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.

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

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

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

where f_y stands for $\exists x(\text{f} \wedge x = y)$, $\exists!x(\text{f})$ stands for $\exists y(\text{f}_y \wedge \forall x(\text{f} \Longrightarrow x = y))$, 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 $({}_0 x \text{ ORELSE } y =$
3 IF $x \neq$ UNDEF THEN x 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 $({}_0\text{STEP\_BL\_VALUE}(cst, doby, att) =$
3 IF 'layout style' $\in \text{NAMS}^{1.18}(cst)$ *and* $att \in \text{NAMS}^{1.18}(\text{REF\_LAY\_STYLE}^{9.4}(cst, doby))$
4 THEN C ^$(\text{REF\_LAY\_STYLE}^{9.4}(cst, doby)) \bullet att$
5 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   ∀ cst, doby, att
2   (₀STEP_BP_VALUE(cst, doby, att) =
3     IF 'presentation style' ∈ NAMS^1.18(cst) and att ∈ NAMS^1.18(REF_PRES_STYLE^9.5(cst, doby))
4     THEN C ^(REF_PRES_STYLE^9.5(cst, doby)) • att
5     ELSE UNDEF₀)
```

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   ∀ cst, doby
2   (₀REF_LAY_STYLE(cst, doby) =
3     C ^doby • <IsLayoutStyle^2.60(C xs) and C xs • 'layout style identifier' = C ^cst • 'layout style'>
4     ORELSE^9.1 [ : ]₀)
```

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   ∀ cst, doby
2   (₀REF_PRES_STYLE(cst, doby) =
3     C ^doby • <IsPresentationStyle^2.61(C xs) and
4               C xs • 'presentation style identifier' = C ^cst • 'presentation style'>
5     ORELSE^9.1 [ : ]₀)
```

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   ∀ cst, doby, att
2   (₀STEP_C_VALUE(cst, doby, att) =
3     IF 'object class' ∈ NAMS^1.18(cst) and att ∈ NAMS^1.18(REF_OBJECT_CLASS^9.7(cst, doby))
4     THEN C ^(REF_OBJECT_CLASS^9.7(cst, doby)) • att
5     ELSE UNDEF₀)
```

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   ∀ cst, doby
2   (₀REF_OBJECT_CLASS(cst, doby) =
3     C ^doby • <IsObjectClassDescription^2.37(C xs) and C xs • 'object class identifier' = C ^cst • 'object class'>
4     ORELSE^9.1 [ : ]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   ∀ cst, doby, att
2   (₀STEP_DP_VALUE(cst, doby, att) =
3     IF 'object class' ∈ NAMS^1.18(cst) and
4        'presentation style' ∈ NAMS^1.18(REF_OBJECT_CLASS^9.7(cst, doby)) and
5        att ∈ NAMS^1.18(REF_PRES_STYLE^9.5(REF_OBJECT_CLASS^9.7(cst, doby), doby))
6     THEN C ^(REF_PRES_STYLE^9.5(REF_OBJECT_CLASS^9.7(cst, doby), doby)) • att
7     ELSE UNDEF₀)
```

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  ∀ cst, doby, att
2  (₀STEP_DL_VALUE(cst, doby, att) =
3    IF 'object class' ∈ NAMS^1.18(cst) and
4       'layout style' ∈ NAMS^1.18(REF_OBJECT_CLASS^9.7(cst, doby)) and
5       att ∈ NAMS^1.18(REF_LAY_STYLE^9.4(REF_OBJECT_CLASS^9.7(cst, doby), doby))
6    THEN C ^(REF_LAY_STYLE^9.4(REF_OBJECT_CLASS^9.7(cst, doby), doby)) • att
7    ELSE UNDEF₀)
```

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  ∀ cst, doby, rdoby, prof, att
2  (₀STEP_E_VALUE(cst, doby, rdoby, prof, att) =
3    IF 'object class' ∈ NAMS^1.18(cst) and
4       'resource' ∈ NAMS^1.18(REF_OBJECT_CLASS^9.7(cst, doby)) and
5       att ∈ NAMS^1.18(RESOURCE_OBJECT_CLASS^9.11(rdoby,
6                       OBJECT_CLASSID_INRES^9.12(prof, REF_OBJECT_CLASS^9.7(cst, doby))))
7    THEN C ^(RESOURCE_OBJECT_CLASS^9.11(rdoby,
8               OBJECT_CLASSID_INRES^9.12(prof, REF_OBJECT_CLASS^9.7(cst, doby)))) • att
9    ELSE UNDEF₀)
```

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  ∀ rdoby, objcid
2  (₀RESOURCE_OBJECT_CLASS(rdoby, objcid) =
3    C ^rdoby • <IsObjectClassDescription^2.37(C xs) and C xs • 'object class identifier' = objcid>
4    ORELSE^9.1 [ : ]₀)
```

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

$\forall prof, cst$
$(_0\text{OBJECT\_CLASSID\_INRES}(prof, cst) =$
$\quad \text{LASTC}^{1.14}\ (\text{C}\ \hat{}\ prof \bullet \text{'resources'} \bullet <\text{HEAD}^{1.13}(\text{C}\ xs) = \text{C}\ \hat{}\ cst \bullet \text{'resource'} >) \bullet {}_0)$

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

$\forall cst, doby, rdoby, prof, att$
$(_0\text{STEP\_FP\_VALUE}(cst, doby, rdoby, prof, att) =$
$\quad \text{IF 'object class'} \in \text{NAMS}^{1.18}(cst)\ \underline{and}$
$\quad\quad \text{'resource'} \in \text{NAMS}^{1.18}(\text{REF\_OBJECT\_CLASS}^{9.7}(cst, doby))\ \underline{and}$
$\quad\quad \text{'presentation style'} \in \text{NAMS}^{1.18}(\text{RESOURCE\_OBJECT\_CLASS}^{9.11}(rdoby,$
$\quad\quad\quad \text{OBJECT\_CLASSID\_INRES}^{9.12}(prof, \text{REF\_OBJECT\_CLASS}^{9.7}(cst, doby))))\ \underline{and}$
$\quad\quad att \in \text{NAMS}^{1.18}(\text{RESOURCE\_PRES\_STYLE}^{9.14}(rdoby,$
$\quad\quad\quad \text{OBJECT\_CLASSID\_INRES}^{9.12}(prof, \text{REF\_OBJECT\_CLASS}^{9.7}(cst, doby))))$
$\quad \text{THEN C}\ \hat{}\ (\text{RESOURCE\_PRES\_STYLE}^{9.14}(rdoby,$
$\quad\quad \text{OBJECT\_CLASSID\_INRES}^{9.12}(prof, \text{REF\_OBJECT\_CLASS}^{9.7}(cst, doby))))) \bullet att$
$\quad \text{ELSE UNDEF}_0)$

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 $({}_0\,\mathrm{RESOURCE\_PRES\_STYLE}(rdoby, objcid) =$
3 THAT *rpres*
4 $({}_1\, rpres \in rdoby$ *and* $\mathrm{IsPresentationStyle}^{2.61}(rpres)$ *and*
5 $\exists\, rcst \in rdoby$
6 $({}_2\, \mathrm{C}\ \hat{}\, rpres \bullet$ 'presentation style identifier' $= \mathrm{C}\ \hat{}\, rcst \bullet$ 'presentation style' *and*
7 $\mathrm{C}\ \hat{}\, rcst \bullet$ 'object class identifier' $= objcid_2)_1)_0)$

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 $({}_0\,\mathrm{STEP\_FL\_VALUE}(cst, doby, rdoby, prof, att) =$
3 IF 'object class' $\in \mathrm{NAMS}^{1.18}(cst)$ *and*
4 'resource' $\in \mathrm{NAMS}^{1.18}(\mathrm{REF\_OBJECT\_CLASS}^{9.7}(cst, doby))$ *and*
5 'layout style' $\in \mathrm{NAMS}^{1.18}(\mathrm{RESOURCE\_OBJECT\_CLASS}^{9.11}(rdoby,$
6 $\mathrm{OBJECT\_CLASSID\_INRES}^{9.12}(prof, \mathrm{REF\_OBJECT\_CLASS}^{9.7}(cst, doby))))$ *and*
7 $att \in \mathrm{NAMS}^{1.18}(\mathrm{RESOURCE\_LAY\_STYLE}^{9.16}(rdoby,$
8 $\mathrm{OBJECT\_CLASSID\_INRES}^{9.12}(prof, \mathrm{REF\_OBJECT\_CLASS}^{9.7}(cst, doby))))$
9 THEN $\mathrm{C}\ \hat{}\, (\mathrm{RESOURCE\_LAY\_STYLE}^{9.16}(rdoby,$
10 $\mathrm{OBJECT\_CLASSID\_INRES}^{9.12}(prof, \mathrm{REF\_OBJECT\_CLASS}^{9.7}(cst, doby)))) \bullet att$
11 ELSE $\mathrm{UNDEF}_0)$

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

$\forall\, rdoby, objcid$
$(_0\text{RESOURCE\_LAY\_STYLE}(rdoby, objcid) =$
THAT $rlay$
$(_1 rlay \in rdoby$ *and* $\text{IsLayoutStyle}^{2.60}(rlay)$ *and*
$\exists\, rcst \in rdoby$
$(_2 \text{C}\ \hat{}\ rlay \bullet \text{'layout style identifier'} = \text{C}\ \hat{}\ rcst \bullet \text{'layout style'}$ *and*
$\text{C}\ \hat{}\ rcst \bullet \text{'object class identifier'} = objcid_2)_1)_0)$

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

$\forall\, cst, doby, rdoby, prof, att, ctype$
$(_0\text{STEP\_G\_VALUE}(cst, doby, rdoby, prof, att, ctype) =$
$\text{SUPERIOR\_VALUE}^{9.18}(_1$ THAT $sup\ (_2 sup \in doby$ *and* $(cst)\text{DescribesImSubOf}^{2.152}(sup)_2),$
$doby, rdoby, prof, att, ctype_1)$
$\text{ORELSE}^{9.1}\ \text{STEP\_G\_VALUE}^{9.17}(_3$ THAT $sup\ (_4 sup \in doby$ *and* $(cst)\text{DescribesImSubOf}^{2.152}(sup)_4),$
$doby, rdoby, prof, att, ctype_3)_0)$

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 $({}_0\text{SUPERIOR\_VALUE}(cst, doby, rdoby, prof, att, ctype) =$
3 $\text{DEFAULT\_VALUE}^{9.19}(cst, att, ctype, doby)\ \text{ORELSE}^{9.1}$
4 $({}_1\text{IF 'object class'} \in \text{NAMS}^{1.18}(cst)$
5 $\text{THEN } ({}_2\text{DEFAULT\_VALUE}^{9.19}(\text{REF\_OBJECT\_CLASS}^{9.7}(cst, doby), att, ctype, doby)\ \text{ORELSE}^{9.1}$
6 $({}_3\text{IF 'resource'} \in \text{NAMS}^{1.18}(\text{REF\_OBJECT\_CLASS}^{9.7}(cst, doby))$
7 $\text{THEN } ({}_4\text{RES\_DEFAULT\_VALUE}^{9.21}(\text{REF\_OBJECT\_CLASS}^{9.7}(cst, doby), rdoby, prof, att, ctype)$
8 $\text{ORELSE}^{9.1}\ \text{UNDEF}_4)$
9 $\text{ELSE UNDEF}_3)_2)$
10 $\text{ELSE UNDEF}_1)_0)$

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 $({}_0\text{DEFAULT\_VALUE}(cst, att, ctype, doby) =$
3 $\text{C } ({}_1\text{THAT } a$
4 $({}_2\text{'default value lists'} \in \text{NAMS}^{1.18}(cst)\ \underline{and}$
5 $\exists\, v$
6 $({}_3 v \in \hat{}cst\text{.'default value lists'} \bullet\ \underline{and}\ v = ctype\ \underline{and}$
7 $({}_4({}_5\text{IsNom(C } v)\ \underline{and}\ a = v \bullet att_5)\ \underline{or}$
8 $({}_6\text{IsCol(C } v)\ \underline{and}\ a = v \bullet\bullet att_6)_4)_3)_2)$
9 $\text{ORELSE}^{9.1}\ \text{DEFAULT\_VALUE\_BY\_STYLE}^{9.20}(cst, doby, att, ctype)_1)_0)$

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   ∀ cst, doby, att, ctype
2   (0 DEFAULT_VALUE_BY_STYLE(cst, doby, att, ctype) =
3     IF 'default value lists' ∈ NAMS^1.18(cst) and
4     ∃ v
5     (1 v ∈̂ ^cst • 'default value lists' • and v = ctype and
6       ∃ a
7       (2 (3 IsNom(C v) and a = v • 'layout style' and
8           att ∈ NAMS^1.18(REF_LAY_STYLE^9.4(C v, doby))
9           THEN C ^(REF_LAY_STYLE^9.4(C v, doby)) • att3) or
10        (4 IsNom(C v) and a = v • 'presentation style' and
11          att ∈ NAMS^1.18(REF_PRES_STYLE^9.5(C v, doby))
12          THEN C ^(REF_PRES_STYLE^9.5(C v, doby)) • att4) or
13        (5 IsCol(C v) and a = v •• 'presentation style' and
14          att ∈ NAMS^1.18(REF_PRES_STYLE^9.5(C v•, doby))
15          THEN C ^(REF_PRES_STYLE^9.5(C v•, doby)) • att5)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   ∀ cst, rdoby, prof, att, ctype
2   (0 RES_DEFAULT_VALUE(cst, rdoby, prof, att, ctype) =
3     DEFAULT_VALUE^9.19(RESOURCE_OBJECT_CLASS^9.11(rdoby,
4                 OBJECT_CLASSID_INRES^9.12(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   ∀ prof, att
2   (0 STEP_H_VALUE(prof, att) =
3     C (1 THAT a (2 a ∈̂ ^prof • 'document application profile defaults' and N a = att2)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 $({}_{0}\mathrm{STEP\_BL\_PAR\_VALUE}(cst, doby, att, par) =$
3 IF 'layout style' $\in \mathrm{NAMS}^{1.18}(cst)$ *and* $att \in \mathrm{NAMS}^{1.18}(\mathrm{REF\_LAY\_STYLE}^{9.4}(cst, doby))$ *and*
4 $par \in \mathrm{NAMS}^{1.18}(\mathrm{C}\ \hat{}\mathrm{REF\_LAY\_STYLE}^{9.4}(cst, doby) \bullet att)$
5 THEN C $\hat{}(\mathrm{REF\_LAY\_STYLE}^{9.4}(cst, doby)) \bullet att \bullet par$
6 ELSE UNDEF_{0})

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 $({}_{0}\mathrm{STEP\_C\_PAR\_VALUE}(cst, doby, att, par) =$
3 IF 'object class' $\in \mathrm{NAMS}^{1.18}(cst)$ *and* $att \in \mathrm{NAMS}^{1.18}(\mathrm{REF\_OBJECT\_CLASS}^{9.7}(cst, doby))$ *and*
4 $par \in \mathrm{NAMS}^{1.18}(\mathrm{C}\ \hat{}\mathrm{REF\_OBJECT\_CLASS}^{9.7}(cst, doby) \bullet att)$
5 THEN C $\hat{}(\mathrm{REF\_OBJECT\_CLASS}^{9.7}(cst, doby)) \bullet att \bullet par$
6 ELSE UNDEF_{0})

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 $({}_{0}\mathrm{STEP\_DL\_PAR\_VALUE}(cst, doby, att, par) =$
3 IF 'object class' $\in \mathrm{NAMS}^{1.18}(cst)$ *and*
4 'layout style' $\in \mathrm{NAMS}^{1.18}(\mathrm{REF\_OBJECT\_CLASS}^{9.7}(cst, doby))$ *and*
5 $att \in \mathrm{NAMS}^{1.18}(\mathrm{REF\_LAY\_STYLE}^{9.4}(\mathrm{REF\_OBJECT\_CLASS}^{9.7}(cst, doby), doby))$ *and*
6 $par \in \mathrm{NAMS}^{1.18}(\mathrm{C}\ \hat{}\mathrm{REF\_LAY\_STYLE}^{9.4}(\mathrm{REF\_OBJECT\_CLASS}^{9.7}(cst, doby), doby) \bullet att)$
7 THEN C $\hat{}(\mathrm{REF\_LAY\_STYLE}^{9.4}(\mathrm{REF\_OBJECT\_CLASS}^{9.7}(cst, doby), doby) \bullet att \bullet par$
8 ELSE UNDEF_{0})

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

$$
\begin{array}{ll}
1 & \forall\, cst, doby, rdoby, prof, att, par \\
2 & ({}_{o}\mathrm{STEP\_E\_PAR\_VALUE}(cst, doby, rdoby, prof, att, par) = \\
3 & \quad \mathrm{IF}\ \text{'object class'} \in \mathrm{NAMS}^{1.18}(cst)\ \underline{and} \\
4 & \qquad \text{'resource'} \in \mathrm{NAMS}^{1.18}(\mathrm{REF\_OBJECT\_CLASS}^{9.7}(cst, doby))\ \underline{and} \\
5 & \qquad att \in \mathrm{NAMS}^{1.18}(\ \mathrm{RESOURCE\_OBJECT\_CLASS}^{9.11}(rdoby, \\
6 & \qquad\qquad \mathrm{OBJECT\_CLASSID\_INRES}^{9.12}(prof, \mathrm{REF\_OBJECT\_CLASS}^{9.7}(cst, doby))))\ \underline{and} \\
7 & \qquad par \in \mathrm{NAMS}^{1.18}(\mathrm{C}\ \hat{}\mathrm{RESOURCE\_OBJECT\_CLASS}^{9.11}(rdoby, \\
8 & \qquad\qquad \mathrm{OBJECT\_CLASSID\_INRES}^{9.12}(prof, \mathrm{REF\_OBJECT\_CLASS}^{9.7}(cst, doby)))\,.\,att) \\
9 & \quad \mathrm{THEN\ C}\ \hat{}(\mathrm{RESOURCE\_OBJECT\_CLASS}^{9.11}(rdoby, \\
10 & \qquad \mathrm{OBJECT\_CLASSID\_INRES}^{9.12}(prof, \mathrm{REF\_OBJECT\_CLASS}^{9.7}(cst, doby)))\,.\,att\,.\,par \\
11 & \quad \mathrm{ELSE\ UNDEF}_{o})
\end{array}
$$

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

$$
\begin{array}{ll}
1 & \forall\, cst, doby, rdoby, prof, att, par \\
2 & ({}_{o}\mathrm{STEP\_FL\_PAR\_VALUE}(cst, doby, rdoby, prof, att, par) = \\
3 & \quad \mathrm{IF}\ \text{'object class'} \in \mathrm{NAMS}^{1.18}(cst)\ \underline{and} \\
4 & \qquad \text{'resource'} \in \mathrm{NAMS}^{1.18}(\mathrm{REF\_OBJECT\_CLASS}^{9.7}(cst, doby))\ \underline{and} \\
5 & \qquad \text{'layout style'} \in \mathrm{NAMS}^{1.18}(\mathrm{RESOURCE\_OBJECT\_CLASS}^{9.11}(rdoby, \\
6 & \qquad\qquad \mathrm{OBJECT\_CLASSID\_INRES}^{9.12}(prof, \mathrm{REF\_OBJECT\_CLASS}^{9.7}(cst, doby))))\ \underline{and} \\
7 & \qquad att \in \mathrm{NAMS}^{1.18}(\ \mathrm{RESOURCE\_LAY\_STYLE}^{9.16}(rdoby, \\
8 & \qquad\qquad \mathrm{OBJECT\_CLASSID\_INRES}^{9.12}(prof, \mathrm{REF\_OBJECT\_CLASS}^{9.7}(cst, doby))))\ \underline{and} \\
9 & \qquad par \in \mathrm{NAMS}^{1.18}(\mathrm{C}\ \hat{}\mathrm{RESOURCE\_LAY\_STYLE}^{9.16}(rdoby, \\
10 & \qquad\qquad \mathrm{OBJECT\_CLASSID\_INRES}^{9.12}(prof, \mathrm{REF\_OBJECT\_CLASS}^{9.7}(cst, doby)))\,.\,att) \\
11 & \quad \mathrm{THEN\ C}\ \hat{}(\mathrm{RESOURCE\_LAY\_STYLE}^{9.16}(rdoby, \\
12 & \qquad \mathrm{OBJECT\_CLASSID\_INRES}^{9.12}(prof, \mathrm{REF\_OBJECT\_CLASS}^{9.7}(cst, doby)))\,.\,att\,.\,par \\
13 & \quad \mathrm{ELSE\ UNDEF}_{o})
\end{array}
$$

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

$$
\begin{aligned}
&\forall\, cst, doby, rdoby, prof, att, par, ctype \\
&({}_0\text{STEP\_G\_PAR\_VALUE}(cst, doby, rdoby, prof, att, par, ctype) = \\
&\quad \text{SUPERIOR\_PAR\_VALUE}^{9.30}({}_1\text{THAT } sup\ ({}_2 sup \in doby\ \underline{and}\ (cst)\text{DescribesImSubOf}^{2.152}(sup)_2), \\
&\qquad\qquad doby, rdoby, prof, att, par, ctype_1) \\
&\quad \text{ORELSE}^{9.1}\ \text{STEP\_G\_PAR\_VALUE}^{9.28}({}_3\text{THAT } sup\ ({}_4 sup \in doby\ \underline{and}\ (cst)\text{DescribesImSubOf}^{2.152}(sup)_4), \\
&\qquad\qquad doby, rdoby, prof, att, par, ctype_3)_0)
\end{aligned}
$$

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

$$
\begin{aligned}
&\forall\, prof, att, par \\
&({}_0\text{STEP\_H\_PAR\_VALUE}(prof, att, par) = \\
&\quad \text{C}\ ({}_1\text{THAT } a \\
&\qquad ({}_2 a \hat{\in} \hat{}prof \bullet \text{'document application profile defaults'}\ \underline{and}\ \text{N}\ a = att\ \underline{and} \\
&\qquad\ \exists p\ ({}_3 p \hat{\in} \hat{}prof \bullet \text{'document application profile defaults'} \bullet att\ \underline{and}\ \text{N}\ p = par_3)_2)_1)_0)
\end{aligned}
$$

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   ∀ cst, doby, rdoby, prof, att, ctype, par
2   (_0 SUPERIOR_PAR_VALUE(cst, doby, rdoby, prof, att, ctype, par) =
3     DEFAULT_PAR_VALUE^9.31(cst, att, ctype, par, doby) ORELSE^9.1
4     (_1 IF 'object class' ∈ NAMS^1.18(cst)
5       THEN (_2 DEFAULT_PAR_VALUE^9.31(REF_OBJECT_CLASS^9.7(cst, doby), att, ctype, par, doby)
6       ORELSE^9.1 (_3 IF 'resource' ∈ NAMS^1.18(REF_OBJECT_CLASS^9.7(cst, doby))
7                  THEN (_4 RES_DEFAULT_PAR_VALUE^9.33(REF_OBJECT_CLASS^9.7(cst, doby),
8                                                      rdoby, prof, att, ctype, par)
9                         ORELSE^9.1 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   ∀ 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' ∈ NAMS^1.18(cst) and
5           ∃ v
6           (_3 v ∈ ^cst • 'default value lists' • and v = ctype and
7             (_4 (_5 IsNom(C v) and a = v • att • par_5) or
8               (_6 IsCol(C v) and a = v • • att • par_6)_4)_3)_2)
9     ORELSE^9.1 DEFAULT_PAR_VALUE_BY_STYLE^9.32(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   ∀ cst, doby, att, ctype, par
2   (0 DEFAULT_PAR_VALUE_BY_STYLE(cst, doby, att, ctype, par) =
3     IF 'default value lists' ∈ NAMS^1.18(cst) and
4     ∃ v
5     (1 v ∈̂ ~cst • 'default value lists' • and v = ctype and
6       ∃ a
7       (2 (3 IsNom(C v) and a = v • 'layout style' and
8           att ∈ NAMS^1.18(REF_LAY_STYLE^9.4(C v, doby)) and
9           par ∈ NAMS^1.18(REF_LAY_STYLE^9.4(C v, doby) • par)
10          THEN C ~(REF_LAY_STYLE^9.4(C v, doby)) • att • par3) or
11        (4 IsNom(C v) and a = v • 'presentation style' and
12          att ∈ NAMS^1.18(REF_PRES_STYLE^9.5(C v, doby)) and
13          par ∈ NAMS^1.18(REF_PRES_STYLE^9.5(C v, doby) • att)
14          THEN C ~(REF_PRES_STYLE^9.5(C v, doby)) • att • par4) or
15        (5 IsCol(C v) and a = v • • 'presentation style' and
16          att ∈ NAMS^1.18(REF_PRES_STYLE^9.5(C v • , doby)) and
17          par ∈ NAMS^1.18(REF_PRES_STYLE^9.5(C v • , doby) • att)
18          THEN C ~(REF_PRES_STYLE^9.5(C v • , doby)) • att • par5)2)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   ∀ cst, rdoby, prof, att, ctype, par
2   (0 RES_DEFAULT_PAR_VALUE(cst, rdoby, prof, att, ctype, par) =
3     DEFAULT_PAR_VALUE^9.31(RESOURCE_OBJECT_CLASS^9.11(rdoby,
4                                 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 $(_0\text{STEP\_C\_SUBPAR\_VALUE}(cst, doby, att, par, subpar) =$
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 $par \in \text{NAMS}^{1.18}(C \text{ ^}\text{REF\_OBJECT\_CLASS}^{9.7}(cst, doby) \bullet att)$ *and*
5 $subpar \in \text{NAMS}^{1.18}(C \text{ ^}\text{REF\_OBJECT\_CLASS}^{9.7}(cst, doby) \bullet att \bullet par)$
6 THEN $C \text{ ^}(\text{REF\_OBJECT\_CLASS}^{9.7}(cst, doby)) \bullet att \bullet par \bullet subpar$
7 ELSE UNDEF_0)

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 $(_0\text{STEP\_E\_SUBPAR\_VALUE}(cst, doby, rdoby, prof, att, par, subpar) =$
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 $par \in \text{NAMS}^{1.18}(C \text{ ^}\text{RESOURCE\_OBJECT\_CLASS}^{9.11}(rdoby,$
8 $\text{OBJECT\_CLASSID\_INRES}^{9.12}(prof, \text{REF\_OBJECT\_CLASS}^{9.7}(cst, doby))) \bullet att)$ *and*
9 $subpar \in \text{NAMS}^{1.18}(C \text{ ^}\text{RESOURCE\_OBJECT\_CLASS}^{9.11}(rdoby,$
10 $\text{OBJECT\_CLASSID\_INRES}^{9.12}(prof, \text{REF\_OBJECT\_CLASS}^{9.7}(cst, doby))) \bullet att \bullet par)$
11 THEN $C \text{ ^}(\text{RESOURCE\_OBJECT\_CLASS}^{9.11}(rdoby,$
12 $\text{OBJECT\_CLASSID\_INRES}^{9.12}(prof, \text{REF\_OBJECT\_CLASS}^{9.7}(cst, doby)))) \bullet att \bullet par \bullet subpar$
13 ELSE UNDEF_0)

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

$\forall\, cst, doby, rdoby, prof, att, par, subpar, ctype$
$(_0\text{STEP\_G\_SUBPAR\_VALUE}(cst, doby, rdoby, prof, att, par, ctype) =$
$\quad \text{SUPERIOR\_PAR\_VALUE}^{9.30}(_1 \text{THAT } sup\ (_2 sup \in doby\ \underline{and}$
$\qquad (cst)\text{DescribesImSubOf}^{2.152}(sup)_2),$
$\qquad doby, rdoby, prof, att, par \bullet subpar, ctype_1)$
$\quad \text{ORELSE}^{9.1}\ \text{STEP\_G\_SUBPAR\_VALUE}^{9.36}(_3 \text{THAT } sup\ (_4 sup \in doby\ \underline{and}$
$\qquad (cst)\text{DescribesImSubOf}^{2.152}(sup)_4),$
$\qquad doby, rdoby, prof, att, par, subpar, ctype_3)_0)$

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

$\forall\, cst, doby, prof, rdoby$
$(_0\text{APPLICATION\_COMMENTS\_VALUE}(cst, doby, prof, rdoby) =$
$\quad \text{IF IsAttributeSet}^{2.62}(cst)\ \underline{and}\ \text{'application comments'} \in \text{NAMS}^{1.18}(cst)\ \underline{and}$
$\quad \underline{not}\ \text{IsPlaceholder}^{1.19}(\text{C } \hat{}cst \bullet \text{'application comments'})$
$\quad \text{THEN C } \hat{}cst \bullet \text{'application comments'}$
$\quad \text{ELSE STEP\_C\_VALUE}^{9.6}(cst, doby, \text{'application comments'})\ \text{ORELSE}^{9.1}$
$\quad (_1\text{STEP\_E\_VALUE}^{9.10}(cst, doby, rdoby, prof, \text{'application comments'})$
$\qquad \text{ORELSE}^{9.1}\ [\rightarrow]_1)_0)$

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 IsAttributeSet^2.62(cst) and 'balance' ∈ NAMS^1.18(cst) and
4        not IsPlaceholder^1.19(C ^cst • 'balance')
5     THEN C ^cst • 'balance'
6     ELSE STEP_C_VALUE^9.6(cst, doby, 'balance') ORELSE^9.1
7     (1 STEP_E_VALUE^9.10(cst, doby, rdoby, prof, 'balance')
8       ORELSE^9.1 'null'1)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 IsAttributeSet^2.62(cst) and 'bindings' ∈ NAMS^1.18(cst) and
4        not IsPlaceholder^1.19(C ^cst • 'bindings')
5     THEN C ^cst • 'bindings'
6     ELSE STEP_C_VALUE^9.6(cst, doby, 'bindings') ORELSE^9.1
7     (1 STEP_E_VALUE^9.10(cst, doby, rdoby, prof, 'bindings')
8       ORELSE^9.1 []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 IsAttributeSet^2.62(cst) and 'block alignment' ∈ NAMS^1.18(cst) and
4        not IsPlaceholder^1.19(C ^cst . 'block alignment')
5     THEN C ^cst . 'block alignment'
6     ELSE STEP_BL_VALUE^9.2(cst, doby, 'block alignment') ORELSE^9.1
7     (1 STEP_DL_VALUE^9.9(cst, doby, 'block alignment') ORELSE^9.1
8       (2 STEP_FL_VALUE^9.15(cst, doby, rdoby, prof, 'block alignment') ORELSE^9.1
9         (3 STEP_G_VALUE^9.17(cst, doby, rdoby, prof, 'block alignment', OBJECT_TYPE_VALUE^9.61(cst, doby))
10          ORELSE^9.1 (4 STEP_H_VALUE^9.22(prof, 'block alignment')
11           ORELSE^9.1 'right-hand aligned' 4)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 IsAttributeSet^2.62(cst) and 'border' ∈ NAMS^1.18(cst) and
4        not IsPlaceholder^1.19(C ^cst . 'border')
5     THEN C ^cst . 'border'
6     ELSE STEP_BP_VALUE^9.3(cst, doby, 'border') ORELSE^9.1
7     (1 STEP_C_VALUE^9.6(cst, doby, 'border') ORELSE^9.1
8       (2 STEP_DP_VALUE^9.8(cst, doby, 'border') ORELSE^9.1
9         (3 STEP_E_VALUE^9.10(cst, doby, rdoby, prof, 'border') ORELSE^9.1
10          (4 STEP_FP_VALUE^9.13(cst, doby, rdoby, prof, 'border') ORELSE^9.1
11            (5 STEP_G_VALUE^9.17(cst, doby, rdoby, prof, 'border', OBJECT_TYPE_VALUE^9.61(cst, doby))
12              ORELSE^9.1 (6 STEP_H_VALUE^9.22(prof, 'border')
13               ORELSE^9.1 ['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   ∀ cst, doby, prof, rdoby
2   (0 COLOUR_VALUE(cst, doby, prof, rdoby) =
3     IF IsAttributeSet^2.62(cst) and 'colour' ∈ NAMS^1.18(cst) and
4        not IsPlaceholder^1.19(C ^cst • 'colour')
5     THEN C ^cst • 'colour'
6     ELSE STEP_BP_VALUE^9.3(cst, doby, 'colour') ORELSE^9.1
7     (1 STEP_C_VALUE^9.6(cst, doby, 'colour') ORELSE^9.1
8       (2 STEP_DP_VALUE^9.8(cst, doby, 'colour') ORELSE^9.1
9         (3 STEP_E_VALUE^9.10(cst, doby, rdoby, prof, 'colour') ORELSE^9.1
10          (4 STEP_FP_VALUE^9.13(cst, doby, rdoby, prof, 'colour') ORELSE^9.1
11            (5 STEP_G_VALUE^9.17(cst, doby, rdoby, prof, 'colour', OBJECT_TYPE_VALUE^9.61(cst, doby))
12              ORELSE^9.1 (6 STEP_H_VALUE^9.22(prof, 'colour')
13                ORELSE^9.1 'colourless' 6) 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   ∀ cst, doby, prof, rdoby
2   (0 CONCATENATION_VALUE(cst, doby, prof, rdoby) =
3     IF IsAttributeSet^2.62(cst) and 'concatenation' ∈ NAMS^1.18(cst) and
4        not IsPlaceholder^1.19(C ^cst • 'concatenation')
5     THEN C ^cst • 'concatenation'
6     ELSE STEP_BL_VALUE^9.2(cst, doby, 'concatenation') ORELSE^9.1
7     (1 STEP_DL_VALUE^9.9(cst, doby, 'concatenation') ORELSE^9.1
8       (2 STEP_FL_VALUE^9.15(cst, doby, rdoby, prof, 'concatenation') ORELSE^9.1
9         (3 STEP_G_VALUE^9.17(cst, doby, rdoby, prof, 'concatenation', OBJECT_TYPE_VALUE^9.61(cst, doby))
10          ORELSE^9.1 '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 $({}_{0}$CONTENT_ARCHITECTURE_CLASS_VALUE$(cst, doby, prof, rdoby) =$
3 IF IsAttributeSet$^{2.62}(cst)$ *and* 'content architecture class' $\in$ NAMS$^{1.18}(cst)$ *and*
4 *not* IsPlaceholder$^{1.19}$(C ^cst • 'content architecture class')
5 THEN C ^cst • 'content architecture class'
6 ELSE STEP_C_VALUE$^{9.6}(cst, doby,$ 'content architecture class') ORELSE$^{9.1}$
7 $({}_{1}$STEP_E_VALUE$^{9.10}(cst, doby, rdoby, prof,$ 'content architecture class') ORELSE$^{9.1}$
8 $({}_{2}$STEP_G_VALUE$^{9.17}(cst, doby, rdoby, prof,$ 'content architecture class',
9 OBJECT_TYPE_VALUE$^{9.61}(cst, doby)$) ORELSE$^{9.1}$
10 $({}_{3}$STEP_H_VALUE$^{9.22}(prof,$ 'content architecture class')
11 ORELSE$^{9.1}$ '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 $({}_{0}$DIMENSIONS_VALUE$(cst, doby, prof, rdoby) =$
3 ['horizontal dimension' : HORIZONTAL_DIMENSION_VALUE$^{9.46}(cst, doby, prof, rdoby)$;
4 'vertical dimension' : VERTICAL_DIMENSION_VALUE$^{9.47}(cst, doby, prof, rdoby)$]$_{0})$

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   ∀ cst, doby, prof, rdoby
2   (0 HORIZONTAL_DIMENSION_VALUE(cst, doby, prof, rdoby) =
3     IF IsAttributeSet^2.62(cst) and 'dimensions' ∈ NAMS^1.18(cst) and
4        'horizontal dimension' ∈ NAMS^1.18(C ^cst • 'dimensions') and
5        not IsPlaceholder^1.19(C ^cst • 'dimensions' • 'horizontal dimension')
6     THEN C ^cst • 'dimensions' • 'horizontal dimension'
7     ELSE STEP_C_PAR_VALUE^9.24(cst, doby, 'dimensions', 'horizontal dimension') ORELSE^9.1
8     (1 STEP_E_PAR_VALUE^9.26(cst, doby, rdoby, prof, 'dimensions', 'horizontal dimension') ORELSE^9.1
9       (2 STEP_G_PAR_VALUE^9.28(cst, doby, rdoby, prof, 'dimensions', 'horizontal dimension',
10                                OBJECT_TYPE_VALUE^9.61(cst, doby)) ORELSE^9.1
11        (3 (4 IF OBJECT_TYPE_VALUE^9.61(cst, doby) = 'composite or basic page'
12            THEN STEP_H_PAR_VALUE^9.29(prof, 'dimensions', 'horizontal dimension')
13                 ORELSE^9.1 UNDEF4)
14          ORELSE^9.1 ['fixed dimension' : HORIZONTAL_FIXED_DIMENSION_VALUE^9.48(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   ∀ cst, doby, prof, rdoby
2   (0 VERTICAL_DIMENSION_VALUE(cst, doby, prof, rdoby) =
3     IF IsAttributeSet^2.62(cst) and 'dimensions' ∈ NAMS^1.18(cst) and
4        'vertical dimension' ∈ NAMS^1.18(C ^cst • 'dimensions') and
5        not IsPlaceholder^1.19(C ^cst • 'dimensions' • 'vertical dimension')
6     THEN C ^cst • 'dimensions' • 'vertical dimension'
7     ELSE STEP_C_PAR_VALUE^9.24(cst, doby, 'dimensions', 'vertical dimension') ORELSE^9.1
8     (1 STEP_E_PAR_VALUE^9.26(cst, doby, rdoby, prof, 'dimensions', 'vertical dimension') ORELSE^9.1
9       (2 STEP_G_PAR_VALUE^9.28(cst, doby, rdoby, prof, 'dimensions', 'vertical dimension',
10                                OBJECT_TYPE_VALUE^9.61(cst, doby)) ORELSE^9.1
11        (3 (4 IF OBJECT_TYPE_VALUE^9.61(cst, doby) = 'composite or basic page'
12            THEN STEP_H_PAR_VALUE^9.29(prof, 'dimensions', 'vertical dimension')
13                 ORELSE^9.1 UNDEF4)
14          ORELSE^9.1 ['fixed dimension' : VERTICAL_FIXED_DIMENSION_VALUE^9.50(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 $({}_0\,\text{HORIZONTAL\_FIXED\_DIMENSION\_VALUE}(cst) =$
3 IF $\text{OBJECT\_TYPE\_VALUE}^{9.61}(cst, doby)$ = 'composite or basic page'
4 THEN 9240
5 ELSE $\text{AVAILABLE\_HORIZONTAL\_DIMENSION\_VALUE}^{9.49}(cst)_0)$

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 $({}_0\,\text{IsAtom}(\text{AVAILABLE\_HORIZONTAL\_DIMENSION\_VALUE}(cst))_0)$

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 $({}_0\,\text{VERTICAL\_FIXED\_DIMENSION\_VALUE}(cst) =$
3 IF $\text{OBJECT\_TYPE\_VALUE}^{9.61}(cst, doby)$ = 'composite or basic page'
4 THEN 13200
5 ELSE $\text{AVAILABLE\_VERTICAL\_DIMENSION\_VALUE}^{9.51}(cst)_0)$

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 $({}_0\,\text{IsAtom}(\text{AVAILABLE\_VERTICAL\_DIMENSION\_VALUE}(cst))_0)$

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   ∀ cst, doby, prof, rdoby
2   (0 FILL_ORDER_VALUE(cst, doby, prof, rdoby) =
3     IF IsAttributeSet^2.62(cst) and 'fill order' ∈ NAMS^1.18(cst) and
4       not IsPlaceholder^1.19(C ^cst . 'fill order')
5     THEN C ^cst . 'fill order'
6     ELSE STEP_BL_VALUE^9.2(cst, doby, 'fill order') ORELSE^9.1
7     (1 STEP_DL_VALUE^9.9(cst, doby, 'fill order') ORELSE^9.1
8       (2 STEP_FL_VALUE^9.15(cst, doby, rdoby, prof, 'fill order') ORELSE^9.1
9         (3 STEP_G_VALUE^9.17(cst, doby, rdoby, prof, 'fill order', OBJECT_TYPE_VALUE^9.61(cst, doby))
10          ORELSE^9.1 '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   ∀ cst, doby, prof, rdoby
2   (0 INDIVISIBILITY_VALUE(cst, doby, prof, rdoby) =
3     IF IsAttributeSet^2.62(cst) and 'indivisibility' ∈ NAMS^1.18(cst) and
4       not IsPlaceholder^1.19(C ^cst . 'indivisibility')
5     THEN C ^cst . 'indivisibility'
6     ELSE STEP_BL_VALUE^9.2(cst, doby, 'indivisibility') ORELSE^9.1
7     (1 STEP_DL_VALUE^9.9(cst, doby, 'indivisibility') ORELSE^9.1
8       (2 STEP_FL_VALUE^9.15(cst, doby, rdoby, prof, 'indivisibility') ORELSE^9.1
9         (3 STEP_G_VALUE^9.17(cst, doby, rdoby, prof, 'indivisibility', OBJECT_TYPE_VALUE^9.61(cst, doby))
10          ORELSE^9.1 '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 $({}_{0}\text{LAYOUT\_CATEGORY\_VALUE}(cst, doby, prof, rdoby) =$
3 IF $\text{IsAttributeSet}^{2.62}(cst)$ *and* 'layout category' $\in \text{NAMS}^{1.18}(cst)$ *and*
4 *not* $\text{IsPlaceholder}^{1.19}(\text{C}\ \hat{}cst \bullet \text{'layout category'})$
5 THEN C $\hat{}cst \bullet$ 'layout category'
6 ELSE $\text{STEP\_BL\_VALUE}^{9.2}(cst, doby,$ 'layout category') $\text{ORELSE}^{9.1}$
7 $({}_{1}\text{STEP\_DL\_VALUE}^{9.9}(cst, doby,$ 'layout category') $\text{ORELSE}^{9.1}$
8 $({}_{2}\text{STEP\_FL\_VALUE}^{9.15}(cst, doby, rdoby, prof,$ 'layout category') $\text{ORELSE}^{9.1}$
9 $({}_{3}\text{STEP\_G\_VALUE}^{9.17}(cst, doby, rdoby, prof,$ 'layout category', $\text{OBJECT\_TYPE\_VALUE}^{9.61}(cst, doby))$
10 $\text{ORELSE}^{9.1}$ '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 $({}_{0}\text{LAYOUT\_OBJECT\_CLASS\_VALUE}(cst, doby, prof, rdoby) =$
3 IF $\text{IsAttributeSet}^{2.62}(cst)$ *and* 'layout object class' $\in \text{NAMS}^{1.18}(cst)$ *and*
4 *not* $\text{IsPlaceholder}^{1.19}(\text{C}\ \hat{}cst \bullet \text{'layout object class'})$
5 THEN C $\hat{}cst \bullet$ 'layout object class'
6 ELSE $\text{STEP\_BL\_VALUE}^{9.2}(cst, doby,$ 'layout object class') $\text{ORELSE}^{9.1}$
7 $({}_{1}\text{STEP\_DL\_VALUE}^{9.9}(cst, doby,$ 'layout object class') $\text{ORELSE}^{9.1}$
8 $({}_{2}\text{STEP\_FL\_VALUE}^{9.15}(cst, doby, rdoby, prof,$ 'layout object class') $\text{ORELSE}^{9.1}$
9 $({}_{3}\text{STEP\_G\_VALUE}^{9.17}(cst, doby, rdoby, prof,$ 'layout object class',
10 $\text{OBJECT\_TYPE\_VALUE}^{9.61}(cst, doby))$
11 $\text{ORELSE}^{9.1}$ '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   ∀ cst, doby, prof, rdoby
2   (0 LAYOUT_PATH_VALUE(cst, doby, prof, rdoby) =
3     IF IsAttributeSet^2.62(cst) and 'layout path' ∈ NAMS^1.18(cst) and
4        not IsPlaccholder^1.19(C ^cst . 'layout path')
5     THEN C ^cst . 'layout path'
6     ELSE STEP_C_VALUE^9.6(cst, doby, 'layout path') ORELSE^9.1
7     (1 STEP_E_VALUE^9.10(cst, doby, rdoby, prof, 'layout path') ORELSE^9.1
8       (2 STEP_G_VALUE^9.17(cst, doby, rdoby, prof, 'layout path', OBJECT_TYPE_VALUE^9.61(cst, doby))
9         ORELSE^9.1 (3 STEP_H_VALUE^9.22(prof, 'layout path')
10          ORELSE^9.1 '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   ∀ cst, doby, prof, rdoby
2   (0 MEDIUM_TYPE_VALUE(cst, doby, prof, rdoby) =
3     ['nominal page size' : NOMINAL_PAGE_SIZE_VALUE^9.58(cst, doby, prof, rdoby);
4      'side of sheet' : SIDE_OF_SHEET_VALUE^9.59(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   ∀ cst, doby, prof, rdoby
2   (0 NOMINAL_PAGE_SIZE_VALUE(cst, doby, prof, rdoby) =
3     IF IsAttributeSet^2.62(cst) and 'medium type' ∈ NAMS^1.18(cst) and
4         'nominal page size' ∈ NAMS^1.18(C ^cst • 'medium type') and
5         not IsPlaceholder^1.19(C ^cst • 'medium type' • 'nominal page size')
6     THEN C ^cst • 'medium type' • 'nominal page size'
7     ELSE STEP_C_PAR_VALUE^9.24(cst, doby, 'medium type', 'nominal page size') ORELSE^9.1
8     (1 STEP_E_PAR_VALUE^9.26(cst, doby, rdoby, prof, 'medium type', 'nominal page size') ORELSE^9.1
9       (2 STEP_G_PAR_VALUE^9.28(cst, doby, rdoby, prof, 'medium type', 'nominal page size',
10                                 OBJECT_TYPE_VALUE^9.61(cst, doby)) ORELSE^9.1
11        (3 STEP_H_PAR_VALUE^9.29(prof, 'medium type', 'nominal page size')
12          ORELSE^9.1 [ → 9920 → 14030 → ]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   ∀ cst, doby, prof, rdoby
2   (0 SIDE_OF_SHEET_VALUE(cst, doby, prof, rdoby) =
3     IF IsAttributeSet^2.62(cst) and 'medium type' ∈ NAMS^1.18(cst) and
4         'side of sheet' ∈ NAMS^1.18(C ^cst • 'medium type') and
5         not IsPlaceholder^1.19(C ^cst • 'medium type' • 'side of sheet')
6     THEN C ^cst • 'medium type' • 'side of sheet'
7     ELSE STEP_C_PAR_VALUE^9.24(cst, doby, 'medium type', 'side of sheet') ORELSE^9.1
8     (1 STEP_E_PAR_VALUE^9.26(cst, doby, rdoby, prof, 'medium type', 'side of sheet') ORELSE^9.1
9       (2 STEP_G_PAR_VALUE^9.28(cst, doby, rdoby, prof, 'medium type', 'side of sheet',
10                                 OBJECT_TYPE_VALUE^9.61(cst, doby)) ORELSE^9.1
11        (3 STEP_H_PAR_VALUE^9.29(prof, 'medium type', 'side of sheet')
12          ORELSE^9.1 '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 $\forall\, cst, doby, prof, rdoby$
2 $({}_{0}$NEW_LAYOUT_OBJECT_VALUE$(cst, doby, prof, rdoby) =$
3 IF IsAttributeSet$^{2.62}(cst)$ *and* 'new layout object' $\in$ NAMS$^{1.18}(cst)$ *and*
4 *not* IsPlaceholder$^{1.19}$(C ^cst • 'new layout object')
5 THEN C ^cst • 'new layout object'
6 ELSE STEP_BL_VALUE$^{9.2}(cst, doby,$ 'new layout object') ORELSE$^{9.1}$
7 $({}_{1}$STEP_DL_VALUE$^{9.9}(cst, doby,$ 'new layout object') ORELSE$^{9.1}$
8 $({}_{2}$STEP_FL_VALUE$^{9.15}(cst, doby, rdoby, prof,$ 'new layout object') ORELSE$^{9.1}$
9 $({}_{3}$STEP_G_VALUE$^{9.17}(cst, doby, rdoby, prof,$ 'new layout object'
10 OBJECT_TYPE_VALUE$^{9.61}(cst, doby)$)
11 ORELSE$^{9.1}$ 'null'$_{3})_{2})_{1})_{0})$

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 $\forall\, cst, doby$
2 $({}_{0}$OBJECT_TYPE_VALUE$(cst, doby) =$
3 IF IsAttributeSet$^{2.62}(cst)$ *and* 'object type' $\in$ NAMS$^{1.18}(cst)$ *and*
4 *not* IsPlaceholder$^{1.19}$(C ^cst • 'object type')
5 THEN C ^cst • 'object type'
6 ELSE STEP_C_VALUE$^{9.6}(cst, doby,$ '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 $\forall\, cst, doby, prof, rdoby$
2 $({}_{0}$OFFSET_VALUE$(cst, doby, prof, rdoby) =$
3 ['leading offset' : LEADING_OFFSET_VALUE$^{9.63}(cst, doby, prof, rdoby)$;
4 'trailing offset' : TRAILING_OFFSET_VALUE$^{9.64}(cst, doby, prof, rdoby)$;
5 'left-hand offset' : LEFTHAND_OFFSET_VALUE$^{9.65}(cst, doby, prof, rdoby)$;
6 'right-hand offset' : RIGHTHAND_OFFSET_VALUE$^{9.66}(cst, doby, prof, rdoby)$]$_{0})$

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   ∀ cst, doby, prof, rdoby
2   (0 LEADING_OFFSET_VALUE(cst, doby, prof, rdoby) =
3     IF IsAttributeSet^2.62(cst) and 'offset' ∈ NAMS^1.18(cst) and
4        'leading offset' ∈ NAMS^1.18(C ^cst • 'offset') and
5        not IsPlaceholder^1.19(C ^cst • 'offset' • 'leading offset')
6     THEN C ^cst • 'offset' • 'leading offset'
7     ELSE STEP_BL_PAR_VALUE^9.23(cst, doby, 'offset', 'leading offset') ORELSE^9.1
8     (1 STEP_DL_PAR_VALUE^9.25(cst, doby, 'offset', 'leading offset') ORELSE^9.1
9       (2 STEP_FL_PAR_VALUE^9.27(cst, doby, rdoby, prof, 'offset', 'leading offset') ORELSE^9.1
10        (3 STEP_G_PAR_VALUE^9.28(cst, doby, rdoby, prof, 'offset', 'leading offset',
11                                 OBJECT_TYPE_VALUE^9.61(cst, doby))
12          ORELSE^9.1 0_3)_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   ∀ cst, doby, prof, rdoby
2   (0 TRAILING_OFFSET_VALUE(cst, doby, prof, rdoby) =
3     IF IsAttributeSet^2.62(cst) and 'offset' ∈ NAMS^1.18(cst) and
4        'trailing offset' ∈ NAMS^1.18(C ^cst • 'offset') and
5        not IsPlaceholder^1.19(C ^cst • 'offset' • 'trailing offset')
6     THEN C ^cst • 'offset' • 'trailing offset'
7     ELSE STEP_BL_PAR_VALUE^9.23(cst, doby, 'offset', 'trailing offset') ORELSE^9.1
8     (1 STEP_DL_PAR_VALUE^9.25(cst, doby, 'offset', 'trailing offset') ORELSE^9.1
9       (2 STEP_FL_PAR_VALUE^9.27(cst, doby, rdoby, prof, 'offset', 'trailing offset') ORELSE^9.1
10        (3 STEP_G_PAR_VALUE^9.28(cst, doby, rdoby, prof, 'offset', 'trailing offset',
11                                 OBJECT_TYPE_VALUE^9.61(cst, doby))
12          ORELSE^9.1 0_3)_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   ∀ cst, doby, prof, rdoby
2   (0 LEFTHAND_OFFSET_VALUE(cst, doby, prof, rdoby) =
3     IF IsAttributeSet^2.62(cst) and 'offset' ∈ NAMS^1.18(cst) and
4         'left-hand offset' ∈ NAMS^1.18(C ^cst . 'offset') and
5         not IsPlaceholder^1.19(C ^cst . 'offset' . 'left-hand offset')
6     THEN C ^cst . 'offset' . 'left-hand offset'
7     ELSE STEP_BL_PAR_VALUE^9.23(cst, doby, 'offset', 'left-hand offset') ORELSE^9.1
8     (1 STEP_DL_PAR_VALUE^9.25(cst, doby, 'offset', 'left-hand offset') ORELSE^9.1
9       (2 STEP_FL_PAR_VALUE^9.27(cst, doby, rdoby, prof, 'offset', 'left-hand offset') ORELSE^9.1
10        (3 STEP_G_PAR_VALUE^9.28(cst, doby, rdoby, prof, 'offset', 'left-hand offset',
11                                 OBJECT_TYPE_VALUE^9.61(cst, doby))
12          ORELSE^9.1 0 3) 2) 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   ∀ cst, doby, prof, rdoby
2   (0 RIGHTHAND_OFFSET_VALUE(cst, doby, prof, rdoby) =
3     IF IsAttributeSet^2.62(cst) and 'offset' ∈ NAMS^1.18(cst) and
4         'right-hand offset' ∈ NAMS^1.18(C ^cst . 'offset') and
5         not IsPlaceholder^1.19(C ^cst . 'offset' . 'right-hand offset')
6     THEN C ^cst . 'offset' . 'right-hand offset'
7     ELSE STEP_BL_PAR_VALUE^9.23(cst, doby, 'offset', 'right-hand offset') ORELSE^9.1
8     (1 STEP_DL_PAR_VALUE^9.25(cst, doby, 'offset', 'right-hand offset') ORELSE^9.1
9       (2 STEP_FL_PAR_VALUE^9.27(cst, doby, rdoby, prof, 'offset', 'right-hand offset') ORELSE^9.1
10        (3 STEP_G_PAR_VALUE^9.28(cst, doby, rdoby, prof, 'offset', 'right-hand offset',
11                                 OBJECT_TYPE_VALUE^9.61(cst, doby))
12          ORELSE^9.1 0 3) 2) 1) 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   ∀ cst, doby, prof, rdoby
2   (0 PAGE_POSITION_VALUE(cst, doby, prof, rdoby) =
3     IF IsAttributeSet^2.62(cst) and 'page position' ∈ NAMS^1.18(cst) and
4       not IsPlaceholder^1.19(C ^cst . 'page position')
5     THEN C ^cst . 'page position'
6     ELSE STEP_C_VALUE^9.6(cst, doby, 'page position') ORELSE^9.1
7     (1 STEP_E_VALUE^9.10(cst, doby, rdoby, prof, 'page position') ORELSE^9.1
8       (2 STEP_G_VALUE^9.17(cst, doby, rdoby, prof, 'page position',
9                            OBJECT_TYPE_VALUE^9.61(cst, doby)) ORELSE^9.1
10        (3 STEP_H_VALUE^9.22(prof, 'page position')
11          ORELSE^9.1 MINIMUM_EDGE_LOSS ^9.68(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   ∀ 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   ∀ cst, doby, prof, rdoby
2   (0 PERMITTED_CATEGORIES_VALUE(cst, doby, prof, rdoby) =
3     IF IsAttributeSet^2.62(cst) and 'permitted categories' ∈ NAMS^1.18(cst) and
4       not IsPlaceholder^1.19(C ^cst . 'permitted categories')
5     THEN C ^cst . 'permitted categories'
6     ELSE STEP_C_VALUE^9.6(cst, doby, 'permitted categories') ORELSE^9.1
7     (1 STEP_E_VALUE^9.10(cst, doby, rdoby, prof, 'permitted categories') ORELSE^9.1
8       (2 STEP_G_VALUE^9.17(cst, doby, rdoby, prof, 'permitted categories',
9                            OBJECT_TYPE_VALUE^9.61(cst, doby))
10        ORELSE^9.1 '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

$\forall\, cst, doby, prof, rdoby$
$({}_0\,\mathrm{POSITION\_VALUE}(cst, doby, prof, rdoby) =$
  ['fixed position' : $\mathrm{FIXED\_POSITION\_VALUE}^{9.71}(cst, doby, prof, rdoby)]\,{}_0)$

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

$\forall\, cst, doby, prof, rdoby$
$({}_0\,\mathrm{FIXED\_POSITION\_VALUE}(cst, doby, prof, rdoby) =$
  ['horizontal position' : $\mathrm{HORIZONTAL\_POSITION\_VALUE}^{9.72}(cst, doby, prof, rdoby)$;
   'vertical position' : $\mathrm{VERTICAL\_POSITION\_VALUE}^{9.73}(cst, doby, prof, rdoby)]\,{}_0)$

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

$\forall\, cst, doby, prof, rdoby$
$({}_0\,\mathrm{HORIZONTAL\_POSITION\_VALUE}(cst, doby, prof, rdoby) =$
  IF $\mathrm{IsAttributeSet}^{2.62}(cst)$ *and* 'position' $\in \mathrm{NAMS}^{1.18}(cst)$ *and*
    'horizontal position' $\in \mathrm{NAMS}^{1.18}(\mathrm{C}\,\hat{}\,cst$. 'position' . 'fixed position') *and*
    not $\mathrm{IsPlaceholder}^{1.19}(\mathrm{C}\,\hat{}\,cst$. 'position' . 'fixed position' . 'horizontal position')
  THEN C $\hat{}\,cst$. 'position' . 'fixed position' . 'horizontal position'
  ELSE $\mathrm{STEP\_C\_SUBPAR\_VALUE}^{9.34}(cst, doby,$ 'position', 'fixed position', 'horizontal position') $\mathrm{ORELSE}^{9.1}$
  $({}_1\,\mathrm{STEP\_E\_SUBPAR\_VALUE}^{9.35}(cst, doby, rdoby, prof,$ 'position', 'fixed position',
    'horizontal position') $\mathrm{ORELSE}^{9.1}$
    $({}_2\,\mathrm{STEP\_G\_SUBPAR\_VALUE}^{9.36}(cst, doby, rdoby, prof,$ 'position', 'fixed position',
      'horizontal position', $\mathrm{OBJECT\_TYPE\_VALUE}^{9.61}(cst, doby))$
      $\mathrm{ORELSE}^{9.1}\ 0_2)_1)_0)$

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   ∀ cst, doby, prof, rdoby
2   (0 VERTICAL_POSITION_VALUE(cst, doby, prof, rdoby) =
3     IF IsAttributeSet^2.62(cst) and 'position' ∈ NAMS^1.18(cst) and
4         'vertical position' ∈ NAMS^1.18(C ^cst • 'position' • 'fixed position') and
5         not IsPlaceholder^1.19(C ^cst • 'position' • 'fixed position' • 'vertical position')
6     THEN C ^cst • 'position' • 'fixed position' • 'vertical position'
7     ELSE STEP_C_SUBPAR_VALUE^9.34(cst, doby, 'position', 'fixed position', 'vertical position') ORELSE^9.1
8     (1 STEP_E_SUBPAR_VALUE^9.35(cst, doby, rdoby, prof, 'position', 'fixed position',
9                                 'vertical position') ORELSE^9.1
10      (2 STEP_G_SUBPAR_VALUE^9.36(cst, doby, rdoby, prof, 'position', 'fixed position',
11                                  'vertical position', OBJECT_TYPE_VALUE^9.61(cst, doby))
12        ORELSE^9.1 0 2) 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   ∀ cst, doby, prof, rdoby
2   (0 PROTECTION_VALUE(cst, doby, prof, rdoby) =
3     IF IsAttributeSet^2.62(cst) and 'protection' ∈ NAMS^1.18(cst) and
4         not IsPlaceholder^1.19(C ^cst • 'protection')
5     THEN C ^cst • 'protection'
6     ELSE STEP_C_VALUE^9.6(cst, doby, 'protection') ORELSE^9.1
7     (1 STEP_E_VALUE^9.10(cst, doby, rdoby, prof, 'protection') ORELSE^9.1
8       (2 STEP_G_VALUE^9.17(cst, doby, rdoby, prof, 'protection', OBJECT_TYPE_VALUE^9.61(cst, doby))
9         ORELSE^9.1 '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 IsAttributeSet$^{2.62}(cst)$ *and* 'same layout object' $\in$ NAMS$^{1.18}(cst)$ *and*
4 *not* IsPlaceholder$^{1.19}$(C ^cst • 'same layout object')
5 THEN C ^cst • 'same layout object'
6 ELSE STEP_BL_VALUE$^{9.2}(cst, doby,$ 'same layout object') ORELSE$^{9.1}$
7 $({}_1$STEP_DL_VALUE$^{9.9}(cst, doby,$ 'same layout object') ORELSE$^{9.1}$
8 $({}_2$STEP_FL_VALUE$^{9.15}(cst, doby, rdoby, prof,$ 'same layout object') ORELSE$^{9.1}$
9 $({}_3$STEP_G_VALUE$^{9.17}(cst, doby, rdoby, prof,$ 'same layout object',
10 OBJECT_TYPE_VALUE$^{9.61}(cst, doby)$)
11 ORELSE$^{9.1}$ ['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_VALUE$^{9.77}(cst, doby, prof, rdoby)$;
4 'trailing edge' : TRAILING_EDGE_VALUE$^{9.78}(cst, doby, prof, rdoby)$;
5 'centre separation' : CENTRE_SEPARATION_VALUE$^{9.79}(cst, doby, prof, rdoby)]_0$)

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 $({}_0\text{LEADING\_EDGE\_VALUE}(cst, doby, prof, rdoby) =$
3 IF $\text{IsAttributeSet}^{2.62}(cst)$ *and* 'separation' $\in \text{NAMS}^{1.18}(cst)$ *and*
4 'leading edge' $\in \text{NAMS}^{1.18}(\text{C}\ \hat{}cst \bullet \text{'separation'})$ *and*
5 *not* $\text{IsPlaceholder}^{1.19}(\text{C}\ \hat{}cst \bullet \text{'separation'} \bullet \text{'leading edge'})$
6 THEN C $\hat{}cst \bullet$ 'separation' $\bullet$ 'leading edge'
7 ELSE $\text{STEP\_BL\_PAR\_VALUE}^{9.23}(cst, doby, \text{'separation'}, \text{'leading edge'})$ $\text{ORELSE}^{9.1}$
8 $({}_1\text{STEP\_DL\_PAR\_VALUE}^{9.25}(cst, doby, \text{'separation'}, \text{'leading edge'})$ $\text{ORELSE}^{9.1}$
9 $({}_2\text{STEP\_FL\_PAR\_VALUE}^{9.27}(cst, doby, rdoby, prof, \text{'separation'}, \text{'leading edge'})$ $\text{ORELSE}^{9.1}$
10 $({}_3\text{STEP\_G\_PAR\_VALUE}^{9.28}(cst, doby, rdoby, prof, \text{'separation'}, \text{'leading edge'},$
11 $\text{OBJECT\_TYPE\_VALUE}^{9.61}(cst, doby))$
12 $\text{ORELSE}^{9.1}\ 0_3)_2)_1)_0)$

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 $({}_0\text{TRAILING\_EDGE\_VALUE}(cst, doby, prof, rdoby) =$
3 IF $\text{IsAttributeSet}^{2.62}(cst)$ *and* 'separation' $\in \text{NAMS}^{1.18}(cst)$ *and*
4 'trailing edge' $\in \text{NAMS}^{1.18}(\text{C}\ \hat{}cst \bullet \text{'separation'})$ *and*
5 *not* $\text{IsPlaceholder}^{1.19}(\text{C}\ \hat{}cst \bullet \text{'separation'} \bullet \text{'trailing edge'})$
6 THEN C $\hat{}cst \bullet$ 'separation' $\bullet$ 'trailing edge'
7 ELSE $\text{STEP\_BL\_PAR\_VALUE}^{9.23}(cst, doby, \text{'separation'}, \text{'trailing edge'})$ $\text{ORELSE}^{9.1}$
8 $({}_1\text{STEP\_DL\_PAR\_VALUE}^{9.25}(cst, doby, \text{'separation'}, \text{'trailing edge'})$ $\text{ORELSE}^{9.1}$
9 $({}_2\text{STEP\_FL\_PAR\_VALUE}^{9.27}(cst, doby, rdoby, prof, \text{'separation'}, \text{'trailing edge'})$ $\text{ORELSE}^{9.1}$
10 $({}_3\text{STEP\_G\_PAR\_VALUE}^{9.28}(cst, doby, rdoby, prof, \text{'separation'}, \text{'trailing edge'},$
11 $\text{OBJECT\_TYPE\_VALUE}^{9.61}(cst, doby))$
12 $\text{ORELSE}^{9.1}\ 0_3)_2)_1)_0)$

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   ∀ cst, doby, prof, rdoby
2   (0 CENTRE_SEPARATION_VALUE(cst, doby, prof, rdoby) =
3     IF IsAttributeSet^2.62(cst) and 'separation' ∈ NAMS^1.18(cst) and
4         'centre separation' ∈ NAMS^1.18(C ^cst . 'separation') and
5         not IsPlaceholder^1.19(C ^cst . 'separation' . 'centre separation')
6     THEN C ^cst . 'separation' . 'centre separation'
7     ELSE STEP_BL_PAR_VALUE^9.23(cst, doby, 'separation', 'centre separation') ORELSE^9.1
8     (1 STEP_DL_PAR_VALUE^9.25(cst, doby, 'separation', 'centre separation') ORELSE^9.1
9       (2 STEP_FL_PAR_VALUE^9.27(cst, doby, rdoby, prof, 'separation', 'centre separation') ORELSE^9.1
10        (3 STEP_G_PAR_VALUE^9.28(cst, doby, rdoby, prof, 'separation', 'centre separation',
11                                 OBJECT_TYPE_VALUE^9.61(cst, doby))
12          ORELSE^9.1 0 3) 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   ∀ cst, doby, prof, rdoby
2   (0 SYNCHRONIZATION_VALUE(cst, doby, prof, rdoby) =
3     IF IsAttributeSet^2.62(cst) and 'synchronization' ∈ NAMS^1.18(cst) and
4         not IsPlaceholder^1.19(C ^cst . 'synchronization')
5     THEN C ^cst . 'synchronization'
6     ELSE STEP_BL_VALUE^9.2(cst, doby, 'synchronization') ORELSE^9.1
7     (1 STEP_DL_VALUE^9.9(cst, doby, 'synchronization') ORELSE^9.1
8       (2 STEP_FL_VALUE^9.15(cst, doby, rdoby, prof, 'synchronization') ORELSE^9.1
9         (3 STEP_G_VALUE^9.17(cst, doby, rdoby, prof, 'synchronization', OBJECT_TYPE_VALUE^9.61(cst, doby))
10          ORELSE^9.1 '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   ∀ cst, doby, prof, rdoby
2   (0 TRANSPARENCY_VALUE(cst, doby, prof, rdoby) =
3     IF IsAttributeSet^2.62(cst) and 'transparency' ∈ NAMS^1.18(cst) and
4       not IsPlaccholder^1.19(C ^cst . 'transparency')
5     THEN C ^cst . 'transparency'
6     ELSE STEP_BP_VALUE^9.3(cst, doby, 'transparency') ORELSE^9.1
7     (1 STEP_C_VALUE^9.6(cst, doby, 'transparency') ORELSE^9.1
8       (2 STEP_DP_VALUE^9.8(cst, doby, 'transparency') ORELSE^9.1
9         (3 STEP_E_VALUE^9.10(cst, doby, rdoby, prof, 'transparency') ORELSE^9.1
10          (4 STEP_FP_VALUE^9.13(cst, doby, rdoby, prof, 'transparency') ORELSE^9.1
11            (5 STEP_G_VALUE^9.17(cst, doby, rdoby, prof, 'transparency',
12                                 OBJECT_TYPE_VALUE^9.61(cst, doby))
13              ORELSE^9.1(6 STEP_H_VALUE^9.22(prof, 'transparency')
14               ORELSE^9.1 '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   ∀ cst, doby, prof
2   (0 TYPE_OF_CODING_VALUE(cst, doby, prof) =
3     IF IsAttributeSet^2.62(cst) and 'type of coding' ∈ NAMS^1.18(cst) and
4       not IsPlaceholder^1.19(C ^cst . 'type of coding')
5     THEN C ^cst . 'type of coding'
6     ELSE STEP_C_VALUE^9.6(cst, doby, 'type of coding') ORELSE^9.1
7     (1 STEP_H_VALUE^9.22(prof, 'type of coding') ORELSE^9.1
8       (2 IF IsCharacterContentPortionDescription^2.57(cst) THEN '2 8 3 6 0' ELSE
9         (3 IF IsRasterGraphicsContentPortionDescription^2.58(cst) THEN '2 8 3 7 0' ELSE
10          (4 IF IsGeometricGraphicsContentPortionDescription^2.59(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 IsAttributeSet$^{2.62}(cst)$ *and* 'user-readable comments' $\in$ NAMS$^{1.18}(cst)$ *and*
4 *not* IsPlaceholder$^{1.19}$(C ^cst • 'user-readable comments')
5 THEN C ^cst • 'user-readable comments'
6 ELSE STEP_C_VALUE$^{9.6}(cst, doby,$ 'user-readable comments') ORELSE$^{9.1}$
7 $(_1$STEP_E_VALUE$^{9.10}(cst, doby, rdoby, prof,$ 'user-readable comments')
8 ORELSE$^{9.1}$ [$\rightarrow$] $_1)_0)$

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 IsAttributeSet$^{2.62}(cst)$ *and* 'user-visible name' $\in$ NAMS$^{1.18}(cst)$ *and*
4 *not* IsPlaceholder$^{1.19}$(C ^cst • 'user-visible name')
5 THEN C ^cst • 'user-visible name'
6 ELSE STEP_C_VALUE$^{9.6}(cst, doby,$ 'user-visible name') ORELSE$^{9.1}$
7 $(_1$STEP_E_VALUE$^{9.10}(cst, doby, rdoby, prof,$ 'user-visible name')
8 ORELSE$^{9.1}$ [$\rightarrow$] $_1)_0)$

F.3.2 Defaultable attributes of ISO 8613-6

Semiformal Description 9.85

Function "Value of the attribute 'alignment'"

The value of the attribute '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 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 'alignment' is 'start-aligned'.

Definition 9.85

1 $\forall\, cst, doby, prof, rdoby$
2 $({}_0\text{ALIGNMENT\_VALUE}(cst, doby, prof, rdoby) =$
3 IF $\text{IsAttributeSet}^{2.62}(cst)$ *and* 'alignment' $\in \text{NAMS}^{1.18}(cst)$ *and*
4 *not* $\text{IsPlaceholder}^{1.19}(\text{C}\ \hat{}cst \bullet \text{'alignment'})$
5 THEN C $\hat{}cst \bullet$ 'alignment'
6 ELSE $\text{STEP\_BP\_VALUE}^{9.3}(cst, doby, \text{'alignment'})$ $\text{ORELSE}^{9.1}$
7 $({}_1$IF $\text{IsBasicLayoutObjectDescription}^{2.54}(cst)$
8 THEN $\text{STEP\_C\_VALUE}^{9.6}(cst, doby, \text{'alignment'})$ $\text{ORELSE}^{9.1}$
9 $({}_2\text{STEP\_DP\_VALUE}^{9.8}(cst, doby, \text{'alignment'})$ $\text{ORELSE}^{9.1}$
10 $({}_3$IF $\text{IsBasicLayoutObjectDescription}^{2.54}(cst)$
11 THEN $\text{STEP\_E\_VALUE}^{9.10}(cst, doby, rdoby, prof, \text{'alignment'})$ $\text{ORELSE}^{9.1}$
12 $({}_4\text{STEP\_FP\_VALUE}^{9.13}(cst, doby, rdoby, prof, \text{'alignment'})$ $\text{ORELSE}^{9.1}$
13 $({}_5\text{STEP\_G\_VALUE}^{9.17}(cst, doby, rdoby, prof, \text{'alignment'}, \text{OBJECT\_TYPE\_VALUE}^{9.61}(cst, doby))$
14 $\text{ORELSE}^{9.1}({}_6\text{STEP\_H\_VALUE}^{9.22}(prof, \text{'alignment'})$
15 $\text{ORELSE}^{9.1}$ 'start-aligned' $)_6)_5)_4)_3)_2)_1)_0)$

Semiformal Description 9.86

Function "Value of the attribute 'character fonts'"

The value of the attribute 'character fonts' 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 fonts' is the empty nomination.

Definition 9.86

```
1   ∀ cst, doby, prof, rdoby
2   (0 CHARACTER_FONTS_VALUE(cst, doby, prof, rdoby) =
3     IF IsAttributeSet^2.62(cst) and 'character fonts' ∈ NAMS^1.18(cst) and
4        not IsPlaceholder^1.19(C ^cst . 'character fonts')
5     THEN C ^cst . 'character fonts'
6     ELSE STEP_BP_VALUE^9.3(cst, doby, 'character fonts') ORELSE^9.1
7     (1 IF IsBasicLayoutObjectDescription^2.54(cst)
8       THEN STEP_C_VALUE^9.6(cst, doby, 'character fonts') ORELSE^9.1
9       (2 STEP_DP_VALUE^9.8(cst, doby, 'character fonts') ORELSE^9.1
10        (3 IF IsBasicLayoutObjectDescription^2.54(cst)
11          THEN STEP_E_VALUE^9.10(cst, doby, rdoby, prof, 'character fonts') ORELSE^9.1
12          (4 STEP_FP_VALUE^9.13(cst, doby, rdoby, prof, 'character fonts') ORELSE^9.1
13            (5 STEP_G_VALUE^9.17(cst, doby, rdoby, prof, 'character fonts',
14                                 OBJECT_TYPE_VALUE^9.61(cst, doby))
15              ORELSE^9.1 (6 STEP_H_VALUE^9.22(prof, 'character fonts')
16               ORELSE^9.1 [ : ]6 )5 )4 )3 )2 )1 )0
```

Semiformal Description 9.87

Function "Value of the attribute 'character orientation'"

The value of the attribute 'character 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 'character orientation' is 0°.

Definition 9.87

```
1   ∀ cst, doby, prof, rdoby
2   (0 CHARACTER_ORIENTATION_VALUE(cst, doby, prof, rdoby) =
3     IF IsAttributeSet^2.62(cst) and 'character orientation' ∈ NAMS^1.18(cst) and
4        not IsPlaceholder^1.19(C ^cst . 'character orientation')
5     THEN C ^cst . 'character orientation'
6     ELSE STEP_BP_VALUE^9.3(cst, doby, 'character orientation') ORELSE^9.1
7     (1 IF IsBasicLayoutObjectDescription^2.54(cst)
8       THEN STEP_C_VALUE^9.6(cst, doby, 'character orientation') ORELSE^9.1
9       (2 STEP_DP_VALUE^9.8(cst, doby, 'character orientation') ORELSE^9.1
10        (3 IF IsBasicLayoutObjectDescription^2.54(cst)
11          THEN STEP_E_VALUE^9.10(cst, doby, rdoby, prof, 'character orientation') ORELSE^9.1
12          (4 STEP_FP_VALUE^9.13(cst, doby, rdoby, prof, 'character orientation') ORELSE^9.1
13            (5 STEP_G_VALUE^9.17(cst, doby, rdoby, prof, 'character orientation',
14                                 OBJECT_TYPE_VALUE^9.61(cst, doby))
15              ORELSE^9.1(6 STEP_H_VALUE^9.22(prof, 'character orientation')
16               ORELSE^9.1 '0°' )5 )4 )3 )2 )1 )0)
```

Semiformal Description 9.88

Function "Value of the attribute 'character path'"

The value of the attribute 'character 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 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 path' is 0°.

Definition 9.88

```
1   ∀ cst, doby, prof, rdoby
2   (0 CHARACTER_PATH_VALUE(cst, doby, prof, rdoby) =
3     IF IsAttributeSet^2.62(cst) and 'character path' ∈ NAMS^1.18(cst) and
4       not IsPlaceholder^1.19(C ^cst.'character path')
5     THEN C ^cst.'character path'
6     ELSE STEP_BP_VALUE^9.3(cst, doby, 'character path') ORELSE^9.1
7     (1 IF IsBasicLayoutObjectDescription^2.54(cst)
8       THEN STEP_C_VALUE^9.6(cst, doby, 'character path') ORELSE^9.1
9       (2 STEP_DP_VALUE^9.8(cst, doby, 'character path') ORELSE^9.1
10        (3 IF IsBasicLayoutObjectDescription^2.54(cst)
11          THEN STEP_E_VALUE^9.10(cst, doby, rdoby, prof, 'character path') ORELSE^9.1
12          (4 STEP_FP_VALUE^9.13(cst, doby, rdoby, prof, 'character path') ORELSE^9.1
13            (5 STEP_G_VALUE^9.17(cst, doby, rdoby, prof, 'character path',
14                                 OBJECT_TYPE_VALUE^9.61(cst, doby))
15              ORELSE^9.1(6 STEP_H_VALUE^9.22(prof, 'character path')
16               ORELSE^9.1 '0°' 6)5)4)3)2)1)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   ∀ cst, doby, prof, rdoby
2   (0 CHARACTER_SPACING_VALUE(cst, doby, prof, rdoby) =
3     IF IsAttributeSet^2.62(cst) and 'character spacing' ∈ NAMS^1.18(cst) and
4       not IsPlaceholder^1.19(C ^cst . 'character spacing')
5     THEN C ^cst . 'character spacing'
6     ELSE STEP_BP_VALUE^9.3(cst, doby, 'character spacing') ORELSE^9.1
7     (1 IF IsBasicLayoutObjectDescription^2.54(cst)
8       THEN STEP_C_VALUE^9.6(cst, doby, 'character spacing') ORELSE^9.1
9       (2 STEP_DP_VALUE^9.8(cst, doby, 'character spacing') ORELSE^9.1
10        (3 IF IsBasicLayoutObjectDescription^2.54(cst)
11          THEN STEP_E_VALUE^9.10(cst, doby, rdoby, prof, 'character spacing') ORELSE^9.1
12          (4 STEP_FP_VALUE^9.13(cst, doby, rdoby, prof, 'character spacing') ORELSE^9.1
13            (5 STEP_G_VALUE^9.17(cst, doby, rdoby, prof, 'character spacing',
14                                 OBJECT_TYPE_VALUE^9.61(cst, doby))
15              ORELSE^9.1(6 STEP_H_VALUE^9.22(prof, 'character spacing')
16               ORELSE^9.1 120 × BMU_EQUIVALENT^9.90(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   ∀ prof
2   (0 BMU_EQUIVALENT(prof) =
3     IF 'unit scaling' ∈ NAMS^1.18(prof)
4     THEN TAIL^1.15(prof . 'unit scaling')/HEAD^1.13(prof . '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   ∀ cst, doby, prof, rdoby
2   (0 CODE_EXTENSION_ANNOUNCERS_VALUE(cst, doby, prof, rdoby) =
3     IF IsAttributeSet2.62(cst) and 'code extension announcers' ∈ NAMS1.18(cst) and
4       not IsPlaceholder1.19(C ^cst . 'code extension announcers')
5     THEN C ^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 6)5)4)3)2)1)0)
```

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   ∀ 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   ∀ cst, doby, prof, rdoby
2   (0 FIRST_LINE_OFFSET_VALUE(cst, doby, prof, rdoby) =
3     IF IsAttributeSet^2.62(cst) and 'first line offset' ∈ NAMS^1.18(cst) and
4       not IsPlaceholder^1.19(C ^cst . 'first line offset')
5     THEN C ^cst . 'first line offset'
6     ELSE STEP_BP_VALUE^9.3(cst, doby, 'first line offset') ORELSE^9.1
7     (1 IF IsBasicLayoutObjectDescription^2.54(cst)
8       THEN STEP_C_VALUE^9.6(cst, doby, 'first line offset') ORELSE^9.1
9       (2 STEP_DP_VALUE^9.8(cst, doby, 'first line offset') ORELSE^9.1
10        (3 IF IsBasicLayoutObjectDescription^2.54(cst)
11          THEN STEP_E_VALUE^9.10(cst, doby, rdoby, prof, 'first line offset') ORELSE^9.1
12          (4 STEP_FP_VALUE^9.13(cst, doby, rdoby, prof, 'first line offset') ORELSE^9.1
13            (5 STEP_G_VALUE^9.17(cst, doby, rdoby, prof, 'first line offset',
14                                 OBJECT_TYPE_VALUE^9.61(cst, doby))
15              ORELSE^9.1(6 STEP_H_VALUE^9.22(prof, 'first line offset')
16               ORELSE^9.1 0 6) 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 IsAttributeSet^2.62(cst) and 'graphic character sets' ∈ NAMS^1.18(cst) and
4       not IsPlaceholder^1.19(C ^cst . 'graphic character sets')
5     THEN C ^cst . 'graphic character sets'
6     ELSE STEP_BP_VALUE^9.3(cst, doby, 'graphic character sets') ORELSE^9.1
7     (1 IF IsBasicLayoutObjectDescription^2.54(cst)
8       THEN STEP_C_VALUE^9.6(cst, doby, 'graphic character sets') ORELSE^9.1
9       (2 STEP_DP_VALUE^9.8(cst, doby, 'graphic character sets') ORELSE^9.1
10        (3 IF IsBasicLayoutObjectDescription^2.54(cst)
11          THEN STEP_E_VALUE^9.10(cst, doby, rdoby, prof, 'graphic character sets') ORELSE^9.1
12          (4 STEP_FP_VALUE^9.13(cst, doby, rdoby, prof, 'graphic character sets') ORELSE^9.1
13            (5 STEP_G_VALUE^9.17(cst, doby, rdoby, prof, 'graphic character sets',
14                                 OBJECT_TYPE_VALUE^9.61(cst, doby))
15              ORELSE^9.1 (6 STEP_H_VALUE^9.22(prof, 'graphic character sets')
16              ORELSE^9.1 G0_G2_ESCAPE_SEQUENCE^9.95 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 9.96

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

The value of the attribute 'graphic character subrepertoire' 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 subrepertoire' is 0.

Definition 9.96

```
1   ∀ cst, doby, prof, rdoby
2   (0 GRAPHIC_CHARACTER_SUBREPERTOIRE_VALUE(cst, doby, prof, rdoby) =
3     IF IsAttributeSet^2.62(cst) and 'graphic character subrepertoire' ∈ NAMS^1.18(cst) and
4        not IsPlaceholder^1.19(C ^cst . 'graphic character subrepertoire')
5     THEN C ^cst . 'graphic character subrepertoire'
6     ELSE STEP_BP_VALUE^9.3(cst, doby, 'graphic character subrepertoire') ORELSE^9.1
7     (1 IF IsBasicLayoutObjectDescription^2.54(cst)
8       THEN STEP_C_VALUE^9.6(cst, doby, 'graphic character subrepertoire') ORELSE^9.1
9       (2 STEP_DP_VALUE^9.8(cst, doby, 'graphic character subrepertoire') ORELSE^9.1
10        (3 IF IsBasicLayoutObjectDescription^2.54(cst)
11          THEN STEP_E_VALUE^9.10(cst, doby, rdoby, prof, 'graphic character subrepertoire') ORELSE^9.1
12          (4 STEP_FP_VALUE^9.13(cst, doby, rdoby, prof, 'graphic character subrepertoire') ORELSE^9.1
13            (5 STEP_G_VALUE^9.17(cst, doby, rdoby, prof, 'graphic character subrepertoire',
14                                 OBJECT_TYPE_VALUE^9.61(cst, doby))
15              ORELSE^9.1(6 STEP_H_VALUE^9.22(prof, 'graphic character subrepertoire')
16               ORELSE^9.1 0 6)5)4)3)2)1)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   ∀ cst, doby, prof, rdoby
2   (0 GRAPHIC_RENDITION_VALUE(cst, doby, prof, rdoby) =
3     IF IsAttributeSet^2.62(cst) and 'graphic rendition' ∈ NAMS^1.18(cst) and
4        not IsPlaceholder^1.19(C ^cst . 'graphic rendition')
5     THEN C ^cst . 'graphic rendition'
6     ELSE STEP_BP_VALUE^9.3(cst, doby, 'graphic rendition') ORELSE^9.1
7     (1 IF IsBasicLayoutObjectDescription^2.54(cst)
8       THEN STEP_C_VALUE^9.6(cst, doby, 'graphic rendition') ORELSE^9.1
9       (2 STEP_DP_VALUE^9.8(cst, doby, 'graphic rendition') ORELSE^9.1
10        (3 IF IsBasicLayoutObjectDescription^2.54(cst)
11          THEN STEP_E_VALUE^9.10(cst, doby, rdoby, prof, 'graphic rendition') ORELSE^9.1
12          (4 STEP_FP_VALUE^9.13(cst, doby, rdoby, prof, 'graphic rendition') ORELSE^9.1
13            (5 STEP_G_VALUE^9.17(cst, doby, rdoby, prof, 'graphic rendition',
14                                 OBJECT_TYPE_VALUE^9.61(cst, doby))
15              ORELSE^9.1(6 STEP_H_VALUE^9.22(prof, 'graphic rendition')
16               ORELSE^9.1 0 6) 5) 4) 3) 2) 1) 0)
```

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   ∀ cst, doby, prof, rdoby
2   (0 ITEMIZATION_VALUE(cst, doby, prof, rdoby) =
3     IF IsAttributeSet^2.62(cst) and 'itemization' ∈ NAMS^1.18(cst) and
4       not IsPlaceholder^1.19(C ~cst • 'itemization')
5     THEN C ~cst • 'itemization'
6     ELSE STEP_BP_VALUE^9.3(cst, doby, 'itemization') ORELSE^9.1
7     (1 IF IsBasicLayoutObjectDescription^2.54(cst)
8       THEN STEP_C_VALUE^9.6(cst, doby, 'itemization') ORELSE^9.1
9       (2 STEP_DP_VALUE^9.8(cst, doby, 'itemization') ORELSE^9.1
10        (3 IF IsBasicLayoutObjectDescription^2.54(cst)
11          THEN STEP_E_VALUE^9.10(cst, doby, rdoby, prof, 'itemization') ORELSE^9.1
12          (4 STEP_FP_VALUE^9.13(cst, doby, rdoby, prof, 'itemization') ORELSE^9.1
13            (5 STEP_G_VALUE^9.17(cst, doby, rdoby, prof, 'itemization', OBJECT_TYPE_VALUE^9.61(cst, doby))
14              ORELSE^9.1(6 STEP_H_VALUE^9.22(prof, 'itemization')
15                ORELSE^9.1 ['identifier alignment' : 'no itemization';
16                            'identifier start offset' : 0;
17                            'identifier end offset' : 0]6)5)4)3)2)1)0)
```

Semiformal Description 9.99

Function "Value of the attribute 'kerning offset'"

The value of the attribute 'kerning 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 'kerning offset' comprises the parameters 'start edge offset' and 'end edge offset' both with a value of 0.

Definition 9.99

```
1   ∀ cst, doby, prof, rdoby
2   (0 KERNING_OFFSET_VALUE(cst, doby, prof, rdoby) =
3     IF IsAttributeSet^2.62(cst) and 'kerning offset' ∈ NAMS^1.18(cst) and
4       not IsPlaceholder^1.19(C ^cst.'kerning offset')
5     THEN C ^cst.'kerning offset'
6     ELSE STEP_BP_VALUE^9.3(cst, doby, 'kerning offset') ORELSE^9.1
7     (1 IF IsBasicLayoutObjectDescription^2.54(cst)
8       THEN STEP_C_VALUE^9.6(cst, doby, 'kerning offset') ORELSE^9.1
9       (2 STEP_DP_VALUE^9.8(cst, doby, 'kerning offset') ORELSE^9.1
10        (3 IF IsBasicLayoutObjectDescription^2.54(cst)
11          THEN STEP_E_VALUE^9.10(cst, doby, rdoby, prof, 'kerning offset') ORELSE^9.1
12          (4 STEP_FP_VALUE^9.13(cst, doby, rdoby, prof, 'kerning offset') ORELSE^9.1
13            (5 STEP_G_VALUE^9.17(cst, doby, rdoby, prof, 'kerning offset',
14                                 OBJECT_TYPE_VALUE^9.61(cst, doby))
15              ORELSE^9.1(6 STEP_H_VALUE^9.22(prof, 'kerning offset')
16               ORELSE^9.1 ['start edge offset' : 0; 'end edge offset' : 0]6)5)4)3)2)1)0)
```

Semiformal Description 9.100

Function "Value of the attribute 'line layout table'"

The value of the attribute 'line layout table' 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 layout table' is the empty collection.

Definition 9.100

```
1   ∀ cst, doby, prof, rdoby
2   (0 LINE_LAYOUT_TABLE_VALUE(cst, doby, prof, rdoby) =
3     IF IsAttributeSet^2.62(cst) and 'line layout table' ∈ NAMS^1.18(cst) and
4        not IsPlaceholder^1.19(C ^cst • 'line layout table')
5     THEN C ^cst • 'line layout table'
6     ELSE STEP_BP_VALUE^9.3(cst, doby, 'line layout table') ORELSE^9.1
7     (1 IF IsBasicLayoutObjectDescription^2.54(cst)
8       THEN STEP_C_VALUE^9.6(cst, doby, 'line layout table') ORELSE^9.1
9       (2 STEP_DP_VALUE^9.8(cst, doby, 'line layout table') ORELSE^9.1
10        (3 IF IsBasicLayoutObjectDescription^2.54(cst)
11          THEN STEP_E_VALUE^9.10(cst, doby, rdoby, prof, 'line layout table') ORELSE^9.1
12          (4 STEP_FP_VALUE^9.13(cst, doby, rdoby, prof, 'line layout table') ORELSE^9.1
13            (5 STEP_G_VALUE^9.17(cst, doby, rdoby, prof, 'line layout table',
14                                 OBJECT_TYPE_VALUE^9.61(cst, doby))
15              ORELSE^9.1(6 STEP_H_VALUE^9.22(prof, 'line layout table')
16               ORELSE^9.1 []6)5)4)3)2)1)0)
```

Semiformal Description 9.101

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

The value of the attribute 'line progression' (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, 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 9.101

```
1   ∀ cst, doby, prof, rdoby
2   (0 CCA_LINE_PROGRESSION_VALUE(cst, doby, prof, rdoby) =
3     IF IsAttributeSet^2.62(cst) and 'line progression' ∈ NAMS^1.18(cst) and
4        not IsPlaceholder^1.19(C ^cst . 'line progression')
5     THEN C ^cst . 'line progression'
6     ELSE STEP_BP_VALUE^9.3(cst, doby, 'line progression') ORELSE^9.1
7     (1 IF IsBasicLayoutObjectDescription^2.54(cst)
8       THEN STEP_C_VALUE^9.6(cst, doby, 'line progression') ORELSE^9.1
9       (2 STEP_DP_VALUE^9.8(cst, doby, 'line progression') ORELSE^9.1
10        (3 IF IsBasicLayoutObjectDescription^2.54(cst)
11          THEN STEP_E_VALUE^9.10(cst, doby, rdoby, prof, 'line progression') ORELSE^9.1
12          (4 STEP_FP_VALUE^9.13(cst, doby, rdoby, prof, 'line progression') ORELSE^9.1
13            (5 STEP_G_VALUE^9.17(cst, doby, rdoby, prof, 'line progression',
14                                 OBJECT_TYPE_VALUE^9.61(cst, doby))
15              ORELSE^9.1(6 STEP_H_VALUE^9.22(prof, 'line progression')
16               ORELSE^9.1 '270°'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   ∀ cst, doby, prof, rdoby
2   (0 LINE_SPACING_VALUE(cst, doby, prof, rdoby) =
3     IF IsAttributeSet^2.62(cst) and 'line spacing' ∈ NAMS^1.18(cst) and
4        not IsPlaceholder^1.19(C ^cst . 'line spacing')
5     THEN C ^cst . 'line spacing'
6     ELSE STEP_BP_VALUE^9.3(cst, doby, 'line spacing') ORELSE^9.1
7     (1 IF IsBasicLayoutObjectDescription^2.54(cst)
8       THEN STEP_C_VALUE^9.6(cst, doby, 'line spacing') ORELSE^9.1
9       (2 STEP_DP_VALUE^9.8(cst, doby, 'line spacing') ORELSE^9.1
10        (3 IF IsBasicLayoutObjectDescription^2.54(cst)
11          THEN STEP_E_VALUE^9.10(cst, doby, rdoby, prof, 'line spacing') ORELSE^9.1
12          (4 STEP_FP_VALUE^9.13(cst, doby, rdoby, prof, 'line spacing') ORELSE^9.1
13            (5 STEP_G_VALUE^9.17(cst, doby, rdoby, prof, 'line spacing', OBJECT_TYPE_VALUE^9.61(cst, doby))
14              ORELSE^9.1 (6 STEP_H_VALUE^9.22(prof, 'line spacing')
15                ORELSE^9.1 200 × BMU_EQUIVALENT^9.90(prof)6)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   ∀ cst, doby, prof, rdoby
2   (0 PAIRWISE_KERNING_VALUE(cst, doby, prof, rdoby) =
3     IF IsAttributeSet^2.62(cst) and 'pairwise kerning' ∈ NAMS^1.18(cst) and
4       not IsPlaceholder^1.19(C ^cst • 'pairwise kerning')
5     THEN C ^cst • 'pairwise kerning'
6     ELSE STEP_BP_VALUE^9.3(cst, doby, 'pairwise kerning') ORELSE^9.1
7     (1 IF IsBasicLayoutObjectDescription^2.54(cst)
8       THEN STEP_C_VALUE^9.6(cst, doby, 'pairwise kerning') ORELSE^9.1
9       (2 STEP_DP_VALUE^9.8(cst, doby, 'pairwise kerning') ORELSE^9.1
10        (3 IF IsBasicLayoutObjectDescription^2.54(cst)
11          THEN STEP_E_VALUE^9.10(cst, doby, rdoby, prof, 'pairwise kerning') ORELSE^9.1
12          (4 STEP_FP_VALUE^9.13(cst, doby, rdoby, prof, 'pairwise kerning') ORELSE^9.1
13            (5 STEP_G_VALUE^9.17(cst, doby, rdoby, prof, 'pairwise kerning',
14                                 OBJECT_TYPE_VALUE^9.61(cst, doby))
15              ORELSE^9.1(6 STEP_H_VALUE^9.22(prof, 'pairwise kerning')
16               ORELSE^9.1 'no'6)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   ∀ cst, doby, prof, rdoby
2   (0 FORMATTING_INDICATOR_VALUE(cst, doby, prof, rdoby) =
3     IF IsAttributeSet^2.62(cst) and 'formatting indicator' ∈ NAMS^1.18(cst) and
4        not IsPlaceholder^1.19(C ^cst . 'formatting indicator')
5     THEN C ^cst . 'formatting indicator'
6     ELSE STEP_BP_VALUE^9.3(cst, doby, 'formatting indicator') ORELSE^9.1
7     (1 IF IsBasicLayoutObjectDescription^2.54(cst)
8       THEN STEP_C_VALUE^9.6(cst, doby, 'formatting indicator') ORELSE^9.1
9       (2 STEP_DP_VALUE^9.8(cst, doby, 'formatting indicator') ORELSE^9.1
10        (3 IF IsBasicLayoutObjectDescription^2.54(cst)
11          THEN STEP_E_VALUE^9.10(cst, doby, rdoby, prof, 'formatting indicator') ORELSE^9.1
12          (4 STEP_FP_VALUE^9.13(cst, doby, rdoby, prof, 'formatting indicator') ORELSE^9.1
13            (5 STEP_G_VALUE^9.17(cst, doby, rdoby, prof, 'formatting indicator',
14                                 OBJECT_TYPE_VALUE^9.61(cst, doby))
15              ORELSE^9.1(6 STEP_H_VALUE^9.22(prof, 'formatting indicator')
16               ORELSE^9.1 'no' 6)5)4)3)2)1)0)
```

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   ∀ cst, doby, prof, rdoby
2   (0 CCA_INITIAL_OFFSET_VALUE(cst, doby, prof, rdoby) =
3     IF IsAttributeSet^2.62(cst) and 'initial offset' ∈ NAMS^1.18(cst) and
4       not IsPlaceholder^1.19(C ^cst • 'initial offset')
5     THEN C ^cst • 'initial offset'
6     ELSE STEP_BP_VALUE^9.3(cst, doby, 'initial offset') ORELSE^9.1
7     (1 IF IsBasicLayoutObjectDescription^2.54(cst)
8       THEN STEP_C_VALUE^9.6(cst, doby, 'initial offset') ORELSE^9.1
9       (2 STEP_DP_VALUE^9.8(cst, doby, 'initial offset') ORELSE^9.1
10        (3 IF IsBasicLayoutObjectDescription^2.54(cst)
11          THEN STEP_E_VALUE^9.10(cst, doby, rdoby, prof, 'initial offset') ORELSE^9.1
12          (4 STEP_FP_VALUE^9.13(cst, doby, rdoby, prof, 'initial offset') ORELSE^9.1
13            (5 STEP_G_VALUE^9.17(cst, doby, rdoby, prof, 'initial offset', OBJECT_TYPE_VALUE^9.61(cst, doby))
14              ORELSE^9.1(6 STEP_H_VALUE^9.22(prof, 'initial offset')
15                ORELSE^9.1 CCA_INITIAL_OFFSET_DEFAULT_VALUE^9.106(cst, doby, prof, rdoby)6)5)4)3)2)1)0)
```

Semiformal Description 9.106

Function "Initial offset default value (for the character content architecture)"

The default value of the initial offset (for the character content architecture) depends on the values of the attributes 'character path' and 'line progression' (for character content) as indicated.

Definition 9.106

```
1   ∀ cst, doby, prof, rdoby
2   (0 CCA_INITIAL_OFFSET_DEFAULT_VALUE(cst, doby, prof, rdoby) =
3     IF CHARACTER_PATH_VALUE^9.88(cst, doby, prof, rdoby) = '0°' and
4        CCA_LINE_PROGRESSION_VALUE^9.101(cst, doby, prof, rdoby) = '270°'
5        THEN ['horizontal coordinate' : 0;
6              'vertical coordinate' : FIRST_LINE_BOX_BACKWARD_EXTENT^9.109(cst)] ELSE
7     (1 IF CHARACTER_PATH_VALUE^9.88(cst, doby, prof, rdoby) = '0°' and
8        CCA_LINE_PROGRESSION_VALUE^9.101(cst, doby, prof, rdoby) = '90°'
9        THEN ['horizontal coordinate' : 0;
10             'vertical coordinate' : VERTICAL_DIMENSION^9.108(cst, doby, prof, rdoby)−
11                                     FIRST_LINE_BOX_BACKWARD_EXTENT^9.109(cst)] ELSE
12      (2 IF CHARACTER_PATH_VALUE^9.88(cst, doby, prof, rdoby) = '90°' and
13         CCA_LINE_PROGRESSION_VALUE^9.101(cst, doby, prof, rdoby) = '270°'
14         THEN ['horizontal coordinate' : FIRST_LINE_BOX_BACKWARD_EXTENT^9.109(cst);
15               'vertical coordinate' : VERTICAL_DIMENSION^9.108(cst, doby, prof, rdoby)] ELSE
16       (3 IF CHARACTER_PATH_VALUE^9.88(cst, doby, prof, rdoby) = '90°' and
17          CCA_LINE_PROGRESSION_VALUE^9.101(cst, doby, prof, rdoby) = '90°'
18          THEN ['horizontal coordinate' : HORIZONTAL_DIMENSION^9.107(cst, doby, prof, rdoby)−
19                                          FIRST_LINE_BOX_BACKWARD_EXTENT^9.109(cst);
20                'vertical coordinate' : VERTICAL_DIMENSION^9.108(cst, doby, prof, rdoby)] ELSE
21        (4 IF CHARACTER_PATH_VALUE^9.88(cst, doby, prof, rdoby) = '180°' and
22           CCA_LINE_PROGRESSION_VALUE^9.101(cst, doby, prof, rdoby) = '270°'
23           THEN ['horizontal coordinate' : HORIZONTAL_DIMENSION^9.107(cst, doby, prof, rdoby);
24                 'vertical coordinate' : VERTICAL_DIMENSION^9.108(cst, doby, prof, rdoby)−
25                                         FIRST_LINE_BOX_BACKWARD_EXTENT^9.109(cst)] ELSE
26          (5 IF CHARACTER_PATH_VALUE^9.88(cst, doby, prof, rdoby) = '180°' and
27             CCA_LINE_PROGRESSION_VALUE^9.101(cst, doby, prof, rdoby) = '90°'
28             THEN ['horizontal coordinate' : HORIZONTAL_DIMENSION^9.107(cst, doby, prof, rdoby);
29                   'vertical coordinate' : FIRST_LINE_BOX_BACKWARD_EXTENT^9.109(cst)] ELSE
30            (6 IF CHARACTER_PATH_VALUE^9.88(cst, doby, prof, rdoby) = '270°' and
31               CCA_LINE_PROGRESSION_VALUE^9.101(cst, doby, prof, rdoby) = '270°'
32               THEN ['horizontal coordinate' : HORIZONTAL_DIMENSION^9.107(cst, doby, prof, rdoby)−
33                                               FIRST_LINE_BOX_BACKWARD_EXTENT^9.109(cst);
34                     'vertical coordinate' : 0] ELSE
35              (7 IF CHARACTER_PATH_VALUE^9.88(cst, doby, prof, rdoby) = '270°' and
36                 CCA_LINE_PROGRESSION_VALUE^9.101(cst, doby, prof, rdoby) = '90°'
37                 THEN ['horizontal coordinate' : FIRST_LINE_BOX_BACKWARD_EXTENT^9.109(cst);
38                        'vertical coordinate' : 0] 7)6)5)4)3)2)1)0)
```

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 $(_{0}\,\mathrm{HORIZONTAL\_DIMENSION}(cst, doby, prof, rdoby) =$
3 C ^DIMENSIONS_VALUE$^{9.45}(cst, doby, prof, rdoby) \bullet$ 'horizontal dimension' $_{0})$

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 $(_{0}\,\mathrm{VERTICAL\_DIMENSION}(cst, doby, prof, rdoby) =$
3 C ^DIMENSIONS_VALUE$^{9.45}(cst, doby, prof, rdoby) \bullet$ 'vertical dimension' $_{0})$

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 $(_{0}\,\mathrm{IsAtom}(\mathrm{FIRST\_LINE\_BOX\_BACKWARD\_EXTENT}(cst))_{0})$

Semiformal Description 9.110

Function "Value of the attribute 'indentation'"

The value of the attribute 'indentation' 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 'indentation' is 0.

Definition 9.110

```
1   ∀ cst, doby, prof, rdoby
2   (0 INDENTATION_VALUE(cst, doby, prof, rdoby) =
3     IF IsAttributeSet^2.62(cst) and 'indentation' ∈ NAMS^1.18(cst) and
4       not IsPlaccholder^1.19(C ^cst . 'indentation')
5     THEN C ^cst . 'indentation'
6     ELSE STEP_BP_VALUE^9.3(cst, doby, 'indentation') ORELSE^9.1
7     (1 IF IsBasicLayoutObjectDescription^2.54(cst)
8       THEN STEP_C_VALUE^9.6(cst, doby, 'indentation') ORELSE^9.1
9       (2 STEP_DP_VALUE^9.8(cst, doby, 'indentation') ORELSE^9.1
10        (3 IF IsBasicLayoutObjectDescription^2.54(cst)
11          THEN STEP_E_VALUE^9.10(cst, doby, rdoby, prof, 'indentation') ORELSE^9.1
12          (4 STEP_FP_VALUE^9.13(cst, doby, rdoby, prof, 'indentation') ORELSE^9.1
13            (5 STEP_G_VALUE^9.17(cst, doby, rdoby, prof, 'indentation', OBJECT_TYPE_VALUE^9.61(cst, doby))
14              ORELSE^9.1 (6 STEP_H_VALUE^9.22(prof, 'indentation')
15                ORELSE^9.1 0 6) 5) 4) 3) 2) 1) 0)
```

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   ∀ cst, doby, prof, rdoby
2   (0 ORPHAN_SIZE_VALUE(cst, doby, prof, rdoby) =
3     IF IsAttributeSet^2.62(cst) and 'orphan size' ∈ NAMS^1.18(cst) and
4       not IsPlaceholder^1.19(C ^cst . 'orphan size')
5     THEN C ^cst . 'orphan size'
6     ELSE STEP_BP_VALUE^9.3(cst, doby, 'orphan size') ORELSE^9.1
7     (1 IF IsBasicLayoutObjectDescription^2.54(cst)
8       THEN STEP_C_VALUE^9.6(cst, doby, 'orphan size') ORELSE^9.1
9       (2 STEP_DP_VALUE^9.8(cst, doby, 'orphan size') ORELSE^9.1
10        (3 IF IsBasicLayoutObjectDescription^2.54(cst)
11          THEN STEP_E_VALUE^9.10(cst, doby, rdoby, prof, 'orphan size') ORELSE^9.1
12          (4 STEP_FP_VALUE^9.13(cst, doby, rdoby, prof, 'orphan size') ORELSE^9.1
13            (5 STEP_G_VALUE^9.17(cst, doby, rdoby, prof, 'orphan size', OBJECT_TYPE_VALUE^9.61(cst, doby))
14              ORELSE^9.1 (6 STEP_H_VALUE^9.22(prof, 'orphan size')
15                ORELSE^9.1 1 6) 5) 4) 3) 2) 1) 0)
```

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   ∀ cst, doby, prof, rdoby
2   (0 PROPORTIONAL_LINE_SPACING_VALUE(cst, doby, prof, rdoby) =
3     IF IsAttributeSet^2.62(cst) and 'proportional line spacing' ∈ NAMS^1.18(cst) and
4       not IsPlaceholder^1.19(C ^cst • 'proportional line spacing')
5     THEN C ^cst • 'proportional line spacing'
6     ELSE STEP_BP_VALUE^9.3(cst, doby, 'proportional line spacing') ORELSE^9.1
7     (1 IF IsBasicLayoutObjectDescription^2.54(cst)
8       THEN STEP_C_VALUE^9.6(cst, doby, 'proportional line spacing') ORELSE^9.1
9       (2 STEP_DP_VALUE^9.8(cst, doby, 'proportional line spacing') ORELSE^9.1
10        (3 IF IsBasicLayoutObjectDescription^2.54(cst)
11          THEN STEP_E_VALUE^9.10(cst, doby, rdoby, prof, 'proportional line spacing') ORELSE^9.1
12          (4 STEP_FP_VALUE^9.13(cst, doby, rdoby, prof, 'proportional line spacing') ORELSE^9.1
13            (5 STEP_G_VALUE^9.17(cst, doby, rdoby, prof, 'proportional line spacing',
14                                 OBJECT_TYPE_VALUE^9.61(cst, doby))
15              ORELSE^9.1 (6 STEP_H_VALUE^9.22(prof, 'proportional line spacing')
16               ORELSE^9.1 'no' 6)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   ∀ cst, doby, prof, rdoby
2   (0 WIDOW_SIZE_VALUE(cst, doby, prof, rdoby) =
3     IF IsAttributeSet^2.62(cst) and 'widow size' ∈ NAMS^1.18(cst) and
4        not IsPlaceholder^1.19(C ^cst . 'widow size')
5     THEN C ^cst . 'widow size'
6     ELSE STEP_BP_VALUE^9.3(cst, doby, 'widow size') ORELSE^9.1
7     (1 IF IsBasicLayoutObjectDescription^2.54(cst)
8       THEN STEP_C_VALUE^9.6(cst, doby, 'widow size') ORELSE^9.1
9       (2 STEP_DP_VALUE^9.8(cst, doby, 'widow size') ORELSE^9.1
10        (3 IF IsBasicLayoutObjectDescription^2.54(cst)
11          THEN STEP_E_VALUE^9.10(cst, doby, rdoby, prof, 'widow size') ORELSE^9.1
12          (4 STEP_FP_VALUE^9.13(cst, doby, rdoby, prof, 'widow size') ORELSE^9.1
13            (5 STEP_G_VALUE^9.17(cst, doby, rdoby, prof, 'widow size', OBJECT_TYPE_VALUE^9.61(cst, doby))
14              ORELSE^9.1(6 STEP_H_VALUE^9.22(prof, 'widow size')
15               ORELSE^9.1 1 6) 5) 4) 3) 2) 1) 0)
```

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   ∀ cst, doby, prof, rdoby
2   (0 CLIPPING_VALUE(cst, doby, prof, rdoby) =
3     IF IsAttributeSet^2.62(cst) and 'clipping' ∈ NAMS^1.18(cst) and
4       not IsPlaceholder^1.19(C ^cst • 'clipping')
5     THEN C ^cst • 'clipping'
6     ELSE STEP_BP_VALUE^9.3(cst, doby, 'clipping') ORELSE^9.1
7     (1 IF IsBasicLayoutObjectDescription^2.54(cst)
8       THEN STEP_C_VALUE^9.6(cst, doby, 'clipping') ORELSE^9.1
9       (2 STEP_DP_VALUE^9.8(cst, doby, 'clipping') ORELSE^9.1
10        (3 IF IsBasicLayoutObjectDescription^2.54(cst)
11          THEN STEP_E_VALUE^9.10(cst, doby, rdoby, prof, 'clipping') ORELSE^9.1
12          (4 STEP_FP_VALUE^9.13(cst, doby, rdoby, prof, 'clipping') ORELSE^9.1
13            (5 STEP_G_VALUE^9.17(cst, doby, rdoby, prof, 'clipping', OBJECT_TYPE_VALUE^9.61(cst, doby))
14              ORELSE^9.1 (6 STEP_H_VALUE^9.22(prof, 'clipping')
15                ORELSE^9.1 [→ [ → 0 → 0 → ] →
16                            [→ (NUMBER_OF_PELS_PER_LINE_VALUE^9.124(THAT cont
17                              (7 (cont)DescribesContPortOf^2.153(cst)7), doby, prof, rdoby) − 1) →
18                            (C ^(THAT cont
19                              (8 (cont)DescribesContPortOf^2.153(cst)8) • 'number of lines' − 1) → ] → ]
20  6)5)4)3)2)1)0)
```

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   (0 RGCA_LINE_PROGRESSION_VALUE(cst, doby, prof, rdoby) =
3     IF IsAttributeSet^2.62(cst) and 'line progression' ∈ NAMS^1.18(cst) and
4        not IsPlaceholder^1.19(C ^cst • 'line progression')
5     THEN C ^cst • 'line progression'
6     ELSE STEP_BP_VALUE^9.3(cst, doby, 'line progression') ORELSE^9.1
7     (1 IF IsBasicLayoutObjectDescription^2.54(cst)
8       THEN STEP_C_VALUE^9.6(cst, doby, 'line progression') ORELSE^9.1
9       (2 STEP_DP_VALUE^9.8(cst, doby, 'line progression') ORELSE^9.1
10        (3 IF IsBasicLayoutObjectDescription^2.54(cst)
11          THEN STEP_E_VALUE^9.10(cst, doby, rdoby, prof, 'line progression') ORELSE^9.1
12          (4 STEP_FP_VALUE^9.13(cst, doby, rdoby, prof, 'line progression') ORELSE^9.1
13            (5 STEP_G_VALUE^9.17(cst, doby, rdoby, prof, 'line progression',
14                                 OBJECT_TYPE_VALUE^9.61(cst, doby))
15              ORELSE^9.1 (6 STEP_H_VALUE^9.22(prof, 'line progression')
16               ORELSE^9.1 '270°'6)5)4()3)2)1)
```

Semiformal Description 9.116

Function "Value of the attribute 'pel path'"

The value of the attribute 'pel 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 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 'pel path' is 0°.

Definition 9.116

```
1   ∀ cst, doby, prof, rdoby
2   (0 PEL_PATH_VALUE(cst, doby, prof, rdoby) =
3     IF IsAttributeSet^2.62(cst) and 'pel path' ∈ NAMS^1.18(cst) and
4       not IsPlaceholder^1.19(C ^cst.'pel path')
5     THEN C ^cst.'pel path'
6     ELSE STEP_BP_VALUE^9.3(cst, doby, 'pel path') ORELSE^9.1
7     (1 IF IsBasicLayoutObjectDescription^2.54(cst)
8       THEN STEP_C_VALUE^9.6(cst, doby, 'pel path') ORELSE^9.1
9       (2 STEP_DP_VALUE^9.8(cst, doby, 'pel path') ORELSE^9.1
10        (3 IF IsBasicLayoutObjectDescription^2.54(cst)
11          THEN STEP_E_VALUE^9.10(cst, doby, rdoby, prof, 'pel path') ORELSE^9.1
12          (4 STEP_FP_VALUE^9.13(cst, doby, rdoby, prof, 'pel path') ORELSE^9.1
13            (5 STEP_G_VALUE^9.17(cst, doby, rdoby, prof, 'pel path', OBJECT_TYPE_VALUE^9.61(cst, doby))
14              ORELSE^9.1 (6 STEP_H_VALUE^9.22(prof, 'pel path')
15                ORELSE^9.1 '0°' 6)5)4)3)2)1)0)
```

Semiformal Description 9.117

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

The value of the attribute 'initial offset' (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 'initial offset' is the initial offset default value (for raster graphics content).

Definition 9.117

```
1   ∀ cst, doby, prof, rdoby
2   (0 RGCA_INITIAL_OFFSET_VALUE(cst, doby, prof, rdoby) =
3     IF IsAttributeSet^2.62(cst) and 'initial offset' ∈ NAMS^1.18(cst) and
4        not IsPlaceholder^1.19(C ^cst . 'initial offset')
5     THEN C ^cst . 'initial offset'
6     ELSE STEP_BP_VALUE^9.3(cst, doby, 'initial offset') ORELSE^9.1
7     (1 IF IsBasicLayoutObjectDescription^2.54(cst)
8       THEN STEP_C_VALUE^9.6(cst, doby, 'initial offset') ORELSE^9.1
9       (2 STEP_DP_VALUE^9.8(cst, doby, 'initial offset') ORELSE^9.1
10        (3 IF IsBasicLayoutObjectDescription^2.54(cst)
11          THEN STEP_E_VALUE^9.10(cst, doby, rdoby, prof, 'initial offset') ORELSE^9.1
12          (4 STEP_FP_VALUE^9.13(cst, doby, rdoby, prof, 'initial offset') ORELSE^9.1
13            (5 STEP_G_VALUE^9.17(cst, doby, rdoby, prof, 'initial offset',
14                                 OBJECT_TYPE_VALUE^9.61(cst, doby))
15              ORELSE^9.1(6 STEP_H_VALUE^9.22(prof, 'initial offset')
16               ORELSE^9.1 RGCA_INITIAL_OFFSET_DEFAULT_VALUE^9.118(cst, doby, prof, rdoby)
17  6)5)4)3)2)1)0)
```

Semiformal Description 9.118

Function "Initial offset default value (for the raster graphics content architecture)"

The default value of the initial offset (for the raster graphics content architecture) depends on the values of the attributes 'pel path' and 'line progression' (for raster graphics content) as indicated.

Definition 9.118

```
1   ∀ cst, doby, prof, rdoby
2   (0 RGCA_INITIAL_OFFSET_DEFAULT_VALUE(cst, doby, prof, rdoby) =
3     IF PEL_PATH_VALUE^9.116(cst, doby, prof, rdoby) = '0°' and
4        RGCA_LINE_PROGRESSION_VALUE^9.115(cst, doby, prof, rdoby) = '270°'
5        THEN ['horizontal coordinate' : 0; 'vertical coordinate' : 0] ELSE
6     (1 IF PEL_PATH_VALUE^9.116(cst, doby, prof, rdoby) = '0°' and
7          RGCA_LINE_PROGRESSION_VALUE^9.115(cst, doby, prof, rdoby) = '90°'
8          THEN ['horizontal coordinate' : 0;
9                'vertical coordinate' : VERTICAL_DIMENSION^9.108(cst, doby, prof, rdoby)] ELSE
10    (2 IF PEL_PATH_VALUE^9.116(cst, doby, prof, rdoby) = '270°' and
11         RGCA_LINE_PROGRESSION_VALUE^9.115(cst, doby, prof, rdoby) = '270°'
12         THEN ['horizontal coordinate' : HORIZONTAL_DIMENSION^9.107(cst, doby, prof, rdoby);
13               'vertical coordinate' : 0] ELSE
14    (3 IF PEL_PATH_VALUE^9.116(cst, doby, prof, rdoby) = '270°' and
15         RGCA_LINE_PROGRESSION_VALUE^9.115(cst, doby, prof, rdoby) = '90°'
16         THEN ['horizontal coordinate' : 0; 'vertical coordinate' : 0] ELSE
17    (4 IF PEL_PATH_VALUE^9.116(cst, doby, prof, rdoby) = '180°' and
18         RGCA_LINE_PROGRESSION_VALUE^9.115(cst, doby, prof, rdoby) = '270°'
19         THEN ['horizontal coordinate' : HORIZONTAL_DIMENSION^9.107(cst, doby, prof, rdoby);
20               'vertical coordinate' : VERTICAL_DIMENSION^9.108(cst, doby, prof, rdoby)] ELSE
21    (5 IF PEL_PATH_VALUE^9.116(cst, doby, prof, rdoby) = '180°' and
22         RGCA_LINE_PROGRESSION_VALUE^9.115(cst, doby, prof, rdoby) = '90°'
23         THEN ['horizontal coordinate' : HORIZONTAL_DIMENSION^9.107(cst, doby, prof, rdoby);
24               'vertical coordinate' : 0] ELSE
25    (6 IF PEL_PATH_VALUE^9.116(cst, doby, prof, rdoby) = '90°' and
26         RGCA_LINE_PROGRESSION_VALUE^9.115(cst, doby, prof, rdoby) = '270°'
27         THEN ['horizontal coordinate' : 0;
28               'vertical coordinate' : VERTICAL_DIMENSION^9.108(cst, doby, prof, rdoby)] ELSE
29    (7 IF PEL_PATH_VALUE^9.116(cst, doby, prof, rdoby) = '90°' and
30         RGCA_LINE_PROGRESSION_VALUE^9.115(cst, doby, prof, rdoby) = '90°'
31         THEN ['horizontal coordinate' : HORIZONTAL_DIMENSION^9.107(cst, doby, prof, rdoby);
32               'vertical coordinate' : VERTICAL_DIMENSION^9.108(cst, doby, prof, rdoby)]
33  7)6)5)4)3)2)1)0)
```

Semiformal Description 9.119

Function "Value of the attribute 'pel transmission density'"

The value of the attribute 'pel transmission density' 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 'pel transmission density' is 6 BMU.

Definition 9.119

```
1   ∀ cst, doby, prof, rdoby
2   (0 PEL_TRANSMISSION_DENSITY_VALUE(cst, doby, prof, rdoby) =
3     IF IsAttributeSet^2.62(cst) and 'pel transmission density' ∈ NAMS^1.18(cst) and
4        not IsPlaceholder^1.19(C ^cst • 'pel transmission density')
5     THEN C ^cst • 'pel transmission density'
6     ELSE STEP_BP_VALUE^9.3(cst, doby, 'pel transmission density') ORELSE^9.1
7     (1 IF IsBasicLayoutObjectDescription^2.54(cst)
8       THEN STEP_C_VALUE^9.6(cst, doby, 'pel transmission density') ORELSE^9.1
9       (2 STEP_DP_VALUE^9.8(cst, doby, 'pel transmission density') ORELSE^9.1
10        (3 IF IsBasicLayoutObjectDescription^2.54(cst)
11          THEN STEP_E_VALUE^9.10(cst, doby, rdoby, prof, 'pel transmission density') ORELSE^9.1
12          (4 STEP_FP_VALUE^9.13(cst, doby, rdoby, prof, 'pel transmission density') ORELSE^9.1
13            (5 STEP_G_VALUE^9.17(cst, doby, rdoby, prof, 'pel transmission density',
14                                 OBJECT_TYPE_VALUE^9.61(cst, doby))
15              ORELSE^9.1 (6 STEP_H_VALUE^9.22(prof, 'pel transmission density')
16               ORELSE^9.1 '6 BMU' 6)5)4)3)2)1)0)
```

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   ∀ cst, doby, prof, rdoby
2   (0 IMAGE_DIMENSIONS_VALUE(cst, doby, prof, rdoby) =
3     IF IsAttributeSet^2.62(cst) and 'image dimensions' ∈ NAMS^1.18(cst) and
4       not IsPlaceholder^1.19(C ^cst • 'image dimensions')
5     THEN C ^cst • 'image dimensions'
6     ELSE STEP_BP_VALUE^9.3(cst, doby, 'image dimensions') ORELSE^9.1
7     (1 IF IsBasicLayoutObjectDescription^2.54(cst)
8       THEN STEP_C_VALUE^9.6(cst, doby, 'image dimensions') ORELSE^9.1
9       (2 STEP_DP_VALUE^9.8(cst, doby, 'image dimensions') ORELSE^9.1
10        (3 IF IsBasicLayoutObjectDescription^2.54(cst)
11          THEN STEP_E_VALUE^9.10(cst, doby, rdoby, prof, 'image dimensions') ORELSE^9.1
12          (4 STEP_FP_VALUE^9.13(cst, doby, rdoby, prof, 'image dimensions') ORELSE^9.1
13            (5 STEP_G_VALUE^9.17(cst, doby, rdoby, prof, 'image dimensions',
14                                 OBJECT_TYPE_VALUE^9.61(cst, doby))
15              ORELSE^9.1(6 STEP_H_VALUE^9.22(prof, 'image dimensions')
16               ORELSE^9.1 'automatic'6)5)4)3)2)1)0)
```

Semiformal Description 9.121

Function "Value of the attribute 'pel spacing'"

The value of the attribute 'pel 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 'pel spacing' are the parameter 'length' with the value 4 and the parameter 'pel spaces' with the value 1.

Definition 9.121

```
1   ∀ cst, doby, prof, rdoby
2   (0 PEL_SPACING_VALUE(cst, doby, prof, rdoby) =
3     IF IsAttributeSet^2.62(cst) and 'pel spacing' ∈ NAMS^1.18(cst) and
4       not IsPlaceholder^1.19(C ^cst . 'pel spacing')
5     THEN C ^cst . 'pel spacing'
6     ELSE STEP_BP_VALUE^9.3(cst, doby, 'pel spacing') ORELSE^9.1
7     (1 IF IsBasicLayoutObjectDescription^2.54(cst)
8       THEN STEP_C_VALUE^9.6(cst, doby, 'pel spacing') ORELSE^9.1
9       (2 STEP_DP_VALUE^9.8(cst, doby, 'pel spacing') ORELSE^9.1
10        (3 IF IsBasicLayoutObjectDescription^2.54(cst)
11          THEN STEP_E_VALUE^9.10(cst, doby, rdoby, prof, 'pel spacing') ORELSE^9.1
12          (4 STEP_FP_VALUE^9.13(cst, doby, rdoby, prof, 'pel spacing') ORELSE^9.1
13            (5 STEP_G_VALUE^9.17(cst, doby, rdoby, prof, 'pel spacing', OBJECT_TYPE_VALUE^9.61(cst, doby))
14              ORELSE^9.1(6 STEP_H_VALUE^9.22(prof, 'pel spacing')
15                ORELSE^9.1 ['length' : 4; 'pel spaces' : 1]6)5)4)3)2)1)0)
```

Semiformal Description 9.122

Function "Value of the attribute 'spacing ratio'"

The value of the attribute 'spacing ratio' 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 'spacing ratio' are the parameters 'line spacing value' and 'pel spacing value' both with a value of 1.

Definition 9.122

```
1   ∀ cst, doby, prof, rdoby
2   (0SPACING_RATIO_VALUE(cst, doby, prof, rdoby) =
3     IF IsAttributeSet^2.62(cst) and 'spacing ratio' ∈ NAMS^1.18(cst) and
4       not IsPlaceholder^1.19(C ^cst • 'spacing ratio')
5     THEN C ^cst • 'spacing ratio'
6     ELSE STEP_BP_VALUE^9.3(cst, doby, 'spacing ratio') ORELSE^9.1
7     (1IF IsBasicLayoutObjectDescription^2.54(cst)
8       THEN STEP_C_VALUE^9.6(cst, doby, 'spacing ratio') ORELSE^9.1
9       (2STEP_DP_VALUE^9.8(cst, doby, 'spacing ratio') ORELSE^9.1
10        (3IF IsBasicLayoutObjectDescription^2.54(cst)
11          THEN STEP_E_VALUE^9.10(cst, doby, rdoby, prof, 'spacing ratio') ORELSE^9.1
12          (4STEP_FP_VALUE^9.13(cst, doby, rdoby, prof, 'spacing ratio') ORELSE^9.1
13            (5STEP_G_VALUE^9.17(cst, doby, rdoby, prof, 'spacing ratio', OBJECT_TYPE_VALUE^9.61(cst, doby))
14              ORELSE^9.1(6STEP_H_VALUE^9.22(prof, 'spacing ratio')
15                ORELSE^9.1 ['line spacing value' : 1; 'pel spacing value' : 1]6)5)4)3)2)1)0)
```

Semiformal Description 9.123

Function "Value of the attribute 'compression'"

The value of the attribute 'compression' 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 'compression' is 'compressed'.

Definition 9.123

```
1   ∀ cst, doby, prof
2   (0COMPRESSION_VALUE(cst, doby, prof) =
3     IF IsAttributeSet^2.62(cst) and 'compression' ∈ NAMS^1.18(cst) and
4       not IsPlaceholder^1.19(C ^cst • 'compression')
5     THEN C ^cst • 'compression'
6     ELSE STEP_C_VALUE^9.6(cst, doby, 'compression') ORELSE^9.1
7     (1STEP_H_VALUE^9.22(prof, 'compression') ORELSE^9.1 'compressed'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   ∀ cst, doby, prof, rdoby
2   (0 NUMBER_OF_PELS_PER_LINE_VALUE(cst, doby, prof, rdoby) =
3     IF IsAttributeSet^2.62(cst) and 'number of pels per line' ∈ NAMS^1.18(cst) and
4        not IsPlaceholder^1.19(C ^cst • 'number of pels per line')
5     THEN C ^cst • 'number of pels per line'
6     ELSE STEP_C_VALUE^9.6(cst, doby, 'number of pels per line') ORELSE^9.1
7     (1 STEP_II_VALUE^9.22(prof, 'number of pels per line') ORELSE^9.1
8       (2 IF PEL_TRANSMISSION_DENSITY_VALUE^9.119(cst, doby, prof, rdoby) = '1 BMU'
9         THEN 10368 ELSE
10        (3 IF PEL_TRANSMISSION_DENSITY_VALUE^9.119(cst, doby, prof, rdoby) = '2 BMU'
11          THEN 5184 ELSE
12          (4 IF PEL_TRANSMISSION_DENSITY_VALUE^9.119(cst, doby, prof, rdoby) = '3 BMU'
13            THEN 3456 ELSE
14            (5 IF PEL_TRANSMISSION_DENSITY_VALUE^9.119(cst, doby, prof, rdoby) = '4 BMU'
15              THEN 2592 ELSE
16              (6 IF PEL_TRANSMISSION_DENSITY_VALUE^9.119(cst, doby, prof, rdoby) = '5 BMU'
17                THEN 2074 ELSE
18                (7 IF PEL_TRANSMISSION_DENSITY_VALUE^9.119(cst, doby, prof, rdoby) = '6 BMU'
19                  THEN 1728 7)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   ∀ cst, doby, prof, rdoby
2   (0 NUMBER_OF_DISCARDED_PELS_VALUE(cst, doby, prof, rdoby) =
3     IF IsAttributeSet^2.62(cst) and 'number of discarded pels' ∈ NAMS^1.18(cst) and
4        not IsPlaceholder^1.19(C ^cst • 'number of discarded pels')
5     THEN C ^cst • 'number of discarded pels'
6     ELSE STEP_C_VALUE^9.6(cst, doby, 'number of discarded pels') ORELSE^9.1
7     (1 IF MAX_PELS_PER_LINE^9.126(cst, doby, prof, rdoby) <
8          NUMBER_OF_PELS_PER_LINE_VALUE^9.124(cst, doby, prof, rdoby)
9       THEN (NUMBER_OF_PELS_PER_LINE_VALUE^9.124(cst, doby, prof, rdoby)−
10            MAX_PELS_PER_LINE^9.126(cst, doby, prof, rdoby))/2
11      ELSE 0 1)
```

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   ∀ cst, doby, prof, rdoby
2   (0 IsAtom(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   ∀ cst, doby, prof, rdoby
2   (0 REGION_OF_INTEREST_SPECIFICATION_VALUE(cst, doby, prof, rdoby) =
3     IF IsAttributeSet2.62(cst) and 'region of interest specification' ∈ NAMS1.18(cst) and
4       not IsPlaceholder1.19(C ^cst • 'region of interest specification')
5     THEN C ^cst • 'region of interest specification'
6     ELSE STEP_BP_VALUE9.3(cst, doby, 'region of interest specification') ORELSE9.1
7     (1 IF IsBasicLayoutObjectDescription2.54(cst)
8       THEN STEP_C_VALUE9.6(cst, doby, 'region of interest specification') ORELSE9.1
9       (2 STEP_DP_VALUE9.8(cst, doby, 'region of interest specification') ORELSE9.1
10        (3 IF IsBasicLayoutObjectDescription2.54(cst)
11          THEN STEP_E_VALUE9.10(cst, doby, rdoby, prof, 'region of interest specification') ORELSE9.1
12          (4 STEP_FP_VALUE9.13(cst, doby, rdoby, prof, 'region of interest specification') ORELSE9.1
13            (5 STEP_G_VALUE9.17(cst, doby, rdoby, prof, 'region of interest specification',
14                                OBJECT_TYPE_VALUE9.61(cst, doby))
15              ORELSE9.1(6 STEP_H_VALUE9.22(prof, 'region of interest specification')
16                ORELSE9.1 ['automatic' : '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 IsAttributeSet^2.62(cst) and 'picture orientation' ∈ NAMS^1.18(cst) and
4       not IsPlaceholder^1.19(C ^cst . 'picture orientation')
5     THEN C ^cst . 'picture orientation'
6     ELSE STEP_BP_VALUE^9.3(cst, doby, 'picture orientation') ORELSE^9.1
7     (1 IF IsBasicLayoutObjectDescription^2.54(cst)
8       THEN STEP_C_VALUE^9.6(cst, doby, 'picture orientation') ORELSE^9.1
9       (2 STEP_DP_VALUE^9.8(cst, doby, 'picture orientation') ORELSE^9.1
10        (3 IF IsBasicLayoutObjectDescription^2.54(cst)
11          THEN STEP_E_VALUE^9.10(cst, doby, rdoby, prof, 'picture orientation') ORELSE^9.1
12          (4 STEP_FP_VALUE^9.13(cst, doby, rdoby, prof, 'picture orientation') ORELSE^9.1
13            (5 STEP_G_VALUE^9.17(cst, doby, rdoby, prof, 'picture orientation',
14                                 OBJECT_TYPE_VALUE^9.61(cst, doby))
15              ORELSE^9.1(6 STEP_H_VALUE^9.22(prof, 'picture orientation')
16               ORELSE^9.1 '0°' 6)5)4)3)2)1)0)
```

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   ∀ cst, doby, prof, rdoby
2   (0 PICTURE_DIMENSIONS_VALUE(cst, doby, prof, rdoby) =
3     IF IsAttributeSet^2.62(cst) and 'picture dimensions' ∈ NAMS^1.18(cst) and
4       not IsPlaceholder^1.19(C ^cst • 'picture dimensions')
5     THEN C ^cst • 'picture dimensions'
6     ELSE STEP_BP_VALUE^9.3(cst, doby, 'picture dimensions') ORELSE^9.1
7     (1 IF IsBasicLayoutObjectDescription^2.54(cst)
8       THEN STEP_C_VALUE^9.6(cst, doby, 'picture dimensions') ORELSE^9.1
9       (2 STEP_DP_VALUE^9.8(cst, doby, 'picture dimensions') ORELSE^9.1
10        (3 IF IsBasicLayoutObjectDescription^2.54(cst)
11          THEN STEP_E_VALUE^9.10(cst, doby, rdoby, prof, 'picture dimensions') ORELSE^9.1
12          (4 STEP_FP_VALUE^9.13(cst, doby, rdoby, prof, 'picture dimensions') ORELSE^9.1
13            (5 STEP_G_VALUE^9.17(cst, doby, rdoby, prof, 'picture dimensions',
14                                 OBJECT_TYPE_VALUE^9.61(cst, doby))
15              ORELSE^9.1(6 STEP_H_VALUE^9.22(prof, 'picture dimensions')
16               ORELSE^9.1 ['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

$\forall\, cst, doby, prof, rdoby$
$(_0\text{GEOMETRIC\_GRAPHICS\_ENCODING\_ANNOUNCER\_VALUE}(cst, doby, prof, rdoby) =$
IF $\text{IsAttributeSet}^{2.62}(cst)$ *and* 'geometric graphics encoding announcer' $\in \text{NAMS}^{1.18}(cst)$ *and*
not $\text{IsPlaceholder}^{1.19}$(C ^$cst$ • 'geometric graphics encoding announcer')
THEN C ^cst • 'geometric graphics encoding announcer'
ELSE $\text{STEP\_BP\_VALUE}^{9.3}(cst, doby,$ 'geometric graphics encoding announcer') $\text{ORELSE}^{9.1}$
$(_1$IF $\text{IsBasicLayoutObjectDescription}^{2.54}(cst)$
THEN $\text{STEP\_C\_VALUE}^{9.6}(cst, doby,$ 'geometric graphics encoding announcer') $\text{ORELSE}^{9.1}$
$(_2\text{STEP\_DP\_VALUE}^{9.8}(cst, doby,$ 'geometric graphics encoding announcer') $\text{ORELSE}^{9.1}$
$(_3$IF $\text{IsBasicLayoutObjectDescription}^{2.54}(cst)$
THEN $\text{STEP\_E\_VALUE}^{9.10}(cst, doby, rdoby, prof,$ 'geometric graphics encoding announcer') $\text{ORELSE}^{9.1}$
$(_4\text{STEP\_FP\_VALUE}^{9.13}(cst, doby, rdoby, prof,$ 'geometric graphics encoding announcer') $\text{ORELSE}^{9.1}$
$(_5\text{STEP\_G\_VALUE}^{9.17}(cst, doby, rdoby, prof,$ 'geometric graphics encoding announcer',
$\text{OBJECT\_TYPE\_VALUE}^{9.61}(cst, doby)$)
$\text{ORELSE}^{9.1}(_6\text{STEP\_H\_VALUE}^{9.22}(prof,$ 'geometric graphics encoding announcer') $\text{ORELSE}^{9.1}$
['Colour Index Precision' : 8; 'Colour Precision' : 8;
'Colour Selection Mode' : 'indexed';
'Colour Value Extent' : [→ [→ 0 → 0 → 0 →] → [→ 255 → 255 → 255 →] →];
'Index Precision' : 16; 'Integer Precision' : 16;
'Maximum Colour Index' : 63; 'Real Precision' : 'fixed point format, 16, 16';
'VDC Integer Precision' : 16; 'VDC Real Precision' : 'fixed point format, 16, 16';
'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   ∀ cst, doby, prof, rdoby
2   (0 LINE_RENDITION_VALUE(cst, doby, prof, rdoby) =
3     IF IsAttributeSet^2.62(cst) and 'line rendition' ∈ NAMS^1.18(cst) and
4       not IsPlaceholder^1.19(C ~cst • 'line rendition')
5     THEN C ~cst • 'line rendition'
6     ELSE STEP_BP_VALUE^9.3(cst, doby, 'line rendition') ORELSE^9.1
7     (1 IF IsBasicLayoutObjectDescription^2.54(cst)
8       THEN STEP_C_VALUE^9.6(cst, doby, 'line rendition') ORELSE^9.1
9       (2 STEP_DP_VALUE^9.8(cst, doby, 'line rendition') ORELSE^9.1
10        (3 IF IsBasicLayoutObjectDescription^2.54(cst)
11          THEN STEP_E_VALUE^9.10(cst, doby, rdoby, prof, 'line rendition') ORELSE^9.1
12          (4 STEP_FP_VALUE^9.13(cst, doby, rdoby, prof, 'line rendition') ORELSE^9.1
13            (5 STEP_G_VALUE^9.17(cst, doby, rdoby, prof, 'line rendition', OBJECT_TYPE_VALUE^9.61(cst, doby))
14              ORELSE^9.1 (6 STEP_H_VALUE^9.22(prof, 'line rendition') ORELSE^9.1
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_DEFAULT^9.132(cst, doby, prof, rdoby);
19               'Line Type' : 1;
20               'Line Width' : LINE_WIDTH_DEFAULT^9.133(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   ∀ cst, doby, prof, rdoby
2   (0 COLOUR_DEFAULT(cst, doby, prof, rdoby) =
3     IF C ~GEOMETRIC_GRAPHICS_ENCODING_ANNOUNCER_VALUE^9.130(cst, doby, prof.rdoby) •
4           'Colour Selection Mode' = 'direct'
5     THEN 'foreground' ELSE 1 0)
```

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 $({}_0\,\text{LINE\_WIDTH\_DEFAULT}(cst, doby, prof, rdoby) =$
3 IF $\text{LINE\_WIDTH\_SPECIFICATION\_MODE}^{9.134}(cst, doby, prof, rdoby)$ = 'scaled'
4 THEN 1.0 ELSE 0.001 * $\text{LONGEST\_VDC\_EXTENT}^{9.135}(cst, doby, prof, rdoby)_0)$

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 $({}_0\,\text{LINE\_WIDTH\_SPECIFICATION\_MODE}(cst, doby, prof, rdoby) =$
3 C ^$\text{LINE\_RENDITION\_VALUE}^{9.131}(cst, doby, prof, rdoby)$.'Line Width Specification Mode'$_0$)

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 $({}_0\,\text{LONGEST\_VDC\_EXTENT}(cst, doby, prof, rdoby) =$
3 IF $({}_1\,\text{HEIGHT}^{9.137}$(C ^$\text{TRANSFORMATION\_SPECIFICATION\_VALUE}^{9.155}(cst, doby, prof.rdoby)$.
4 'VDC Extent') >
5 $\text{WIDTH}^{9.136}$(C ^$\text{TRANSFORMATION\_SPECIFICATION\_VALUE}^{9.155}(cst, doby, prof.rdoby)$.
6 'VDC Extent')$_1$)
7 THEN $\text{HEIGHT}^{9.137}$(C ^$\text{TRANSFORMATION\_SPECIFICATION\_VALUE}^{9.155}(cst, doby, prof.rdoby)$.
8 'VDC Extent')
9 ELSE $\text{WIDTH}^{9.136}$(C ^$\text{TRANSFORMATION\_SPECIFICATION\_VALUE}^{9.155}(cst, doby, prof.rdoby)$.
10 'VDC Extent')$_0$)

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 $({}_{0}\mathrm{WIDTH}(v) =$
3 $\mathrm{HEAD}^{1.13}(\mathrm{LASTC}^{1.14}(v)) - \mathrm{HEAD}^{1.13}(\mathrm{HEAD}^{1.13}(v))_{0})$

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 $({}_{0}\mathrm{HEIGHT}(v) =$
3 $\mathrm{LASTC}^{1.14}(\mathrm{LASTC}^{1.14}(v)) - \mathrm{LASTC}^{1.14}(\mathrm{HEAD}^{1.13}(v))_{0})$

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   ∀ cst, doby, prof, rdoby
2   (0 MARKER_RENDITION_VALUE(cst, doby, prof, rdoby) =
3     IF IsAttributeSet^2.62(cst) and 'marker rendition' ∈ NAMS^1.18(cst) and
4       not IsPlaceholder^1.19(C ^cst • 'marker rendition')
5     THEN C ^cst • 'marker rendition'
6     ELSE STEP_BP_VALUE^9.3(cst, doby, 'marker rendition') ORELSE^9.1
7     (1 IF IsBasicLayoutObjectDescription^2.54(cst)
8       THEN STEP_C_VALUE^9.6(cst, doby, 'marker rendition') ORELSE^9.1
9       (2 STEP_DP_VALUE^9.8(cst, doby, 'marker rendition') ORELSE^9.1
10        (3 IF IsBasicLayoutObjectDescription^2.54(cst)
11          THEN STEP_E_VALUE^9.10(cst, doby, rdoby, prof, 'marker rendition') ORELSE^9.1
12          (4 STEP_FP_VALUE^9.13(cst, doby, rdoby, prof, 'marker rendition') ORELSE^9.1
13            (5 STEP_G_VALUE^9.17(cst, doby, rdoby, prof, 'marker rendition',
14                                 OBJECT_TYPE_VALUE^9.61(cst, doby))
15              ORELSE^9.1(6 STEP_H_VALUE^9.22(prof, 'marker rendition') ORELSE^9.1
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_DEFAULT^9.132(cst, doby, prof, rdoby);
20               'Marker Size' : MARKER_SIZE_DEFAULT^9.139(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   ∀ cst, doby, prof, rdoby
2   (0 MARKER_SIZE_DEFAULT(cst, doby, prof, rdoby) =
3     IF MARKER_SIZE_SPECIFICATION_MODE^9.140(cst, doby, prof, rdoby) = 'scaled'
4     THEN 1.0 ELSE 0.01 * LONGEST_VDC_EXTENT^9.135(cst, doby, prof, rdoby)0)
```