetOfLineBundleSpecifications$^{8.12}(v)$ ...
and ... IsLineBundleSpecification$^{8.13}(v)$ ...
and ... IsGeometricGraphicsColourValue$^{8.14}(v)$ ...
and ... IsWidthValue$^{8.15}(v)$ ...
and ... IsSpecificationModeValue$^{8.16}(v)$ ...
and ... IsLineTypeValue$^{8.17}(v)$ ...
and ... IsMarkerRenditionValue$^{8.18}(v)$ ...
and ... IsSetOfMarkerBundleSpecifications$^{8.19}(v)$ ...
and ... IsMarkerBundleSpecification$^{8.20}(v)$ ...
and ... IsMarkerTypeValue$^{8.21}(v)$ ...
and ... IsTextRenditionValue$^{8.22}(v)$ ...
and ... IsCharacterSetList$^{8.23}(v)$ ...
and ... IsCharacterSetSpecification$^{8.24}(v)$ ...
and ... IsSetOfTextBundleSpecifications$^{8.25}(v)$ ...
and ... IsRegisteredDesignationSequenceTail$^{8.26}(v)$ ...

and ... IsTextBundleSpecification$^{8.27}(v)$ ...
and ... IsOrientationVectorPair$^{8.28}(v)$ ...
and ... IsFontListSet$^{8.29}(v)$ ...
and ... IsTextAlignmentSpecification$^{8.30}(v)$ ...
and ... IsFilledAreaRenditionValue$^{8.31}(v)$ ...
and ... IsSetOfFillBundleSpecifications$^{8.32}(v)$ ...
and ... IsFillBundleSpecification$^{8.33}(v)$ ...
and ... IsSetOfPatternTableSpecifications$^{8.34}(v)$ ...
and ... IsPatternTableSpecification$^{8.35}(v)$ ...
and ... IsSeqOfDirectColourValue$^{8.36}(v)$ ...
and ... IsHatchIndexValue$^{8.37}(v)$ ...
and ... IsPatternSizeValue$^{8.38}(v)$ ...
and ... IsEdgeRenditionValue$^{8.39}(v)$ ...
and ... IsSetOfEdgeBundleSpecifications$^{8.40}(v)$ ...
and ... IsEdgeBundleSpecification$^{8.41}(v)$ ...
and ... IsEdgeTypeValue$^{8.42}(v)$ ...
and ... IsColourRepresentationsValue$^{8.43}(v)$ ...
and ... IsSetOfColourTableSpecifications$^{8.44}(v)$ ...
and ... IsColourTableSpecification$^{8.45}(v)$ ...
and ... IsTransparencySpecificationValue$^{8.46}(v)$ ...
and ... IsTransformationSpecificationValue$^{8.47}(v)$ ...
and ... IsRegionOfInterestSpecificationValue$^{8.48}(v)$ ...
and ... IsPictureOrientationValue$^{8.49}(v)$ ...
and ... IsPictureDimensionsValue$^{8.50}(v)$ ...
and ... IsGeometricGraphicsContentArchitectureClassValue$^{8.51}(v)$ ...
and ... IsGeometricGraphicsContentTypeOfCodingValue$^{8.52}(v)$ ...
and ... IsGeometricGraphicsContentInformationValue$^{8.53}(v)$ ...
and ... IsDirectColourValue$^{8.54}(v)$ ...
and ... IsVDCPair$^{8.55}(v)$ ...
and ... IsVDCValue$^{8.56}(v)$ ...
and ... IsNnVDCValue$^{8.57}(v)$ ...
and ... IsRegisteredLineType$^{8.58}(v)$ ...
and ... IsRegisteredMarkerType$^{8.59}(v)$ ...
and ... IsRegisteredHatchIndex$^{8.60}(v)$ ...
and ... IsRegisteredEdgeType$^{8.61}(v)$ ...

NOTE: Other predicates or operators which are used here, but are defined in clause 6, are not listed here.

E.2 Interface to the Document Profile

Semiformal Description 8.1

Predicate "is a profile defaultable geometric graphics content architecture attribute"

A profile defaultable geometric graphics content architecture attribute is a geometric graphics content presentation attribute.

NOTE: This predicate is used in annex B.

Definition 8.1

```
1   ∀ att
2   (₀ IsProfileDefaultableGeometricGraphicsContentArchitectureAttribute(att) iff
3     IsGeometricGraphicsContentPresentationAttribute^8.6(att) ₀)
```

Semiformal Description 8.2

Predicate "is a profile geometric graphics coding specification"

A profile geometric graphics coding specification is a nomination where each element is a geometric graphics content coding attribute.

NOTE: This predicate is used in annex B.

Definition 8.2

```
1   ∀ v
2   (₀ IsProfileGeometricGraphicsCodingSpecification(v) iff
3     IsNom(v) and
4     ∀ b ∈ ~v • (IsGeometricGraphicsContentCodingAttribute^8.7(C b)) ₀)
```

Semiformal Description 8.3

Predicate "is a profile geometric graphics presentation feature"

A profile geometric graphics presentation feature is a nomination where each element is a geometric graphics content presentation attribute.

NOTE: This predicate is used in annex B.

Definition 8.3

```
1   ∀ v
2   (₀ IsProfileGeometricGraphicsPresentationFeature(v) iff
3     IsNom(v) and
4     ∀ b ∈ ~v • (IsGeometricGraphicsContentPresentationAttribute^8.6(C b)) ₀)
```

E.3 Interface to the Document Architecture

Semiformal Description 8.4

Predicate "satisfies geometric graphics content architecture constraints"

A document profile *prof* and a document body *doby* satisfy the constraints imposed by the geometric graphics content architecture if the following holds: For all constituents *cst* and content portions *cont* in the document body for which the content portion is described by the constituent and for which the attribute 'content architecture class' of the constituent has the value '2 8 2 8 0', the content portion is a geometric graphics content portion description and the value '2 8 2 8 0' appears in the document profile attribute 'content architecture classes'.

NOTE: This predicate is used in clause 7 of this part of ISO 8613.

Definition 8.4

1 $\forall\, prof, doby$
2 $({}_{0}\,\text{SatisfiesGeometricGraphicsContentArchitectureConstraints}(prof, doby)\ \underline{iff}$
3 $\quad \forall\, cst, cont \in doby$
4 $\quad ({}_{1}\,({}_{2}\,(cont)\text{DescribesContPortOf}^{2.153}(cst)\ \underline{and}$
5 $\quad\quad C\ \hat{}\ cst \bullet \text{'content architecture class'} = \text{'2 8 2 8 0'}_{2})\ \underline{impl}$
6 $\quad\quad ({}_{3}\,\text{IsGeometricGraphicsContentPortionDescription}^{8.5}(cont)\ \underline{and}$
7 $\quad\quad\quad \text{'2 8 2 8 0'} \in C\ \hat{}\ prof \bullet \text{'content architecture classes'}_{3})_{1})_{0})$

Semiformal Description 8.5

Predicate "is a geometric graphics content portion description"

A geometric graphics content portion description is a set of geometric graphics content portion attributes.

NOTE: This predicate is used in clause 7 of this part of ISO 8613.

Definition 8.5

1 $\forall\, cont$
2 $({}_{0}\,\text{IsGeometricGraphicsContentPortionDescription}(cont)\ \underline{iff}$
3 $\quad \text{IsGeometricGraphicsContentPortionAttributeSet}^{8.8}(cont)_{0})$

Semiformal Description 8.6

Predicate "is a geometric graphics content presentation attribute"

A geometric graphics content presentation attribute is one of the attributes 'colour representations', 'edge rendition', 'filled area rendition', 'geometric graphics encoding announcer', 'line rendition', 'marker rendition', 'picture dimensions', 'picture orientation', 'region of interest specification', 'text rendition', 'transparency specification' and 'transformation specification' with an appropriate value.

NOTE: This predicate is used in clause 7 of this part of ISO 8613.

Definition 8.6

$\forall att$
$(_0$IsGeometricGraphicsContentPresentationAttribute(att) *iff*
$\exists n, c$
$(_1\ att = [n : c]$ *and*
$n \in$ ['colour representations'; 'edge rendition';
'filled area rendition'; 'geometric graphics encoding announcer';
'line rendition'; 'marker rendition';
'picture dimensions'; 'picture orientation';
'region of interest specification'; 'text rendition';
'transformation specification'; 'transparency specification'] *and*
$(_2\ n =$ 'colour representations' *impl*
$(_3$IsColourRepresentationsValue$^{8.43}(c)$ *or* IsPlaceholder$^{1.19}(c)_3)_2)$ *and*
$(_4\ n =$ 'edge rendition' *impl*
$(_5$IsEdgeRenditionValue$^{8.39}(c)$ *or* IsPlaceholder$^{1.19}(c)_5)_4)$ *and*
$(_6\ n =$ 'filled area rendition' *impl*
$(_7$IsFilledAreaRenditionValue$^{8.31}(c)$ *or* IsPlaceholder$^{1.19}(c)_7)_6)$ *and*
$(_8\ n =$ 'geometric graphics encoding announcer' *impl*
$(_9$IsGeometricGraphicsEncodingAnnouncerValue$^{8.9}(c)$ *or* IsPlaceholder$^{1.19}(c)_9)_8)$ *and*
$(_{10}\ n =$ 'line rendition' *impl*
$(_{11}$IsLineRenditionValue$^{8.11}(c)$ *or* IsPlaceholder$^{1.19}(c)_{11})_{10})$ *and*
$(_{12}\ n =$ 'marker rendition' *impl*
$(_{13}$IsMarkerRenditionValue$^{8.18}(c)$ *or* IsPlaceholder$^{1.19}(c)_{13})_{12})$ *and*
$(_{14}\ n =$ 'picture dimensions' *impl*
$(_{15}$IsPictureDimensionsValue$^{8.50}(c)$ *or* IsPlaceholder$^{1.19}(c)_{15})_{14})$ *and*
$(_{16}\ n =$ 'picture orientation' *impl*
$(_{17}$IsPictureOrientationValue$^{8.49}(c)$ *or* IsPlaceholder$^{1.19}(c)_{17})_{16})$ *and*
$(_{18}\ n =$ 'region of interest specification' *impl*
$(_{19}$IsRegionOfInterestSpecificationValue$^{8.48}(c)$ *or* IsPlaceholder$^{1.19}(c)_{19})_{18})$ *and*
$(_{20}\ n =$ 'text rendition' *impl*
$(_{21}$IsTextRenditionValue$^{8.22}(c)$ *or* IsPlaceholder$^{1.19}(c)_{21})_{20})$ *and*
$(_{22}\ n =$ 'transformation specification' *impl*
$(_{23}$IsTransformationSpecificationValue$^{8.47}(c)$ *or* IsPlaceholder$^{1.19}(c)_{23})_{22})$ *and*
$(_{24}\ n =$ 'transparency specification' *impl*
$(_{25}$IsTransparencySpecificationValue$^{8.46}(c)$ *or* IsPlaceholder$^{1.19}(c)_{25})_{24})_1)_0)$

Semiformal Description 8.7

Predicate "is a geometric graphics content coding attribute"

A geometric graphics content coding attribute is the attribute 'type of coding' with an appropriate value.

NOTE: This predicate is used in clause 7 of this part of ISO 8613.

Definition 8.7

1 $\forall att$
2 $(_0$ IsGeometricGraphicsContentCodingAttribute(att) *iff*
3 $\exists n, c$
4 $(_1 att = [n : c]$ *and*
5 $n =$ 'type of coding' *and* IsGeometricGraphicsContentTypeOfCodingValue$^{8.52}(c)\ _1)_0)$

E.4 Attributes of the Geometric Graphics Content Architecture

Semiformal Description 8.8

Predicate "is a geometric graphics content portion attribute set"

For a geometric graphics content portion attribute set the value of the attribute 'content information' is a geometric graphics content information value.

Definition 8.8

1 $\forall as$
2 $(_0$ IsGeometricGraphicsContentPortionAttributeSet(as) *iff*
3 IsNeNom$^{1.2}(as)$ *and*
4 $\forall a \in \hat{} as \bullet$
5 $(_1$ N $a =$ 'content information' *impl* IsGeometricGraphicsContentInformationValue$^{8.53}$(C a) $_1)_0)$

Semiformal Description 8.9

Predicate "is a geometric graphics encoding announcer value" (clause 6.1.1.1)

The value of the attribute 'geometric graphics encoding announcer' is a nomination with the names 'Colour Index Precision', 'Colour Precision', 'Colour Selection Mode', 'Colour Value Extent', 'Index Precision', 'Integer Precision', 'Maximum Colour Index', 'Real Precision', 'VDC Integer Precision', 'VDC Real Precision' and 'VDC Type' (3-3). For the names 'Colour Index Precision', 'Colour Precision', 'Index Precision' and 'Integer Precision' the value is either 8, 16, 24 or 32 (11, 12). For the name 'Colour Selection Mode' the value is either 'indexed' or 'direct' (13, 14). For the name 'Colour Value Extent' the value is a direct colour value pair (15, 16). For the name 'Maximum Colour Index' the value is a non-negative integer (17, 18). For the names 'Real Precision' and 'VDC Real Precision' the value is either 'floating point format, 9, 23', 'floating point format, 12, 52', 'fixed point format, 16, 16' or 'fixed point format, 32, 32' (19-22). For the name 'VDC Integer Precision' the value is either 16, 24 or 32 (23, 24). For the name 'VDC Type' the value is either 'integer' or 'real' (25, 26). All parameters are defaultable.

Definition 8.9

```
1   ∀ v
2   (0 IsGeometricGraphicsEncodingAnnouncerValue(v) iff
3     IsNom(v) and
4     NAMS1.18(v) = ['Colour Index Precision';  'Colour Precision';
5                    'Colour Selection Mode';  'Colour Value Extent';
6                    'Index Precision';        'Integer Precision';
7                    'Maximum Colour Index';   'Real Precision';
8                    'VDC Integer Precision';  'VDC Real Precision';
9                    'VDC Type'] and
10  ∀ a ∈̂ v •
11  (1 (2 N a ∈ ['Colour Index Precision'; 'Colour Precision'; 'Index Precision'; 'Integer Precision'] impl
12       (3 IsPlaceholder1.19(C a) or C a ∈ [8; 16; 24; 32]3)2) and
13     (4 N a = 'Colour Selection Mode' impl
14       (5 IsPlaceholder1.19(C a) or C a ∈ ['indexed'; 'direct']5)4) and
15     (6 N a = 'Colour Value Extent' impl
16       (7 IsPlaceholder1.19(C a) or IsDirectColourValuePair8.10(C a)7)6) and
17     (8 N a = 'Maximum Colour Index' impl
18       (9 IsPlaceholder1.19(C a) or IsNnInt1.7(C a)9)8) and
19     (10 N a ∈ ['Real Precision'; 'VDC Real Precision'] impl
20       (11 IsPlaceholder1.19(C a) or
21         C a ∈ ['floating point format, 9, 23'; 'floating point format, 12, 52';
22                'fixed point format, 16, 16'; 'fixed point format, 32, 32']11)10) and
23     (12 N a = 'VDC Integer Precision' impl
24       (13 IsPlaceholder1.19(C a) or C a ∈ [16; 24; 32]13)12) and
25     (14 N a = 'VDC Type' impl
26       (15 IsPlaceholder1.19(C a) or C a ∈ ['integer'; 'real']15)14)1)0)
```

Semiformal Description 8.10

Predicate "is a direct colour value pair" (clause 6.1.1.1)

A direct colour value pair is a sequence of two direct colour values.

Definition 8.10

```
1   ∀ v
2   (0 IsDirectColourValuePair(v) iff
3     v = [ → l → r → ] and
4     IsDirectColourValue^8.54(l) and IsDirectColourValue^8.54(r)0)
```

Semiformal Description 8.11

Predicate "is a line rendition value" (clause 6.1.1.2)

The value of the attribute 'line rendition' is a nomination with the names 'line aspect source flags', 'line bundle specifications', 'Line Bundle Index', 'Line Colour', 'Line Type', 'Line Width' and 'Line Width Specification Mode'. For the name 'line aspect source flags' the value is again a nomination with the names 'line colour asf', 'line type asf' and 'line width asf' whose values are either 'bundled' or 'individual' (9-14). For the name 'line bundle specifications' the value is a set of line bundle specifications (15, 16). For the name 'Line Bundle Index' the value is a positive integer (17, 18). For the name 'Line Colour' the value is a geometric graphics colour value (19, 20). For the name 'Line Type' the value is a line type value (21, 22). For the name 'Line Width' the value is a width value (23, 24). For the name 'Line Width Specification Mode' the value is a specification mode value (25, 26). All parameters are defaultable.

Definition 8.11

```
1   ∀ v
2   (0 IsLineRenditionValue(v) iff
3     IsNom(v) and
4     NAMS^1.18(v) = ['line aspect source flags'; 'line bundle specifications';
5                     'Line Bundle Index';      'Line Colour';
6                     'Line Type';              'Line Width';
7                     'Line Width Specification Mode'] and
8     ∀ a ∈ ~v •
9     (1 (2 N a = 'line aspect source flags' impl
10        (3 IsPlaceholder^1.19(C a) or
11          (4 IsNom(C a) and
12            NAMS^1.18(C a) = ['line colour asf'; 'line type asf'; 'line width asf'] and
13            ∀ b ∈ ~(C a) •
14              (5 IsPlaceholder^1.19(C b) or C b ∈ ['bundled'; 'individual'])5)4)3)2) and
15      (6 N a = 'line bundle specifications' impl
16        (7 IsPlaceholder^1.19(C a) or IsSetOfLineBundleSpecifications^8.12(C a)7)6) and
17      (8 N a = 'Line Bundle Index' impl
18        (9 IsPlaceholder^1.19(C a) or IsNat(C a)9)8) and
19      (10 N a = 'Line Colour' impl
20        (11 IsPlaceholder^1.19(C a) or IsGeometricGraphicsColourValue^8.14(C a)11)10) and
21      (12 N a = 'Line Type' impl
22        (13 IsPlaceholder^1.19(C a) or IsLineTypeValue^8.17(C a)13)12) and
23      (14 N a = 'Line Width' impl
24        (15 IsPlaceholder^1.19(C a) or IsWidthValue^8.15(C a)15)14) and
25      (16 N a = 'Line Width Specification Mode' impl
26        (17 IsPlaceholder^1.19(C a) or IsSpecificationModeValue^8.16(C a)17)16)1)0)
```

Semiformal Description 8.12

Predicate "is a set of line bundle specifications" (clause 6.1.1.2)

A set of line bundle specifications is a collection of elements which are line bundle specifications.

Definition 8.12

```
1   ∀ v
2   (0 IsSetOfLineBundleSpecifications(v) iff
3     IsCol(v) and
4     ∀ a ∈ v (IsLineBundleSpecification^8.13(a))0)
```

Semiformal Description 8.13

Predicate "is a line bundle specification" (clause 6.1.1.2)

A line bundle specification is a nomination with the names 'line bundle representation' and 'Line Bundle Index'. For the name 'Line Bundle Index' the value is a positive integer (6). For the name 'line bundle representation' the value is again a nomination with the names 'Line Colour', 'Line Type' and 'Line Width' (7-9). For the name 'Line Colour' the value is a geometric graphics colour value (11). For the name 'Line Type' the value is a line type value (12). For the name 'Line Width' the value is a width value (13).

Definition 8.13

```
1   ∀ v
2   (0 IsLineBundleSpecification(v) iff
3     IsNom(v) and
4     NAMS^1.18(v) = ['line bundle representation'; 'Line Bundle Index'] and
5     ∀ a ∈̂ ^v •
6     (1 (2 N a = 'Line Bundle Index' impl IsNat(C a)2) and
7       (3 N a = 'line bundle representation' impl
8         (4 IsNom(C a) and
9           NAMS^1.18(C a) = ['Line Colour'; 'Line Type'; 'Line Width'] and
10          ∀ b ∈̂ ^(C a) •
11          (5 (6 N b = 'Line Colour' impl IsGeometricGraphicsColourValue^8.14(C b)6) and
12            (7 N b = 'Line Type' impl IsLineTypeValue^8.17(C b)7) and
13            (8 N b = 'Line Width' impl IsWidthValue^8.15(C b)8)5)4)3)1)0)
```

Semiformal Description 8.14

Predicate "is a geometric graphics colour value" (clauses 6.1.1.2, 6.1.1.3, 6.1.1.4, 6.1.1.5, 6.1.1.6)

A geometric graphics colour value is either a non-negative integer or a direct colour value.

NOTE: Since the attribute 'colour' is used within the document structures (ISO 8613-2) and an associated predicate "IsColourValue" is defined in clause 7 of this part of ISO 8613, the qualifier "GeometricGraphics" has been added to the predicate name.

Definition 8.14

```
1   ∀ v
2   (0 IsGeometricGraphicsColourValue(v) iff
3     IsNnInt^1.7(v) or IsDirectColourValue^8.54(v) 0)
```

Semiformal Description 8.15

Predicate "is a width value" (clauses 6.1.1.2, 6.1.1.3, 6.1.1.6)

A width value is either a non-negative real number or a non-negative VDC value.

Definition 8.15

1 $\forall v$
2 ($_0$IsWidthValue(v) *iff*
3 IsNnReal$^{1.24}$(v) *or* IsNnVDCValue$^{8.57}$(v) $_0$)

Semiformal Description 8.16

Predicate "is a specification mode value" (clauses 6.1.1.2, 6.1.1.3, 6.1.1.6)

A specification mode value is either `'absolute'` or `'scaled'`.

Definition 8.16

1 $\forall v$
2 ($_0$IsSpecificationModeValue(v) *iff*
3 $v \in$ ['absolute'; 'scaled'] $_0$)

Semiformal Description 8.17

Predicate "is a line type value" (clause 6.1.1.2)

A line type value is 1, 2, 3, 4, 5 or any other registered line type.

Definition 8.17

1 $\forall v$
2 ($_0$IsLineTypeValue(v) *iff*
3 $v \in$ [1; 2; 3; 4; 5] *or* IsRegisteredLineType$^{8.58}$(v)$_0$)

Semiformal Description 8.18

Predicate "is a marker rendition value" (clause 6.1.1.3)

The value of the attribute 'marker rendition' is a nomination with the names 'marker aspect source flags', 'marker bundle specifications', 'Marker Bundle Index', 'Marker Colour', 'Marker Size', 'Marker Size Specification Mode' and 'Marker Type'. For the name 'marker aspect source flags' the value is again a nomination with the names 'marker colour asf', 'marker size asf' and 'marker type asf' whose values are either 'bundled' or 'individual' (9-14). For the name 'marker bundle specifications' the value is a set of marker bundle specifications (15, 16). For the name 'Marker Bundle Index' the value is a positive integer (17, 18). For the name 'Marker Colour' the value is a geometric graphics colour value (19, 20). For the name 'Marker Size' the value is a width value (21, 22). For the name 'Marker Size Specification Mode' the value is a specification mode value (23, 24). For the name 'Marker Type' the value is a marker type value (25, 26). All parameters are defaultable.

Definition 8.18

1 $\forall v$
2 $({}_{0}\,\text{IsMarkerRenditionValue}(v)$ *iff*
3 IsNom(v) *and*
4 $\text{NAMS}^{1.18}(v)$ = ['marker aspect source flags'; 'marker bundle specifications';
5 'Marker Bundle Index'; 'Marker Colour';
6 'Marker Size'; 'Marker Size Specification Mode';
7 'Marker Type'] *and*
8 $\forall a \hat{\in} \hat{}v \bullet$
9 $({}_{1}({}_{2}\ \text{N}\ a$ = 'marker aspect source flags' *impl*
10 $({}_{3}\,\text{IsPlaceholder}^{1.19}(\text{C}\ a)$ *or*
11 $({}_{4}\,\text{IsNom}(\text{C}\ a)$ *and*
12 $\text{NAMS}^{1.18}(\text{C}\ a)$ = ['marker colour asf'; 'marker size asf'; 'marker type asf'] *and*
13 $\forall b \hat{\in} \hat{}(\text{C}\ a) \bullet$
14 $({}_{5}\,\text{IsPlaceholder}^{1.19}(\text{C}\ b)$ *or* C $b \in$ ['bundled'; 'individual'])$_{5})_{4})_{3})_{2})$ *and*
15 $({}_{6}\ \text{N}\ a$ = 'marker bundle specifications' *impl*
16 $({}_{7}\,\text{IsPlaceholder}^{1.19}(\text{C}\ a)$ *or* $\text{IsSctOfMarkerBundleSpecifications}^{8.19}(\text{C}\ a)_{7})_{6})$ *and*
17 $({}_{8}\ \text{N}\ a$ = 'Marker Bundle Index' *impl*
18 $({}_{9}\,\text{IsPlaceholder}^{1.19}(\text{C}\ a)$ *or* $\text{IsNat}(\text{C}\ a)_{9})_{8})$ *and*
19 $({}_{10}\,\text{N}\ a$ = 'Marker Colour' *impl*
20 $({}_{11}\,\text{IsPlaceholder}^{1.19}(\text{C}\ a)$ *or* $\text{IsGeometricGraphicsColourValue}^{8.14}(\text{C}\ a)_{11})_{10})$ *and*
21 $({}_{12}\,\text{N}\ a$ = 'Marker Size' *impl*
22 $({}_{13}\,\text{IsPlaceholder}^{1.19}(\text{C}\ a)$ *or* $\text{IsWidthValue}^{8.15}(\text{C}\ a)_{13})_{12})$ *and*
23 $({}_{14}\,\text{N}\ a$ = 'Marker Size Specification Mode' *impl*
24 $({}_{15}\,\text{IsPlaceholder}^{1.19}(\text{C}\ a)$ *or* $\text{IsSpecificationModeValue}^{8.16}(\text{C}\ a)_{15})_{14})$ *and*
25 $({}_{16}\,\text{N}\ a$ = 'Marker Type' *impl*
26 $({}_{17}\,\text{IsPlaceholder}^{1.19}(\text{C}\ a)$ *or* $\text{IsMarkerTypeValue}^{8.21}(\text{C}\ a)_{17})_{16})_{1})_{0})$

Semiformal Description 8.19

Predicate "is a set of marker bundle specifications" (clause 6.1.1.3)

A set of marker bundle specifications is a collection of elements which are marker bundle specifications.

Definition 8.19

1 $\forall v$
2 $({}_{0}\,\text{IsSetOfMarkerBundleSpecifications}(v)$ *iff*
3 IsCol(v) *and*
4 $\forall a \in v\ (\text{IsMarkerBundleSpecification}^{8.20}(a))_{0})$

Semiformal Description 8.20

Predicate "is a marker bundle specification" (clause 6.1.1.3)

A marker bundle specification is a nomination with the names 'marker bundle representation' and 'Marker Bundle Index'. For the name 'Marker Bundle Index' the value is a positive integer (6). For the name 'marker bundle representation' the value is again a nomination with the names 'Marker Colour', 'Marker Size' and 'Marker Type' (7-9). For the name 'Marker Colour' the value is a geometric graphics colour value (11). For the name 'Marker Size' the value is a width value (12). For the name 'Marker Type' the value is a marker type value (13).

Definition 8.20

```
1   ∀ v
2   (0 IsMarkerBundleSpecification(v) iff
3     IsNom(v) and
4     NAMS^1.18(v) = ['marker bundle representation'; 'Marker Bundle Index'] and
5     ∀ a ∈̂ v •
6     (1 (2 N a = 'Marker Bundle Index' impl IsNat(C a)2) and
7       (3 N a = 'marker bundle representation' impl
8         (4 IsNom(C a) and
9           NAMS^1.18(C a) = ['Marker Colour'; 'Marker Size'; 'Marker Type'] and
10          ∀ b ∈̂ (C a) •
11          (5 (6 N b = 'Marker Colour' impl IsGeometricGraphicsColourValue^8.14(C b)6) and
12            (7 N b = 'Marker Size' impl IsWidthValue^8.15(C b)7) and
13            (8 N b = 'Marker Type' impl IsMarkerTypeValue^8.21(C b)8)5)4)3)1)0)
```

Semiformal Description 8.21

Predicate "is a marker type value" (clause 6.1.1.3)

A marker type value is 1, 2, 3, 4, 5 or any other registered marker type.

Definition 8.21

```
1   ∀ v
2   (0 IsMarkerTypeValue(v) iff
3     v ∈ [1; 2; 3; 4; 5] or IsRegisteredMarkerType^8.59(v)0)
```

Semiformal Description 8.22

Predicate "is a text rendition value" (clause 6.1.1.4)

A text rendition value is a nomination with the names 'text aspect source flags', 'text bundle specifications', 'Alternate Character Set Index', 'Character Coding Announcer', 'Character Expansion Factor', 'Character Height', 'Character Orientation', 'Character Set Index', 'Character Set List', 'Character Spacing', 'Font List', 'TextAlignment', 'Text Bundle Index', 'Text Colour', 'Text Font Index', 'Text Path' and 'Text Precision'. For the name 'text aspect source flags' the value is again a nomination with the names 'character expansion factor asf', 'character spacing asf', 'text colour asf', 'text font asf' and 'text precision asf' whose values are either 'bundled' or 'individual' (14–20). For the name 'text bundle specifications' the value is a set of text bundle specifications (21, 22). For the name 'Alternate Character Set Index' the value is a positive integer (23, 24). For the name 'Character Coding Announcer' the value is either 'basic 7-bit', 'basic 8-bit', 'extended 7-bit' or 'extended 8-bit' (25–27). For the name 'Character Expansion Factor' the value is a positive real number (28, 29). For the name 'Character Height' the value is a non-negative VDC value (30, 31). For the name 'Character Orientation' the value is an orientation vector pair (32, 33). For the name 'Character Set Index' the value is a positive integer (34, 35). For the name 'Character Set List' the value is a character set list (36, 37). For the name 'Character Spacing' the value is a real number (38, 39). For the name 'Font List' the value is a font list set (40, 41). For the name 'Text Alignment' the value is a text alignment specification (42, 43). For the name 'Text Bundle Index' the value is a positive integer (44, 45). For the name 'Text Colour' the value is a geometric graphics colour value (46, 47). For the name 'Text Font Index' the value is a positive integer (48). For the name 'Text Path' the value is 'down', 'left', 'right' or 'up' (49, 50). For the name 'Text Precision' the value is 'character', 'string' or 'stroke' (51, 52). All parameters are defaultable.

Definition 8.22

```
1   ∀ v
2   (0 IsTextRenditionValue(v) iff
3     IsNom(v) and
4     NAMS1.18(v) = ['text aspect source flags';      'text bundle specifications';
5                    'Alternate Character Set Index'; 'Character Coding Announcer';
6                    'Character Expansion Factor';    'Character Height';
7                    'Character Orientation';         'Character Set Index';
8                    'Character Set List';            'Character Spacing';
9                    'Font List';                     'Text Alignment';
10                   'Text Bundle Index';             'Text Colour';
11                   'Text Font Index';               'Text Path';
12                   'Text Precision'] and
13    ∀ a ∈̂ v •
14    (1 (2 N a = 'text aspect source flags' impl
15        (3 IsPlaceholder1.19(C a) or
16          (4 IsNom(C a) and
17            NAMS1.18(C a) = ['character expansion factor asf'; 'character spacing asf';
18                             'text colour asf'; 'text font asf'; 'text precision asf'] and
19            ∀ b ∈̂ (C a) •
20            (5 IsPlaceholder1.19(C b) or C b ∈ ['bundled'; 'individual'])5)4)3)2) and
21      (6 N a = 'text bundle specifications' impl
22        (7 IsPlaceholder1.19(C a) or IsSetOfTextBundleSpecifications8.25(C a)7)6) and
23      (8 N a = 'Alternate Character Set Index' impl
24        (9 IsPlaceholder1.19(C a) or IsNat(C a)9)8) and
25      (10 N a = 'Character Coding Announcer' impl
26        (11 IsPlaceholder1.19(C a) or
27          C a ∈ ['basic 7-bit'; 'basic 8-bit'; 'extended 7-bit'; 'extended 8-bit']11)10) and
28      (12 N a = 'Character Expansion Factor' impl
29        (13 IsPlaceholder1.19(C a) or IsPosReal1.25(C a)13)12) and
30      (14 N a = 'Character Height' impl
31        (15 IsPlaceholder1.19(C a) or IsNnVDCValue8.57(C a)15)14) and
32      (16 N a = 'Character Orientation' impl
33        (17 IsPlaceholder1.19(C a) or IsOrientationVectorPair8.28(C a)17)16) and
34      (18 N a = 'Character Set Index' impl
35        (19 IsPlaceholder1.19(C a) or IsNat(C a)19)18) and
36      (20 N a = 'Character Set List' impl
37        (21 IsPlaceholder1.19(C a) or IsCharacterSetList8.23(C a)21)20) and
38      (22 N a = 'Character Spacing' impl
39        (23 IsPlaceholder1.19(C a) or IsReal(C a)23)22) and
40      (24 N a = 'Font List' impl
41        (25 IsPlaceholder1.19(C a) or IsFontListSet8.29(C a)25)24) and
42      (26 N a = 'Text Alignment' impl
43        (27 IsPlaceholder1.19(C a) or IsTextAlignmentSpecification8.30(C a)27)26) and
44      (28 N a = 'Text Bundle Index' impl
45        (29 IsPlaceholder1.19(C a) or IsNat(C a)29)28) and
46      (30 N a = 'Text Colour' impl
47        (31 IsPlaceholder1.19(C a) or IsGeometricGraphicsColourValue8.14(C a)31)30) and
48      (32 N a = 'Text Font Index' impl (33 IsPlaceholder1.19(C a) or IsNat(C a)33)32) and
49      (34 N a = 'Text Path' impl
50        (35 IsPlaceholder1.19(C a) or C a ∈ ['down'; 'left'; 'right'; 'up']35)34) and
51      (36 N a = 'Text Precision' impl
52        (37 IsPlaceholder1.19(C a) or C a ∈ ['character'; 'string'; 'stroke']37)36)1)0)
```

Semiformal Description 8.23

Predicate "is a character set list" (clause 6.1.1.4)

A character set list is a set of elements which are character set specifications.

Definition 8.23

1 $\forall v$
2 $(_0$IsCharacterSetList(v) *iff*
3 IsCol(v) *and*
4 $\forall a \in v$ (IsCharacterSetSpecification$^{8.24}(a))_0)$

Semiformal Description 8.24

Predicate "is a character set specification" (clause 6.1.1.4)

A character set specification is a nomination with the names 'character set type' and 'designation sequence tail'. For the name 'character set type' the value is either '94-character sets', '96-character sets', '94-character multibyte sets', '96-character multibyte sets' or 'complete code' (6–8). For the name 'designation sequence tail' the value is a registered designation sequence tail.

Definition 8.24

1 $\forall v$
2 $(_0$IsCharacterSetSpecification(v) *iff*
3 IsNom(v) *and*
4 NAMS$^{1.18}(v)$ = ['character set type'; 'designation sequence tail'] *and*
5 $\forall a \hat{\in} \hat{}v$ •
6 $(_1(_2$N a = 'character set type' *impl*
7 C $a \in$ ['94-character sets'; '96-character sets'; '94-character multibyte sets';
8 '96-character multibyte sets'; 'complete code']$_2)$ *and*
9 $(_3$N a = 'designation sequence tail' *impl* IsRegisteredDesignationSequenceTail$^{8.26}$(C $a))_3)_1)_0)$

Semiformal Description 8.25

Predicate "is a set of text bundle specifications" (clause 6.1.1.4)

A set of text bundle specifications is a collection of elements which are text bundle specifications.

Definition 8.25

1 $\forall v$
2 $(_0$IsSetOfTextBundleSpecifications(v) *iff*
3 IsCol(v) *and*
4 $\forall a \in v$ (IsTextBundleSpecification$^{8.27}(a))_0)$

Semiformal Description 8.26

Predicate "is a registered designation sequence tail" (clause 6.1.1.4)

A registered designation sequence tail is considered an atomic construct.

Definition 8.26

1 $\forall v$
2 $(_0$IsRegisteredDesignationSequenceTail(v) *iff*
3 IsAtom$(v)_0)$

Semiformal Description 8.27

Predicate "is a text bundle specification" (clause 6.1.1.4)

A text bundle specification is a nomination with the names 'text bundle representation' and 'Text Bundle Index'. For the name 'Text Bundle Index' the value is a positive integer (6). For the name 'text bundle representation' the value is again a nomination with the names 'Character Expansion Factor', 'Character Spacing', 'Text Colour', 'Text Font Index' and 'Text Precision' (7–10). For the names 'Character Expansion Factor' the value is a positive real number (12). For the name 'Character Spacing' the value is a real number (13). For the name 'Text Colour' the value is a geometric graphics colour value (14). For the name 'Text Font Index' the value is a positive integer (15). For the name 'Text Precision' the value is either 'character', 'string' or 'stroke'.

Definition 8.27

$\forall v$
$({}_0\text{IsTextBundleSpecification}(v)$ *iff*
  $\text{IsNom}(v)$ *and*
  $\text{NAMS}^{1.18}(v) = [\text{'text bundle representation'}; \text{'Text Bundle Index'}]$ *and*
  $\forall a \hat{\in} \hat{}v \bullet$
  $({}_1({}_2\text{N } a = \text{'Text Bundle Index'}$ *impl* $\text{IsNat}(\text{C } a)_2)$ *and*
    $({}_3\text{N } a = \text{'text bundle representation'}$ *impl*
      $({}_4\text{IsNom}(\text{C } a)$ *and*
        $\text{NAMS}^{1.18}(\text{C } a) = [\text{'Character Expansion Factor'}; \text{'Character Spacing'};$
          $\text{'Text Colour'}; \text{'Text Font Index'}; \text{'Text Precision'}]_4)$ *and*
      $\forall b \hat{\in} \hat{}(\text{C } a) \bullet$
      $({}_5({}_6\text{N } b = \text{'Character Expansion Factor'}$ *impl* $\text{IsPosReal}^{1.25}(\text{C } b)_6)$ *and*
        $({}_7\text{N } b = \text{'Character Spacing'}$ *impl* $\text{IsReal}(\text{C } b)_7)$ *and*
        $({}_8\text{N } b = \text{'Text Colour'}$ *impl* $\text{IsGeometricGraphicsColourValue}^{8.14}(\text{C } b)_8)$ *and*
        $({}_9\text{N } b = \text{'Text Font Index'}$ *impl* $\text{IsNat}(\text{C } b)_9)$ *and*
        $({}_{10}\text{N } b = \text{'Text Precision'}$ *impl* $\text{C } b \in [\text{'character'}; \text{'string'}; \text{'stroke'}]_{10})_5)_3)_1)_0)$

Semiformal Description 8.28

Predicate "is an orientation vector pair" (clause 6.1.1.4)

An orientation vector pair is a VDC pair.

Definition 8.28

$\forall v$
$({}_0\text{IsOrientationVectorPair}(v)$ *iff*
  $\text{IsVDCPair}^{8.55}(v)_0)$

Semiformal Description 8.29

Predicate "is a font list set" (clause 6.1.1.4)

A font list set is a set of elements which are considered atomic. (Each atom represents a registered font name.)

Definition 8.29

$\forall v$
$({}_0\text{IsFontListSet}(v)$ *iff*
  $\text{IsCol}(v)$ *and*
  $\forall a \in v\ (\text{IsAtom}(a))_0)$

Semiformal Description 8.30

Predicate "is a text alignment specification" (clause 6.1.1.4)

A text alignment specification is a nomination with the names 'horizontal alignment', 'vertical alignment', 'continuous horizontal alignment' and 'continuous vertical alignment' (3-5). For the name 'horizontal alignment' the value is either 'normal horizontal', 'left', 'centre', 'right' or 'continuous horizontal' (7, 8). For the name 'vertical alignment' the value is either 'normal vertical', 'top', 'cap', 'half', 'base', 'bottom' or 'continuous vertical' (9, 10). For the names 'continuous horizontal alignment' and 'continuous vertical alignment' the value is a real number (11, 12).

Definition 8.30

```
1   ∀ v
2   (0 IsTextAlignmentSpecification(v) iff
3     IsNom(v) and
4     NAMS1.18(v) = ['horizontal alignment'; 'vertical alignment';
5                    'continuous horizontal alignment'; 'continuous vertical alignment'] and
6     ∀ a ∈̂ v •
7     (1 (2 N a = 'horizontal alignment' impl
8         C a ∈ ['normal horizontal'; 'left'; 'centre'; 'right'; 'continuous horizontal']2) and
9       (3 N a = 'vertical alignment' impl
10        C a ∈ ['normal vertical'; 'top'; 'cap'; 'half'; 'base'; 'bottom'; 'continuous vertical']3) and
11      (4 N a ∈ ['continuous horizontal alignment'; 'continuous vertical alignment'] impl
12        IsReal(C a)4)1)0)
```

Semiformal Description 8.31

Predicate "is a filled area rendition value" (clause 6.1.1.5)

A filled area rendition value is a nomination with the names 'fill aspect source flags', 'fill bundle specifications', 'pattern table specifications', 'Fill Bundle Index', 'Fill Colour', 'Fill Reference Point', 'Hatch Index', 'Interior Style', 'Pattern Index' and 'Pattern Size'. For the name 'fill aspect source flags' the value is again a nomination with the names 'fill colour asf', 'interior style asf', 'hatch index asf' and 'pattern index asf' whose value is either 'bundled' or 'individual' (10–16). For the name 'fill bundle specifications' the value is a set of fill bundle specifications (17, 18). For the name 'pattern table specifications' the value is a set of pattern table specifications (19, 20). For the name 'Fill Bundle Index' the value is a positive integer (21, 22). For the name 'Fill Colour' the value is a geometric graphics colour value (23, 24). For the name 'Fill Reference Point' the value is a VDC pair (25, 26). For the name 'Interior Style' the value is either 'empty', 'hatch', 'hollow', 'pattern' or 'solid' (27, 28). For the name 'Hatch Index' the value is a hatch index value (29, 30). For the name 'Pattern Index' the value is a positive integer (31, 32). For the name 'Pattern Size' the value is a pattern size value (33, 34). All parameters are defaultable.

Definition 8.31

```
1   ∀ v
2   (0 IsFilledAreaRenditionValue(v) iff
3     IsNom(v) and
4     NAMS1.18(v) = ['fill aspect source flags';      'fill bundle specifications';
5                    'pattern table specifications'; 'Fill Bundle Index';
6                    'Fill Colour';                  'Fill Reference Point';
7                    'Hatch Index';                  'Interior Style';
8                    'Pattern Index';                'Pattern Size'] and
9     ∀ a ∈̂ v •
10    (1 (2 N a = 'fill aspect source flags' impl
11        (3 IsPlaceholder1.19(C a) or
12          (4 IsNom(C a) and
13            NAMS1.18(C a) = ['fill colour asf'; 'interior style asf'; 'hatch index asf';
14                             'pattern index asf'] and
15            ∀ b ∈̂ (C a) •
16            (5 IsPlaceholder1.19(C b) or C b ∈ ['bundled'; 'individual'])5)4)3)2) and
17      (6 N a = 'fill bundle specifications' impl
18        (7 IsPlaceholder1.19(C a) or IsSetOfFillBundleSpecifications8.32(C a)7)6) and
19      (8 N a = 'pattern table specifications' impl
20        (9 IsPlaceholder1.19(C a) or IsSetOfPatternTableSpecifications8.34(C a)9)8) and
21      (10 N a = 'Fill Bundle Index' impl
22        (11 IsPlaceholder1.19(C a) or IsNat(C a)11)10) and
23      (12 N a = 'Fill Colour' impl
24        (13 IsPlaceholder1.19(C a) or IsGeometricGraphicsColourValue8.14(C a)13)12) and
25      (14 N a = 'Fill Reference Point' impl
26        (15 IsPlaceholder1.19(C a) or IsVDCPair8.55(C a)15)14) and
27      (16 N a = 'Interior Style' impl
28        (17 IsPlaceholder1.19(C a) or C a ∈ ['empty'; 'hatch'; 'hollow'; 'pattern'; 'solid']17)16) and
29      (18 N a = 'Hatch Index' impl
30        (19 IsPlaceholder1.19(C a) or IsHatchIndexValue8.37(C a)19)18) and
31      (20 N a = 'Pattern Index' impl
32        (21 IsPlaceholder1.19(C a) or IsNat(C a)21)20) and
33      (22 N a = 'Pattern Size' impl
34        (23 IsPlaceholder1.19(C a) or IsPatternSizeValue8.38(C a)23)22)1)0)
```

Semiformal Description 8.32

Predicate "is a set of fill bundle specifications" (clause 6.1.1.5)

A set of fill bundle specifications is a collection of elements which are fill bundle specifications.

Definition 8.32

```
1   ∀ v
2   (0 IsSetOfFillBundleSpecifications(v) iff
3     IsCol(v) and
4     ∀ a ∈ v (IsFillBundleSpecification^8.33(a)) 0)
```

Semiformal Description 8.33

Predicate "is a fill bundle specification" (clause 6.1.1.5)

A fill bundle specification is a nomination with the names 'fill bundle representation' and 'Fill Bundle Index'. For the name 'Fill Bundle Index' the value is a positive integer (6). For the name 'fill bundle representation' the value is again a nomination with the names 'Fill Colour', 'Hatch Index', 'Interior Style' and 'Pattern Index'. For the name 'Fill Colour' the value is a geometric graphics colour value (11). For the name 'Hatch Index' the value is a hatch index value (12). For the name 'Interior Style' the value is either 'empty', 'hatch', 'hollow', 'pattern' or 'solid' (13). For the name 'Pattern Index' the value is a positive integer (14).

Definition 8.33

```
1   ∀ v
2   (0 IsFillBundleSpecification(v) iff
3     IsNom(v) and
4     NAMS^1.18(v) = ['fill bundle representation'; 'Fill Bundle Index'] and
5     ∀ a ∈̂ ^v •
6     (1 (2 N a = 'Fill Bundle Index' impl IsNat(C a)2) and
7       (3 N a = 'fill bundle representation' impl
8         (4 IsNom(C a) and
9           NAMS^1.18(C a) = ['Fill Colour'; 'Hatch Index'; 'Interior Style'; 'Pattern Index'] and
10          ∀ b ∈̂ ^(C a) •
11          (5 (6 N b = 'Fill Colour' impl IsGeometricGraphicsColourValue^8.14(C b)6) and
12            (7 N b = 'Hatch Index' impl IsHatchIndexValue^8.37(C b)7) and
13            (8 N b = 'Interior Style' impl C b ∈ ['empty'; 'hatch'; 'hollow'; 'pattern'; 'solid']8) and
14            (9 N b = 'Pattern Index' impl IsNat(C b)9)5)4)3)
```

Semiformal Description 8.34

Predicate "is a set of pattern table specifications" (clause 6.1.1.5)

A set of pattern table specifications is a collection of elements which are pattern table specifications.

Definition 8.34

```
1   ∀ v
2   (0 IsSetOfPatternTableSpecifications(v) iff
3     IsCol(v) and
4     ∀ a ∈ v (IsPatternTableSpecification^8.35(a)) 0)
```

Semiformal Description 8.35

Predicate "is a pattern table specification" (clause 6.1.1.5)

A pattern table specification is a nomination with the names 'colour', 'local colour precision', 'nx', 'ny' and 'pattern table index'. For the name 'colour' the value is again a nomination with either the name 'index array' or 'value array'; in the first case, the value is a sequence of non-negative integers, in the second case, a sequence of direct colour values (6–11). For the name 'local colour precision' the value is either 0, 1, 2, 4, 8, 16, 24 or 32 (12). For the names 'nx', 'ny' and 'pattern table index' the value is a positive integer (13).

Definition 8.35

```
1   ∀ v
2   (0 IsPatternTableSpecification(v) iff
3     IsNom(v) and
4     NAMS1.18(v) = ['colour'; 'local colour precision'; 'nx'; 'ny'; 'pattern table index'] and
5     ∀ a ∈̂ v •
6     (1 (2 N a = 'colour' impl
7         (3 IsNom(C a) and
8           (4 NAMS1.18(C a) = ['index array'] or NAMS1.18(C a) = ['value array']4) and
9           ∀ b ∈̂ (C a) •
10          (5 (6 N b = 'index array' impl IsSeqOfNnInt1.23(C b)6) and
11            (7 N b = 'value array' impl IsSeqOfDirectColourValue8.36(C b)7)5)3)2) and
12      (8 N a = 'local colour precision' impl C a ∈ [0; 1; 2; 4; 8; 16; 24; 32]8) and
13      (9 N a ∈ ['nx', 'ny', 'pattern table index'] impl IsNat(C a)9)1)0)
```

Semiformal Description 8.36

Predicate "is a sequence of direct colour values" (clause 6.1.1.5)

A sequence of direct colour values is a non-empty catenation whose components are direct colour values.

Definition 8.36

```
1   ∀ v
2   (0 IsSeqOfDirectColourValue(v) iff
3     IsNeCat1.3(v) and
4     ∀ a ∈̂ v • (IsDirectColourValue8.54(C a)) 0)
```

Semiformal Description 8.37

Predicate "is a hatch index value" (clause 6.1.1.5)

A hatch index value is 1, 2, 3, 4, 5, 6 or any other registered hatch index.

Definition 8.37

```
1   ∀ v
2   (0 IsHatchIndexValue(v) iff
3     v ∈ [1; 2; 3; 4; 5; 6] or IsRegisteredHatchIndex8.60(v)0)
```

Semiformal Description 8.38

Predicate "is a pattern size value" (clause 6.1.1.5)

A pattern size value is a nomination with the names 'height vector x component', 'height vector y component' 'width vector x component' and 'width vector y component' whose values are VDC values.

Definition 8.38

```
1   ∀ v
2   (₀IsPatternSizeValue(v) iff
3     IsNom(v) and
4     NAMS^1.18(v) = ['height vector x component'; 'height vector y component';
5                     'width vector x component'; 'width vector y component'] and
6     ∀ a ∈̂ v • (IsVDCValuc^8.56(C a)) ₀)
```

Semiformal Description 8.39

Predicate "is an edge rendition value" (clause 6.1.1.6)

An edge rendition value is a nomination with the names 'edge aspect source flags', 'edge bundle specifications', 'Edge Bundle Index', 'Edge Colour', 'Edge Type', 'Edge Visibility', 'Edge Width' and 'Edge Width Specification Mode'. For the name 'edge aspect source flags' the value is again a nomination with the names 'edge colour asf', 'edge type asf' and 'edge width asf' whose value is either 'bundled' or 'individual' (9–14). For the name 'edge bundle specifications' the value is a set of edge bundle specifications (15, 16). For the name 'Edge Bundle Index' the value is positive integer (17, 18). For the name 'Edge Colour' the value is a geometric graphics colour value (19, 20). For the name 'Edge Type' the value is an edge type value (21, 22). For the name 'Edge Visibility' the value is either 'off' or 'on' (23, 24). For the name 'Edge Width' the value is a width value (25, 26). For the name 'Edge Width Specification Mode' the value is a specification mode value (27, 28). All values of all parameters are defaultable.

Definition 8.39

```
1   ∀ v
2   (0 IsEdgeRenditionValue(v) iff
3     IsNom(v) and
4     NAMS1.18(v) = ['edge aspect source flags'; 'edge bundle specifications';
5                    'Edge Bundle Index';       'Edge Colour';
6                    'Edge Type';               'Edge Visibility';
7                    'Edge Width';              'Edge Width Specification Mode'] and
8     ∀ a ∈̂ v •
9     ((1 (2 N a = 'edge aspect source flags' impl
10        (3 IsPlaceholder1.19(C a) or
11          (4 IsNom(C a) and
12            NAMS1.18(C a) = ['edge colour asf'; 'edge type asf'; 'edge width asf'] and
13            ∀ b ∈̂ (C a) •
14            (5 IsPlaceholder1.19(C b) or C b ∈ ['bundled'; 'individual'])5)4)3)2) and
15      (6 N a = 'edge bundle specifications' impl
16        (7 IsPlaceholder1.19(C a) or IsSetOfEdgeBundleSpecifications8.40(C a)7)6) and
17      (8 N a = 'Edge Bundle Index' impl
18        (9 IsPlaceholder1.19(C a) or IsNat(C a)9)8) and
19      (10 N a = 'Edge Colour' impl
20        (11 IsPlaceholder1.19(C a) or IsGeometricGraphicsColourValue8.14(C a)11)10) and
21      (12 N a = 'Edge Type' impl
22        (13 IsPlaceholder1.19(C a) or IsEdgeTypeValue8.42(C a)13)12) and
23      (14 N a = 'Edge Visibility' impl
24        (15 IsPlaceholder1.19(C a) or C a ∈ ['off'; 'on']15)14) and
25      (16 N a = 'Edge Width' impl
26        (17 IsPlaceholder1.19(C a) or IsWidthValue8.15(C a)17)16) and
27      (18 N a = 'Edge Width Specification Mode' impl
28        (19 IsPlaceholder1.19(C a) or IsSpecificationModeValue8.16(C a)19)18)1)0)
```

Semiformal Description 8.40

Predicate "is a set of edge bundle specifications" (clause 6.1.1.6)

A set of edge bundle specifications is a collection of elements which are edge bundle specifications.

Definition 8.40

```
1   ∀ v
2   (0 IsSetOfEdgeBundleSpecifications(v) iff
3     IsCol(v) and
4     ∀ a ∈ v (IsEdgeBundleSpecification^8.41(a)) 0)
```

Semiformal Description 8.41

Predicate "is an edge bundle specification" (clause 6.1.1.6)

An edge bundle specification is a nomination with the names 'edge bundle representation' and 'Edge Bundle Index'. For the name 'Edge Bundle Index' the value is a positive integer (6). For the name 'edge bundle representation' the value is again a nomination with the names 'Edge Colour', 'Edge Type' and 'Edge Width'. For the name 'Edge Colour' the value is a geometric graphics colour value (11). For the name 'Edge Type' the value is an edge type value (12). For the name 'Edge Width' the value is a width value (13).

Definition 8.41

```
1   ∀ v
2   (0 IsEdgeBundleSpecification(v) iff
3     IsNom(v) and
4     NAMS^1.18(v) = ['edge bundle representation';'Edge Bundle Index'] and
5     ∀ a ∈̂ ^v •
6     (1 (2 N a = 'Edge Bundle Index' impl IsNat(C a)2) and
7       (3 N a = 'edge bundle representation' impl
8         (4 IsNom(C a) and
9           NAMS^1.18(C a) = ['Edge Colour';'Edge Type';'Edge Width'] and
10          ∀ b ∈̂ ^(C a) •
11          (5 (6 N b = 'Edge Colour' impl IsGeometricGraphicsColourValue^8.14(C b)6) and
12            (7 N b = 'Edge Type' impl IsEdgeTypeValue^8.42(C b)7) and
13            (8 N b = 'Edge Width' impl IsWidthValue^8.15(C b)8)5)4)3)1)0)
```

Semiformal Description 8.42

Predicate "is a edge type value" (clause 6.1.1.6)

A edge type value is 1, 2, 3, 4, 5 or any other registered edge type.

Definition 8.42

```
1   ∀ v
2   (0 IsEdgeTypeValue(v) iff
3     v ∈ [1; 2; 3; 4; 5] or IsRegisteredEdgeType^8.61(v)0)
```

Semiformal Description 8.43

Predicate "is a colour representations value" (clause 6.1.1.7)

A colour representations value is a nomination with the names 'colour table specifications' and 'Background Colour'. For the name 'colour table specifications' the value is a set of colour table specifications (6, 7). For the name 'Background Colour' the value is a direct colour value (8, 9). The values are defaultable.

Definition 8.43

```
1  ∀ v
2  (0 IsColourRepresentationsValue(v) iff
3    IsNom(v) and
4    NAMS1.18(v) = ['colour table specifications'; 'Background Colour'] and
5    ∀ a ∈̂ v •
6    (1 (2 N a = 'colour table specifications' impl
7        (3 IsPlaceholder1.19(C a) or IsSetOfColourTableSpecifications8.44(C a)3)2) and
8      (4 N a = 'Background Colour' impl
9        (5 IsPlaceholder1.19(C a) or IsDirectColourValue8.54(C a)5)4)1)0)
```

Semiformal Description 8.44

Predicate "is a set of colour table specifications" (clause 6.1.1.7)

A set of colour table specifications is a collection of elements which are colour table specifications.

Definition 8.44

```
1  ∀ v
2  (0 IsSetOfColourTableSpecifications(v) iff
3    IsCol(v) and
4    ∀ a ∈ v (IsColourTableSpecification8.45(a)) 0)
```

Semiformal Description 8.45

Predicate "is a colour table specification" (clause 6.1.1.7)

A colour table specification is a nomination with the names 'starting index' and 'colour list'. For the name 'starting index' the value is a non-negative integer (6). For the name 'colour list' the value is a non-empty collection whose components are direct colour values (7–9).

Definition 8.45

```
1  ∀ v
2  (0 IsColourTableSpecification(v) iff
3    IsNom(v) and
4    NAMS1.18(v) = ['starting index'; 'colour list'] and
5    ∀ a ∈̂ v •
6    (1 (2 N a = 'starting index' impl IsNnInt1.7(C a)2) and
7      (3 N b = 'colour list' impl
8        (4 IsNeCol1.1(C a) and
9          ∀ b ∈ C a (IsDirectColourValue8.54(b)) 4)3)1)0)
```

Semiformal Description 8.46

Predicate "is a transparency specification value" (clause 6.1.1.8)

A transparency specification value is a nomination with the names 'Auxiliary Colour' and 'Transparency'. For the name 'Auxiliary Colour' the value is a geometric graphics colour value (6, 7). For the name 'Transparency' the value is either 'off' or 'on' (8, 9). Both values are defaultable.

Definition 8.46

```
1   ∀ v
2   (0 IsTransparencySpecificationValue(v) iff
3     IsNom(v) and
4     NAMS1.18(v) = ['Auxiliary Colour'; 'Transparency'] and
5     ∀ a ∈̂ v •
6     (1 (2 N a = 'Auxiliary Colour' impl
7         (3 IsPlaceholder1.19(C a) or IsGeometricGraphicsColourValue8.14(C a)3)2) and
8       (4 N a = 'Transparency' impl
9         (5 IsPlaceholder1.19(C a) or C a ∈ ['off'; 'on']5)4)1)0)
```

Semiformal Description 8.47

Predicate "is a transformation specification value" (clause 6.1.1.9)

A transformation specification value is a nomination with the names 'Clip Indicator', 'Clip Rectangle' and 'VDC Extent'. For the name 'Clip Indicator' the value is either 'off' or 'on' (6, 7). For the names 'Clip Rectangle' and 'VDC Extent' the value is a VDC pair (8–11). All three values are defaultable.

Definition 8.47

```
1   ∀ v
2   (0 IsTransformationSpecificationValue(v) iff
3     IsNom(v) and
4     NAMS1.18(v) = ['Clip Indicator'; 'Clip Rectangle'; 'VDC Extent'] and
5     ∀ a ∈̂ v •
6     (1 (2 N a = 'Clip Indicator' impl
7         (3 IsPlaceholder1.19(C a) or C a ∈ ['off'; 'on']3)2) and
8       (4 N a = 'Clip Rectangle' impl
9         (5 IsPlaceholder1.19(C a) or IsVDCPair8.55(C a)5)4) and
10      (6 N a = 'VDC Extent' impl
11        (7 IsPlaceholder1.19(C a) or IsVDCPair8.55(C a)7)6)1)0)
```

Semiformal Description 8.48

Predicate "is a region of interest specification value" (clause 6.1.2)

A region of interest specification value is a nomination with either the name 'rectangle' or 'automatic'. For the name 'rectangle' the value is a VDC pair (6). For the name 'automatic' the value is 'null' (7).

Definition 8.48

1 $\forall v$
2 $(_0$IsRegionOfInterestSpecificationValue(v) *iff*
3 IsNom(v) *and*
4 $(_1$NAMS$^{1.18}(v) =$ ['rectangle'] *or* NAMS$^{1.18}(v) =$ ['automatic']$_1)$ *and*
5 $\forall a \in \hat{} v \bullet$
6 $(_2(_3$N $a =$ 'rectangle' *impl* IsVDCPair$^{8.55}$(C $a)_3)$ *and*
7 $(_4$N $a =$ 'automatic' *impl* C $a =$ 'null'$_4)_2)_0)$

Semiformal Description 8.49

Predicate "is a picture orientation value" (clause 6.1.3)

A picture orientation value is '0°', '90°', '180°' or '270°'.

Definition 8.49

1 $\forall v$
2 $(_0$IsPictureOrientationValue(v) *iff*
3 $v \in$ ['0°';'90°';'180°';'270°']$_0)$

Semiformal Description 8.50

Predicate "is a picture dimensions value" (clause 6.3.1)

The value of the attribute 'picture dimensions' is a nomination where the name is 'width controlled', 'height controlled', 'area controlled' and/or 'automatic'. For the names 'width controlled' and 'height controlled' the corresponding component is a non-empty nomination with the names 'minimum width' and/or 'preferred width' and the associated components are non-negative integers.. For the name 'area controlled' the corresponding component is a non-empty nomination with the names 'minimum width', 'preferred width', 'minimum height', 'preferred height' (the component are non-negative integers) and/or 'aspect ratio flag' (the component is either 'fixed' or 'variable'). For the name 'automatic' the component is 'null'.

Definition 8.50

```
1   ∀ v
2   (0 IsPictureDimensionsValue(v) iff
3     IsNom(v) and
4     NAMS1.18(v) = ['width controlled'; 'height controlled';
5                    'area controlled'; 'automatic'] and
6     ∀ a ∈̂ v •
7     (1 (2 N a = 'width controlled' impl
8         (3 IsNeNom1.2(C a) and
9           NAMS1.18(C a) = ['minimum width'; 'preferred width'] and
10          ∀ b ∈̂ (C a) • (IsNnInt1.7(C b)) 3)2) and
11      (4 N a = 'height controlled' impl
12        (5 IsNeNom1.2(C a) and
13          NAMS1.18(C a) = ['minimum height'; 'preferred height'] and
14          ∀ b ∈̂ (C a) • IsNnInt1.7(C b) 5)4) and
15      (6 N a = 'area controlled' impl
16        (7 IsNeNom1.2(C a) and
17          NAMS1.18(C a) = ['minimum height'; 'preferred height';
18                           'minimum width'; 'preferred width'; 'aspect ratio flag'] and
19          ∀ b ∈̂ (C a) •
20          (8 N b ∈ ['minimum width'; 'preferred width';
21                    'minimum height'; 'preferred height'] impl
22            IsNnInt1.7(C b) 8) and
23            N b = 'aspect ratio flag' impl C b ∈ ['fixed'; 'variable'] 7)6) and
24      (9 N a = 'automatic' impl C a = 'null' 9)1)0)
```

Semiformal Description 8.51

Predicate "is a geometric graphics content architecture class value" (clause 6.4.1)

The value of the attribute 'content architecture class' is an ASN.1 object identifier with the value '2 8 2 8 0'.

NOTE: This predicate is used in clause 7.

Definition 8.51

```
1   ∀ v
2   (0 IsGeometricGraphicsContentArchitectureClassValue(v) iff
3     v = '2 8 2 8 0'0)
```

Semiformal Description 8.52

Predicate "is a geometric graphics type of coding value" (clause 7.1)

For a geometric graphics content portion the value of the attribute **'type of coding'** is an ASN.1 object identifier with the value '2 8 3 8 0'.

NOTE: This predicate is used in clause 7.

Definition 8.52

1 $\forall v$
2 $({}_0$IsGeometricGraphicsContentTypeOfCodingValue(v) *iff*
3 $v =$ '2 8 3 8 0'$_0)$

Semiformal Description 8.53

Predicate "is a geometric graphics content information value" (clause 7.2)

For a geometric graphics content portion the value of the attribute **'content information'** is an octet string, representing the binary encoding of a Computer Graphics Metafile according to ISO 8632.

NOTE: This predicate is used in clause 7.

Definition 8.53

1 $\forall v$
2 $({}_0$IsGeometricGraphicsContentInformationValue(v) *iff*
3 IsOctetString$^{1.10}(v)_0)$

Semiformal Description 8.54

Predicate "is a direct colour value"

A direct colour value is considered an atomic construct.

Definition 8.54

1 $\forall v$
2 $({}_0$IsDirectColourValue(v) *iff*
3 IsAtom$(v)_0)$

Semiformal Description 8.55

Predicate "is a VDC Pair"

A VDC pair is catenation of two components which are VDC values.

Definition 8.55

1 $\forall v$
2 $({}_0$IsVDCPair(v) *iff*
3 $v = [\rightarrow l \rightarrow r \rightarrow]$ *and*
4 IsVDCValue$^{8.56}(l)$ *and* IsVDCValue$^{8.56}(r)_0)$

Semiformal Description 8.56

Predicate "is a VDC value"

A VDC value is considered an atomic construct.

Definition 8.56

1 $\forall v$
2 $({}_{0}\text{IsVDCValue}(v)$ *iff*
3 $\text{IsAtom}(v)_{0})$

Semiformal Description 8.57

Predicate "is a non-negative VDC value"

A non-negative VDC value is considered an atomic construct.

Definition 8.57

1 $\forall v$
2 $({}_{0}\text{IsNnVDCValue}(v)$ *iff*
3 $\text{IsAtom}(v)_{0})$

NOTE — For the following four definitions the precise specifications are outside the scope of this Part of ISO 8613 since they are not given in ISO 8613-8.

Semiformal Description 8.58

Predicate "is a registered line type"

A registered line type is any value that has been officially registered. It is considered atomic within the formal specification of the geometric graphics content architectures.

Definition 8.58

1 $\forall v$
2 $({}_{0}\text{IsRegisteredLineType}(v)$ *iff*
3 $\text{IsAtom}(v)_{0})$

Semiformal Description 8.59

Predicate "is a registered marker type"

A registered marker type is any value that has been officially registered. It is considered atomic within the formal specification of the geometric graphics content architectures.

Definition 8.59

1 $\forall v$
2 $({}_{0}\text{IsRegisteredMarkerType}(v)$ *iff*
3 $\text{IsAtom}(v)_{0})$