

INTERNATIONAL STANDARD

ISO/IEC
8613-10

First edition
1991-06-01

AMENDMENT 5
1993-05-15

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

Part 10:

Formal specifications

**AMENDMENT 5 : Formal specification of the
defaulting mechanism for defaultable attributes**

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

Partie 10: Spécifications formelles

*AMENDEMENT 5 : Spécification formelle du mécanisme par défaut pour les
attributs par défaut*



Reference number
ISO/IEC 8613-10:1991/Amd.5:1993 (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 5 to International Standard ISO/IEC 8613-10:1991 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Sub-Committee SC 18, *Document processing and related communication*.

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 F forms an integral part of this part of ISO/IEC 8613.

© ISO/IEC 1993

All rights reserved. No part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from the publisher.

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

Printed in Switzerland

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

Part 10:

Formal specifications

AMENDMENT 5 : Formal specification of the defaulting mechanism for defaultable attributes

IECNORM.COM : Click to view the full PDF of ISO/IEC 8613-10:1991/Amd.5:1993

Instructions for merging this Amendment into ISO/IEC 8613-10:

— In Contents add:

Annex F: Formal specification of the defaulting mechanism for defaultable attributes

— In the definition of the production rules (subclause 4.2), the following line is added to the definition of a "term":

“particularization-term |”

— The following production rule is added:

“particularization-term ::=
 THAT var (formula)

NOTE – The semantics of the terminal symbol **THAT** in this production rule are specified in subclause 4.5”.

— In subclause 4.5, the following text is added:

“**THAT** var (formula) The particularizator **THAT** makes a term from a formula. For a formula f , the term “**THAT** x (f)” denotes that entity which, if assigned to x , makes f true. In general, this makes sense only if f contains x as a free variable. If there is not exactly one entity (i.e., none or more than one) satisfying f , the term is undefined. Formally, particularization is characterized by

$$\forall y (y = \text{THAT } x (f) \iff (\exists !x(f) \wedge f_y) \vee (\neg \exists !x(f) \wedge y = \text{UNDEF}))$$

where f_y stands for $\exists x(f \wedge x = y)$, $\exists !x(f)$ stands for $\exists y(f_y \wedge \forall x(f \implies x = y))$, and f is assumed not to contain y as a free variable.”

The Annex contained on the following pages is to be added to ISO/IEC 8613-10.

Annex F

(normative)

Formal specification of the defaulting mechanism for defaultable attributes

F.1 Introduction

This annex gives a formal specification of the defaulting mechanism for defaultable attributes.

Clause F.2 defines a set of general functions which are used in clause F.3.

Clause F.3 defines a set of functions which describe the derivation of all defaultable attributes in ISO 8613. It is substructured into clauses F.3.1 (defaultable attributes of ISO 8613-2), clauses F.3.2 (defaultable attributes of ISO 8613-6), clauses F.3.3 (defaultable attributes of ISO 8613-7) and clauses F.3.4 (defaultable attributes of ISO 8613-8).

Clause F.4 is an index for the predicate symbols, operator symbols and attribute names used in clauses F.2 and F.3.

F.2 General functions

This clause defines the general functions for determining the values of defaultable attributes according to clause 5.1.2.4 of ISO 8613-2.

Semiformal Description 9.1

Function “ x ORELSE y ”

The function x ORELSE y returns the value x if x is not undefined, otherwise the value y .

Definition 9.1

- 1 $\forall x, y$
- 2 $(x \text{ ORELSE } y =$
- 3 $\text{ IF } x \neq \text{UNDEF THEN } x \text{ ELSE } y_0)$

Semiformal Description 9.2

Function “Attribute value by step BL”

If a layout style is referenced by a constituent cst and if the attribute att is specified for this layout style (3) then its value is the value determined by step BL (4). Otherwise the value is undefined by step BL.

Definition 9.2

- 1 $\forall cst, doby, att$
- 2 $(\text{STEP_BL_VALUE}(cst, doby, att) =$
- 3 $\text{ IF 'layout style' } \in \text{NAMS}^{1.18}(cst) \text{ and } att \in \text{NAMS}^{1.18}(\text{REF_LAY_STYLE}^{9.4}(cst, doby))$
- 4 $\text{ THEN } C \text{ } \neg(\text{REF_LAY_STYLE}^{9.4}(cst, doby)) \bullet att$
- 5 $\text{ ELSE UNDEF}_0)$

Semiformal Description 9.3

Function "Attribute value by step BP"

If a presentation style is referenced by a constituent *cst* and if the attribute *att* is specified for this presentation style (3) then its value is the value determined by step BP (4). Otherwise the value is undefined by step BP.

Definition 9.3

```

1   $\forall cst, doby, att$ 
2  ( $\circ$ STEP_BP_VALUE(cst, doby, att) =
3  IF 'presentation style'  $\in$  NAMS1.18(cst) and att  $\in$  NAMS1.18(REF_PRES_STYLE9.5(cst, doby))
4  THEN C  $\wedge$ (REF_PRES_STYLE9.5(cst, doby)).att
5  ELSE UNDEF $\circ$ )

```

Semiformal Description 9.4

Function "Referenced layout style"

If a layout style exists within the document body *doby* for which the value of the attribute 'layout style identifier' is the same as the value of the attribute 'layout style' of a particular constituent *cst* this is called the referenced layout style (3) which is returned by this function. Otherwise, the function returns the empty nomination.

Definition 9.4

```

1   $\forall cst, doby$ 
2  ( $\circ$ REF_LAY_STYLE(cst, doby) =
3  C  $\wedge$  doby.<IsLayoutStyle2.60(C xs) and C xs. 'layout style identifier' = C  $\wedge$  cst. 'layout style'>
4  ORELSE9.1 [ : ] $\circ$ )

```

Semiformal Description 9.5

Function "Referenced presentation style"

If a presentation style exists within the document body *doby* for which the value of the attribute 'presentation style identifier' is the same as the value of the attribute 'presentation style' of a particular constituent *cst* this is called the referenced presentation style (3, 4) which is returned by this function. Otherwise, the function returns the empty nomination.

Definition 9.5

```

1   $\forall cst, doby$ 
2  ( $\circ$ REF_PRES_STYLE(cst, doby) =
3  C  $\wedge$  doby.<IsPresentationStyle2.61(C xs) and
4  C xs. 'presentation style identifier' = C  $\wedge$  cst. 'presentation style'>
5  ORELSE9.1 [ : ] $\circ$ )

```

Semiformal Description 9.6

Function “Attribute value by step C”

If an object class is referenced by a constituent *cst* and if the attribute *att* is specified for this object class (3) then its value is the value determined by step C (4). Otherwise the value is undefined by step C.

Definition 9.6

```

1   $\forall cst, doby, att$ 
2  ( ${}_0$ STEP_C_VALUE(cst, doby, att) =
3  IF 'object class'  $\in$  NAMS1.18(cst) and att  $\in$  NAMS1.18(REF_OBJECT_CLASS9.7(cst, doby))
4  THEN C  $\wedge$  (REF_OBJECT_CLASS9.7(cst, doby)) • att
5  ELSE UNDEF ${}_0$ )

```

Semiformal Description 9.7

Function “Referenced object class”

If an object class exists within the document body *doby* for which the value of the attribute 'object class identifier' is the same as the value of the attribute 'object class' of a particular constituent *cst* this is called the referenced object class (3) which is returned by this function. Otherwise, the function returns the empty nomination.

Definition 9.7

```

1   $\forall cst, doby$ 
2  ( ${}_0$ REF_OBJECT_CLASS(cst, doby) =
3  C  $\wedge$  doby • <IsObjectClassDescription2.37(C xs) and C xs • 'object class identifier' = C  $\wedge$  cst • 'object class'>
4  ORELSE9.1 [ : ] ${}_0$ )

```

Semiformal Description 9.8

Function “Attribute value by step DP”

If an object class is referenced by a constituent *cst*, and a presentation style is referenced by this object class and the attribute *att* is specified for this presentation style (3-5) then its value is the value determined by step DP (6). Otherwise the value is undefined by step DP.

Definition 9.8

```

1   $\forall cst, doby, att$ 
2  ( ${}_0$ STEP_DP_VALUE(cst, doby, att) =
3  IF 'object class'  $\in$  NAMS1.18(cst) and
4  'presentation style'  $\in$  NAMS1.18(REF_OBJECT_CLASS9.7(cst, doby)) and
5  att  $\in$  NAMS1.18(REF_PRES_STYLE9.5(REF_OBJECT_CLASS9.7(cst, doby), doby))
6  THEN C  $\wedge$  (REF_PRES_STYLE9.5(REF_OBJECT_CLASS9.7(cst, doby), doby)) • att
7  ELSE UNDEF ${}_0$ )

```

Semiformal Description 9.9

Function “Attribute value by step DL”

If an object class is referenced by a constituent *cst*, and a layout style is referenced by this object class and the attribute *att* is specified for this layout style (3-5) then its value is the value determined by step DL (6). Otherwise the value is undefined by step DL.

Definition 9.9

```

1   $\forall cst, doby, att$ 
2  ( $\circ$ STEP_DL_VALUE(cst, doby, att) =
3  IF 'object class'  $\in$  NAMS1.18(cst) and
4  'layout style'  $\in$  NAMS1.18(REF_OBJECT_CLASS9.7(cst, doby)) and
5  att  $\in$  NAMS1.18(REF_LAY_STYLE9.4(REF_OBJECT_CLASS9.7(cst, doby), doby))
6  THEN C  $\sim$ (REF_LAY_STYLE9.4(REF_OBJECT_CLASS9.7(cst, doby), doby)) • att
7  ELSE UNDEF $\circ$ )

```

Semiformal Description 9.10

Function “Attribute value by step E”

If an object class is referenced by a constituent *cst*, and a (resource) object class in a resource document is referenced by this object class and the attribute *att* is specified for this (resource) object class (3–6) then this attribute value is the value determined by step E (7, 8). Otherwise the value is undefined by step E.

Definition 9.10

```

1   $\forall cst, doby, rdoby, prof, att$ 
2  ( $\circ$ STEP_E_VALUE(cst, doby, rdoby, prof, att) =
3  IF 'object class'  $\in$  NAMS1.18(cst) and
4  'resource'  $\in$  NAMS1.18(REF_OBJECT_CLASS9.7(cst, doby)) and
5  att  $\in$  NAMS1.18(RESOURCE_OBJECT_CLASS9.11(rdoby,
6  OBJECT_CLASSID_INRES9.12(prof, REF_OBJECT_CLASS9.7(cst, doby))))
7  THEN C  $\sim$ (RESOURCE_OBJECT_CLASS9.11(rdoby,
8  OBJECT_CLASSID_INRES9.12(prof, REF_OBJECT_CLASS9.7(cst, doby)))) • att
9  ELSE UNDEF $\circ$ )

```

Semiformal Description 9.11

Function “Resource object class”

If an object class exists within the document body *rdoby* (of a resource document) for which the value of the attribute 'object class identifier' corresponds to *objcid*, this is called the resource object class (3) which is returned by this function. Otherwise, the function returns the empty nomination.

Definition 9.11

```

1   $\forall rdoby, objcid$ 
2  ( $\circ$ RESOURCE_OBJECT_CLASS(rdoby, objcid) =
3  C  $\sim$ rdoby • <IsObjectClassDescription2.37(C xs) and C xs • 'object class identifier' = objcid>
4  ORELSE9.1 [ : ] $\circ$ )

```

Semiformal Description 9.12

Function "Object class identifier in a resource document"

The object class identifier of an object class in a resource document is determined by the attribute 'resources' in the document profile by taking the second component of that construct in the value of this attribute for which the first component corresponds to the value of the attribute 'resource' for a particular constituent *cst*. If such a construct cannot be found the value of the function is undefined.

Definition 9.12

```

1   $\forall prof, cst$ 
2   $(\text{OBJECT\_CLASSID\_INRES}(prof, cst) =$ 
3   $\text{LASTC}^{1.14}(C \sim prof \bullet 'resources' \bullet \text{HEAD}^{1.13}(C \text{ xs}) = C \wedge cst \bullet 'resource') \bullet \text{.})$ 

```

Semiformal Description 9.13

Function "Attribute value by step FP"

If an object class is referenced by a constituent *cst*, and a (resource) object class in a resource document is referenced by this object class, and a presentation style is referenced by the (resource) object class and the attribute *att* is specified for this presentation style (3-8) then this attribute value is the value determined by step FP (9, 10). Otherwise the value is undefined by step FP.

Definition 9.13

```

1   $\forall cst, doby, rdoby, prof, att$ 
2   $(\text{STEP\_FP\_VALUE}(cst, doby, rdoby, prof, att) =$ 
3   $\text{IF 'object class' } \in \text{NAMS}^{1.18}(cst) \text{ and}$ 
4   $\text{'resource' } \in \text{NAMS}^{1.18}(\text{REF\_OBJECT\_CLASS}^{9.7}(cst, doby)) \text{ and}$ 
5   $\text{'presentation style' } \in \text{NAMS}^{1.18}(\text{RESOURCE\_OBJECT\_CLASS}^{9.11}(rdoby,$ 
6   $\text{OBJECT\_CLASSID\_INRES}^{9.12}(prof, \text{REF\_OBJECT\_CLASS}^{9.7}(cst, doby))) \text{ and}$ 
7   $att \in \text{NAMS}^{1.18}(\text{RESOURCE\_PRES\_STYLE}^{9.14}(rdoby,$ 
8   $\text{OBJECT\_CLASSID\_INRES}^{9.12}(prof, \text{REF\_OBJECT\_CLASS}^{9.7}(cst, doby)))$ 
9   $\text{THEN } C \sim (\text{RESOURCE\_PRES\_STYLE}^{9.14}(rdoby,$ 
10  $\text{OBJECT\_CLASSID\_INRES}^{9.12}(prof, \text{REF\_OBJECT\_CLASS}^{9.7}(cst, doby))) \bullet att$ 
11  $\text{ELSE UNDEF}_\text{.})$ 

```

Semiformal Description 9.14

Function "Presentation style in a resource document"

The presentation style in a resource document belonging to an object class with the object class identifier *objcid* is that constituent *rpres* in a resource document *rdoby* which is a presentation style (3, 4) and for which there exists a constituent *rcst* in the resource document (5) such that the value of the attribute 'presentation style identifier' of the presentation style is equal to the value of the attribute 'presentation style' of the constituent *rcst* and the value of the attribute 'object class identifier' of the constituent *rcst* equals *objcid* (6, 7). If such a presentation style cannot be found the resource presentation style is undefined.

Definition 9.14

```

1   $\forall rdoby, objcid$ 
2  ( ${}_0$  RESOURCE_PRES_STYLE(rdoby, objcid) =
3  THAT rpres
4  ( ${}_1$  rpres  $\in$  rdoby and IsPresentationStyle2.61(rpres) and
5   $\exists rcst \in rdoby$ 
6  ( ${}_2$   $C \hat{\sim} rpres \cdot$  'presentation style identifier' =  $C \hat{\sim} rcst \cdot$  'presentation style' and
7   $C \hat{\sim} rcst \cdot$  'object class identifier' = objcid2)1)0)
```

Semiformal Description 9.15

Function "Attribute value by step FL"

If an object class is referenced by a constituent *cst*, and a (resource) object class in a resource document is referenced by this object class, and a layout style is referenced by the (resource) object class and the attribute *att* is specified for this layout style (3-8) then this attribute value is the value determined by step FL (9, 10). Otherwise the value is undefined by step FL.

Definition 9.15

```

1   $\forall cst, doby, rdoby, prof, att$ 
2  ( ${}_0$  STEP_FL_VALUE(cst, doby, rdoby, prof, att) =
3  IF 'object class'  $\in$  NAMS1.18(cst) and
4  'resource'  $\in$  NAMS1.18(REF_OBJECT_CLASS9.7(cst, doby)) and
5  'layout style'  $\in$  NAMS1.18(RESOURCE_OBJECT_CLASS9.11(rdoby,
6  OBJECT_CLASSID_INRES9.12(prof, REF_OBJECT_CLASS9.7(cst, doby)))) and
7  att  $\in$  NAMS1.18(RESOURCE_LAY_STYLE9.16(rdoby,
8  OBJECT_CLASSID_INRES9.12(prof, REF_OBJECT_CLASS9.7(cst, doby))))
9  THEN  $C \hat{\sim}$ (RESOURCE_LAY_STYLE9.16(rdoby,
10  OBJECT_CLASSID_INRES9.12(prof, REF_OBJECT_CLASS9.7(cst, doby))))  $\cdot att$ 
11  ELSE UNDEF0)
```

Semiformal Description 9.16

Function “Layout style in a resource document”

The layout style in a resource document belonging to an object class with the object class identifier *objcid* is that constituent *rlay* in a resource document *rdoby* which is a layout style (3, 4) and for which there exists a constituent *rcst* in the resource document (5) such that the value of the attribute 'layout style identifier' of the layout style is equal to the value of the attribute 'layout style' of the constituent *rcst* and the value of the attribute 'object class identifier' of the constituent *rcst* equals *objcid* (6, 7). If such a layout style cannot be found the resource layout style is undefined.

Definition 9.16

$$\begin{aligned}
 &1 \quad \forall rdoby, objcid \\
 &2 \quad ({}_0 \text{RESOURCE_LAY_STYLE}(rdoby, objcid) = \\
 &3 \quad \text{THAT } rlay \\
 &4 \quad ({}_1 rlay \in rdoby \text{ \underline{and} } \text{IsLayoutStyle}^{2.60}(rlay) \text{ \underline{and} } \\
 &5 \quad \exists rcst \in rdoby \\
 &6 \quad ({}_2 C \sim rlay \cdot \text{'layout style identifier'} = C \sim rcst \cdot \text{'layout style'} \text{ \underline{and} } \\
 &7 \quad C \sim rcst \cdot \text{'object class identifier'} = objcid_2)_1)_0
 \end{aligned}$$

Semiformal Description 9.17

Function “Attribute value by step G”

If there exists a constituent *sup* in the document body to which the constituent *cst* is immediately subordinate (3) and if the attribute value can be found (in a default value list) on the superior constituent *sup* (3, 4) then this attribute value is the value determined by step G. Otherwise, if a superior constituent exists, the search for the attribute value according to step G is continued (recursively) on the superior constituent *cst* (6). Otherwise, if no superior constituent exists or if the attribute value cannot be found there, the value is undefined by step G.

Definition 9.17

$$\begin{aligned}
 &1 \quad \forall cst, doby, rdoby, prof, att, ctype \\
 &2 \quad ({}_0 \text{STEP_G_VALUE}(cst, doby, rdoby, prof, att, ctype) = \\
 &3 \quad \text{SUPERIOR_VALUE}^{9.18}({}_1 \text{THAT } sup ({}_2 sup \in doby \text{ \underline{and} } (cst)\text{DescribesImSubOf}^{2.152}(sup)_2), \\
 &4 \quad doby, rdoby, prof, att, ctype_1) \\
 &5 \quad \text{ORELSE}^{9.1} \text{STEP_G_VALUE}^{9.17}({}_3 \text{THAT } sup ({}_4 sup \in doby \text{ \underline{and} } (cst)\text{DescribesImSubOf}^{2.152}(sup)_4), \\
 &6 \quad doby, rdoby, prof, att, ctype_3)_0)
 \end{aligned}$$

Semiformal Description 9.18

Function "Superior value"

If the attribute value can be found in the default value list of the constituent *cst*, then this value is returned as the superior value (3). Otherwise, if the constituent refers to an object class (4) and the attribute value can be found in the default value list of the referenced object class, then this value is returned as the superior value (5). Otherwise, if the referenced object class refers to a resource document (6) and the attribute value can be found in the resource default value list of the referenced object class, then this value is returned as the superior value (7). Otherwise the superior value is undefined.

Definition 9.18

```

1   $\forall cst, doby, rdoby, prof, att, ctype$ 
2  ( $_0$  SUPERIOR_VALUE( $cst, doby, rdoby, prof, att, ctype$ ) =
3  DEFAULT_VALUE9.19( $cst, att, ctype, doby$ ) ORELSE9.1
4  ( $_1$  IF 'object class'  $\in$  NAMS1.18( $cst$ )
5  THEN ( $_2$  DEFAULT_VALUE9.19(REF_OBJECT_CLASS9.7( $cst, doby$ ),  $att, ctype, doby$ ) ORELSE9.1
6  ( $_3$  IF 'resource'  $\in$  NAMS1.18(REF_OBJECT_CLASS9.7( $cst, doby$ ))
7  THEN ( $_4$  RES_DEFAULT_VALUE9.21(REF_OBJECT_CLASS9.7( $cst, doby$ ),  $rdoby, prof, att, ctype$ )
8  ORELSE9.1 UNDEF $_4$ )
9  ELSE UNDEF $_3$ ) $_2$ )
10 ELSE UNDEF $_1$ ) $_0$ )

```

Semiformal Description 9.19

Function "Default value"

If a default value list is specified for the constituent *cst* (4), and default values for the particular type of constituent are specified (5, 6) and a default value for the particular attribute is specified (7, 8) then this value is the default value. Otherwise it is tested whether a style is referenced from the default value list which provides a value for the attribute (9). This function reflects the two cases that one (7) or more (8) sets of default values are specified, i.e., (a) one set is specified for composite object types (7); (b) one or more sets may be specified for basic object types (8).

Definition 9.19

```

1   $\forall cst, att, ctype, doby$ 
2  ( $_0$  DEFAULT_VALUE( $cst, att, ctype, doby$ ) =
3  C ( $_1$  THAT  $a$ 
4  ( $_2$  'default value lists'  $\in$  NAMS1.18( $cst$ ) and
5   $\exists v$ 
6  ( $_3$   $v \in \sim cst \cdot$  'default value lists'  $\cdot$  and  $v = ctype$  and
7  ( $_4$  (IsNom(C  $v$ ) and  $a = v \cdot att$ ) $_5$  or
8  ( $_6$  (IsCol(C  $v$ ) and  $a = v \cdot att$ ) $_4$ ) $_3$ ) $_2$ )
9  ORELSE9.1 DEFAULT_VALUE_BY_STYLE9.20( $cst, doby, att, ctype$ ) $_1$ ) $_0$ )

```

Semiformal Description 9.20

Function "Default value by style"

If the attribute 'default value lists' is specified (3), contains an entry for the particular type of constituent (4, 5), there exists a reference to a layout style (7) and this layout style contains the attribute (8), then the value of this attribute is the default value supplied by a style. Similarly, the value may also be supplied by a reference to a presentation style from within a default value list (10–15). This function reflects the two cases that one (10–12) or more (13–15) sets of default values are specified, i.e., (a) one set is specified for composite object types; (b) one or more sets may be specified for basic object types.

Definition 9.20

```

1   $\forall cst, doby, att, ctype$ 
2  ( ${}_0$ DEFAULT_VALUE_BY_STYLE( $cst, doby, att, ctype$ ) =
3  IF 'default value lists'  $\in$  NAMS1.18( $cst$ ) and
4   $\exists v$ 
5  ( ${}_1 v \in \sim cst \cdot$  'default value lists'  $\cdot$  and  $v = ctype$  and
6   $\exists a$ 
7  ( ${}_2$ ( ${}_3$ IsNom( $C v$ ) and  $a = v \cdot$  'layout style' and
8   $att \in$  NAMS1.18(REF_LAY_STYLE9.4( $C v, doby$ )))
9  THEN  $C \sim$ (REF_LAY_STYLE9.4( $C v, doby$ ))  $\cdot att$ ) or
10 ( ${}_4$ IsNom( $C v$ ) and  $a = v \cdot$  'presentation style' and
11  $att \in$  NAMS1.18(REF_PRES_STYLE9.5( $C v, doby$ )))
12 THEN  $C \sim$ (REF_PRES_STYLE9.5( $C v, doby$ ))  $\cdot att$ ) or
13 ( ${}_5$ IsCol( $C v$ ) and  $a = v \cdot$  'presentation style' and
14  $att \in$  NAMS1.18(REF_PRES_STYLE9.5( $C v, doby$ )))
15 THEN  $C \sim$ (REF_PRES_STYLE9.5( $C v, doby$ ))  $\cdot att$ ) ${}_2$ ) ${}_1$ ) ${}_0$ )

```

Semiformal Description 9.21

Function "Resource default value"

If a default value is specified in a resource object class referenced by the object class cst , then this value is the resource default value. Otherwise the resource default value is undefined.

Definition 9.21

```

1   $\forall cst, rdoby, prof, att, ctype$ 
2  ( ${}_0$ RES_DEFAULT_VALUE( $cst, rdoby, prof, att, ctype$ ) =
3  DEFAULT_VALUE9.19(RESOURCE_OBJECT_CLASS9.11( $rdoby$ ),
4  OBJECT_CLASSID_INRES9.12( $prof, cst$ )),  $att, ctype, rdoby$ ) ${}_0$ )

```

Semiformal Description 9.22

Function "Attribute value by step H"

If the attribute 'document application profile defaults' is specified in the document profile and this attribute specifies a value for the attribute att (3) then this is the value determined by step H. Otherwise the value is undefined by step H.

Definition 9.22

```

1   $\forall prof, att$ 
2  ( ${}_0$ STEP_H_VALUE( $prof, att$ ) =
3   $C$  ( ${}_1$  THAT  $a$  ( ${}_2 a \in \sim prof \cdot$  'document application profile defaults' and  $N a = att$ ) ${}_2$ ) ${}_1$ ) ${}_0$ )

```

Semiformal Description 9.23

Function “Parameter value by step BL”

If a layout style is referenced by a constituent *cst* and if the attribute *att* is specified for this layout style (3) and if the parameter *par* is specified for this attribute (4), then its value is the value determined by step BL (5). Otherwise the value is undefined by step BL.

Definition 9.23

```

1   $\forall cst, doby, att, par$ 
2  ( $\text{STEP\_BL\_PAR\_VALUE}(cst, doby, att, par) =$ 
3    IF 'layout style'  $\in$   $\text{NAMS}^{1.18}(cst)$  and  $att \in \text{NAMS}^{1.18}(\text{REF\_LAY\_STYLE}^{9.4}(cst, doby))$  and
4       $par \in \text{NAMS}^{1.18}(C \sim \text{REF\_LAY\_STYLE}^{9.4}(cst, doby) \cdot att)$ 
5    THEN  $C \sim (\text{REF\_LAY\_STYLE}^{9.4}(cst, doby) \cdot att) \cdot par$ 
6    ELSE UNDEF0)
```

Semiformal Description 9.24

Function “Parameter value by step C”

If an object class is referenced by a constituent *cst* and if the attribute *att* is specified for this object class (3) and if the parameter *par* is specified for the attribute (4), then its value is the value determined by step C (5). Otherwise the value is undefined by step C.

Definition 9.24

```

1   $\forall cst, doby, att, par$ 
2  ( $\text{STEP\_C\_PAR\_VALUE}(cst, doby, att, par) =$ 
3    IF 'object class'  $\in$   $\text{NAMS}^{1.18}(cst)$  and  $att \in \text{NAMS}^{1.18}(\text{REF\_OBJECT\_CLASS}^{9.7}(cst, doby))$  and
4       $par \in \text{NAMS}^{1.18}(C \sim \text{REF\_OBJECT\_CLASS}^{9.7}(cst, doby) \cdot att)$ 
5    THEN  $C \sim (\text{REF\_OBJECT\_CLASS}^{9.7}(cst, doby) \cdot att) \cdot par$ 
6    ELSE UNDEF0)
```

Semiformal Description 9.25

Function “Parameter value by step DL”

If an object class is referenced by a constituent *cst*, and a layout style is referenced by this object class and the attribute *att* is specified for this layout style (3-5) and the parameter *par* is specified for this attribute (6), then its value is the value determined by step DL (7). Otherwise the value is undefined by step DL.

Definition 9.25

```

1   $\forall cst, doby, att, par$ 
2  ( $\text{STEP\_DL\_PAR\_VALUE}(cst, doby, att, par) =$ 
3    IF 'object class'  $\in$   $\text{NAMS}^{1.18}(cst)$  and
4      'layout style'  $\in$   $\text{NAMS}^{1.18}(\text{REF\_OBJECT\_CLASS}^{9.7}(cst, doby))$  and
5       $att \in \text{NAMS}^{1.18}(\text{REF\_LAY\_STYLE}^{9.4}(\text{REF\_OBJECT\_CLASS}^{9.7}(cst, doby), doby))$  and
6       $par \in \text{NAMS}^{1.18}(C \sim \text{REF\_LAY\_STYLE}^{9.4}(\text{REF\_OBJECT\_CLASS}^{9.7}(cst, doby), doby) \cdot att)$ 
7    THEN  $C \sim (\text{REF\_LAY\_STYLE}^{9.4}(\text{REF\_OBJECT\_CLASS}^{9.7}(cst, doby), doby) \cdot att) \cdot par$ 
8    ELSE UNDEF0)
```

Semiformal Description 9.26

Function "Parameter value by step E"

If an object class is referenced by a constituent *cst*, and a (resource) object class in a resource document is referenced by this object class and the attribute *att* is specified for this (resource) object class (3–6) and the parameter *par* is specified for this attribute (7, 8) then this attribute value is the value determined by step E (9, 10). Otherwise the value is undefined by step E.

Definition 9.26

```

1   $\forall cst, doby, rdoby, prof, att, par$ 
2  ( $\text{STEP\_E\_PAR\_VALUE}(cst, doby, rdoby, prof, att, par) =$ 
3  IF 'object class'  $\in$   $\text{NAMS}^{1.18}(cst)$  and
4  'resource'  $\in$   $\text{NAMS}^{1.18}(\text{REF\_OBJECT\_CLASS}^{9.7}(cst, doby))$  and
5   $att \in \text{NAMS}^{1.18}(\text{RESOURCE\_OBJECT\_CLASS}^{9.11}(rdoby,$ 
6   $\text{OBJECT\_CLASSID\_INRES}^{9.12}(prof, \text{REF\_OBJECT\_CLASS}^{9.7}(cst, doby))))$  and
7   $par \in \text{NAMS}^{1.18}(C \text{ ^-RESOURCE\_OBJECT\_CLASS}^{9.11}(rdoby,$ 
8   $\text{OBJECT\_CLASSID\_INRES}^{9.12}(prof, \text{REF\_OBJECT\_CLASS}^{9.7}(cst, doby))) \cdot att)$ 
9  THEN  $C \text{ ^-}(\text{RESOURCE\_OBJECT\_CLASS}^{9.11}(rdoby,$ 
10  $\text{OBJECT\_CLASSID\_INRES}^{9.12}(prof, \text{REF\_OBJECT\_CLASS}^{9.7}(cst, doby))) \cdot att \cdot par$ 
11 ELSE UNDEF0)

```

Semiformal Description 9.27

Function "Parameter value by step FL"

If an object class is referenced by a constituent *cst*, and a (resource) object class in a resource document is referenced by this object class, and a layout style is referenced by the (resource) object class and the attribute *att* is specified for this layout style (3–8) and the parameter *par* is specified for this attribute (9, 10), then this attribute value is the value determined by step FL (11, 12). Otherwise the value is undefined by step FL.

Definition 9.27

```

1   $\forall cst, doby, rdoby, prof, att, par$ 
2  ( $\text{STEP\_FL\_PAR\_VALUE}(cst, doby, rdoby, prof, att, par) =$ 
3  IF 'object class'  $\in$   $\text{NAMS}^{1.18}(cst)$  and
4  'resource'  $\in$   $\text{NAMS}^{1.18}(\text{REF\_OBJECT\_CLASS}^{9.7}(cst, doby))$  and
5  'layout style'  $\in$   $\text{NAMS}^{1.18}(\text{RESOURCE\_OBJECT\_CLASS}^{9.11}(rdoby,$ 
6   $\text{OBJECT\_CLASSID\_INRES}^{9.12}(prof, \text{REF\_OBJECT\_CLASS}^{9.7}(cst, doby))))$  and
7   $att \in \text{NAMS}^{1.18}(\text{RESOURCE\_LAY\_STYLE}^{9.16}(rdoby,$ 
8   $\text{OBJECT\_CLASSID\_INRES}^{9.12}(prof, \text{REF\_OBJECT\_CLASS}^{9.7}(cst, doby))))$  and
9   $par \in \text{NAMS}^{1.18}(C \text{ ^-RESOURCE\_LAY\_STYLE}^{9.16}(rdoby,$ 
10  $\text{OBJECT\_CLASSID\_INRES}^{9.12}(prof, \text{REF\_OBJECT\_CLASS}^{9.7}(cst, doby))) \cdot att)$ 
11 THEN  $C \text{ ^-}(\text{RESOURCE\_LAY\_STYLE}^{9.16}(rdoby,$ 
12  $\text{OBJECT\_CLASSID\_INRES}^{9.12}(prof, \text{REF\_OBJECT\_CLASS}^{9.7}(cst, doby))) \cdot att \cdot par$ 
13 ELSE UNDEF0)

```

Semiformal Description 9.28

Function "Parameter value by step G"

If there exists a constituent *sup* in the document body to which the constituent *cst* is immediately subordinate (3) and if the parameter value can be found (in a default value list) on the superior constituent *sup* (3, 4) then this parameter value is the value determined by step G. Otherwise, if a superior constituent exists, the search for the parameter value according to step G is continued (recursively) on the superior constituent *cst* (6). Otherwise, if no superior constituent exists or if the parameter value cannot be found there, the value is undefined by step G.

Definition 9.28

$$\begin{aligned}
 &1 \quad \forall cst, doby, rdoby, prof, att, par, ctype \\
 &2 \quad ({}_0 \text{STEP_G_PAR_VALUE}(cst, doby, rdoby, prof, att, par, ctype) = \\
 &3 \quad \text{SUPERIOR_PAR_VALUE}^{9.30}({}_1 \text{ THAT } sup ({}_2 sup \in doby \text{ and } (cst) \text{DescribesImSubOf}^{2.152}(sup)_2), \\
 &4 \quad \quad doby, rdoby, prof, att, par, ctype)_1) \\
 &5 \quad \text{ORELSE}^{9.1} \text{STEP_G_PAR_VALUE}^{9.28}({}_3 \text{ THAT } sup ({}_4 sup \in doby \text{ and } (cst) \text{DescribesImSubOf}^{2.152}(sup)_4), \\
 &6 \quad \quad doby, rdoby, prof, att, par, ctype)_3)_0)
 \end{aligned}$$

Semiformal Description 9.29

Function "Parameter value by step H"

If the attribute 'document application profile defaults' is specified in the document profile and this attribute specifies a value for the attribute *att* (4) and if this attribute specifies a value for the parameter *par* (5), then this is the value determined by step H. Otherwise the value is undefined by step H.

Definition 9.29

$$\begin{aligned}
 &1 \quad \forall prof, att, par \\
 &2 \quad ({}_0 \text{STEP_H_PAR_VALUE}(prof, att, par) = \\
 &3 \quad \text{C} ({}_1 \text{ THAT } a \\
 &4 \quad \quad ({}_2 a \in \sim prof \cdot \text{'document application profile defaults'} \text{ and } N a = att \text{ and} \\
 &5 \quad \quad \exists p ({}_3 p \in \sim prof \cdot \text{'document application profile defaults'} \cdot att \text{ and } N p = par)_2)_1)_0)
 \end{aligned}$$

Semiformal Description 9.30

Function “Superior parameter value”

If the parameter value can be found in the default value list of the constituent *cst*, then this value is returned as the superior value (3). Otherwise, if the constituent refers to an object class (4) and the parameter value can be found in the default value list of the referenced object class, then this value is returned as the superior value (5). Otherwise, if the referenced object class refers to a resource document (6) and the parameter value can be found in the resource default value list of the referenced object class, then this value is returned as the superior value (7, 8). Otherwise the superior value is undefined.

Definition 9.30

```

1   $\forall cst, doby, rdoby, prof, att, ctype, par$ 
2  ( $_0$  SUPERIOR_PAR_VALUE(cst, doby, rdoby, prof, att, ctype, par) =
3  DEFAULT_PAR_VALUE9.31(cst, att, ctype, par, doby) ORELSE9.1
4  ( $_1$  IF 'object class'  $\in$  NAMS1.18(cst)
5  THEN ( $_2$  DEFAULT_PAR_VALUE9.31(REF_OBJECT_CLASS9.7(cst, doby), att, ctype, par, doby)
6  ORELSE9.1 ( $_3$  IF 'resource'  $\in$  NAMS1.18(REF_OBJECT_CLASS9.7(cst, doby))
7  THEN ( $_4$  RES_DEFAULT_PAR_VALUE9.33(REF_OBJECT_CLASS9.7(cst, doby),
8  rdoby, prof, att, ctype, par)
9  ORELSE9.1 UNDEF $_4$ )
10 ELSE UNDEF $_3$ ) $_2$ )
11 ELSE UNDEF $_1$ ) $_0$ )

```

Semiformal Description 9.31

Function “Default value of a parameter”

If a default value list is specified for the constituent *cst* (4), and default values for the particular type of constituent are specified (5, 6) and a default value for the particular parameter is specified (7, 8) then this value is the default value. Otherwise it is tested whether a style is referenced from the default value list which provides a value for the parameter (9). This function reflects the two cases that one (7) or more (8) sets of default values are specified, i.e., (a) one set is specified for composite object types (7); (b) one or more sets may be specified for basic object types (8).

Definition 9.31

```

1   $\forall cst, att, ctype, par, doby$ 
2  ( $_0$  DEFAULT_PAR_VALUE(cst, att, ctype, par, doby) =
3  C ( $_1$  THAT a
4  ( $_2$  'default value lists'  $\in$  NAMS1.18(cst) and
5   $\exists v$ 
6  ( $_3$   $\hat{v} \in \neg cst \cdot$  'default value lists'  $\cdot$  and  $v = ctype$  and
7  ( $_4$  (IsNom(C v) and  $a = v \cdot att \cdot par$ ) or
8  ( $_5$  IsCol(C v) and  $a = v \cdot \cdot att \cdot par$ ) $_4$ ) $_3$ ) $_2$ )
9  ORELSE9.1 DEFAULT_PAR_VALUE_BY_STYLE9.32(cst, doby, att, ctype, par) $_1$ ) $_0$ )

```

Semiformal Description 9.32

Function "Default parameter value by style"

If the attribute 'default value lists' is specified (3), contains an entry for the particular type of constituent (4, 5), there exists a reference to a layout style (7) and this layout style contains the attribute (8) and the parameter (9), then the value of this parameter is the default parameter value supplied by a style (10). Similarly, the value may also be supplied by a reference to a presentation style from within a default value list (11–18). This function reflects the two cases that one (11–14) or more (15–18) sets of default values are specified, i.e., (a) one set is specified for composite object types; (b) one or more sets may be specified for basic object types.

Definition 9.32

```

1   $\forall cst, doby, att, ctype, par$ 
2   $(\text{DEFAULT\_PAR\_VALUE\_BY\_STYLE}(cst, doby, att, ctype, par) =$ 
3    IF 'default value lists'  $\in$   $\text{NAMS}^{1.18}(cst)$  and
4     $\exists v$ 
5     $(v \in \sim cst \cdot \text{'default value lists'} \cdot \text{and } v = ctype \text{ and}$ 
6     $\exists a$ 
7     $(\text{IsNom}(C v) \text{ and } a = v \cdot \text{'layout style'} \text{ and}$ 
8     $att \in \text{NAMS}^{1.18}(\text{REF\_LAY\_STYLE}^{9.4}(C v, doby)) \text{ and}$ 
9     $par \in \text{NAMS}^{1.18}(\text{REF\_LAY\_STYLE}^{9.4}(C v, doby) \cdot par)$ 
10    $\text{THEN } C \sim(\text{REF\_LAY\_STYLE}^{9.4}(C v, doby)) \cdot att \cdot par_3) \text{ or}$ 
11    $(\text{IsNom}(C v) \text{ and } a = v \cdot \text{'presentation style'} \text{ and}$ 
12    $att \in \text{NAMS}^{1.18}(\text{REF\_PRES\_STYLE}^{9.5}(C v, doby)) \text{ and}$ 
13    $par \in \text{NAMS}^{1.18}(\text{REF\_PRES\_STYLE}^{9.5}(C v, doby) \cdot att)$ 
14    $\text{THEN } C \sim(\text{REF\_PRES\_STYLE}^{9.5}(C v, doby)) \cdot att \cdot par_4) \text{ or}$ 
15    $(\text{IsCol}(C v) \text{ and } a = v \cdot \text{'presentation style'} \text{ and}$ 
16    $att \in \text{NAMS}^{1.18}(\text{REF\_PRES\_STYLE}^{9.5}(C v, doby)) \text{ and}$ 
17    $par \in \text{NAMS}^{1.18}(\text{REF\_PRES\_STYLE}^{9.5}(C v, doby) \cdot att)$ 
18    $\text{THEN } C \sim(\text{REF\_PRES\_STYLE}^{9.5}(C v, doby)) \cdot att \cdot par_5)_1)_0)$ 

```

Semiformal Description 9.33

Function "Resource default value of a parameter"

If a default value is specified in a resource object class referenced by the object class *cst*, then this value is the resource default value. Otherwise the resource default value is undefined.

Definition 9.33

```

1   $\forall cst, rdoby, prof, att, ctype, par$ 
2   $(\text{RES\_DEFAULT\_PAR\_VALUE}(cst, rdoby, prof, att, ctype, par) =$ 
3     $\text{DEFAULT\_PAR\_VALUE}^{9.31}(\text{RESOURCE\_OBJECT\_CLASS}^{9.11}(rdoby,$ 
4     $\text{OBJECT\_CLASSID\_INRES}^{9.12}(prof, cst)), att, ctype, par, rdoby)_0)$ 

```

Semiformal Description 9.34

Function “Sub-parameter value by step C”

If an object class is referenced by a constituent *cst* and if the attribute *att* is specified for this object class (3) and if the parameter *par* and sub-parameter *subpar* are specified for the attribute (4, 5), then the value of the sub-parameter is the value determined by step C (6). Otherwise the value is undefined by step C.

Definition 9.34

```

1   $\forall cst, doby, att, par, subpar$ 
2  ( $_0$  STEP_C_SUBPAR_VALUE(cst, doby, att, par, subpar) =
3  IF 'object class'  $\in$  NAMS1.18(cst) and att  $\in$  NAMS1.18(REF_OBJECT_CLASS9.7(cst, doby)) and
4  par  $\in$  NAMS1.18(C  $\wedge$  REF_OBJECT_CLASS9.7(cst, doby)  $\cdot$  att) and
5  subpar  $\in$  NAMS1.18(C  $\wedge$  REF_OBJECT_CLASS9.7(cst, doby)  $\cdot$  att  $\cdot$  par)
6  THEN C  $\wedge$  (REF_OBJECT_CLASS9.7(cst, doby)  $\cdot$  att  $\cdot$  par  $\cdot$  subpar)
7  ELSE UNDEF0)

```

Semiformal Description 9.35

Function “Sub-parameter value by step E”

If an object class is referenced by a constituent *cst*, and a (resource) object class in a resource document is referenced by this object class and the attribute *att* is specified for this (resource) object class (3–6) and the parameter *par* and sub-parameter *subpar* are specified for this attribute (7–10) then the value of the sub-parameter is the value determined by step E (11, 12). Otherwise the value is undefined by step E.

Definition 9.35

```

1   $\forall cst, doby, rdoby, prof, att, par, subpar$ 
2  ( $_0$  STEP_E_SUBPAR_VALUE(cst, doby, rdoby, prof, att, par, subpar) =
3  IF 'object class'  $\in$  NAMS1.18(cst) and
4  'resource'  $\in$  NAMS1.18(REF_OBJECT_CLASS9.7(cst, doby)) and
5  att  $\in$  NAMS1.18(RESOURCE_OBJECT_CLASS9.11(rdoby,
6  OBJECT_CLASSID_INRES9.12(prof, REF_OBJECT_CLASS9.7(cst, doby)))) and
7  par  $\in$  NAMS1.18(C  $\wedge$  RESOURCE_OBJECT_CLASS9.11(rdoby,
8  OBJECT_CLASSID_INRES9.12(prof, REF_OBJECT_CLASS9.7(cst, doby)))  $\cdot$  att) and
9  subpar  $\in$  NAMS1.18(C  $\wedge$  RESOURCE_OBJECT_CLASS9.11(rdoby,
10 OBJECT_CLASSID_INRES9.12(prof, REF_OBJECT_CLASS9.7(cst, doby)))  $\cdot$  att  $\cdot$  par)
11 THEN C  $\wedge$  (RESOURCE_OBJECT_CLASS9.11(rdoby,
12 OBJECT_CLASSID_INRES9.12(prof, REF_OBJECT_CLASS9.7(cst, doby))))  $\cdot$  att  $\cdot$  par  $\cdot$  subpar)
13 ELSE UNDEF0)

```

Semiformal Description 9.36

Function "Sub-parameter value by step G"

If there exists a constituent *sup* in the document body to which the constituent *cst* is immediately subordinate (3) and if the sub-parameter value can be found (in a default value list) on the superior constituent *sup* (3, 4) then this sub-parameter value is the value determined by step G. Otherwise, if a superior constituent exists, the search for the sub-parameter value according to step G is continued (recursively) on the superior constituent *cst* (6-8). Otherwise, if no superior constituent exists or if the sub-parameter value cannot be found there, the value is undefined by step G.

Definition 9.36

```

1   $\forall cst, doby, rdoby, prof, att, par, subpar, ctype$ 
2  ( $\circ$ STEP_G_SUBPAR_VALUE(cst, doby, rdoby, prof, att, par, ctype) =
3  SUPERIOR_PAR_VALUE9.30( $\circ$  THAT sup ( $\circ$  sup  $\in$  doby and
4  (cst)DescribesImSubOf2.152(sup) $\circ$ ),
5  doby, rdoby, prof, att, par • subpar, ctype $\circ$ )
6  ORELSE9.1 STEP_G_SUBPAR_VALUE9.36( $\circ$  THAT sup ( $\circ$  sup  $\in$  doby and
7  (cst)DescribesImSubOf2.152(sup) $\circ$ ),
8  doby, rdoby, prof, att, par, subpar, ctype $\circ$ )
```

F.3 Functions for the values of defaultable attributes

This clause defines for each defaultable attribute the rules to determine its value.

NOTE — Default values for attributes are only derived for objects or content portions, not for object classes. Therefore, the argument *cst* in the functions defined below denotes always an object or content portion, never an object class.

F.3.1 Defaultable attributes of ISO 8613-2

Semiformal Description 9.37

Function "Value of the attribute 'application comments'"

The value of the attribute 'application comments' may be specified directly on a particular constituent (3-5). If this is not the case, test if the attribute value can be found by step C (6). If this is not the case, test if the attribute value can be found by step E (7). If this is not the case, the value of the attribute 'application comments' is the empty catenation.

Definition 9.37

```

1   $\forall cst, doby, prof, rdoby$ 
2  ( $\circ$ APPLICATION_COMMENTS_VALUE(cst, doby, prof, rdoby) =
3  IF IsAttributeSet2.62(cst) and 'application comments'  $\in$  NAMS1.18(cst) and
4  not IsPlaceholder1.19(C  $\sim$  cst • 'application comments')
5  THEN C  $\sim$  cst • 'application comments'
6  ELSE STEP_C_VALUE9.6(cst, doby, 'application comments') ORELSE9.1
7  ( $\circ$ STEP_E_VALUE9.10(cst, doby, rdoby, prof, 'application comments')
8  ORELSE9.1 [  $\rightarrow$  ] $\circ$ )
```

Semiformal Description 9.38

Function "Value of the attribute 'balance'"

The value of the attribute 'balance' may be specified directly on a particular constituent (3-5). If this is not the case, test if the attribute value can be found by step C (6). If this is not the case, test if the attribute value can be found by step E (7). If this is not the case, the value of the attribute 'balance' is 'null'.

Definition 9.38

```

1  ∀ cst, doby, prof, rdoby
2  (₀ BALANCE_VALUE(cst, doby, prof, rdoby) =
3    IF IsAttributeSet2.62(cst) and 'balance' ∈ NAMS1.18(cst) and
4      not IsPlaceholder1.19(C ^cst. 'balance')
5    THEN C ^cst. 'balance'
6    ELSE STEP_C_VALUE9.6(cst, doby, 'balance') ORELSE9.1
7    (₁ STEP_E_VALUE9.10(cst, doby, rdoby, prof, 'balance')
8    ORELSE9.1 'null' )₀)

```

Semiformal Description 9.39

Function "Value of the attribute 'bindings'"

The value of the attribute 'bindings' may be specified directly on a particular constituent (3-5). If this is not the case, test if the attribute value can be found by step C (6). If this is not the case, test if the attribute value can be found by step E (7). If this is not the case, the value of the attribute 'bindings' is the empty set.

Definition 9.39

```

1  ∀ cst, doby, prof, rdoby
2  (₀ BINDINGS_VALUE(cst, doby, prof, rdoby) =
3    IF IsAttributeSet2.62(cst) and 'bindings' ∈ NAMS1.18(cst) and
4      not IsPlaceholder1.19(C ^cst. 'bindings')
5    THEN C ^cst. 'bindings'
6    ELSE STEP_C_VALUE9.6(cst, doby, 'bindings') ORELSE9.1
7    (₁ STEP_E_VALUE9.10(cst, doby, rdoby, prof, 'bindings')
8    ORELSE9.1 [],)₀)

```

Semiformal Description 9.40

Function "Value of the attribute 'block alignment'"

The value of the attribute 'block alignment' may be specified directly on a particular constituent (3-5). If this is not the case, test if the attribute value can be found by step BL (6). If this is not the case, test if the attribute value can be found by step DL (7). If this is not the case, test if the attribute value can be found by step FL (8). If this is not the case, test if the attribute value can be found by step G (9). If this is not the case, test if the attribute value can be found by step H (10). If this is not the case, the value of the attribute 'block alignment' is 'right-hand aligned'.

Definition 9.40

```

1  ∀ cst, doby, prof, rdoby
2  (₀ BLOCK_ALIGNMENT_VALUE(cst, doby, prof, rdoby) =
3    IF IsAttributeSet2.62(cst) and 'block alignment' ∈ NAMS1.18(cst) and
4      not IsPlaceholder1.19(C ^cst . 'block alignment')
5    THEN C ^cst . 'block alignment'
6    ELSE STEP_BL_VALUE9.2(cst, doby, 'block alignment') ORELSE9.1
7      (₁ STEP_DL_VALUE9.9(cst, doby, 'block alignment') ORELSE9.1
8        (₂ STEP_FL_VALUE9.15(cst, doby, rdoby, prof, 'block alignment') ORELSE9.1
9          (₃ STEP_G_VALUE9.17(cst, doby, rdoby, prof, 'block alignment', OBJECT_TYPE_VALUE9.61(cst, doby))
10         ORELSE9.1 (₄ STEP_H_VALUE9.22(prof, 'block alignment')
11         ORELSE9.1 'right-hand aligned' )₃)₂)₁)₀)

```

Semiformal Description 9.41

Function "Value of the attribute 'border'"

The value of the attribute 'border' may be specified directly on a particular constituent (3-5). If this is not the case, test if the attribute value can be found by step BP (6). If this is not the case, test if the attribute value can be found by step C (7). If this is not the case, test if the attribute value can be found by step DP (8). If this is not the case, test if the attribute value can be found by step E (9). If this is not the case, test if the attribute value can be found by step FP (10). If this is not the case, test if the attribute value can be found by step G (11). If this is not the case, test if the attribute value can be found by step H (12). If this is not the case, the value comprises three parameters where the value of the parameters 'border line width' and 'border freespace width' is 0 and the value of the parameter 'border line type' is 'solid'.

Definition 9.41

```

1  ∀ cst, doby, prof, rdoby
2  (₀ BORDER_VALUE(cst, doby, prof, rdoby) =
3    IF IsAttributeSet2.62(cst) and 'border' ∈ NAMS1.18(cst) and
4      not IsPlaceholder1.19(C ^cst . 'border')
5    THEN C ^cst . 'border'
6    ELSE STEP_BP_VALUE9.3(cst, doby, 'border') ORELSE9.1
7      (₁ STEP_C_VALUE9.6(cst, doby, 'border') ORELSE9.1
8        (₂ STEP_DP_VALUE9.8(cst, doby, 'border') ORELSE9.1
9          (₃ STEP_E_VALUE9.10(cst, doby, rdoby, prof, 'border') ORELSE9.1
10         (₄ STEP_FP_VALUE9.13(cst, doby, rdoby, prof, 'border') ORELSE9.1
11         (₅ STEP_G_VALUE9.17(cst, doby, rdoby, prof, 'border', OBJECT_TYPE_VALUE9.61(cst, doby))
12         ORELSE9.1 (₆ STEP_H_VALUE9.22(prof, 'border')
13         ORELSE9.1 ['border line width' : 0; 'border line type' : 'solid';
14         'border freespace width' : 0] )₆)₅)₄)₃)₂)₁)₀)

```

Semiformal Description 9.42

Function “Value of the attribute 'colour'”

The value of the attribute 'colour' may be specified directly on a particular constituent (3-5). If this is not the case, test if the attribute value can be found by step BP (6). If this is not the case, test if the attribute value can be found by step C (7). If this is not the case, test if the attribute value can be found by step DP (8). If this is not the case, test if the attribute value can be found by step E (9). If this is not the case, test if the attribute value can be found by step FP (10). If this is not the case, test if the attribute value can be found by step G (11). If this is not the case, test if the attribute value can be found by step H (12). If this is not the case, the value of the attribute 'colour' is 'colourless'.

Definition 9.42

```

1   $\forall cst, doby, prof, rdoby$ 
2  (0 COLOUR_VALUE(cst, doby, prof, rdoby) =
3  IF IsAttributeSet2.62(cst) and 'colour' ∈ NAMS1.18(cst) and
4  not IsPlaceholder1.19(C  $\hat{cst}$  • 'colour')
5  THEN C  $\hat{cst}$  • 'colour'
6  ELSE STEP_BP_VALUE9.3(cst, doby, 'colour') ORELSE9.1
7  (1 STEP_C_VALUE9.6(cst, doby, 'colour') ORELSE9.1
8  (2 STEP_DP_VALUE9.8(cst, doby, 'colour') ORELSE9.1
9  (3 STEP_E_VALUE9.10(cst, doby, rdoby, prof, 'colour') ORELSE9.1
10 (4 STEP_FP_VALUE9.13(cst, doby, rdoby, prof, 'colour') ORELSE9.1
11 (5 STEP_G_VALUE9.17(cst, doby, rdoby, prof, 'colour', OBJECT_TYPE_VALUE9.61(cst, doby))
12 ORELSE9.1 (6 STEP_H_VALUE9.22(prof, 'colour')
13 ORELSE9.1 'colourless' )5)4)3)2)1)0)

```

Semiformal Description 9.43

Function “Value of the attribute 'concatenation'”

The value of the attribute 'concatenation' may be specified directly on a particular constituent (3-5). If this is not the case, test if the attribute value can be found by step BL (6). If this is not the case, test if the attribute value can be found by step DL (7). If this is not the case, test if the attribute value can be found by step FL (8). If this is not the case, test if the attribute value can be found by step G (9). If this is not the case, the value of the attribute 'concatenation' is 'non-concatenated'.

Definition 9.43

```

1   $\forall cst, doby, prof, rdoby$ 
2  (0 CONCATENATION_VALUE(cst, doby, prof, rdoby) =
3  IF IsAttributeSet2.62(cst) and 'concatenation' ∈ NAMS1.18(cst) and
4  not IsPlaceholder1.19(C  $\hat{cst}$  • 'concatenation')
5  THEN C  $\hat{cst}$  • 'concatenation'
6  ELSE STEP_BL_VALUE9.2(cst, doby, 'concatenation') ORELSE9.1
7  (1 STEP_DL_VALUE9.9(cst, doby, 'concatenation') ORELSE9.1
8  (2 STEP_FL_VALUE9.15(cst, doby, rdoby, prof, 'concatenation') ORELSE9.1
9  (3 STEP_G_VALUE9.17(cst, doby, rdoby, prof, 'concatenation', OBJECT_TYPE_VALUE9.61(cst, doby))
10 ORELSE9.1 'non-concatenated' )3)2)1)0)

```

Semiformal Description 9.44

Function "Value of the attribute 'content architecture class'"

The value of the attribute 'content architecture class' may be specified directly on a particular constituent (3-5). If this is not the case, test if the attribute value can be found by step C (6). If this is not the case, test if the attribute value can be found by step E (7). If this is not the case, test if the attribute value can be found by step G (8, 9). If this is not the case, test if the attribute value can be found by step H (10). If this is not the case, the value of the attribute 'content architecture class' is '2 8 2 6 0'.

Definition 9.44

```

1   $\forall cst, doby, prof, rdoby$ 
2  (0CONTENT_ARCHITECTURE_CLASS_VALUE(cst, doby, prof, rdoby) =
3  IF IsAttributeSet2.62(cst) and 'content architecture class'  $\in$  NAMS1.18(cst) and
4  not IsPlaceholder1.19( $C \sim cst \cdot$  'content architecture class')
5  THEN  $C \sim cst \cdot$  'content architecture class'
6  ELSE STEP_C_VALUE9.6(cst, doby, 'content architecture class') ORELSE9.1
7  (1STEP_E_VALUE9.10(cst, doby, rdoby, prof, 'content architecture class') ORELSE9.1
8  (2STEP_G_VALUE9.17(cst, doby, rdoby, prof, 'content architecture class',
9  OBJECT_TYPE_VALUE9.61(cst, doby)) ORELSE9.1
10 (3STEP_H_VALUE9.22(prof, 'content architecture class')
11 ORELSE9.1 '2 8 2 6 0' 3)2)1)0)

```

Semiformal Description 9.45

Function "Value of the attribute 'dimensions'"

The value of the attribute 'dimensions' comprises the parameter 'horizontal dimension' whose value is a horizontal dimension value, and the parameter 'vertical dimension' whose value is a vertical dimension value.

Definition 9.45

```

1   $\forall cst, doby, prof, rdoby$ 
2  (0DIMENSIONS_VALUE(cst, doby, prof, rdoby) =
3  ['horizontal dimension' : HORIZONTAL_DIMENSION_VALUE9.46(cst, doby, prof, rdoby);
4  'vertical dimension' : VERTICAL_DIMENSION_VALUE9.47(cst, doby, prof, rdoby)]0)

```

Semiformal Description 9.46

Function "Value of the parameter 'horizontal dimension'"

The value of the parameter 'horizontal dimension' may be specified directly on a particular constituent (3-6). If this is not the case, test if the parameter value can be found by step C (7). If this is not the case, test if the parameter value can be found by step E (8). If this is not the case, test if the parameter value can be found by step G (9). If this is not the case and the constituent is a page, test if the parameter value can be found by step H (11, 12). If step H provides no value or if the constituent is not a page, the value is the parameter 'horizontal dimension' whose value is the sub-parameter 'fixed dimension' with a fixed dimension value (14).

Definition 9.46

```

1   $\forall cst, doby, prof, rdoby$ 
2  (0 HORIZONTAL_DIMENSION_VALUE(cst, doby, prof, rdoby) =
3    IF IsAttributeSet2.62(cst) and 'dimensions' ∈ NAMS1.18(cst) and
4      'horizontal dimension' ∈ NAMS1.18(C ^ cst . 'dimensions') and
5      not IsPlaceholder1.19(C ^ cst . 'dimensions' . 'horizontal dimension')
6    THEN C ^ cst . 'dimensions' . 'horizontal dimension'
7    ELSE STEP_C_PAR_VALUE9.24(cst, doby, 'dimensions', 'horizontal dimension') ORELSE9.1
8    (1 STEP_E_PAR_VALUE9.26(cst, doby, rdoby, prof, 'dimensions', 'horizontal dimension') ORELSE9.1
9      (2 STEP_G_PAR_VALUE9.28(cst, doby, rdoby, prof, 'dimensions', 'horizontal dimension',
10         OBJECT_TYPE_VALUE9.61(cst, doby)) ORELSE9.1
11         (3 (4 IF OBJECT_TYPE_VALUE9.61(cst, doby) = 'composite or basic page'
12            THEN STEP_H_PAR_VALUE9.29(prof, 'dimensions', 'horizontal dimension')
13            ORELSE9.1 UNDEF4))
14         ORELSE9.1 ['fixed dimension' : HORIZONTAL_FIXED_DIMENSION_VALUE9.48(cst)]3)2)1)0)

```

Semiformal Description 9.47

Function "Value of the parameter 'vertical dimension'"

The value of the parameter 'vertical dimension' may be specified directly on a particular constituent (3-6). If this is not the case, test if the parameter value can be found by step C (7). If this is not the case, test if the parameter value can be found by step E (8). If this is not the case, test if the parameter value can be found by step G (9). If this is not the case and the constituent is a page, test if the parameter value can be found by step H (11, 12). If step H provides no value or if the constituent is not a page, the value is the parameter 'vertical dimension' whose value is the sub-parameter 'fixed dimension' with a fixed dimension value (14).

Definition 9.47

```

1   $\forall cst, doby, prof, rdoby$ 
2  (0 VERTICAL_DIMENSION_VALUE(cst, doby, prof, rdoby) =
3    IF IsAttributeSet2.62(cst) and 'dimensions' ∈ NAMS1.18(cst) and
4      'vertical dimension' ∈ NAMS1.18(C ^ cst . 'dimensions') and
5      not IsPlaceholder1.19(C ^ cst . 'dimensions' . 'vertical dimension')
6    THEN C ^ cst . 'dimensions' . 'vertical dimension'
7    ELSE STEP_C_PAR_VALUE9.24(cst, doby, 'dimensions', 'vertical dimension') ORELSE9.1
8    (1 STEP_E_PAR_VALUE9.26(cst, doby, rdoby, prof, 'dimensions', 'vertical dimension') ORELSE9.1
9      (2 STEP_G_PAR_VALUE9.28(cst, doby, rdoby, prof, 'dimensions', 'vertical dimension',
10         OBJECT_TYPE_VALUE9.61(cst, doby)) ORELSE9.1
11         (3 (4 IF OBJECT_TYPE_VALUE9.61(cst, doby) = 'composite or basic page'
12            THEN STEP_H_PAR_VALUE9.29(prof, 'dimensions', 'vertical dimension')
13            ORELSE9.1 UNDEF4))
14         ORELSE9.1 ['fixed dimension' : VERTICAL_FIXED_DIMENSION_VALUE9.50(cst)]3)2)1)0)

```

Semiformal Description 9.48

Function “horizontal fixed dimension value”

The horizontal fixed dimension value is 9240 (the horizontal dimension of the assured reproduction area of an ISO A4 page) for a page and an available horizontal dimension value otherwise.

Definition 9.48

```

1   $\forall cst$ 
2  ( $_o$  HORIZONTAL_FIXED_DIMENSION_VALUE( $cst$ ) =
3  IF OBJECT_TYPE_VALUE9.61( $cst, doby$ ) = 'composite or basic page'
4  THEN 9240
5  ELSE AVAILABLE_HORIZONTAL_DIMENSION_VALUE9.49( $cst$ ) $_o$ )

```

Semiformal Description 9.49

Function “available horizontal dimension value”

The available horizontal dimension is determined during the layout process. It is considered an atom-valued function here.

Definition 9.49

```

1   $\forall cst$ 
2  ( $_o$  IsAtom(AVAILABLE_HORIZONTAL_DIMENSION_VALUE( $cst$ )) $_o$ )

```

Semiformal Description 9.50

Function “vertical fixed dimension value”

The vertical fixed dimension value is 13200 (the vertical dimension of the reassured reproduction area of an ISO A4 page) for a page and an available vertical dimension value otherwise.

Definition 9.50

```

1   $\forall cst$ 
2  ( $_o$  VERTICAL_FIXED_DIMENSION_VALUE( $cst$ ) =
3  IF OBJECT_TYPE_VALUE9.61( $cst, doby$ ) = 'composite or basic page'
4  THEN 13200
5  ELSE AVAILABLE_VERTICAL_DIMENSION_VALUE9.51( $cst$ ) $_o$ )

```

Semiformal Description 9.51

Function “available vertical dimension value”

The available vertical dimension is determined during the layout process. It is considered an atom-valued function here.

Definition 9.51

```

1   $\forall cst$ 
2  ( $_o$  IsAtom(AVAILABLE_VERTICAL_DIMENSION_VALUE( $cst$ )) $_o$ )

```

Semiformal Description 9.52

Function “Value of the attribute ‘fill order’”

The value of the attribute ‘fill order’ may be specified directly on a particular constituent (3-5). If this is not the case, test if the attribute value can be found by step BL (6). If this is not the case, test if the attribute value can be found by step DL (7). If this is not the case, test if the attribute value can be found by step FL (8). If this is not the case, test if the attribute value can be found by step G (9). If this is not the case, the value of the attribute ‘fill order’ is ‘normal order’.

Definition 9.52

```

1   $\forall cst, doby, prof, rdoby$ 
2  (0 FILL_ORDER_VALUE(cst, doby, prof, rdoby) =
3  IF IsAttributeSet2.62(cst) and ‘fill order’ ∈ NAMS1.18(cst) and
4  not IsPlaceholder1.19(C ^ cst . ‘fill order’)
5  THEN C ^ cst . ‘fill order’
6  ELSE STEP_BL_VALUE9.2(cst, doby, ‘fill order’) ORELSE9.1
7  (1 STEP_DL_VALUE9.9(cst, doby, ‘fill order’) ORELSE9.1
8  (2 STEP_FL_VALUE9.15(cst, doby, rdoby, prof, ‘fill order’) ORELSE9.1
9  (3 STEP_G_VALUE9.17(cst, doby, rdoby, prof, ‘fill order’, OBJECT_TYPE_VALUE9.61(cst, doby))
10  ORELSE9.1 ‘normal order’3)2)1)0)

```

Semiformal Description 9.53

Function “Value of the attribute ‘indivisibility’”

The value of the attribute ‘indivisibility’ may be specified directly on a particular constituent (3-5). If this is not the case, test if the attribute value can be found by step BL (6). If this is not the case, test if the attribute value can be found by step DL (7). If this is not the case, test if the attribute value can be found by step FL (8). If this is not the case, test if the attribute value can be found by step G (9). If this is not the case, the value of the attribute ‘indivisibility’ is ‘null’.

Definition 9.53

```

1   $\forall cst, doby, prof, rdoby$ 
2  (0 INDIVISIBILITY_VALUE(cst, doby, prof, rdoby) =
3  IF IsAttributeSet2.62(cst) and ‘indivisibility’ ∈ NAMS1.18(cst) and
4  not IsPlaceholder1.19(C ^ cst . ‘indivisibility’)
5  THEN C ^ cst . ‘indivisibility’
6  ELSE STEP_BL_VALUE9.2(cst, doby, ‘indivisibility’) ORELSE9.1
7  (1 STEP_DL_VALUE9.9(cst, doby, ‘indivisibility’) ORELSE9.1
8  (2 STEP_FL_VALUE9.15(cst, doby, rdoby, prof, ‘indivisibility’) ORELSE9.1
9  (3 STEP_G_VALUE9.17(cst, doby, rdoby, prof, ‘indivisibility’, OBJECT_TYPE_VALUE9.61(cst, doby))
10  ORELSE9.1 ‘null’3)2)1)0)

```

Semiformal Description 9.54

Function “Value of the attribute 'layout category'”

The value of the attribute 'layout category' may be specified directly on a particular constituent (3-5). If this is not the case, test if the attribute value can be found by step BL (6). If this is not the case, test if the attribute value can be found by step DL (7). If this is not the case, test if the attribute value can be found by step FL (8). If this is not the case, test if the attribute value can be found by step G (9). If this is not the case, the value of the attribute 'layout category' is 'null'.

Definition 9.54

```

1   $\forall cst, doby, prof, rdoby$ 
2  (0 LAYOUT_CATEGORY_VALUE(cst, doby, prof, rdoby) =
3  IF IsAttributeSet2.62(cst) and 'layout category' ∈ NAMS1.18(cst) and
4  not IsPlaceholder1.19(C  $\hat{cst}$  . 'layout category')
5  THEN C  $\hat{cst}$  . 'layout category'
6  ELSE STEP_BL_VALUE9.2(cst, doby, 'layout category') ORELSE9.1
7  (1 STEP_DL_VALUE9.9(cst, doby, 'layout category') ORELSE9.1
8  (2 STEP_FL_VALUE9.15(cst, doby, rdoby, prof, 'layout category') ORELSE9.1
9  (3 STEP_G_VALUE9.17(cst, doby, rdoby, prof, 'layout category', OBJECT_TYPE_VALUE9.61(cst, doby))
10  ORELSE9.1 'null' 3)2)1)0)

```

Semiformal Description 9.55

Function “Value of the attribute 'layout object class'”

The value of the attribute 'layout object class' may be specified directly on a particular constituent (3-5). If this is not the case, test if the attribute value can be found by step BL (6). If this is not the case, test if the attribute value can be found by step DL (7). If this is not the case, test if the attribute value can be found by step FL (8). If this is not the case, test if the attribute value can be found by step G (9, 10). If this is not the case, the value of the attribute 'layout object class' is 'null'.

Definition 9.55

```

1   $\forall cst, doby, prof, rdoby$ 
2  (0 LAYOUT_OBJECT_CLASS_VALUE(cst, doby, prof, rdoby) =
3  IF IsAttributeSet2.62(cst) and 'layout object class' ∈ NAMS1.18(cst) and
4  not IsPlaceholder1.19(C  $\hat{cst}$  . 'layout object class')
5  THEN C  $\hat{cst}$  . 'layout object class'
6  ELSE STEP_BL_VALUE9.2(cst, doby, 'layout object class') ORELSE9.1
7  (1 STEP_DL_VALUE9.9(cst, doby, 'layout object class') ORELSE9.1
8  (2 STEP_FL_VALUE9.15(cst, doby, rdoby, prof, 'layout object class') ORELSE9.1
9  (3 STEP_G_VALUE9.17(cst, doby, rdoby, prof, 'layout object class',
10  OBJECT_TYPE_VALUE9.61(cst, doby))
11  ORELSE9.1 'null' 3)2)1)0)

```

Semiformal Description 9.56

Function "Value of the attribute 'layout path'"

The value of the attribute 'layout path' may be specified directly on a particular constituent (3-5). If this is not the case, test if the attribute value can be found by step C (6). If this is not the case, test if the attribute value can be found by step E (7). If this is not the case, test if the attribute value can be found by step G (8, 9). If this is not the case, test if the attribute value can be found by step H (10). If this is not the case, the value of the attribute 'layout path' is '270°'.

Definition 9.56

```

1   $\forall cst, doby, prof, rdoby$ 
2  (0 LAYOUT_PATH_VALUE(cst, doby, prof, rdoby) =
3    IF IsAttributeSet2.62(cst) and 'layout path' ∈ NAMS1.18(cst) and
4      not IsPlaceholder1.19(C ^ cst . 'layout path')
5    THEN C ^ cst . 'layout path'
6    ELSE STEP_C_VALUE9.6(cst, doby, 'layout path') ORELSE9.1
7    (1 STEP_E_VALUE9.10(cst, doby, rdoby, prof, 'layout path') ORELSE9.1
8    (2 STEP_G_VALUE9.17(cst, doby, rdoby, prof, 'layout path', OBJECT_TYPE_VALUE9.61(cst, doby))
9    ORELSE9.1 (3 STEP_H_VALUE9.22(prof, 'layout path')
10   ORELSE9.1 ('270°' 3)2)1)0)

```

Semiformal Description 9.57

Function "Value of the attribute 'medium type'"

The value of the attribute 'medium type' comprises the parameter 'nominal page size' whose value is a nominal page size value, and the parameter 'side of sheet' whose value is a side of sheet value.

Definition 9.57

```

1   $\forall cst, doby, prof, rdoby$ 
2  (0 MEDIUM_TYPE_VALUE(cst, doby, prof, rdoby) =
3    ['nominal page size' : NOMINAL_PAGE_SIZE_VALUE9.58(cst, doby, prof, rdoby);
4    'side of sheet' : SIDE_OF_SHEET_VALUE9.59(cst, doby, prof, rdoby)]0)

```

Semiformal Description 9.58

Function "Value of the parameter 'nominal page size'"

The value of the parameter 'nominal page size' may be specified directly on a particular constituent (3-6). If this is not the case, test if the parameter value can be found by step C (7). If this is not the case, test if the parameter value can be found by step E (8). If this is not the case, test if the parameter value can be found by step G (9). If this is not the case, test if the parameter value can be found by step H (11). If this is not the case, the value of the parameter 'nominal page size' is the pair 9920 and 14030.

Definition 9.58

```

1   $\forall cst, doby, prof, rdoby$ 
2  (0 NOMINAL_PAGE_SIZE_VALUE(cst, doby, prof, rdoby) =
3  IF IsAttributeSet2.62(cst) and 'medium type'  $\in$  NAMS1.18(cst) and
4  'nominal page size'  $\in$  NAMS1.18(C  $\hat{cst}$  . 'medium type') and
5  not IsPlaceholder1.19(C  $\hat{cst}$  . 'medium type' . 'nominal page size')
6  THEN C  $\hat{cst}$  . 'medium type' . 'nominal page size'
7  ELSE STEP_C_PAR_VALUE9.24(cst, doby, 'medium type', 'nominal page size') ORELSE9.1
8  (1 STEP_E_PAR_VALUE9.26(cst, doby, rdoby, prof, 'medium type', 'nominal page size') ORELSE9.1
9  (2 STEP_G_PAR_VALUE9.28(cst, doby, rdoby, prof, 'medium type', 'nominal page size',
10  OBJECT_TYPE_VALUE9.61(cst, doby)) ORELSE9.1
11  (3 STEP_H_PAR_VALUE9.29(prof, 'medium type', 'nominal page size')
12  ORELSE9.1 [  $\rightarrow$  9920  $\rightarrow$  14030  $\rightarrow$  ]3)2)1)0)

```

Semiformal Description 9.59

Function "Value of the parameter 'side of sheet'"

The value of the parameter 'side of sheet' may be specified directly on a particular constituent (3-6). If this is not the case, test if the parameter value can be found by step C (7). If this is not the case, test if the parameter value can be found by step E (8). If this is not the case, test if the parameter value can be found by step G (9). If this is not the case, test if the parameter value can be found by step H (11). If this is not the case, the value of the parameter 'side of sheet' is 'unspecified'.

Definition 9.59

```

1   $\forall cst, doby, prof, rdoby$ 
2  (0 SIDE_OF_SHEET_VALUE(cst, doby, prof, rdoby) =
3  IF IsAttributeSet2.62(cst) and 'medium type'  $\in$  NAMS1.18(cst) and
4  'side of sheet'  $\in$  NAMS1.18(C  $\hat{cst}$  . 'medium type') and
5  not IsPlaceholder1.19(C  $\hat{cst}$  . 'medium type' . 'side of sheet')
6  THEN C  $\hat{cst}$  . 'medium type' . 'side of sheet'
7  ELSE STEP_C_PAR_VALUE9.24(cst, doby, 'medium type', 'side of sheet') ORELSE9.1
8  (1 STEP_E_PAR_VALUE9.26(cst, doby, rdoby, prof, 'medium type', 'side of sheet') ORELSE9.1
9  (2 STEP_G_PAR_VALUE9.28(cst, doby, rdoby, prof, 'medium type', 'side of sheet',
10  OBJECT_TYPE_VALUE9.61(cst, doby)) ORELSE9.1
11  (3 STEP_H_PAR_VALUE9.29(prof, 'medium type', 'side of sheet')
12  ORELSE9.1 'unspecified')3)2)1)0)

```

Semiformal Description 9.60

Function “Value of the attribute 'new layout object'”

The value of the attribute 'new layout object' may be specified directly on a particular constituent (3-5). If this is not the case, test if the attribute value can be found by step BL (6). If this is not the case, test if the attribute value can be found by step DL (7). If this is not the case, test if the attribute value can be found by step FL (8). If this is not the case, test if the attribute value can be found by step G (9, 10). If this is not the case, the value of the attribute 'new layout object' is 'null'.

Definition 9.60

```

1  ∀ cst, doby, prof, rdoby
2  (₀ NEW_LAYOUT_OBJECT_VALUE(cst, doby, prof, rdoby) =
3    IF IsAttributeSet2.62(cst) and 'new layout object' ∈ NAMS1.18(cst) and
4      not IsPlaceholder1.19(C ^cst. 'new layout object')
5    THEN C ^cst. 'new layout object'
6    ELSE STEP_BL_VALUE9.2(cst, doby, 'new layout object') ORELSE9.1
7      (₁ STEP_DL_VALUE9.9(cst, doby, 'new layout object') ORELSE9.1
8        (₂ STEP_FL_VALUE9.15(cst, doby, rdoby, prof, 'new layout object') ORELSE9.1
9          (₃ STEP_G_VALUE9.17(cst, doby, rdoby, prof, 'new layout object'),
10             OBJECT_TYPE_VALUE9.61(cst, doby))
11          ORELSE9.1 'null' )₂)₁)₀)

```

Semiformal Description 9.61

Function “Value of the attribute 'object type'”

The value of the attribute 'object type' may be specified directly on a particular constituent (3-5). If this is not the case, the attribute value is determined by step C (6).

NOTE — An undefined result of step C is not possible for a valid ODA document.

Definition 9.61

```

1  ∀ cst, doby
2  (₀ OBJECT_TYPE_VALUE(cst, doby) =
3    IF IsAttributeSet2.62(cst) and 'object type' ∈ NAMS1.18(cst) and
4      not IsPlaceholder1.19(C ^cst. 'object type')
5    THEN C ^cst. 'object type'
6    ELSE STEP_C_VALUE9.6(cst, doby, 'object type')₀)

```

Semiformal Description 9.62

Function “Value of the attribute 'offset'”

The value of the attribute 'offset' comprises the parameter 'leading offset' whose value is a leading offset value, the parameter 'trailing offset' whose value is a trailing offset value, the parameter 'left-hand offset' whose value is a left-hand offset value, and the parameter 'right-hand offset' whose value is a right-hand offset value.

Definition 9.62

```

1  ∀ cst, doby, prof, rdoby
2  (₀ OFFSET_VALUE(cst, doby, prof, rdoby) =
3    ['leading offset' : LEADING_OFFSET_VALUE9.63(cst, doby, prof, rdoby);
4     'trailing offset' : TRAILING_OFFSET_VALUE9.64(cst, doby, prof, rdoby);
5     'left-hand offset' : LEFTHAND_OFFSET_VALUE9.65(cst, doby, prof, rdoby);
6     'right-hand offset' : RIGHTHAND_OFFSET_VALUE9.66(cst, doby, prof, rdoby)]₀)

```

Semiformal Description 9.63

Function "Value of the parameter 'leading offset'"

The value of the parameter 'leading offset' may be specified directly on a particular constituent (3-6). If this is not the case, test if the parameter value can be found by step BL (7). If this is not the case, test if the parameter value can be found by step DL (8). If this is not the case, test if the parameter value can be found by step FL (9). If this is not the case, test if the parameter value can be found by step G (10, 11). If this is not the case, the value of the parameter 'leading offset' is 0.

Definition 9.63

```

1   $\forall cst, doby, prof, rdoby$ 
2  (0 LEADING_OFFSET_VALUE(cst, doby, prof, rdoby) =
3  IF IsAttributeSet2.62(cst) and 'offset' ∈ NAMS1.18(cst) and
4  'leading offset' ∈ NAMS1.18(C  $\hat{cst}$  . 'offset') and
5  not IsPlaceholder1.19(C  $\hat{cst}$  . 'offset' . 'leading offset')
6  THEN C  $\hat{cst}$  . 'offset' . 'leading offset'
7  ELSE STEP_BL_PAR_VALUE9.23(cst, doby, 'offset', 'leading offset') ORELSE9.1
8  (1 STEP_DL_PAR_VALUE9.25(cst, doby, 'offset', 'leading offset') ORELSE9.1
9  (2 STEP_FL_PAR_VALUE9.27(cst, doby, rdoby, prof, 'offset', 'leading offset') ORELSE9.1
10 (3 STEP_G_PAR_VALUE9.28(cst, doby, rdoby, prof, 'offset', 'leading offset',
11 OBJECT_TYPE_VALUE9.61(cst, doby))
12 ORELSE9.1 03)2)1)0)
```

Semiformal Description 9.64

Function "Value of the parameter 'trailing offset'"

The value of the parameter 'trailing offset' may be specified directly on a particular constituent (3-6). If this is not the case, test if the parameter value can be found by step BL (7). If this is not the case, test if the parameter value can be found by step DL (8). If this is not the case, test if the parameter value can be found by step FL (9). If this is not the case, test if the parameter value can be found by step G (10, 11). If this is not the case, the value of the parameter 'trailing offset' is 0.

Definition 9.64

```

1   $\forall cst, doby, prof, rdoby$ 
2  (0 TRAILING_OFFSET_VALUE(cst, doby, prof, rdoby) =
3  IF IsAttributeSet2.62(cst) and 'offset' ∈ NAMS1.18(cst) and
4  'trailing offset' ∈ NAMS1.18(C  $\hat{cst}$  . 'offset') and
5  not IsPlaceholder1.19(C  $\hat{cst}$  . 'offset' . 'trailing offset')
6  THEN C  $\hat{cst}$  . 'offset' . 'trailing offset'
7  ELSE STEP_BL_PAR_VALUE9.23(cst, doby, 'offset', 'trailing offset') ORELSE9.1
8  (1 STEP_DL_PAR_VALUE9.25(cst, doby, 'offset', 'trailing offset') ORELSE9.1
9  (2 STEP_FL_PAR_VALUE9.27(cst, doby, rdoby, prof, 'offset', 'trailing offset') ORELSE9.1
10 (3 STEP_G_PAR_VALUE9.28(cst, doby, rdoby, prof, 'offset', 'trailing offset',
11 OBJECT_TYPE_VALUE9.61(cst, doby))
12 ORELSE9.1 03)2)1)0)
```

Semiformal Description 9.65

Function “Value of the parameter 'left-hand offset'”

The value of the parameter 'left-hand offset' may be specified directly on a particular constituent (3-6). If this is not the case, test if the parameter value can be found by step BL (7). If this is not the case, test if the parameter value can be found by step DL (8). If this is not the case, test if the parameter value can be found by step FL (9). If this is not the case, test if the parameter value can be found by step G (10, 11). If this is not the case, the value of the parameter 'left-hand offset' is 0.

Definition 9.65

```

1   $\forall cst, doby, prof, rdoby$ 
2  (0 LEFTHAND_OFFSET_VALUE(cst, doby, prof, rdoby) =
3    IF IsAttributeSet2.62(cst) and 'offset' ∈ NAMS1.18(cst) and
4      'left-hand offset' ∈ NAMS1.18(C ^ cst . 'offset') and
5      not IsPlaceholder1.19(C ^ cst . 'offset' . 'left-hand offset')
6    THEN C ^ cst . 'offset' . 'left-hand offset'
7    ELSE STEP_BL_PAR_VALUE9.23(cst, doby, 'offset', 'left-hand offset') ORELSE9.1
8    (1 STEP_DL_PAR_VALUE9.25(cst, doby, 'offset', 'left-hand offset') ORELSE9.1
9    (2 STEP_FL_PAR_VALUE9.27(cst, doby, rdoby, prof, 'offset', 'left-hand offset') ORELSE9.1
10   (3 STEP_G_PAR_VALUE9.28(cst, doby, rdoby, prof, 'offset', 'left-hand offset',
11     OBJECT_TYPE_VALUE9.61(cst, doby))
12   ORELSE9.1 (0))1)0)
```

Semiformal Description 9.66

Function “Value of the parameter 'right-hand offset'”

The value of the parameter 'right-hand offset' may be specified directly on a particular constituent (3-6). If this is not the case, test if the parameter value can be found by step BL (7). If this is not the case, test if the parameter value can be found by step DL (8). If this is not the case, test if the parameter value can be found by step FL (9). If this is not the case, test if the parameter value can be found by step G (10, 11). If this is not the case, the value of the parameter 'right-hand offset' is 0.

Definition 9.66

```

1   $\forall cst, doby, prof, rdoby$ 
2  (0 RIGHTHAND_OFFSET_VALUE(cst, doby, prof, rdoby) =
3    IF IsAttributeSet2.62(cst) and 'offset' ∈ NAMS1.18(cst) and
4      'right-hand offset' ∈ NAMS1.18(C ^ cst . 'offset') and
5      not IsPlaceholder1.19(C ^ cst . 'offset' . 'right-hand offset')
6    THEN C ^ cst . 'offset' . 'right-hand offset'
7    ELSE STEP_BL_PAR_VALUE9.23(cst, doby, 'offset', 'right-hand offset') ORELSE9.1
8    (1 STEP_DL_PAR_VALUE9.25(cst, doby, 'offset', 'right-hand offset') ORELSE9.1
9    (2 STEP_FL_PAR_VALUE9.27(cst, doby, rdoby, prof, 'offset', 'right-hand offset') ORELSE9.1
10   (3 STEP_G_PAR_VALUE9.28(cst, doby, rdoby, prof, 'offset', 'right-hand offset',
11     OBJECT_TYPE_VALUE9.61(cst, doby))
12   ORELSE9.1 (0))2)0)
```

Semiformal Description 9.67

Function “Value of the attribute 'page position'”

The value of the attribute 'page position' may be specified directly on a particular constituent (3-5). If this is not the case, test if the attribute value can be found by step C (6). If this is not the case, test if the attribute value can be found by step E (7). If this is not the case, test if the attribute value can be found by step G (8, 9). If this is not the case, test if the attribute value can be found by step H (10). If this is not the case, the value of the attribute 'page position' is “such that edge losses are minimized” (clause 5.4.3.4).

Definition 9.67

```

1   $\forall cst, doby, prof, rdoby$ 
2  (0 PAGE_POSITION_VALUE( $cst, doby, prof, rdoby$ ) =
3  IF IsAttributeSet2.62( $cst$ ) and 'page position'  $\in$  NAMS1.18( $cst$ ) and
4  not IsPlaceholder1.19( $C \hat{cst} \cdot$  'page position')
5  THEN  $C \hat{cst} \cdot$  'page position'
6  ELSE STEP_C_VALUE9.6( $cst, doby, 'page position'$ ) ORELSE9.1
7  (1 STEP_E_VALUE9.10( $cst, doby, rdoby, prof, 'page position'$ ) ORELSE9.1
8  (2 STEP_G_VALUE9.17( $cst, doby, rdoby, prof, 'page position',$ 
9  OBJECT_TYPE_VALUE9.61( $cst, doby$ )) ORELSE9.1
10 (3 STEP_H_VALUE9.22( $prof, 'page position'$ )
11 ORELSE9.1 MINIMUM_EDGE_LOSS9.68( $cst$ )3)2)1)0)
```

Semiformal Description 9.68

Function “minimum edge loss for the attribute 'page position'”

The computation of a minimum edge loss for the attribute 'page position' is implementation dependent and therefore considered an atom-valued function.

Definition 9.68

```

1   $\forall cst$ 
2  (0 IsAtom(MINIMUM_EDGE_LOSS( $cst$ ))0)
```

Semiformal Description 9.69

Function “Value of the attribute 'permitted categories'”

The value of the attribute 'permitted categories' may be specified directly on a particular constituent (3-5). If this is not the case, test if the attribute value can be found by step C (6). If this is not the case, test if the attribute value can be found by step E (7). If this is not the case, test if the attribute value can be found by step G (8, 9). If this is not the case, the value of the attribute 'permitted categories' is 'null'.

Definition 9.69

```

1   $\forall cst, doby, prof, rdoby$ 
2  (0 PERMITTED_CATEGORIES_VALUE( $cst, doby, prof, rdoby$ ) =
3  IF IsAttributeSet2.62( $cst$ ) and 'permitted categories'  $\in$  NAMS1.18( $cst$ ) and
4  not IsPlaceholder1.19( $C \hat{cst} \cdot$  'permitted categories')
5  THEN  $C \hat{cst} \cdot$  'permitted categories'
6  ELSE STEP_C_VALUE9.6( $cst, doby, 'permitted categories'$ ) ORELSE9.1
7  (1 STEP_E_VALUE9.10( $cst, doby, rdoby, prof, 'permitted categories'$ ) ORELSE9.1
8  (2 STEP_G_VALUE9.17( $cst, doby, rdoby, prof, 'permitted categories',$ 
9  OBJECT_TYPE_VALUE9.61( $cst, doby$ ))
10 ORELSE9.1 'null'2)1)0)
```

Semiformal Description 9.70

Function “Value of the attribute 'position'”

The value of the attribute 'position' is the parameter 'fixed position' whose value is a fixed position value.

NOTE — The parameter 'variable position' can only be specified for certain frame classes, not for objects.

Definition 9.70

- 1 $\forall cst, doby, prof, rdoby$
- 2 $(_0 \text{ POSITION_VALUE}(cst, doby, prof, rdoby) =$
- 3 $[\text{'fixed position'} : \text{FIXED_POSITION_VALUE}^{9.71}(cst, doby, prof, rdoby)]_0)$

Semiformal Description 9.71

Function “Value of the parameter 'fixed position'”

The value of the parameter 'fixed position' is the sub-parameter 'horizontal position' whose value is a horizontal position value, and the sub-parameter 'vertical position' whose value is a vertical position value.

Definition 9.71

- 1 $\forall cst, doby, prof, rdoby$
- 2 $(_0 \text{ FIXED_POSITION_VALUE}(cst, doby, prof, rdoby) =$
- 3 $[\text{'horizontal position'} : \text{HORIZONTAL_POSITION_VALUE}^{9.72}(cst, doby, prof, rdoby);$
- 4 $\text{'vertical position'} : \text{VERTICAL_POSITION_VALUE}^{9.73}(cst, doby, prof, rdoby)]_0)$

Semiformal Description 9.72

Function “Value of the sub-parameter 'horizontal position'”

The value of the sub-parameter 'horizontal position' may be specified directly on a particular constituent (3-6). If this is not the case, test if the sub-parameter value can be found by step C (7). If this is not the case, test if the sub-parameter value can be found by step E (8, 9). If this is not the case, test if the sub-parameter value can be found by step G (10, 11). If this is not the case, the value of the sub-parameter 'horizontal position' is 0.

Definition 9.72

- 1 $\forall cst, doby, prof, rdoby$
- 2 $(_0 \text{ HORIZONTAL_POSITION_VALUE}(cst, doby, prof, rdoby) =$
- 3 $\text{IF IsAttributeSet}^{2.62}(cst) \text{ and 'position'} \in \text{NAMS}^{1.18}(cst) \text{ and}$
- 4 $\text{'horizontal position'} \in \text{NAMS}^{1.18}(C \hat{=} cst \cdot \text{'position'} \cdot \text{'fixed position'}) \text{ and}$
- 5 $\text{not IsPlaceholder}^{1.19}(C \hat{=} cst \cdot \text{'position'} \cdot \text{'fixed position'} \cdot \text{'horizontal position'})$
- 6 $\text{THEN } C \hat{=} cst \cdot \text{'position'} \cdot \text{'fixed position'} \cdot \text{'horizontal position'}$
- 7 $\text{ELSE STEP_C_SUBPAR_VALUE}^{9.34}(cst, doby, \text{'position'}, \text{'fixed position'}, \text{'horizontal position'}) \text{ ORELSE}^{9.1}$
- 8 $(\text{STEP_E_SUBPAR_VALUE}^{9.35}(cst, doby, rdoby, prof, \text{'position'}, \text{'fixed position'},$
- 9 $\text{'horizontal position'}) \text{ ORELSE}^{9.1}$
- 10 $(\text{STEP_G_SUBPAR_VALUE}^{9.36}(cst, doby, rdoby, prof, \text{'position'}, \text{'fixed position'},$
- 11 $\text{'horizontal position'}, \text{OBJECT_TYPE_VALUE}^{9.61}(cst, doby))$
- 12 $\text{ORELSE}^{9.1} 0_2)_0)$

Semiformal Description 9.73

Function "Value of the sub-parameter 'vertical position'"

The value of the sub-parameter 'vertical position' may be specified directly on a particular constituent (3-6). If this is not the case, test if the sub-parameter value can be found by step C (7). If this is not the case, test if the sub-parameter value can be found by step E (8, 9). If this is not the case, test if the sub-parameter value can be found by step G (10, 11). If this is not the case, the value of the sub-parameter 'vertical position' is 0.

Definition 9.73

```

1   $\forall cst, doby, prof, rdoby$ 
2  (0 VERTICAL_POSITION_VALUE(cst, doby, prof, rdoby) =
3  IF IsAttributeSet2.62(cst) and 'position' ∈ NAMS1.18(cst) and
4  'vertical position' ∈ NAMS1.18(C  $\sim$  cst . 'position' . 'fixed position') and
5  not IsPlaceholder1.19(C  $\sim$  cst . 'position' . 'fixed position' . 'vertical position')
6  THEN C  $\sim$  cst . 'position' . 'fixed position' . 'vertical position'
7  ELSE STEP_C_SUBPAR_VALUE9.34(cst, doby, 'position', 'fixed position', 'vertical position') ORELSE9.1
8  (1 STEP_E_SUBPAR_VALUE9.35(cst, doby, rdoby, prof, 'position', 'fixed position',
9  'vertical position') ORELSE9.1
10 (2 STEP_G_SUBPAR_VALUE9.36(cst, doby, rdoby, prof, 'position', 'fixed position',
11 'vertical position', OBJECT_TYPE_VALUE9.61(cst, doby))
12 ORELSE9.1 02),1),0)

```

Semiformal Description 9.74

Function "Value of the attribute 'protection'"

The value of the attribute 'protection' may be specified directly on a particular constituent (3-5). If this is not the case, test if the attribute value can be found by step C (6). If this is not the case, test if the attribute value can be found by step E (7). If this is not the case, test if the attribute value can be found by step G (8). If this is not the case, the value of the attribute 'protection' is 'unprotected'.

Definition 9.74

```

1   $\forall cst, doby, prof, rdoby$ 
2  (0 PROTECTION_VALUE(cst, doby, prof, rdoby) =
3  IF IsAttributeSet2.62(cst) and 'protection' ∈ NAMS1.18(cst) and
4  not IsPlaceholder1.19(C  $\sim$  cst . 'protection')
5  THEN C  $\sim$  cst . 'protection'
6  ELSE STEP_C_VALUE9.6(cst, doby, 'protection') ORELSE9.1
7  (1 STEP_E_VALUE9.10(cst, doby, rdoby, prof, 'protection') ORELSE9.1
8  (2 STEP_G_VALUE9.17(cst, doby, rdoby, prof, 'protection', OBJECT_TYPE_VALUE9.61(cst, doby))
9  ORELSE9.1 'unprotected'2),1),0)

```

Semiformal Description 9.75

Function "Value of the attribute 'same layout object'"

The value of the attribute 'same layout object' may be specified directly on a particular constituent (3-5). If this is not the case, test if the attribute value can be found by step BL (6). If this is not the case, test if the attribute value can be found by step DL (7). If this is not the case, test if the attribute value can be found by step FL (8). If this is not the case, test if the attribute value can be found by step G (9, 10). If this is not the case, the value of the the attribute 'same layout object' is the parameter 'logical object' with the value 'null'.

Definition 9.75

```

1   $\forall cst, doby, prof, rdoby$ 
2  ( $_0$  SAME_LAYOUTOBJECT_VALUE( $cst, doby, prof, rdoby$ ) =
3    IF IsAttributeSet2.62( $cst$ ) and 'same layout object'  $\in$  NAMS1.18( $cst$ ) and
4      not IsPlaceholder1.19( $C \sim cst \bullet$  'same layout object')
5    THEN  $C \sim cst \bullet$  'same layout object'
6    ELSE STEP_BL_VALUE9.2( $cst, doby, 'same layout object'$ ) ORELSE9.1
7    ( $_1$  STEP_DL_VALUE9.9( $cst, doby, 'same layout object'$ ) ORELSE9.1
8      ( $_2$  STEP_FL_VALUE9.15( $cst, doby, rdoby, prof, 'same layout object'$ ) ORELSE9.1
9        ( $_3$  STEP_G_VALUE9.17( $cst, doby, rdoby, prof, 'same layout object'$ )
10         OBJECT_TYPE_VALUE9.61( $cst, doby$ ))
11    ORELSE9.1 ['logical object' : 'null']  $_3)_2)_1)_0$ )

```

Semiformal Description 9.76

Function "Value of the attribute 'separation'"

The value of the attribute 'separation' comprises the parameter 'leading edge' whose value is a leading edge value, the parameter 'trailing edge' whose value is a trailing edge value, and the parameter 'centre separation' whose value is a centre separation value.

Definition 9.76

```

1   $\forall cst, doby, prof, rdoby$ 
2  ( $_0$  SEPARATION_VALUE( $cst, doby, prof, rdoby$ ) =
3    ['leading edge' : LEADING_EDGE_VALUE9.77( $cst, doby, prof, rdoby$ );
4     'trailing edge' : TRAILING_EDGE_VALUE9.78( $cst, doby, prof, rdoby$ );
5     'centre separation' : CENTRE_SEPARATION_VALUE9.79( $cst, doby, prof, rdoby$ )]  $_0$ )

```

Semiformal Description 9.77

Function “Value of the parameter 'leading edge'”

The value of the parameter 'leading edge' may be specified directly on a particular constituent (3-6). If this is not the case, test if the parameter value can be found by step BL (7). If this is not the case, test if the parameter value can be found by step DL (8). If this is not the case, test if the parameter value can be found by step FL (9). If this is not the case, test if the parameter value can be found by step G (10, 11). If this is not the case, the value of the parameter 'leading edge' is 0.

Definition 9.77

```

1   $\forall cst, doby, prof, rdoby$ 
2  (0 LEADING_EDGE_VALUE(cst, doby, prof, rdoby) =
3  IF IsAttributeSet2.62(cst) and 'separation'  $\in$  NAMS1.18(cst) and
4  'leading edge'  $\in$  NAMS1.18( $C \hat{cst} \cdot$  'separation') and
5  not IsPlaceholder1.19( $C \hat{cst} \cdot$  'separation'  $\cdot$  'leading edge')
6  THEN  $C \hat{cst} \cdot$  'separation'  $\cdot$  'leading edge'
7  ELSE STEP_BL_PAR_VALUE9.23(cst, doby, 'separation', 'leading edge') ORELSE9.1
8  (1 STEP_DL_PAR_VALUE9.25(cst, doby, 'separation', 'leading edge') ORELSE9.1
9  (2 STEP_FL_PAR_VALUE9.27(cst, doby, rdoby, prof, 'separation', 'leading edge') ORELSE9.1
10 (3 STEP_G_PAR_VALUE9.28(cst, doby, rdoby, prof, 'separation', 'leading edge',
11 OBJECT_TYPE_VALUE9.61(cst, doby))
12 ORELSE9.1 03)2)1)0)

```

Semiformal Description 9.78

Function “Value of the parameter 'trailing edge'”

The value of the parameter 'trailing edge' may be specified directly on a particular constituent (3-6). If this is not the case, test if the parameter value can be found by step BL (7). If this is not the case, test if the parameter value can be found by step DL (8). If this is not the case, test if the parameter value can be found by step FL (9). If this is not the case, test if the parameter value can be found by step G (10, 11). If this is not the case, the value of the parameter 'trailing edge' is 0.

Definition 9.78

```

1   $\forall cst, doby, prof, rdoby$ 
2  (0 TRAILING_EDGE_VALUE(cst, doby, prof, rdoby) =
3  IF IsAttributeSet2.62(cst) and 'separation'  $\in$  NAMS1.18(cst) and
4  'trailing edge'  $\in$  NAMS1.18( $C \hat{cst} \cdot$  'separation') and
5  not IsPlaceholder1.19( $C \hat{cst} \cdot$  'separation'  $\cdot$  'trailing edge')
6  THEN  $C \hat{cst} \cdot$  'separation'  $\cdot$  'trailing edge'
7  ELSE STEP_BL_PAR_VALUE9.23(cst, doby, 'separation', 'trailing edge') ORELSE9.1
8  (1 STEP_DL_PAR_VALUE9.25(cst, doby, 'separation', 'trailing edge') ORELSE9.1
9  (2 STEP_FL_PAR_VALUE9.27(cst, doby, rdoby, prof, 'separation', 'trailing edge') ORELSE9.1
10 (3 STEP_G_PAR_VALUE9.28(cst, doby, rdoby, prof, 'separation', 'trailing edge',
11 OBJECT_TYPE_VALUE9.61(cst, doby))
12 ORELSE9.1 03)2)1)0)

```

Semiformal Description 9.79

Function “Value of the parameter 'centre separation'”

The value of the parameter 'centre separation' may be specified directly on a particular constituent (3-6). If this is not the case, test if the parameter value can be found by step BL (7). If this is not the case, test if the parameter value can be found by step DL (8). If this is not the case, test if the parameter value can be found by step FL (9). If this is not the case, test if the parameter value can be found by step G (10, 11). If this is not the case, the value of the parameter 'centre separation' is 0.

Definition 9.79

```

1   $\forall cst, doby, prof, rdoby$ 
2  (0 CENTRE_SEPARATION_VALUE(cst, doby, prof, rdoby) =
3  IF IsAttributeSet2.62(cst) and 'separation'  $\in$  NAMS1.18(cst) and
4  'centre separation'  $\in$  NAMS1.18( $C \hat{cst} \cdot$  'separation') and
5  not IsPlaceholder1.19( $C \hat{cst} \cdot$  'separation' . 'centre separation')
6  THEN  $C \hat{cst} \cdot$  'separation' . 'centre separation'
7  ELSE STEP_BL_PAR_VALUE9.23(cst, doby, 'separation', 'centre separation') ORELSE9.1
8  (1 STEP_DL_PAR_VALUE9.25(cst, doby, 'separation', 'centre separation') ORELSE9.1
9  (2 STEP_FL_PAR_VALUE9.27(cst, doby, rdoby, prof, 'separation', 'centre separation') ORELSE9.1
10 (3 STEP_G_PAR_VALUE9.28(cst, doby, rdoby, prof, 'separation', 'centre separation',
11 OBJECT_TYPE_VALUE9.61(cst, doby))
12 ORELSE9.1 03)2)1)0)

```

Semiformal Description 9.80

Function “Value of the attribute 'synchronization'”

The value of the attribute 'synchronization' may be specified directly on a particular constituent (3-5). If this is not the case, test if the attribute value can be found by step BL (6). If this is not the case, test if the attribute value can be found by step DL (7). If this is not the case, test if the attribute value can be found by step FL (8). If this is not the case, test if the attribute value can be found by step G (9). If this is not the case, the value of the attribute 'synchronization' is 'null'.

Definition 9.80

```

1   $\forall cst, doby, prof, rdoby$ 
2  (0 SYNCHRONIZATION_VALUE(cst, doby, prof, rdoby) =
3  IF IsAttributeSet2.62(cst) and 'synchronization'  $\in$  NAMS1.18(cst) and
4  not IsPlaceholder1.19( $C \hat{cst} \cdot$  'synchronization')
5  THEN  $C \hat{cst} \cdot$  'synchronization'
6  ELSE STEP_BL_VALUE9.2(cst, doby, 'synchronization') ORELSE9.1
7  (1 STEP_DL_VALUE9.9(cst, doby, 'synchronization') ORELSE9.1
8  (2 STEP_FL_VALUE9.15(cst, doby, rdoby, prof, 'synchronization') ORELSE9.1
9  (3 STEP_G_VALUE9.17(cst, doby, rdoby, prof, 'synchronization', OBJECT_TYPE_VALUE9.61(cst, doby))
10 ORELSE9.1 'null'3)2)1)0)

```

Semiformal Description 9.81

Function "Value of the attribute 'transparency'"

The value of the attribute 'transparency' may be specified directly on a particular constituent (3-5). If this is not the case, test if the attribute value can be found by step BP (6). If this is not the case, test if the attribute value can be found by step C (7). If this is not the case, test if the attribute value can be found by step DP (8). If this is not the case, test if the attribute value can be found by step E (9). If this is not the case, test if the attribute value can be found by step FP (10). If this is not the case, test if the attribute value can be found by step G (11, 12). If this is not the case, test if the attribute value can be found by step H (13). If this is not the case, the value of the attribute 'transparency' is 'transparent'.

Definition 9.81

```

1   $\forall cst, doby, prof, rdoby$ 
2  (0 TRANSPARENCY_VALUE(cst, doby, prof, rdoby) =
3  IF IsAttributeSet2.62(cst) and 'transparency' ∈ NAMS1.18(cst) and
4  not IsPlaceholder1.19(C  $\neg$  cst . 'transparency')
5  THEN C  $\neg$  cst . 'transparency'
6  ELSE STEP_BP_VALUE9.3(cst, doby, 'transparency') ORELSE9.1
7  (1 STEP_C_VALUE9.6(cst, doby, 'transparency') ORELSE9.1
8  (2 STEP_DP_VALUE9.8(cst, doby, 'transparency') ORELSE9.1
9  (3 STEP_E_VALUE9.10(cst, doby, rdoby, prof, 'transparency') ORELSE9.1
10 (4 STEP_FP_VALUE9.13(cst, doby, rdoby, prof, 'transparency') ORELSE9.1
11 (5 STEP_G_VALUE9.17(cst, doby, rdoby, prof, 'transparency'),
12 OBJECT_TYPE_VALUE9.61(cst, doby))
13 ORELSE9.1(6 STEP_H_VALUE9.22(prof, 'transparency')
14 ORELSE9.1 'transparent' 6)5)4)3)2)1)0)

```

Semiformal Description 9.82

Function "Value of the attribute 'type of coding'"

The value of the attribute 'type of coding' may be specified directly on a particular constituent (3-5). If this is not the case, test if the attribute value can be found by step C (6). If this is not the case, test if the attribute value can be found by step H (7). If this is not the case, the value of the attribute 'type of coding' is '2 8 3 6 0' for character content, '2 8 3 7 0' for raster graphics content, and '2 8 3 8 0' for geometric graphics content (8-10).

Definition 9.82

```

1   $\forall cst, doby, prof$ 
2  (0 TYPE_OF_CODING_VALUE(cst, doby, prof) =
3  IF IsAttributeSet2.62(cst) and 'type of coding' ∈ NAMS1.18(cst) and
4  not IsPlaceholder1.19(C  $\neg$  cst . 'type of coding')
5  THEN C  $\neg$  cst . 'type of coding'
6  ELSE STEP_C_VALUE9.6(cst, doby, 'type of coding') ORELSE9.1
7  (1 STEP_H_VALUE9.22(prof, 'type of coding') ORELSE9.1
8  (2 IF IsCharacterContentPortionDescription2.57(cst) THEN '2 8 3 6 0' ELSE
9  (3 IF IsRasterGraphicsContentPortionDescription2.58(cst) THEN '2 8 3 7 0' ELSE
10 (4 IF IsGeometricGraphicsContentPortionDescription2.59(cst) THEN '2 8 3 8 0' 4)3)2)1)0)

```

Semiformal Description 9.83

Function "Value of the attribute 'user-readable comments'"

The value of the attribute 'user-readable comments' may be specified directly on a particular constituent (3-5). If this is not the case, test if the attribute value can be found by step C (6). If this is not the case, test if the attribute value can be found by step E (7). If this is not the case, the value of the attribute 'user-readable comments' is the empty catenation.

Definition 9.83

```

1   $\forall cst, doby, prof, rdoby$ 
2  ( $_0$  USER_READABLE_COMMENTS_VALUE( $cst, doby, prof, rdoby$ ) =
3  IF IsAttributeSet2.62( $cst$ ) and 'user-readable comments'  $\in$  NAMS1.18( $cst$ ) and
4  not IsPlaceholder1.19( $C \sim cst \bullet$  'user-readable comments')
5  THEN  $C \sim cst \bullet$  'user-readable comments'
6  ELSE STEP_C_VALUE9.6( $cst, doby, 'user-readable comments'$ ) ORELSE9.1
7  ( $_1$  STEP_E_VALUE9.10( $cst, doby, rdoby, prof, 'user-readable comments'$ )
8  ORELSE9.1 [ $\rightarrow$ ]  $_0$ )
```

Semiformal Description 9.84

Function "Value of the attribute 'user-visible name'"

The value of the attribute 'user-visible name' may be specified directly on a particular constituent (3-5). If this is not the case, test if the attribute value can be found by step C (6). If this is not the case, test if the attribute value can be found by step E (7). If this is not the case, the value of the attribute 'user-visible name' is the empty catenation.

Definition 9.84

```

1   $\forall cst, doby, prof, rdoby$ 
2  ( $_0$  USER_VISIBLE_NAME_VALUE( $cst, doby, prof, rdoby$ ) =
3  IF IsAttributeSet2.62( $cst$ ) and 'user-visible name'  $\in$  NAMS1.18( $cst$ ) and
4  not IsPlaceholder1.19( $C \sim cst \bullet$  'user-visible name')
5  THEN  $C \sim cst \bullet$  'user-visible name'
6  ELSE STEP_C_VALUE9.6( $cst, doby, 'user-visible name'$ ) ORELSE9.1
7  ( $_1$  STEP_E_VALUE9.10( $cst, doby, rdoby, prof, 'user-visible name'$ )
8  ORELSE9.1 [ $\rightarrow$ ]  $_0$ )
```


Semiformal Description 9.89

Function “Value of the attribute 'character spacing'”

The value of the attribute 'character spacing' may be specified directly on a particular constituent (3-5). If this is not the case, test if the attribute value can be found by step BP (6). If this is not the case and the constituent is a basic layout object, test if the attribute value can be found by step C (7, 8). If this is not the case, test if the attribute value can be found by step DP (9). If this is not the case and the attribute is a basic layout object, test if the attribute value can be found by step E (10, 11). If this is not the case, test if the attribute value can be found by step FP (12). If this is not the case, test if the attribute value can be found by step G (13, 14). If this is not the case, test if the attribute value can be found by step H (15). If this is not the case, the value of the attribute 'character spacing' is the equivalent of 120 BMUs.

Definition 9.89

```

1   $\forall cst, doby, prof, rdoby$ 
2  ( $_0$  CHARACTER_SPACING_VALUE( $cst, doby, prof, rdoby$ ) =
3  IF IsAttributeSet2.62( $cst$ ) and 'character spacing'  $\in$  NAMS1.18( $cst$ ) and
4  not IsPlaceholder1.19( $C \hat{cst} \cdot$  'character spacing')
5  THEN  $C \hat{cst} \cdot$  'character spacing'
6  ELSE STEP_BP_VALUE9.3( $cst, doby, 'character spacing'$ ) ORELSE9.1
7  ( $_1$  IF IsBasicLayoutObjectDescription2.54( $cst$ )
8  THEN STEP_C_VALUE9.6( $cst, doby, 'character spacing'$ ) ORELSE9.1
9  ( $_2$  STEP_DP_VALUE9.8( $cst, doby, 'character spacing'$ ) ORELSE9.1
10 ( $_3$  IF IsBasicLayoutObjectDescription2.54( $cst$ )
11 THEN STEP_E_VALUE9.10( $cst, doby, rdoby, prof, 'character spacing'$ ) ORELSE9.1
12 ( $_4$  STEP_FP_VALUE9.13( $cst, doby, rdoby, prof, 'character spacing'$ ) ORELSE9.1
13 ( $_5$  STEP_G_VALUE9.17( $cst, doby, rdoby, prof, 'character spacing'$ ,
14 OBJECT_TYPE_VALUE9.61( $cst, doby$ ))
15 ORELSE9.1( $_6$  STEP_H_VALUE9.22( $prof, 'character spacing'$ )
16 ORELSE9.1  $120 \times$  BMU_EQUIVALENT9.90( $prof$ ) $_6$ ) $_5$ ) $_4$ ) $_3$ ) $_2$ ) $_1$ ) $_0$ )

```

Semiformal Description 9.90

Function “BMU equivalent”

If the attribute 'unit scaling' is specified in the document profile, the BMU equivalent is the second component of this attribute divided by the first component of this attribute. Otherwise, the BMU equivalent is 1.

Definition 9.90

```

1   $\forall prof$ 
2  ( $_0$  BMU_EQUIVALENT( $prof$ ) =
3  IF 'unit scaling'  $\in$  NAMS1.18( $prof$ )
4  THEN TAIL1.15( $prof \cdot$  'unit scaling')/HEAD1.13( $prof \cdot$  'unit scaling')
5  ELSE  $1$ ) $_0$ 

```

Semiformal Description 9.91

Function “Value of the attribute 'code extension announcers'”

The value of the attribute 'code extension announcers' may be specified directly on a particular constituent (3-5). If this is not the case, test if the attribute value can be found by step BP (6). If this is not the case and the constituent is a basic layout object, test if the attribute value can be found by step C (7, 8). If this is not the case, test if the attribute value can be found by step DP (9). If this is not the case and the attribute is a basic layout object, test if the attribute value can be found by step E (10, 11). If this is not the case, test if the attribute value can be found by step FP (12). If this is not the case, test if the attribute value can be found by step G (13, 14). If this is not the case, test if the attribute value can be found by step H (15). If this is not the case, the value of the attribute 'code extension announcers' are the escape sequences announcing the use of the G0 and G2 sets.

Definition 9.91

```

1   $\forall cst, doby, prof, rdoby$ 
2  (0 CODE_EXTENSION_ANNOUNCERS_VALUE(cst, doby, prof, rdoby) =
3  IF IsAttributeSet2.62(cst) and 'code extension announcers'  $\in$  NAMS1.18(cst) and
4  not IsPlaceholder1.19(C  $\wedge$  cst . 'code extension announcers')
5  THEN C  $\wedge$  cst . 'code extension announcers'
6  ELSE STEP_BP_VALUE9.3(cst, doby, 'code extension announcers') ORELSE9.1
7  (1 IF IsBasicLayoutObjectDescription2.54(cst)
8  THEN STEP_C_VALUE9.6(cst, doby, 'code extension announcers') ORELSE9.1
9  (2 STEP_DP_VALUE9.8(cst, doby, 'code extension announcers') ORELSE9.1
10 (3 IF IsBasicLayoutObjectDescription2.54(cst)
11 THEN STEP_E_VALUE9.10(cst, doby, rdoby, prof, 'code extension announcers') ORELSE9.1
12 (4 STEP_FP_VALUE9.13(cst, doby, rdoby, prof, 'code extension announcers') ORELSE9.1
13 (5 STEP_G_VALUE9.17(cst, doby, rdoby, prof, 'code extension announcers',
14 OBJECT_TYPE_VALUE9.61(cst, doby))
15 ORELSE9.1(6 STEP_H_VALUE9.22(prof, 'code extension announcers')
16 ORELSE9.1 G0_G2_ESCAPE_SEQUENCE_ANNOUNCER9.92(0)1)2)3)4)5)6)

```

Semiformal Description 9.92

Function “G0-G2 Escape Sequence Announcer”

The G0-G2 Escape Sequence Announcer is considered an atom-valued function since it is defined outside ISO 8613.

Definition 9.92

```

1   $\forall cst$ 
2  (0 IsAtom(G0_G2_ESCAPE_SEQUENCE_ANNOUNCER)0)

```

Semiformal Description 9.93

Function "Value of the attribute 'first line offset'"

The value of the attribute 'first line offset' may be specified directly on a particular constituent (3-5). If this is not the case, test if the attribute value can be found by step BP (6). If this is not the case and the constituent is a basic layout object, test if the attribute value can be found by step C (7, 8). If this is not the case, test if the attribute value can be found by step DP (9). If this is not the case and the attribute is a basic layout object, test if the attribute value can be found by step E (10, 11). If this is not the case, test if the attribute value can be found by step FP (12). If this is not the case, test if the attribute value can be found by step G (13, 14). If this is not the case, test if the attribute value can be found by step H (15). If this is not the case, the value of the attribute 'first line offset' is 0.

Definition 9.93

```

1   $\forall cst, doby, prof, rdoby$ 
2  (0 FIRST_LINE_OFFSET_VALUE(cst, doby, prof, rdoby) =
3  IF IsAttributeSet2.62(cst) and 'first line offset'  $\in$  NAMS1.18(cst) and
4  not IsPlaceholder1.19( $C \hat{cst} \bullet$  'first line offset')
5  THEN  $C \hat{cst} \bullet$  'first line offset'
6  ELSE STEP_BP_VALUE9.3(cst, doby, 'first line offset') ORELSE9.1
7  (1 IF IsBasicLayoutObjectDescription2.54(cst)
8  THEN STEP_C_VALUE9.6(cst, doby, 'first line offset') ORELSE9.1
9  (2 STEP_DP_VALUE9.8(cst, doby, 'first line offset') ORELSE9.1
10 (3 IF IsBasicLayoutObjectDescription2.54(cst)
11 THEN STEP_E_VALUE9.10(cst, doby, rdoby, prof, 'first line offset') ORELSE9.1
12 (4 STEP_FP_VALUE9.13(cst, doby, rdoby, prof, 'first line offset') ORELSE9.1
13 (5 STEP_G_VALUE9.17(cst, doby, rdoby, prof, 'first line offset',
14 OBJECT_TYPE_VALUE9.61(cst, doby))
15 ORELSE9.1(6 STEP_H_VALUE9.22(prof, 'first line offset')
16 ORELSE9.1(06)5)4)3)2)1)0)

```

Semiformal Description 9.94

Function "Value of the attribute 'graphic character sets'"

The value of the attribute 'graphic character sets' may be specified directly on a particular constituent (3-5). If this is not the case, test if the attribute value can be found by step BP (6). If this is not the case and the constituent is a basic layout object, test if the attribute value can be found by step C (7, 8). If this is not the case, test if the attribute value can be found by step DP (9). If this is not the case and the attribute is a basic layout object, test if the attribute value can be found by step E (10, 11). If this is not the case, test if the attribute value can be found by step FP (12). If this is not the case, test if the attribute value can be found by step G (13, 14). If this is not the case, test if the attribute value can be found by step H (15). If this is not the case, the value of the attribute 'graphic character sets' are the escape sequences for invoking character sets of ISO 6937-2 as G0 and G2 sets.

Definition 9.94

```

1  ∀ cst, doby, prof, rdoby
2  (₀ GRAPHIC_CHARACTER_SETS_VALUE(cst, doby, prof, rdoby) =
3    IF IsAttributeSet2.62(cst) and 'graphic character sets' ∈ NAMS1.18(cst) and
4      not IsPlaceholder1.19(C ~cst. 'graphic character sets')
5      THEN C ~cst. 'graphic character sets'
6      ELSE STEP_BP_VALUE9.3(cst, doby, 'graphic character sets') ORELSE9.1
7      (₁ IF IsBasicLayoutObjectDescription2.54(cst)
8        THEN STEP_C_VALUE9.6(cst, doby, 'graphic character sets') ORELSE9.1
9        (₂ STEP_DP_VALUE9.8(cst, doby, 'graphic character sets') ORELSE9.1
10       (₃ IF IsBasicLayoutObjectDescription2.54(cst)
11         THEN STEP_E_VALUE9.10(cst, doby, rdoby, prof, 'graphic character sets') ORELSE9.1
12         (₄ STEP_FP_VALUE9.13(cst, doby, rdoby, prof, 'graphic character sets') ORELSE9.1
13         (₅ STEP_G_VALUE9.17(cst, doby, rdoby, prof, 'graphic character sets',
14           OBJECT_TYPE2_VALUE9.61(cst, doby))
15         ORELSE9.1(₆ STEP_H_VALUE9.22(prof, 'graphic character sets')
16         ORELSE9.1 G0_G2_ESCAPE_SEQUENCE9.95(₆)₅)₄)₃)₂)₁)₀)

```

Semiformal Description 9.95

Function "G0-G2 Escape Sequence"

The G0-G2 Escape Sequence is considered an atom-valued function since it is defined outside ISO 8613.

Definition 9.95

```

1  ∀ cst
2  (₀ IsAtom(G0_G2_ESCAPE_SEQUENCE)₀)

```


Semiformal Description 9.97

Function “Value of the attribute 'graphic rendition'”

The value of the attribute 'graphic rendition' may be specified directly on a particular constituent (3-5). If this is not the case, test if the attribute value can be found by step BP (6). If this is not the case and the constituent is a basic layout object, test if the attribute value can be found by step C (7, 8). If this is not the case, test if the attribute value can be found by step DP (9). If this is not the case and the attribute is a basic layout object, test if the attribute value can be found by step E (10, 11). If this is not the case, test if the attribute value can be found by step FP (12). If this is not the case, test if the attribute value can be found by step G (13, 14). If this is not the case, test if the attribute value can be found by step H (15). If this is not the case, the value of the attribute 'graphic rendition' is 0.

Definition 9.97

```

1   $\forall cst, doby, prof, rdoby$ 
2  (0 GRAPHIC_RENDITION_VALUE( $cst, doby, prof, rdoby$ ) =
3  IF IsAttributeSet2.62( $cst$ ) and 'graphic rendition'  $\in$  NAMS1.18( $cst$ ) and
4  not IsPlaceholder1.19( $C \hat{cst} \cdot$  'graphic rendition')
5  THEN  $C \hat{cst} \cdot$  'graphic rendition'
6  ELSE STEP_BP_VALUE9.3( $cst, doby, 'graphic rendition'$ ) ORELSE9.1
7  (1 IF IsBasicLayoutObjectDescription2.54( $cst$ )
8  THEN STEP_C_VALUE9.6( $cst, doby, 'graphic rendition'$ ) ORELSE9.1
9  (2 STEP_DP_VALUE9.8( $cst, doby, 'graphic rendition'$ ) ORELSE9.1
10 (3 IF IsBasicLayoutObjectDescription2.54( $cst$ )
11 THEN STEP_E_VALUE9.10( $cst, doby, rdoby, prof, 'graphic rendition'$ ) ORELSE9.1
12 (4 STEP_FP_VALUE9.13( $cst, doby, rdoby, prof, 'graphic rendition'$ ) ORELSE9.1
13 (5 STEP_G_VALUE9.17( $cst, doby, rdoby, prof, 'graphic rendition',$ 
14 OBJECT_TYPE_VALUE9.61( $cst, doby$ ))
15 ORELSE9.1(6 STEP_H_VALUE9.22( $prof, 'graphic rendition'$ )
16 ORELSE9.1 (06)5)4)3)2)1)0)

```

IECNORM.COM : Click to view the full text of ISO/IEC 8613-10:1991/Amd 5:1993

Semiformal Description 9.98

Function "Value of the attribute 'itemization'"

The value of the attribute 'itemization' may be specified directly on a particular constituent (3-5). If this is not the case, test if the attribute value can be found by step BP (6). If this is not the case and the constituent is a basic layout object, test if the attribute value can be found by step C (7, 8). If this is not the case, test if the attribute value can be found by step DP (9). If this is not the case and the attribute is a basic layout object, test if the attribute value can be found by step E (10, 11). If this is not the case, test if the attribute value can be found by step FP (12). If this is not the case, test if the attribute value can be found by step G (13). If this is not the case, test if the attribute value can be found by step H (14). If this is not the case, the value of the attribute 'itemization' comprises the parameter 'identifier alignment' with the value 'no itemization' and the parameters 'identifier start offset' and 'identifier end offset' with the value 0.

Definition 9.98

```

1   $\forall cst, doby, prof, rdoby$ 
2  (0ITEMIZATION_VALUE(cst, doby, prof, rdoby) =
3  IF IsAttributeSet2.62(cst) and 'itemization'  $\in$  NAMS1.18(cst) and
4  not IsPlaceholder1.19(C  $\neg$  cst . 'itemization')
5  THEN C  $\neg$  cst . 'itemization'
6  ELSE STEP_BP_VALUE9.3(cst, doby, 'itemization') ORELSE9.1
7  (1 IF IsBasicLayoutObjectDescription2.54(cst)
8  THEN STEP_C_VALUE9.6(cst, doby, 'itemization') ORELSE9.1
9  (2 STEP_DP_VALUE9.8(cst, doby, 'itemization') ORELSE9.1
10 (3 IF IsBasicLayoutObjectDescription2.54(cst)
11 THEN STEP_E_VALUE9.10(cst, doby, rdoby, prof, 'itemization') ORELSE9.1
12 (4 STEP_FP_VALUE9.13(cst, doby, rdoby, prof, 'itemization') ORELSE9.1
13 (5 STEP_G_VALUE9.17(cst, doby, rdoby, prof, 'itemization', OBJECT_TYPE_VALUE9.61(cst, doby))
14 ORELSE9.1(6 STEP_H_VALUE9.22(prof, 'itemization')
15 ORELSE9.1 ['identifier alignment' : 'no itemization';
16 'identifier start offset' : 0;
17 'identifier end offset' : 0])6)5)4)3)2)1)0)

```


Semiformal Description 9.102

Function "Value of the attribute 'line spacing'"

The value of the attribute 'line spacing' may be specified directly on a particular constituent (3-5). If this is not the case, test if the attribute value can be found by step BP (6). If this is not the case and the constituent is a basic layout object, test if the attribute value can be found by step C (7, 8). If this is not the case, test if the attribute value can be found by step DP (9). If this is not the case and the attribute is a basic layout object, test if the attribute value can be found by step E (10, 11). If this is not the case, test if the attribute value can be found by step FP (12). If this is not the case, test if the attribute value can be found by step G (13). If this is not the case, test if the attribute value can be found by step H (14). If this is not the case, the value of the attribute 'line spacing' is the equivalent of 200 BMUs.

Definition 9.102

```

1   $\forall cst, doby, prof, rdoby$ 
2  (0 LINE_SPACING_VALUE(cst, doby, prof, rdoby) =
3  IF IsAttributeSet2.62(cst) and 'line spacing'  $\in$  NAMS1.18(cst) and
4  not IsPlaceholder1.19( $C \sim cst \cdot$  'line spacing')
5  THEN  $C \sim cst \cdot$  'line spacing'
6  ELSE STEP_BP_VALUE9.3(cst, doby, 'line spacing') ORELSE9.1
7  (1 IF IsBasicLayoutObjectDescription2.54(cst)
8  THEN STEP_C_VALUE9.6(cst, doby, 'line spacing') ORELSE9.1
9  (2 STEP_DP_VALUE9.8(cst, doby, 'line spacing') ORELSE9.1
10 (3 IF IsBasicLayoutObjectDescription2.54(cst)
11 THEN STEP_E_VALUE9.10(cst, doby, rdoby, prof, 'line spacing') ORELSE9.1
12 (4 STEP_FP_VALUE9.13(cst, doby, rdoby, prof, 'line spacing') ORELSE9.1
13 (5 STEP_G_VALUE9.17(cst, doby, rdoby, prof, 'line spacing', OBJECT_TYPE_VALUE9.61(cst, doby))
14 ORELSE9.1(6 STEP_H_VALUE9.22(prof, 'line spacing')
15 ORELSE9.1 200  $\times$  BMU_EQUIVALENT9.90(prof)5)4)3)2)1)0)

```

Semiformal Description 9.103

Function “Value of the attribute 'pairwise kerning'”

The value of the attribute 'pairwise kerning' may be specified directly on a particular constituent (3-5). If this is not the case, test if the attribute value can be found by step BP (6). If this is not the case and the constituent is a basic layout object, test if the attribute value can be found by step C (7, 8). If this is not the case, test if the attribute value can be found by step DP (9). If this is not the case and the attribute is a basic layout object, test if the attribute value can be found by step E (10, 11). If this is not the case, test if the attribute value can be found by step FP (12). If this is not the case, test if the attribute value can be found by step G (13, 14). If this is not the case, test if the attribute value can be found by step H (15). If this is not the case, the value of the attribute 'pairwise kerning' is 'no'.

Definition 9.103

```

1   $\forall cst, doby, prof, rdoby$ 
2  (0 PAIRWISE_KERNING_VALUE(cst, doby, prof, rdoby) =
3  IF IsAttributeSet2.62(cst) and 'pairwise kerning'  $\in$  NAMS1.18(cst) and
4  not IsPlaceholder1.19( $C \sim cst \cdot$  'pairwise kerning')
5  THEN  $C \sim cst \cdot$  'pairwise kerning'
6  ELSE STEP_BP_VALUE9.3(cst, doby, 'pairwise kerning') ORELSE9.1
7  (1 IF IsBasicLayoutObjectDescription2.54(cst)
8  THEN STEP_C_VALUE9.6(cst, doby, 'pairwise kerning') ORELSE9.1
9  (2 STEP_DP_VALUE9.8(cst, doby, 'pairwise kerning') ORELSE9.1
10 (3 IF IsBasicLayoutObjectDescription2.54(cst)
11 THEN STEP_E_VALUE9.10(cst, doby, rdoby, prof, 'pairwise kerning') ORELSE9.1
12 (4 STEP_FP_VALUE9.13(cst, doby, rdoby, prof, 'pairwise kerning') ORELSE9.1
13 (5 STEP_G_VALUE9.17(cst, doby, rdoby, prof, 'pairwise kerning',
14 OBJECT_TYPE_VALUE9.61(cst, doby))
15 ORELSE9.1(6 STEP_H_VALUE9.22(prof, 'pairwise kerning')
16 ORELSE9.1 'no' c)5)4)3)2)1)0)

```

Semiformal Description 9.104

Function "Value of the attribute 'formatting indicator'"

The value of the attribute 'formatting indicator' may be specified directly on a particular constituent (3-5). If this is not the case, test if the attribute value can be found by step BP (6). If this is not the case and the constituent is a basic layout object, test if the attribute value can be found by step C (7, 8). If this is not the case, test if the attribute value can be found by step DP (9). If this is not the case and the attribute is a basic layout object, test if the attribute value can be found by step E (10, 11). If this is not the case, test if the attribute value can be found by step FP (12). If this is not the case, test if the attribute value can be found by step G (13, 14). If this is not the case, test if the attribute value can be found by step H (15). If this is not the case, the value of the attribute 'formatting indicator' is 'no'.

Definition 9.104

```

1   $\forall cst, doby, prof, rdoby$ 
2  (0 FORMATTING_INDICATOR_VALUE(cst, doby, prof, rdoby) =
3  IF IsAttributeSet2.62(cst) and 'formatting indicator'  $\in$  NAMS1.18(cst) and
4  not IsPlaceholder1.19(C  $\sim$  cst . 'formatting indicator')
5  THEN C  $\sim$  cst . 'formatting indicator'
6  ELSE STEP_BP_VALUE9.3(cst, doby, 'formatting indicator') ORELSE9.1
7  (1 IF IsBasicLayoutObjectDescription2.54(cst)
8  THEN STEP_C_VALUE9.6(cst, doby, 'formatting indicator') ORELSE9.1
9  (2 STEP_DP_VALUE9.8(cst, doby, 'formatting indicator') ORELSE9.1
10 (3 IF IsBasicLayoutObjectDescription2.54(cst)
11 THEN STEP_E_VALUE9.10(cst, doby, rdoby, prof, 'formatting indicator') ORELSE9.1
12 (4 STEP_FP_VALUE9.13(cst, doby, rdoby, prof, 'formatting indicator') ORELSE9.1
13 (5 STEP_G_VALUE9.17(cst, doby, rdoby, prof, 'formatting indicator',
14 OBJECT_TYPE_VALUE9.61(cst, doby))
15 ORELSE9.1(6 STEP_H_VALUE9.22(prof, 'formatting indicator')
16 ORELSE9.1 'no' 7) 8) 9) 10) 11) 12) 13) 14) 15) 16)

```

Semiformal Description 9.105

Function "Value of the attribute 'initial offset' (for the character content architecture)"

The value of the attribute 'initial offset' (for the character content architecture) may be specified directly on a particular constituent (3-5). If this is not the case, test if the attribute value can be found by step BP (6). If this is not the case and the constituent is a basic layout object, test if the attribute value can be found by step C (7, 8). If this is not the case, test if the attribute value can be found by step DP (9). If this is not the case and the attribute is a basic layout object, test if the attribute value can be found by step E (10, 11). If this is not the case, test if the attribute value can be found by step FP (12). If this is not the case, test if the attribute value can be found by step G (13). If this is not the case, test if the attribute value can be found by step H (14). If this is not the case, the value of the attribute 'initial offset' is the initial offset default value (for character content).

Definition 9.105

```

1   $\forall cst, doby, prof, rdoby$ 
2  (0 CCA_INITIAL_OFFSET_VALUE(cst, doby, prof, rdoby) =
3   IF IsAttributeSet2.62(cst) and 'initial offset'  $\in$  NAMS1.18(cst) and
4   not IsPlaceholder1.19( $C \sim cst \cdot$  'initial offset')
5   THEN  $C \sim cst \cdot$  'initial offset'
6   ELSE STEP_BP_VALUE9.3(cst, doby, 'initial offset') ORELSE9.1
7   (1 IF IsBasicLayoutObjectDescription2.54(cst)
8   THEN STEP_C_VALUE9.6(cst, doby, 'initial offset') ORELSE9.1
9   (2 STEP_DP_VALUE9.8(cst, doby, 'initial offset') ORELSE9.1
10  (3 IF IsBasicLayoutObjectDescription2.54(cst)
11  THEN STEP_E_VALUE9.10(cst, doby, rdoby, prof, 'initial offset') ORELSE9.1
12  (4 STEP_FP_VALUE9.13(cst, doby, rdoby, prof, 'initial offset') ORELSE9.1
13  (5 STEP_G_VALUE9.17(cst, doby, rdoby, prof, 'initial offset', OBJECT_TYPE_VALUE9.61(cst, doby))
14  ORELSE9.1(6 STEP_H_VALUE9.22(prof, 'initial offset')
15  ORELSE9.1 CCA_INITIAL_OFFSET_DEFAULT_VALUE9.106(cst, doby, prof, rdoby)6)5)4)3)2)1)0)

```

IECNORM.COM : Click to view the full PDF of ISO/IEC 8613-10:1991/Amd.5:1993

Semiformal Description 9.107

Function “Horizontal dimension (of a basic object)”

The horizontal dimension (of a basic object) is the value of the parameter 'horizontal dimension' of the dimensions value for the object *cst*.

Definition 9.107

- 1 $\forall cst, doby, prof, rdoby$
- 2 $(\text{HORIZONTAL_DIMENSION}(cst, doby, prof, rdoby) =$
- 3 $\text{C} \wedge \text{DIMENSIONS_VALUE}^{9.45}(cst, doby, prof, rdoby) \bullet \text{'horizontal dimension' } \text{,})$

Semiformal Description 9.108

Function “Vertical dimension (of a basic object)”

The vertical dimension (of a basic object) is the value of the parameter 'vertical dimension' of the dimensions value for the object *cst*.

Definition 9.108

- 1 $\forall cst, doby, prof, rdoby$
- 2 $(\text{VERTICAL_DIMENSION}(cst, doby, prof, rdoby) =$
- 3 $\text{C} \wedge \text{DIMENSIONS_VALUE}^{9.45}(cst, doby, prof, rdoby) \bullet \text{'vertical dimension' } \text{,})$

Semiformal Description 9.109

Function “First line box backward extent”

The backward extent of the first line box is determined when formatting the content portion. It is considered an atom-valued function here.

Definition 9.109

- 1 $\forall cst$
- 2 $(\text{IsAtom}(\text{FIRST_LINE_BOX_BACKWARD_EXTENT}(cst)), \text{,})$

Semiformal Description 9.111

Function "Value of the attribute 'orphan size'"

The value of the attribute 'orphan size' may be specified directly on a particular constituent (3-5). If this is not the case, test if the attribute value can be found by step BP (6). If this is not the case and the constituent is a basic layout object, test if the attribute value can be found by step C (7, 8). If this is not the case, test if the attribute value can be found by step DP (9). If this is not the case and the attribute is a basic layout object, test if the attribute value can be found by step E (10, 11). If this is not the case, test if the attribute value can be found by step FP (12). If this is not the case, test if the attribute value can be found by step G (13). If this is not the case, test if the attribute value can be found by step H (14). If this is not the case, the value of the attribute 'orphan size' is 1.

Definition 9.111

```

1   $\forall cst, doby, prof, rdoby$ 
2  (  $\text{ORPHAN\_SIZE\_VALUE}(cst, doby, prof, rdoby) =$ 
3    IF  $\text{IsAttributeSet}^{2.62}(cst)$  and 'orphan size'  $\in \text{NAMS}^{1.18}(cst)$  and
4      not  $\text{IsPlaceholder}^{1.19}(C \sim cst \cdot \text{'orphan size'})$ 
5      THEN  $C \sim cst \cdot \text{'orphan size'}$ 
6      ELSE  $\text{STEP\_BP\_VALUE}^{9.3}(cst, doby, \text{'orphan size'})$  ORELSE9.1
7      ( $\text{IF } \text{IsBasicLayoutObjectDescription}^{2.54}(cst)$ 
8        THEN  $\text{STEP\_C\_VALUE}^{9.6}(cst, doby, \text{'orphan size'})$  ORELSE9.1
9        ( $\text{STEP\_DP\_VALUE}^{9.8}(cst, doby, \text{'orphan size'})$  ORELSE9.1
10         ( $\text{IF } \text{IsBasicLayoutObjectDescription}^{2.54}(cst)$ 
11           THEN  $\text{STEP\_E\_VALUE}^{9.10}(cst, doby, rdoby, prof, \text{'orphan size'})$  ORELSE9.1
12           ( $\text{STEP\_FP\_VALUE}^{9.13}(cst, doby, rdoby, prof, \text{'orphan size'})$  ORELSE9.1
13             ( $\text{STEP\_G\_VALUE}^{9.17}(cst, doby, rdoby, prof, \text{'orphan size'}, \text{OBJECT\_TYPE\_VALUE}^{9.61}(cst, doby))$ 
14             ORELSE9.1 ( $\text{STEP\_H\_VALUE}^{9.22}(prof, \text{'orphan size'})$ 
15             ORELSE9.1 1)))))

```

IECNORM.COM : Click to view the full PDF of ISO/IEC 8613-10:1991/Amd.5:1993

Semiformal Description 9.112

Function "Value of the attribute 'proportional line spacing'"

The value of the attribute 'proportional line spacing' may be specified directly on a particular constituent (3-5). If this is not the case, test if the attribute value can be found by step BP (6). If this is not the case and the constituent is a basic layout object, test if the attribute value can be found by step C (7, 8). If this is not the case, test if the attribute value can be found by step DP (9). If this is not the case and the attribute is a basic layout object, test if the attribute value can be found by step E (10, 11). If this is not the case, test if the attribute value can be found by step FP (12). If this is not the case, test if the attribute value can be found by step G (13, 14). If this is not the case, test if the attribute value can be found by step H (15). If this is not the case, the value of the attribute 'proportional line spacing' is 'no'.

Definition 9.112

```

1   $\forall cst, doby, prof, rdoby$ 
2  (0PROPORTIONAL_LINE_SPACING_VALUE(cst, doby, prof, rdoby) =
3  IF IsAttributeSet2.62(cst) and 'proportional line spacing'  $\in$  NAMS1.18(cst) and
4  not IsPlaceholder1.19(C  $\wedge$  cst • 'proportional line spacing')
5  THEN C  $\wedge$  cst • 'proportional line spacing'
6  ELSE STEP_BP_VALUE9.3(cst, doby, 'proportional line spacing') ORELSE9.1
7  (1IF IsBasicLayoutObjectDescription2.54(cst)
8  THEN STEP_C_VALUE9.6(cst, doby, 'proportional line spacing') ORELSE9.1
9  (2STEP_DP_VALUE9.8(cst, doby, 'proportional line spacing') ORELSE9.1
10 (3IF IsBasicLayoutObjectDescription2.54(cst)
11 THEN STEP_E_VALUE9.10(cst, doby, rdoby, prof, 'proportional line spacing') ORELSE9.1
12 (4STEP_FP_VALUE9.13(cst, doby, rdoby, prof, 'proportional line spacing') ORELSE9.1
13 (5STEP_G_VALUE9.17(cst, doby, rdoby, prof, 'proportional line spacing',
14 OBJECT_TYPE_VALUE9.61(cst, doby))
15 ORELSE9.1(6STEP_H_VALUE9.22(prof, 'proportional line spacing')
16 ORELSE9.1 'no' )5)4)3)2)1)0)

```

Semiformal Description 9.113

Function "Value of the attribute 'widow size'"

The value of the attribute 'widow size' may be specified directly on a particular constituent (3-5). If this is not the case, test if the attribute value can be found by step BP (6). If this is not the case and the constituent is a basic layout object, test if the attribute value can be found by step C (7, 8). If this is not the case, test if the attribute value can be found by step DP (9). If this is not the case and the attribute is a basic layout object, test if the attribute value can be found by step E (10, 11). If this is not the case, test if the attribute value can be found by step FP (12). If this is not the case, test if the attribute value can be found by step G (13). If this is not the case, test if the attribute value can be found by step H (14). If this is not the case, the value of the attribute 'widow size' is 1.

Definition 9.113

```

1   $\forall cst, doby, prof, rdoby$ 
2  (0 WIDOW_SIZE_VALUE(cst, doby, prof, rdoby) =
3    IF IsAttributeSet2.62(cst) and 'widow size'  $\in$  NAMS1.18(cst) and
4      not IsPlaceholder1.19( $C \sim cst \cdot$  'widow size')
5    THEN  $C \sim cst \cdot$  'widow size'
6    ELSE STEP_BP_VALUE9.3(cst, doby, 'widow size') ORELSE9.1
7    (1 IF IsBasicLayoutObjectDescription2.54(cst)
8      THEN STEP_C_VALUE9.6(cst, doby, 'widow size') ORELSE9.1
9      (2 STEP_DP_VALUE9.8(cst, doby, 'widow size') ORELSE9.1
10     (3 IF IsBasicLayoutObjectDescription2.54(cst)
11       THEN STEP_E_VALUE9.10(cst, doby, rdoby, prof, 'widow size') ORELSE9.1
12       (4 STEP_FP_VALUE9.13(cst, doby, rdoby, prof, 'widow size') ORELSE9.1
13       (5 STEP_G_VALUE9.17(cst, doby, rdoby, prof, 'widow size', OBJECT_TYPE_VALUE9.61(cst, doby))
14       ORELSE9.1(6 STEP_H_VALUE9.22(prof, 'widow size')
15       ORELSE9.1 1(6)5)4)3)2)1)0)

```

IECNORM.COM : Click to view the full PDF of ISO/IEC 8613-10:1991/Amd 5:1993

F.3.3 Defaultable attributes of ISO 8613-7

Semiformal Description 9.114

Function "Value of the attribute 'clipping'"

The value of the attribute 'clipping' may be specified directly on a particular constituent (3-5). If this is not the case, test if the attribute value can be found by step BP (6). If this is not the case and the constituent is a basic layout object, test if the attribute value can be found by step C (7, 8). If this is not the case, test if the attribute value can be found by step DP (9). If this is not the case and the attribute is a basic layout object, test if the attribute value can be found by step E (10, 11). If this is not the case, test if the attribute value can be found by step FP (12). If this is not the case, test if the attribute value can be found by step G (13). If this is not the case, test if the attribute value can be found by step H (14). If this is not the case, the value of the attribute 'clipping' is a sequence of two integer pairs, where the first pair consists of two zeros and the second pair consists of the number of pels per line minus one and the number of lines minus one of that content portion *cont* which is associated with the constituent.

Definition 9.114

```

1   $\forall cst, doby, prof, rdoby$ 
2  (0 CLIPPING_VALUE(cst, doby, prof, rdoby) =
3  IF IsAttributeSet2.62(cst) and 'clipping'  $\in$  NAMS1.18(cst) and
4  not IsPlaceholder1.19( $C \sim cst \cdot$  'clipping')
5  THEN  $C \sim cst \cdot$  'clipping'
6  ELSE STEP_BP_VALUE9.3(cst, doby, 'clipping') ORELSE9.1
7  (1 IF IsBasicLayoutObjectDescription2.54(cst)
8  THEN STEP_C_VALUE9.6(cst, doby, 'clipping') ORELSE9.1
9  (2 STEP_DP_VALUE9.8(cst, doby, 'clipping') ORELSE9.1
10 (3 IF IsBasicLayoutObjectDescription2.54(cst)
11 THEN STEP_E_VALUE9.10(cst, doby, rdoby, prof, 'clipping') ORELSE9.1
12 (4 STEP_FP_VALUE9.13(cst, doby, rdoby, prof, 'clipping') ORELSE9.1
13 (5 STEP_G_VALUE9.17(cst, doby, rdoby, prof, 'clipping', OBJECT_TYPE_VALUE9.61(cst, doby))
14 ORELSE9.1 (6 STEP_H_VALUE9.22(prof, 'clipping')
15 ORELSE9.1 [ $\rightarrow$  [ $\rightarrow$  0  $\rightarrow$  0  $\rightarrow$ ]  $\rightarrow$ 
16 [ $\rightarrow$  (NUMBER_OF_PELS_PER_LINE_VALUE9.124(THAT cont
17 (7 (cont)DescribesContPortOf2.153(cst)7), doby, prof, rdoby) - 1)  $\rightarrow$ 
18 ( $C \sim$  (THAT cont
19 (8 (cont)DescribesContPortOf2.153(cst)8)  $\cdot$  'number of lines' - 1)  $\rightarrow$ ]  $\rightarrow$ ]
20 (0), (1), (2), (3), (4), (5), (6), (7), (8))

```

Semiformal Description 9.115

Function "Value of the attribute 'line progression' (for the raster graphics content architecture)"

The value of the attribute 'line progression' (for the raster graphics content architecture) may be specified directly on a particular constituent (3-5). If this is not the case, test if the attribute value can be found by step BP (6). If this is not the case and the constituent is a basic layout object, test if the attribute value can be found by step C (7, 8). If this is not the case, test if the attribute value can be found by step DP (9). If this is not the case and the attribute is a basic layout object, test if the attribute value can be found by step E (10, 11). If this is not the case, test if the attribute value can be found by step FP (12). If this is not the case, test if the attribute value can be found by step G (13, 14). If this is not the case, test if the attribute value can be found by step H (15). If this is not the case, the value of the attribute 'line progression' is 270°.

Definition 9.115

```

1  ∀ cst, doby, prof, rdoby
2  (RGCA_LINE_PROGRESSION_VALUE(cst, doby, prof, rdoby) =
3    IF IsAttributeSet2.62(cst) and 'line progression' ∈ NAMS1.18(cst) and
4      not IsPlaceholder1.19(C ^ cst . 'line progression')
5    THEN C ^ cst . 'line progression'
6    ELSE STEP_BP_VALUE9.3(cst, doby, 'line progression') ORELSE9.1
7      (1 IF IsBasicLayoutObjectDescription2.54(cst)
8        THEN STEP_C_VALUE9.6(cst, doby, 'line progression') ORELSE9.1
9          (2 STEP_DP_VALUE9.8(cst, doby, 'line progression') ORELSE9.1
10             (3 IF IsBasicLayoutObjectDescription2.54(cst)
11                THEN STEP_E_VALUE9.10(cst, doby, rdoby, prof, 'line progression') ORELSE9.1
12                  (4 STEP_FP_VALUE9.13(cst, doby, rdoby, prof, 'line progression') ORELSE9.1
13                     (5 STEP_G_VALUE9.17(cst, doby, rdoby, prof, 'line progression',
14                        OBJECT_TYPE_VALUE9.61(cst, doby))
15                      ORELSE9.1(6 STEP_H_VALUE9.22(prof, 'line progression')
16                        ORELSE9.1 '270°')))))))

```


Semiformal Description 9.120

Function "Value of the attribute 'image dimensions'"

The value of the attribute 'image dimensions' may be specified directly on a particular constituent (3-5). If this is not the case, test if the attribute value can be found by step BP (6). If this is not the case and the constituent is a basic layout object, test if the attribute value can be found by step C (7, 8). If this is not the case, test if the attribute value can be found by step DP (9). If this is not the case and the attribute is a basic layout object, test if the attribute value can be found by step E (10, 11). If this is not the case, test if the attribute value can be found by step FP (12). If this is not the case, test if the attribute value can be found by step G (13, 14). If this is not the case, test if the attribute value can be found by step H (15). If this is not the case, the value of the attribute 'image dimensions' is 'automatic'.

Definition 9.120

```

1   $\forall cst, doby, prof, rdoby$ 
2  (0 IMAGE_DIMENSIONS_VALUE(cst, doby, prof, rdoby) =
3  IF IsAttributeSet2.62(cst) and 'image dimensions'  $\in$  NAMS1.18(cst) and
4  not IsPlaceholder1.19( $C \hat{cst} \cdot$  'image dimensions')
5  THEN  $C \hat{cst} \cdot$  'image dimensions'
6  ELSE STEP_BP_VALUE9.3(cst, doby, 'image dimensions') ORELSE9.1
7  (1 IF IsBasicLayoutObjectDescription2.54(cst)
8  THEN STEP_C_VALUE9.6(cst, doby, 'image dimensions') ORELSE9.1
9  (2 STEP_DP_VALUE9.8(cst, doby, 'image dimensions') ORELSE9.1
10 (3 IF IsBasicLayoutObjectDescription2.54(cst)
11 THEN STEP_E_VALUE9.10(cst, doby, rdoby, prof, 'image dimensions') ORELSE9.1
12 (4 STEP_FP_VALUE9.13(cst, doby, rdoby, prof, 'image dimensions') ORELSE9.1
13 (5 STEP_G_VALUE9.17(cst, doby, rdoby, prof, 'image dimensions',
14 OBJECT_TYPE_VALUE9.61(cst, doby))
15 ORELSE9.1(6 STEP_H_VALUE9.22(prof, 'image dimensions')
16 ORELSE9.1 'automatic' 6)5)4)3)2)1)0)

```


Semiformal Description 9.124

Function "Value of the attribute 'number of pels per line'"

The value of the attribute 'number of pels per line' may be specified directly on a particular constituent (3-5). If this is not the case, test if the attribute value can be found by step C (6). If this is not the case, test if the attribute value can be found by step H (7). If this is not the case, the value of the attribute 'number of pels per line' depends on the value of the attribute 'pel transmission density' as indicated.

Definition 9.124

```

1   $\forall cst, doby, prof, rdoby$ 
2  (0 NUMBER_OF_PELS_PER_LINE_VALUE(cst, doby, prof, rdoby) =
3  IF IsAttributeSet2.62(cst) and 'number of pels per line'  $\in$  NAMS1.18(cst) and
4  not IsPlaceholder1.19( $C \hat{cst} \cdot$  'number of pels per line')
5  THEN  $C \hat{cst} \cdot$  'number of pels per line'
6  ELSE STEP_C_VALUE9.6(cst, doby, 'number of pels per line') ORELSE9.1
7  (1 STEP_H_VALUE9.22(prof, 'number of pels per line') ORELSE9.1
8  (2 IF PEL_TRANSMISSION_DENSITY_VALUE9.119(cst, doby, prof, rdoby) = '1 BMU'
9  THEN 10368 ELSE
10 (3 IF PEL_TRANSMISSION_DENSITY_VALUE9.119(cst, doby, prof, rdoby) = '2 BMU'
11 THEN 5184 ELSE
12 (4 IF PEL_TRANSMISSION_DENSITY_VALUE9.119(cst, doby, prof, rdoby) = '3 BMU'
13 THEN 3456 ELSE
14 (5 IF PEL_TRANSMISSION_DENSITY_VALUE9.119(cst, doby, prof, rdoby) = '4 BMU'
15 THEN 2592 ELSE
16 (6 IF PEL_TRANSMISSION_DENSITY_VALUE9.119(cst, doby, prof, rdoby) = '5 BMU'
17 THEN 2074 ELSE
18 (7 IF PEL_TRANSMISSION_DENSITY_VALUE9.119(cst, doby, prof, rdoby) = '6 BMU'
19 THEN 17287)6)5)4)3)2)1)0)

```

Semiformal Description 9.125

Function "Value of the attribute 'number of discarded pels'"

The value of the attribute 'number of discarded pels' may be specified directly on a particular constituent (3-5). If this is not the case, test if the attribute value can be found by step C (6). If this is not the case, test if the attribute value can be found by step H (7). If this is not the case, the value of the attribute 'number of discarded pels' is the number of pels per line minus the maximum number of pels in the line divided by 2, if the number of pels per line minus exceeds the maximum number of pels in the line, otherwise 0.

Definition 9.125

```

1   $\forall cst, doby, prof, rdoby$ 
2  (0 NUMBER_OF_DISCARDED_PELS_VALUE(cst, doby, prof, rdoby) =
3  IF IsAttributeSet2.62(cst) and 'number of discarded pels'  $\in$  NAMS1.18(cst) and
4  not IsPlaceholder1.19( $C \hat{cst} \cdot$  'number of discarded pels')
5  THEN  $C \hat{cst} \cdot$  'number of discarded pels'
6  ELSE STEP_C_VALUE9.6(cst, doby, 'number of discarded pels') ORELSE9.1
7  (1 IF MAX_PELS_PER_LINE9.126(cst, doby, prof, rdoby) <
8  NUMBER_OF_PELS_PER_LINE_VALUE9.124(cst, doby, prof, rdoby)
9  THEN (NUMBER_OF_PELS_PER_LINE_VALUE9.124(cst, doby, prof, rdoby) -
10  MAX_PELS_PER_LINE9.126(cst, doby, prof, rdoby))/2
11  ELSE 0,)

```

Semiformal Description 9.126

Function “maximum number of pels per line”

The maximum number of pels per line depends on the size of the block in direction of the pel path and is considered an atomic function for the formal specification.

Definition 9.126

```
1   $\forall cst, doby, prof, rdoby$ 
2  ( $\text{IsAtom}(\text{MAX\_PELS\_PER\_LINE}(cst, doby, prof, rdoby))_0$ )
```

F.3.4 Defaultable attributes of ISO 8613-8

Semiformal Description 9.127

Function “Value of the attribute 'region of interest specification'”

The value of the attribute 'region of interest specification' may be specified directly on a particular constituent (3-5). If this is not the case, test if the attribute value can be found by step BP (6). If this is not the case and the constituent is a basic layout object, test if the attribute value can be found by step C (7, 8). If this is not the case, test if the attribute value can be found by step DP (9). If this is not the case and the attribute is a basic layout object, test if the attribute value can be found by step E (10, 11). If this is not the case, test if the attribute value can be found by step FP (12). If this is not the case, test if the attribute value can be found by step G (13, 14). If this is not the case, test if the attribute value can be found by step H (15). If this is not the case, the value of the attribute 'region of interest specification' is the parameter 'automatic' with the value 'null'.

Definition 9.127

```
1   $\forall cst, doby, prof, rdoby$ 
2  ( $\text{REGION\_OF\_INTEREST\_SPECIFICATION\_VALUE}(cst, doby, prof, rdoby) =$ 
3  IF  $\text{IsAttributeSet}^{2.62}(cst)$  and 'region of interest specification'  $\in \text{NAMS}^{1.18}(cst)$  and
4  not  $\text{IsPlaceholder}^{1.19}(C \sim cst \cdot \text{'region of interest specification'})$ 
5  THEN  $C \sim cst \cdot \text{'region of interest specification'}$ 
6  ELSE  $\text{STEP\_BP\_VALUE}^{9.3}(cst, doby, \text{'region of interest specification'})$  ORELSE9.1
7  ( $\text{IF } \text{IsBasicLayoutObjectDescription}^{2.54}(cst)$ 
8  THEN  $\text{STEP\_C\_VALUE}^{9.6}(cst, doby, \text{'region of interest specification'})$  ORELSE9.1
9  ( $\text{STEP\_DP\_VALUE}^{9.8}(cst, doby, \text{'region of interest specification'})$  ORELSE9.1
10  ( $\text{IF } \text{IsBasicLayoutObjectDescription}^{2.54}(cst)$ 
11  THEN  $\text{STEP\_E\_VALUE}^{9.10}(cst, doby, rdoby, prof, \text{'region of interest specification'})$  ORELSE9.1
12  ( $\text{STEP\_FP\_VALUE}^{9.13}(cst, doby, rdoby, prof, \text{'region of interest specification'})$  ORELSE9.1
13  ( $\text{STEP\_G\_VALUE}^{9.17}(cst, doby, rdoby, prof, \text{'region of interest specification'}),$ 
14  OBJECT_TYPE_VALUE9.61( $cst, doby$ ))
15  ORELSE9.1( $\text{STEP\_H\_VALUE}^{9.22}(prof, \text{'region of interest specification'})$ )
16  ORELSE9.1 [ $\text{'automatic'}$  :  $\text{'null'}$ ]6)5)4)3)2)1)0)
```

Semiformal Description 9.128

Function "Value of the attribute 'picture orientation'"

The value of the attribute 'picture orientation' may be specified directly on a particular constituent (3-5). If this is not the case, test if the attribute value can be found by step BP (6). If this is not the case and the constituent is a basic layout object, test if the attribute value can be found by step C (7, 8). If this is not the case, test if the attribute value can be found by step DP (9). If this is not the case and the attribute is a basic layout object, test if the attribute value can be found by step E (10, 11). If this is not the case, test if the attribute value can be found by step FP (12). If this is not the case, test if the attribute value can be found by step G (13, 14). If this is not the case, test if the attribute value can be found by step H (15). If this is not the case, the value of the attribute 'picture orientation' is 0°.

Definition 9.128

```

1  ∀ cst, doby, prof, rdoby
2  (₀ PICTURE_ORIENTATION_VALUE(cst, doby, prof, rdoby) =
3    IF IsAttributeSet2.62(cst) and 'picture orientation' ∈ NAMS1.18(cst) and
4      not IsPlaceholder1.19(C ~ cst . 'picture orientation')
5      THEN C ~ cst . 'picture orientation'
6      ELSE STEP_BP_VALUE9.3(cst, doby, 'picture orientation') ORELSE9.1
7      (₁ IF IsBasicLayoutObjectDescription2.54(cst)
8        THEN STEP_C_VALUE9.6(cst, doby, 'picture orientation') ORELSE9.1
9        (₂ STEP_DP_VALUE9.8(cst, doby, 'picture orientation') ORELSE9.1
10       (₃ IF IsBasicLayoutObjectDescription2.54(cst)
11         THEN STEP_E_VALUE9.10(cst, doby, rdoby, prof, 'picture orientation') ORELSE9.1
12         (₄ STEP_FP_VALUE9.13(cst, doby, rdoby, prof, 'picture orientation') ORELSE9.1
13         (₅ STEP_G_VALUE9.17(cst, doby, rdoby, prof, 'picture orientation',
14           OBJECT_TYPE_VALUE9.61(cst, doby))
15         ORELSE9.1(₆ STEP_H_VALUE9.22(prof, 'picture orientation')
16         ORELSE9.1 '0°' )))))))

```

IECNORM.COM : Click to view the full PDF of ISO/IEC 8613-10:1991/Amd 5:1993

Semiformal Description 9.129

Function "Value of the attribute 'picture dimensions'"

The value of the attribute 'picture dimensions' may be specified directly on a particular constituent (3-5). If this is not the case, test if the attribute value can be found by step BP (6). If this is not the case and the constituent is a basic layout object, test if the attribute value can be found by step C (7, 8). If this is not the case, test if the attribute value can be found by step DP (9). If this is not the case and the attribute is a basic layout object, test if the attribute value can be found by step E (10, 11). If this is not the case, test if the attribute value can be found by step FP (12). If this is not the case, test if the attribute value can be found by step G (13, 14). If this is not the case, test if the attribute value can be found by step H (15). If this is not the case, the value of the attribute 'picture dimensions' is the parameter 'automatic' with the value 'null'.

Definition 9.129

```

1   $\forall cst, doby, prof, rdoby$ 
2  (0 PICTURE_DIMENSIONS_VALUE( $cst, doby, prof, rdoby$ ) =
3   IF IsAttributeSet2.62( $cst$ ) and 'picture dimensions'  $\in$  NAMS1.18( $cst$ ) and
4   not IsPlaceholder1.19( $C \sim cst \cdot$  'picture dimensions')
5   THEN  $C \sim cst \cdot$  'picture dimensions'
6   ELSE STEP_BP_VALUE9.3( $cst, doby, 'picture dimensions'$ ) ORELSE9.1
7   (1 IF IsBasicLayoutObjectDescription2.54( $cst$ )
8   THEN STEP_C_VALUE9.6( $cst, doby, 'picture dimensions'$ ) ORELSE9.1
9   (2 STEP_DP_VALUE9.8( $cst, doby, 'picture dimensions'$ ) ORELSE9.1
10  (3 IF IsBasicLayoutObjectDescription2.54( $cst$ )
11  THEN STEP_E_VALUE9.10( $cst, doby, rdoby, prof, 'picture dimensions'$ ) ORELSE9.1
12  (4 STEP_FP_VALUE9.13( $cst, doby, rdoby, prof, 'picture dimensions'$ ) ORELSE9.1
13  (5 STEP_G_VALUE9.17( $cst, doby, rdoby, prof, 'picture dimensions',$ 
14  OBJECT_TYPE_VALUE9.61( $cst, doby$ ))
15  ORELSE9.1(6 STEP_H_VALUE9.22( $prof, 'picture dimensions'$ )
16  ORELSE9.1 ['automatic' : 'null'] (6) (5) (4) (3) (2) (1) (0))

```

Semiformal Description 9.130

Function "Value of the attribute 'geometric graphics encoding announcer'"

The value of the attribute 'geometric graphics encoding announcer' may be specified directly on a particular constituent (3-5). If this is not the case, test if the attribute value can be found by step BP (6). If this is not the case and the constituent is a basic layout object, test if the attribute value can be found by step C (7, 8). If this is not the case, test if the attribute value can be found by step DP (9). If this is not the case and the attribute is a basic layout object, test if the attribute value can be found by step E (10, 11). If this is not the case, test if the attribute value can be found by step FP (12). If this is not the case, test if the attribute value can be found by step G (13, 14). If this is not the case, test if the attribute value can be found by step H (15). If this is not the case, the value of the attribute 'geometric graphics encoding announcer' comprises the parameters 'Colour Index Precision' and 'Colour Precision' with the value 8, 'Colour Selection Mode' with the value 'indexed', 'Colour Value Extent' with the value ((0,0,0),(255,255,255)), 'Index Precision', 'Integer Precision' and 'VDC Integer Precision' with the value 16, 'Maximum Colour Index' with the value 63, 'Real Precision' and 'VDC Real Precision' with the value 'fixed point format, 16, 16', and 'VDC Type' with the value 'integer'.

Definition 9.130

```

1  ∀ cst, doby, prof, rdoby
2  (₀ GEOMETRIC_GRAPHICS_ENCODING_ANNOUNCER_VALUE(cst, doby, prof, rdoby) =
3  IF IsAttributeSet2.62(cst) and 'geometric graphics encoding announcer' ∈ NAMS1.18(cst) and
4  not IsPlaceholder1.19(C ^ cst . 'geometric graphics encoding announcer')
5  THEN C ^ cst . 'geometric graphics encoding announcer'
6  ELSE STEP_BP_VALUE9.3(cst, doby, 'geometric graphics encoding announcer') ORELSE9.1
7  (₁ IF IsBasicLayoutObjectDescription2.54(cst)
8  THEN STEP_C_VALUE9.6(cst, doby, 'geometric graphics encoding announcer') ORELSE9.1
9  (₂ STEP_DP_VALUE9.8(cst, doby, 'geometric graphics encoding announcer') ORELSE9.1
10 (₃ IF IsBasicLayoutObjectDescription2.54(cst)
11 THEN STEP_E_VALUE9.10(cst, doby, rdoby, prof, 'geometric graphics encoding announcer') ORELSE9.1
12 (₄ STEP_FP_VALUE9.13(cst, doby, rdoby, prof, 'geometric graphics encoding announcer') ORELSE9.1
13 (₅ STEP_G_VALUE9.17(cst, doby, rdoby, prof, 'geometric graphics encoding announcer',
14 OBJECT_TYPE_VALUE9.61(cst, doby))
15 ORELSE9.1(₆ STEP_H_VALUE9.22(prof, 'geometric graphics encoding announcer') ORELSE9.1
16 ['Colour Index Precision' : 8; 'Colour Precision' : 8;
17 'Colour Selection Mode' : 'indexed';
18 'Colour Value Extent' : [ → [ → 0 → 0 → 0 → ] → [ → 255 → 255 → 255 → ] → ];
19 'Index Precision' : 16; 'Integer Precision' : 16;
20 'Maximum Colour Index' : 63; 'Real Precision' : 'fixed point format, 16, 16';
21 'VDC Integer Precision' : 16; 'VDC Real Precision' : 'fixed point format, 16, 16';
22 'VDC Type' : 'integer']₆)₅)₄)₃)₂)₁)₀)

```

Semiformal Description 9.131

Function "Value of the attribute 'line rendition'"

The value of the attribute 'line rendition' may be specified directly on a particular constituent (3-5). If this is not the case, test if the attribute value can be found by step BP (6). If this is not the case and the constituent is a basic layout object, test if the attribute value can be found by step C (7, 8). If this is not the case, test if the attribute value can be found by step DP (9). If this is not the case and the attribute is a basic layout object, test if the attribute value can be found by step E (10, 11). If this is not the case, test if the attribute value can be found by step FP (12). If this is not the case, test if the attribute value can be found by step G (13). If this is not the case, test if the attribute value can be found by step H (14). If this is not the case, the value of the attribute 'line rendition' comprises the parameters 'line aspect source flags' with the sub-parameters 'line colour asf', 'line type asf' and 'line width asf' whose value is 'individual', 'line bundle specifications' whose value is an empty collection, 'Line Bundle Index' and 'Line Type' with a value of 1, 'Line Colour' whose value is the colour default value, 'Line Width' whose value is the line width default value, and 'Line Width Specification Mode' with the value 'scaled'.

Definition 9.131

```

1   $\forall cst, doby, prof, rdoby$ 
2  (0 LINE_RENDITION_VALUE(cst, doby, prof, rdoby) =
3    IF IsAttributeSet2.62(cst) and 'line rendition'  $\in$  NAMS1.18(cst) and
4      not IsPlaceholder1.19( $C \hat{=} cst \bullet$  'line rendition')
5    THEN  $C \hat{=} cst \bullet$  'line rendition'
6    ELSE STEP_BP_VALUE9.3(cst, doby, 'line rendition') ORELSE9.1
7    (1 IF IsBasicLayoutObjectDescription2.54(cst)
8      THEN STEP_C_VALUE9.6(cst, doby, 'line rendition') ORELSE9.1
9      (2 STEP_DP_VALUE9.8(cst, doby, 'line rendition') ORELSE9.1
10     (3 IF IsBasicLayoutObjectDescription2.54(cst)
11       THEN STEP_E_VALUE9.10(cst, doby, rdoby, prof, 'line rendition') ORELSE9.1
12       (4 STEP_FP_VALUE9.13(cst, doby, rdoby, prof, 'line rendition') ORELSE9.1
13       (5 STEP_G_VALUE9.17(cst, doby, rdoby, prof, 'line rendition', OBJECT_TYPE_VALUE9.61(cst, doby)))
14       ORELSE9.1(6 STEP_H_VALUE9.22(prof, 'line rendition') ORELSE9.1
15         ['line aspect source flags' : ['line colour asf' : 'individual'; 'line type asf' : 'individual';
16           'line width asf' : 'individual'];
17         'line bundle specifications' : []; 'Line Bundle Index' : 1;
18         'Line Colour' : COLOUR_DEFAULT9.132(cst, doby, prof, rdoby);
19         'Line Type' : 1;
20         'Line Width' : LINE_WIDTH_DEFAULT9.133(cst, doby, prof, rdoby);
21         'Line Width Specification Mode' : 'scaled']6)5)4)3)2)1)0)

```

Semiformal Description 9.132

Function "default colour value"

If the colour selection mode is 'direct', the default colour value is 'foreground', otherwise 1.

Definition 9.132

```

1   $\forall cst, doby, prof, rdoby$ 
2  (0 COLOUR_DEFAULT(cst, doby, prof, rdoby) =
3    IF  $C \hat{=} GEOMETRIC_GRAPHICS_ENCODING_ANNOUNCER\_VALUE9.130(cst, doby, prof, rdoby) \bullet$ 
4      'Colour Selection Mode' = 'direct'
5    THEN 'foreground' ELSE 10)

```

Semiformal Description 9.133

Function "default line width value"

If the line width specification mode is 'scaled', the default line width is 1.0. Otherwise, the default line width is 0.001 times the longest side of the VDC extent.

Definition 9.133

```

1   $\forall cst, doby, prof, rdoby$ 
2  (o LINE_WIDTH_DEFAULT(cst, doby, prof, rdoby) =
3    IF LINE_WIDTH_SPECIFICATION_MODE9.134(cst, doby, prof, rdoby) = 'scaled'
4    THEN 1.0 ELSE 0.001 * LONGEST_VDC_EXTENT9.135(cst, doby, prof, rdoby)o)

```

Semiformal Description 9.134

Function "line width specification mode"

The line width specification mode is the value of the parameter 'Line Width Specification Mode' of the attribute 'line rendition'.

Definition 9.134

```

1   $\forall cst, doby, prof, rdoby$ 
2  (o LINE_WIDTH_SPECIFICATION_MODE(cst, doby, prof, rdoby) =
3    C ^LINE_RENDITION_VALUE9.131(cst, doby, prof, rdoby) • 'Line Width Specification Mode'o)

```

Semiformal Description 9.135

Function "longest VDC extent"

If the height of the VDC extent is greater than the width of the VDC extent, the longest VDC extent is the height, otherwise the width.

Definition 9.135

```

1   $\forall cst, doby, prof, rdoby$ 
2  (o LONGEST_VDC_EXTENT(cst, doby, prof, rdoby) =
3    IF ( HEIGHT9.137(C ^TRANSFORMATION_SPECIFICATION_VALUE9.155(cst, doby, prof, rdoby) •
4      'VDC Extent') >
5      WIDTH9.136(C ^TRANSFORMATION_SPECIFICATION_VALUE9.155(cst, doby, prof, rdoby) •
6      'VDC Extent')1)
7    THEN HEIGHT9.137(C ^TRANSFORMATION_SPECIFICATION_VALUE9.155(cst, doby, prof, rdoby) •
8      'VDC Extent')
9    ELSE WIDTH9.136(C ^TRANSFORMATION_SPECIFICATION_VALUE9.155(cst, doby, prof, rdoby) •
10     'VDC Extent')o)

```

Semiformal Description 9.136

Function “width (of a VDC extent)”

The width of a VDC extent is given by the first component of the second coordinate pair minus the first component of the first coordinate pair.

Definition 9.136

1 $\forall v$
 2 $({}_0\text{WIDTH}(v) =$
 3 $\text{HEAD}^{1.13}(\text{LASTC}^{1.14}(v)) - \text{HEAD}^{1.13}(\text{HEAD}^{1.13}(v))_0)$

Semiformal Description 9.137

Function “height (of a VDC extent)”

The width of a VDC extent is given by the second component of the second coordinate pair minus the second component of the first coordinate pair.

Definition 9.137

1 $\forall v$
 2 $({}_0\text{HEIGHT}(v) =$
 3 $\text{LASTC}^{1.14}(\text{LASTC}^{1.14}(v)) - \text{LASTC}^{1.14}(\text{HEAD}^{1.13}(v))_0)$

Semiformal Description 9.138

Function “Value of the attribute 'marker rendition'”

The value of the attribute 'marker rendition' may be specified directly on a particular constituent (3-5). If this is not the case, test if the attribute value can be found by step BP (6). If this is not the case and the constituent is a basic layout object, test if the attribute value can be found by step C (7, 8). If this is not the case, test if the attribute value can be found by step DP (9). If this is not the case and the attribute is a basic layout object, test if the attribute value can be found by step E (10, 11). If this is not the case, test if the attribute value can be found by step FP (12). If this is not the case, test if the attribute value can be found by step G (13, 14). If this is not the case, test if the attribute value can be found by step H (15). If this is not the case, the value of the attribute 'marker rendition' comprises the parameters 'marker aspect source flags' with the sub-parameters 'marker colour asf', 'marker size asf' and 'marker type asf' whose values are 'individual', 'marker bundle specifications' whose value is the empty collection, 'Marker Bundle Index' with a value of 1, 'Marker Colour' whose value is the default colour value, 'Marker Size' whose value is the default marker size value, 'Marker Size Specification Mode' with the value 'scaled', and 'Marker Type' with the value 3.

Definition 9.138

```

1   $\forall cst, doby, prof, rdoby$ 
2  (0 MARKER_RENDITION_VALUE(cst, doby, prof, rdoby) =
3  IF IsAttributeSet2.62(cst) and 'marker rendition' ∈ NAMS1.18(cst) and
4  not IsPlaceholder1.19(C  $\hat{cst}$  . 'marker rendition')
5  THEN C  $\hat{cst}$  . 'marker rendition'
6  ELSE STEP_BP_VALUE9.3(cst, doby, 'marker rendition') ORELSE9.1
7  (1 IF IsBasicLayoutObjectDescription2.54(cst)
8  THEN STEP_C_VALUE9.6(cst, doby, 'marker rendition') ORELSE9.1
9  (2 STEP_DP_VALUE9.8(cst, doby, 'marker rendition') ORELSE9.1
10 (3 IF IsBasicLayoutObjectDescription2.54(cst)
11 THEN STEP_E_VALUE9.10(cst, doby, rdoby, prof, 'marker rendition') ORELSE9.1
12 (4 STEP_FP_VALUE9.13(cst, doby, rdoby, prof, 'marker rendition') ORELSE9.1
13 (5 STEP_G_VALUE9.17(cst, doby, rdoby, prof, 'marker rendition',
14 OBJECT_TYPE_VALUE9.61(cst, doby))
15 ORELSE9.1(6 STEP_H_VALUE9.22(prof, 'marker rendition') ORELSE9.1
16 ['marker aspect source flags' : ['marker colour asf' : 'individual'; 'marker size asf' : 'individual';
17 'marker type asf' : 'individual'];
18 'marker bundle specifications' : []; 'Marker Bundle Index' : 1;
19 'Marker Colour' : COLOUR_DEFAULT9.132(cst, doby, prof, rdoby);
20 'Marker Size' : MARKER_SIZE_DEFAULT9.139(cst, doby, prof, rdoby);
21 'Marker Size Specification Mode' : 'scaled';
22 'Marker Type' : 3])6)5)4)3)2)1)0)

```

Semiformal Description 9.139

Function “default marker size value”

If the marker size specification mode is 'scaled', the default marker size is 1.0. Otherwise, the default marker size is 0.01 times the longest side of the VDC extent.

Definition 9.139

```

1   $\forall cst, doby, prof, rdoby$ 
2  (0 MARKER_SIZE_DEFAULT(cst, doby, prof, rdoby) =
3  IF MARKER_SIZE_SPECIFICATION_MODE9.140(cst, doby, prof, rdoby) = 'scaled'
4  THEN 1.0 ELSE 0.01 * LONGEST_VDC_EXTENT9.135(cst, doby, prof, rdoby)0)

```