INTERNATIONAL STANDARD

ISO/IEC 8613-10

Second edition
1995-10-15

Information technology — Open Document Architecture (ODA) and Interchange Format —

Part 10: Formal specifications

Technologies de l'information — Architecture de document ouverte (ODA) et format de transfert —

Partie 10: Spécifications formelles

Reference number
ISO/IEC 8613-10:1995(E)

--`,,```,,,,````-`-`,,`,,`,`,,`---
Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Contents

Page

Foreword iii
1 Scope 1
2 Normative references 2
3 Definitions 2
4 Syntax and semantics of the specification language 3
4.1 Basic concepts 3
4.2 Syntax of the specification language 4
4.3 Predicate symbols with built-in semantics 7
4.4 Operator symbols with built-in semantics 7
4.5 Other terms 8
4.6 Notational simplifications 9
5 Structure of the formal specifications 11
6 Commonly used definitions 13
7 Formal specification of the document structures 20
7.1 Sets of constituents 24
7.2 Constituents 35
7.3 Attributes 61
7.4 Subsidiary definitions 87
7.4.1 Predicates 87
7.4.2 Functions 95
7.5 Additional terminological definitions 97
7.6 Index of predicate symbols, operator symbols and attribute names 100
Annex A Tutorial on the specification language 107
A.1 Introduction 107
A.2 Atomic constructs 107
A.3 Composite constructs 108
A.4 Spots 109
A.5 Predicates 109
A.6 Operators 110
A.7 Predicates for the formal specifications of ISO 8613 113
A.8 Further examples 113

© ISO/IEC 1991
All rights reserved. No part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from the publisher.

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

Printed in Switzerland

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

11 Formal specification of the geometric graphics content architectures . . . 198
11.1 Interface to the Document Profile . . . 200
11.2 Interface to the Document Architecture . . . 201
11.3 Attributes of the Geometric Graphics Content Architecture . . . 203
12 Formal specification of the defaulting mechanism for defaultable attributes . . . 228
12.1 General functions . . . 228
12.2 Functions for the values of defaultable attributes . . . 245
12.2.1 Defaultable attributes of ISO/IEC 8613-2 . . . 245
12.2.2 Defaultable attributes of ISO/IEC 8613-6 . . . 274
12.2.3 Defaultable attributes of ISO/IEC 8613-7 . . . 298
12.2.4 Defaultable attributes of ISO/IEC 8613-8 . . . 310
13 Index of predicate symbols, operator bols and attribute names . . . 327
Annex A: Tutorial on the specification lang . . . 354
A.1 Introduction . . . 354
A.2 Atomic constructs . . . 354
A.3 Composite constructs . . . 355
A.4 Spots . . . 356
A.5 Predicates . . . 356
A.6 Operators . . . 357
A.7 Predicates for the formal specifications of ISO/IEC 8613 . . . 360
A.8 Further examples . . . 360

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

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 pa ar fields of technical activity. ISO and IEC technical committees collaborat elds of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work.

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

International Standard ISO/IEC 8613-10 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology,* Subcommittee SC 18, *Document processing and related communication.*

This second edition is a revision of the first edition (ISO/IEC 8613-10:1991). It also incorporates amendment 1:1991, amendment 2:1991, amendment 3:1992, amendment 4:1992 and amendment 5:1993.

ISO/IEC 8613 consists of the following parts, under the general title *Information technology — Open Document Architecture (ODA) and Interchange Format*:

- *Part 1: Introduction and general principles*
- *Part 2: Document structures*
- *Part 3: Abstract interface for the manipulation of ODA documents*
- *Part 4: Document profile*
- *Part 5: Open Document Interchange Format*
- *Part 6: Character content architectures*
- *Part 7: Raster graphics content architectures*
- *Part 8: Geometric graphics content architectures*
- *Part 9: Audio content architectures*
- *Part 10: Formal specifications*
- *Part 11: Tabular structures and tabular layout*
- *Part 12: Identification of document fragments*
- *Part 13: Spreadsheet*
- *Part 14: Temporal relationships and non-linear structures*

Annex A of this part of ISO/IEC 8613 is for information only.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Information technology — Open Document Architecture (ODA) and Interchange Format —

Part 10:
Formal specifications

1 Scope

The purpose of ISO/IEC 8613 is to facilitate the interchange of documents.

In the context of ISO/IEC 8613, documents are considered to be items such as memoranda, letters, invoices. forms and reports, which may include pictures and tabular material. The content elements used within the documents may include graphic characters, geometric graphics elements and raster graphics elements, all potentially within one document.

NOTE — ISO/IEC 8613 is designed to allow for extensions, including hypermedia features, spreadsheets and additional types of content such as audio and video.

In addition to the content types defined in this International Standard, ODA also provides for arbitrary content types to be included in documents.

ISO/IEC 8613 applies to the interchange of documents by means of data communications or the exchange of storage media.

It provides for the interchange of documents for either or both of the following purposes:

- to allow presentation as intended by the originator;
- to allow processing such as editing and reformatting.

The composition of a document in interchange can take several forms:

- formatted form, allowing presentation of the document;
- processable form, allowing processing of the document;
- formatted processable form, allowing both presentation and processing.

ISO/IEC 8613 also provides for the interchange of ODA information structures used for the processing of interchanged documents.

This part of ISO/IEC 8613

- specifies a formal description technique appropriate for describing the technical specifications of the document structures (ISO/IEC 8613-2), the document profile (ISO/IEC 8613-4) and the content architectures (currently ISO/IEC 8613-6, -7 and -8);
- gives formal specifications of the document structures, the document profile and the content architectures using this formal description technique.

The aim of the formal specifications of ODA (FODA) is to provide a precise and unambiguous interpretation of the technical specifications in other parts of ISO/IEC 8613 (currently parts 2, 4, 6, 7, and 8), using formal syntax and formal semantics.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

FODA can be used

— as a basis for implementations of ISO/IEC 8613;

— as a validation tool for the verification of conforming systems;

— as a reference point for examining future extensions and revisions to ISO/IEC 8613.

If a discrepancy between the natural language text and the formal specifications should be discovered, the natural language text should be regarded as the valid interpretation of this International Standard until the discrepancy is resolved.

2 Normative references

The following standards contain provisions which, through reference in this text, constitute provisions of this part of ISO/IEC 8613. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this part of ISO/IEC 8613 are encouraged to investigate the possibility of applying the most recent editions of the standards indicated below. Members of IEC and ISO maintain registers of currently valid International Standards.

ISO/IEC 2022:1994, *Information technology — Character code structure and extension techniques.*

ISO/IEC 6429:1992, *Information technology — Control functions for coded character sets.*

ISO/IEC 6937:1994, *Information technology — Coded graphic character set for text communication — Latin alphabet.*

ISO 8601:1988, *Data elements and interchange formats — Information interchange — Representation of dates and times.*

ISO/IEC 8613-1:1994, *Information technology — Open Document Architecture (ODA) and Interchange format: Introduction and general principles.*

ISO/IEC 8613-2:1995, *Information technology — Open Document Architecture (ODA) and Interchange format: Document structures.*

ISO/IEC 8613-4:1994, *Information technology — Open Document Architecture (ODA) and Interchange format: Document profile.*

ISO/IEC 8613-6:1994, *Information technology — Open Document Architecture (ODA) and Interchange format: Character content architectures.*

ISO/IEC 8613-7:1994, *Information technology — Open Document Architecture (ODA) and Interchange format: Raster graphics content architectures.*

ISO/IEC 8613-8:1994, *Information technology — Open Document Architecture (ODA) and Interchange format: Geometric graphics content architectures.*

ISO/IEC 8632-1:1992, *Information technology — Computer graphics — Metafile for the storage and transfer of picture description information — Part 1: Functional specification.*

ISO/IEC 8632-3:1992, *Information technology — Computer graphics — Metafile for the storage and transfer of picture description information — Part 3: Binary encoding.*

ISO/IEC 9541-2:1991, *Information technology — Font information interchange — Part 2: Interchange format.*

3 Definitions

For the purposes of this part of ISO/IEC 8613, the definitions given in ISO/IEC 8613-1 apply.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

4 Syntax and semantics of the specification language

This clause describes the formal description technique used for the formal specifications.

NOTE — A tutorial on this formal description technique is given in annex A.

4.1 Basic concepts

ISO/IEC 8613 describes document structures, the document profile and the content architectures in terms of abstract information constructs which are drawn from the following structural categories:

— An ODA construct may be an **atomic** construct, e.g. an attribute name or a natural number within an object identifier.

— An ODA construct may be a **composite** construct, i.e. may consist of other constructs. With respect to their interrelationship, three kinds of composition are distinguished. An ODA construct may be

a) a **set**;
b) a **function** (mapping);
c) a **sequence** (list, string);

of other ODA constructs.

For example, a specific layout description is a set (of constituents), a constituent is a nomination (see below) which is a function or mapping (from attribute names onto attribute values), and an attribute value of 'subordinates' or of 'object identifier' is a sequence (of atomic natural numbers).

It is these very structures which are captured by the language used for the formal specifications of ISO/IEC 8613. The language used is called IMCL, *Information Modelling by Composition Language*. The semantics of the specification language consist of the following abstract elements:

— the universe which is a non-empty set of **entities** of the following kinds:

a) constructs;
b) spots;
c) spotsets (i.e. sets of spots);
d) the entity UNDEF ("undefined");

— functions from the universe to the universe, that is, operators on entities of the universe;

— predicates in the universe, that is, predicates on entities of the universe.

A **construct** is an information object which is one of the following:

— an atomic construct or **atom**, for short;
— a composite construct or **compound**, for short, which may be
a) a **collection**, which is an unordered set of component constructs;
b) a **nomination**, which is a function that can be regarded as an unordered set of ordered pairs where each pair consists of a *name* and a *value*;
c) a **catenation**, which is a sequence of component constructs.

The special terminology for composite constructs is to distinguish them from other sets, functions or sequences.

In order to be able to address components in constructs of arbitrary compositional structure, the concept of a **spot** is introduced. This concept is an abstract counterpart for the intuitive idea associated with pointing into an information structure at some position and saying "here". However, in general the "here" is not identified uniquely by the component construct as such (e.g. in a word, the same letter may occur several times), but rather by the context in which it appears. To deal conceptually with the idea of "here" requires a way to identify contexts.

The concept of a spot allows the distinction to be made between a considered construct and its position within a comprising composite construct of which it is a component. For example, the character string "data" (a catenation) has the component constructs 'd', 'a' and 't'. Whereas 'd' and 't' appear at one spot each, the 'a' appears at two spots, namely at the second and at the fourth position counted from the front end. So, "data" has four

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

component spots, but only three component constructs. If a construct is considered outside any context, it is said to be at its **ownspot**.

Spots are usually selected by selection criteria. However, a selection criterion need not be unique. Thus, the objects most naturally dealt with are not even spots, but rather sets of spots or **spotsets**. Consequently, there are no expressions for single spots, but for singleton spotsets, instead, i.e. for spotsets with only one spot (see $\in$ in 4.3 and ^ in 4.4).

It should be noted that the specification language is built on first-order predicate logic and mathematical set theory.

4.2 Syntax of the specification language

This subclause defines the syntax of the specification language, i.e., each expression in the formal specifications is built using the syntax rules given in this clause. The semantics of the terminal symbols appearing in the syntax rules are specified in 4.3 to 4.5.

Remarks on the meta-language:

The symbol pairs { }, [] and −. .− as well as the symbols : : = , | , ... , and ≡ belong to the meta-language. They have the following meanings:

: : = separates the meta-variable to be defined (left-hand side) from the meta-language expression which defines it (right-hand side)

{ } delimit a syntactical unit

[ ] delimit a syntactical unit and indicate that this syntactical unit is optional, i.e. may also be absent
NOTE 1: The meta-language symbols [and] are different from the special-characters [and] used in the production rules for empty-constant, explicit-composition-term and extensional-collection-term.

−. .− delimit a comment in the meta-language text

| separates alternative syntactical units, i.e. indicates a choice of exactly one of the syntactical units
e.g. $\{a \mid m \mid p\}x$ means
ax or mx or px
NOTE 2: The meta-language symbol | is different from the special-character | used in terms denoting sets.

... is a convenient notation for recursive definitions: the symbol follows a syntactical unit which may appear one or more times, i.e. which may be repeated several times
e.g. $\{yf\}$... means
yf or $yfyf$ or $yfyfyf$ etc.
e.g. $i[so]$... means
i or iso or $isoso$ etc.

≡ A space in the language definition requires one or more blanks in the defined language expression. Just the reverse is indicated by the symbol ≡, which requires an immediate juxtaposition of the neighboring strings of the specification language. Where syntactical uniqueness is not affected, blanks may be omitted in expressions of the specification language (e.g. before and after parentheses).

For the sake of readability, the symbols "(" and ")" are used as meta-variables (instead of word-symbols such as "left-del" or "right-del"). All other meta-variables are strings of lower-case letters (with the hyphen for linking the components of a meta-variable).

Production rules:

expression : : =
formula | term

formula : : =
prime-formula |
not formula | formula { *and* | *or* | *impl* | *iff* | *xor* } formula |
∃ var (formula) | ∀ var (formula) |
(formula)

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

NOTE 3: The terminal symbols used in this production rule have the usual semantics of first-order predicate logic: *not* is the logical negation, *and*, *or*, *xor* (exclusive or), *impl* (implies) and *iff* (if and only if) are the usual logical connectors, ∀ (for all) and ∃ (exists) are the logical quantifiers.

prime-formula : : =
[parameter-part] predicate-symbol-part ...
[parameter-part predicate-symbol-part ...] ...[parameter-part]

predicate-symbol-part : : =
upper-case-letter [≡ letter |≡ digit]... ≡ lower-case-letter [≡ letter |≡ digit]... |
= |≠|<|≤|>|≥|∈|∉|∈̂|⊂|⊆|⊃|⊇

NOTE 4: The semantics of the terminal symbols (=, ≠, ... ⊇) in this production rule are specified in 4.3.

term : : =
var |
constant |
operator-term |
explicit-composition-term |
conditional-term |
extensional-collection-term |
extensional-spotset-term |
spot-selection-term |
(term)

var : : =
lower-case-letter [≡ letter |≡ digit]... [≡ subscript-digit]...

constant : : =
standard-constant |
nonstandard-constant

standard-constant : : =
UNDEF |
empty-constant |
number-atom-constant

empty-constant : : =
[ ] −. empty collection .− |
[:] −. empty nomination .− |
[→] −. empty catenation .− |
< > −. empty spotset .−

number-atom-constant : : =
[+ ≡| − ≡] digit [≡ digit]... [≡ . ≡ digit [≡ digit]...]

nonstandard-constant : : =
≡ character [≡ character]... ≡ ' −. restriction on apostrophe occurrence .−

operator-term : : =
[parameter-part] operator-symbol-part ...
[parameter-part operator-symbol-part ...]...[parameter-part]

operator-symbol-part : : =
upper-case-letter [≡ upper-case-letter |≡ digit |≡ _]... |
^ | + | − | * | / | ∪ | ∩ | \ | // | • | • |↓|↑

NOTE 5: The semantics of the terminal symbols (^, +, ... ↑) in this production rule are specified in 4.4.

explicit-composition-term : : =
[term [; term]...] −. collection .− |
[term : term [; term : term]...] −. nomination .− |
[→ term [→ term]... →] −. catenation .− |
" ≡ character [≡ character]... ≡ " −. catenation of characters, restriction on quote occurrence .−

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

conditional-term : :=
IF formula THEN term ELSE term

NOTE 6: The semantics of the terminal symbols (IF , THEN and ELSE) in this production rule are specified in 4.5.

extensional-collection-term : :=
[var | formula] –. collection of constructs for which the formula holds.–

extensional-spotset-term : :=
< var | formula > –. union of singleton spotsets for which the formula holds.–

spot-selection-term : :=
term spot-selection-clause |
elliptic-spot-selection-term

spot-selection-clause : :=
<var || formula> |
<formula> –. *xs* is assumed for var .– |
<name-specification [, name-specification] ... >

elliptic-spot-selection-term : :=
term {• | * |↓|↑} name-specification

NOTE 7: The semantics of the terminal symbols (•, *, ↓ and ↑) in this production rule are specified in 4.5.

name-specification : :=
nonstandard-constant | var

parameter-part : :=
term | (term [, term]...)

character : :=
letter | digit | subscript-digit | special-character

letter : :=
upper-case-letter | lower-case-letter

upper-case-letter : :=
A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z

lower-case-letter : :=
a | b | c | d | e | f | g | h | i | j | k | l | m | n | o | p | q | r | s | t | u | v | w | x | y | z

digit : :=
0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

subscript-digit : :=
$_0$ |$_1$|$_2$|$_3$|$_4$|$_5$|$_6$|$_7$|$_8$|$_9$

special-character : :=
. | , | ; | + | - | –. etc. .–

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

4.3 Predicate symbols with built-in semantics

A sequence of predicate-symbol-parts is referred to as a **predicate symbol**. For each n-ary predicate symbol there is an n-ary predicate on the universe of the specification language, i.e. an n-ary relation on entities of the universe. Some predicate symbols have built-in semantics which are introduced by the following.

NOTE — The predicate-symbol-parts are syntactically distinguished from operator-symbol-parts and variables.

True	means	the valid fact (something stated as being true)
False	means	the invalid fact (something stated as being false)
IsAtom(t)	means	t is an atomic construct or atom, for short
IsNat(t)	means	t is a natural number (1, 2, ...; zero excluded)
IsInt(t)	means	t is an integer number (... -2, -1, 0, 1, 2, ...)
IsReal(t)	means	t is a real number
IsCol(t)	means	t is a collection
IsNom(t)	means	t is a nomination
IsCat(t)	means	t is a catenation
IsSpotset(t)	means	t is a spotset
IsSingle(t)	means	t is a singleton spotset
$t_1 = t_2$	means	t_1 is equal to t_2 (all entities)
$t_1 \neq t_2$	means	*not* $t_1 = t_2$
$t_1 < t_2$	means	t_1 is less than t_2 (numbers)
$t_1 \leq t_2$	means	t_1 is less than or equal to t_2
$t_1 > t_2$	means	t_1 is greater than t_2
$t_1 \geq t_2$	means	t_1 is greater than or equal to t_2
$t_1 \in t_2$	means	t_1 is element of t_2 (collections)
$t_1 \notin t_2$	means	*not* $t_1 \in t_2$
$t_1 \in t_2$	means	t_1 is singleton spotset and subset of t_2 (spotsets)
$t_1 \subset t_2$	means	t_1 is subset of t_2 (collections or spotsets)
$t_1 \subseteq t_2$	means	t_1 is subset of or equal to t_2 (collections or spotsets)
$t_1 \supset t_2$	means	t_2 is subset of t_1 (collections or spotsets)
$t_1 \supseteq t_2$	means	t_2 is subset of or equal to t_1 (collections or spotsets)

The unary predicate symbols mean predicates for expressing that an entity belongs to a certain class or "type" of entities, i.e. has a particular property. The binary predicate symbols refer to predicates which indicate whether or not a particular relationship holds for two entities.

4.4 Operator symbols with built-in semantics

A sequence of operator-symbol-parts is referred to as an **operator symbol**. For each n-ary operator symbol there is an n-ary operator or function from the universe to the universe of the specification language, i.e., a mapping from n-ary tuples of entities onto entities of the universe. Some operator symbols have built-in semantics which are introduced by the following.

NOTE — The operator-symbol-parts are syntactically distinguished from predicate-symbol-parts and variables. For all operators it holds that the result is UNDEF, if a parameter term does not meet the requirement stated below.

C t	If t denotes a singleton spotset, C t denotes the component construct at the spot given by t.
N t	If t denotes a singleton spotset of a spot that is a component of a nomination ("immediately inward" of a nomination is the formal term), then N t denotes the name construct of the component as it is within the nomination.
F t	If t denotes a set of exactly one spot immediately inward of a catenation spot, F t denotes the front part of this catenation up to but excluding the component given by t (catenation of components with lower position than t).

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

CARD t	If t denotes a collection or a spotset, CARD t denotes its cardinality, i.e., the number of component constructs or the number of spots.
R t	If t denotes a set of exactly one spot immediately inward of a catenation spot, R t denotes the rear part of this catenation up to but excluding the component given by t (catenation of components with higher position than t).
$t_1 + t_2$ $t_1 - t_2$ $t_1 * t_2$ t_1/t_2	If t_1 and t_2 denote numbers, the terms denote numbers as known from arithmetic. The usual precedence rules for arithmetical operators apply.
$t_1 \cup t_2$ $t_1 \cap t_2$ $t_1 \setminus t_2$	If t_1 and t_2 denote either collections or spotsets, the terms denote their set-theoretic union, intersection or difference, i.e. either collections or spotsets.
$t_1 // t_2$	If t_1 and t_2 denote catenations, $t_1 // t_2$ denotes the catenation obtained by concatenating the two catenations in the given order.
$\hat{}t$	If t denotes a construct, then $\hat{}t$ ("ownspot of t") denotes the singleton spotset containing the ownspot of t.
$t\bullet$	If t denotes a spotset containing no atom spots (spots with atoms), then $t\bullet$ ("next inwards") denotes the set of all spots which are immediately inward of the spots of the spotset t.
$t\downarrow$	If t denotes a spotset, $t\downarrow$ denotes the set of all terminal spots inward of or - for atoms and empty constructs - equal to the spots given by t (read "most inward").
$t^\bullet$	If t denotes a spotset without ownspots, $t^\bullet$ denotes the set of all spots which are immediately outward of the spots given by t (read "next outward").
$t\uparrow$	If t denotes a spotset, $t\uparrow$ denotes the set of all ownspots outward of or - for ownspots - equal to the spots given by t (read "most outward").
$t\bullet$ $t\downarrow$ $t^\bullet$ $t\uparrow$	If t denotes the empty spotset, the operator-terms denote the empty spotset, too.

The normal evaluation order for expressions is from left to right, with the following exceptions:

— If a term is enclosed in opening and closing parentheses this term is evaluated first;
— Operators have precedence over predicates;
— Between operators the precedence order is:

1.: ^ (ownspot of)
2.: • ° ↓ ↑ (next inward, next outward, most inward, most outward)
3.: spot-selection-clause
4.: spot-selection-term
5.: other operators

4.5 Other terms

Apart from operator-terms there are other compound terms which result in constructs or spotsets (or UNDEF). Their built-in semantics are introduced by the following.

$[t_1; t_2; t_3]$	If t_i denote constructs, the whole term denotes the collection which contains the constructs t_i as components. (This is an example for explicit-composition-term)
[ ]	Denotes the empty collection. (This is an example for empty-constant)
$[n_1 : c_1; n_2 : c_2]$	If n_i and c_i denote constructs, where all n_i are distinct, the whole term denotes the nomination which contains the constructs c_i as components under the (unique) names n_i. (This is an example for explicit-composition-term)
[:]	Denotes the empty nomination. (This is an example for empty-constant)

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

[→ m_1 → m_2 → m_3 →]	If m_i denote constructs, the whole term denotes the catenation which contains the constructs m_i as components — also referred to as members — in the indicated sequence. (This is an example for explicit-composition-term)
"ODA Part 2"	Denotes the catenation [→ 'O' → 'D' → 'A' → ' ' → 'P' → 'a' → 'r' → 't' → ' ' → '2' →]. A string of characters enclosed in quotes denotes the catenation of those characters. A pair of quotes in the string stands for a single one in the catenation. (This is an example for explicit-composition-term)
[—]	Denotes the empty catenation. (This is an example for empty-constant)
< >	Denotes the empty spotset. (This is an example for empty-constant)
IF formula THEN t_1 ELSE t_2	If t_1 and t_2 are terms, the whole term denotes the same as t_1 or t_2, depending on whether the formula is True or False, respectively. (This is an example for conditional-term)
[*var* | formula]	Denotes the collection of all constructs *var* which satisfy the formula. (This is an example for extensional-collection-term)
< *var* | formula >	Denotes the spotset which is the union of all singleton spotsets *var* which satisfy the formula. (This is an example for extensional-spotset-term)
t <*var* || formula >	If t denotes a (possibly empty) spotset, the whole term denotes the union of all singleton spotsets *var* which contain a spot taken from t and for which the formula is True. (This is an example for spot-selection-term)
	Three elliptic notations are provided for frequently occurring spot-selection-clauses:
t < formula >	If a variable *var* is not introduced explicitly, the abbreviated term t < formula > is evaluated for the standard variable *xs* (singleton set of the spot under examination or examination spot, for short). (This is an example for spot-selection-term)
t <$n_1, n_2, ...$>	If the formula has the structure N $var = n_1$ *or* N $var = n_2$ *or* ... where n_i are name-specifications, the formula may be abbreviated as a list of name-specifications. (This is an example for spot-selection-term)
$t \bullet n$ $t \downarrow n$ $t^{\bullet} n$ $t \uparrow n$	If there is only one name-specification n used for spot selection, an elliptic-spot-selection-term is provided as an abbreviation of special spot-selection-terms (ending with •, • etc.). The n stands for <N *xs* = n> (see name qualification in programming languages). (This is an example for elliptic-spot-selection-term)

4.6 Notational simplifications

The common notational simplifications for successive logical quantifications can be used. The following examples explain these "short-hand" notations which are usually applied in first-order predicate logic:

The expression	$\forall x(\forall y(\exists z(\text{formula})))$
may be written as	$\forall x \forall y \exists z$ (formula)
or even as	$\forall x, y \exists z$ (formula)

A further abbreviation is used to help emphasize the "essential part" of a quantified formula:

The expression	$\forall x(x \in m$ *impl* formula)
may be written as	$\forall x \in m$ (formula)
and	$\exists x(x \in m$ *and* formula)
may be written as	$\exists x \in m$ (formula)

This notation can be combined with the previous one:

The expression	$\forall x(x \in m$ *impl* $\forall y(\exists z(z \in p$ *and* formula)))

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

may be written as $\forall x \in m,\ y\ \exists z \in p$ (formula)

However, note that the expression

$\forall y, x \in m\ \exists z \in m$ (formula)

includes the restriction $y \in$ m and has therefore to be expanded to

$\forall y(y \in m\ \underline{impl}\ \forall x(x \in m\ \underline{impl}\ \exists z \in m$ (formula)))

The same abbreviations are used for $\hat{\in}$.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

5 Structure of the formal specifications

This clause outlines the general concepts for the formal specifications. Those terms which are used at several places throughout the formal specifications are contained in clause 6.

Formally speaking, the formal specifications consist of a single formula in first-order predicate logic. This formula is called the overall formula. The formula consists of other formulae which are connected by *and*:

$$\text{formula}_1 \ \underline{and} \ \text{formula}_2 \ \underline{and} \ \text{formula}_3 \ \underline{and} \ \ldots \ \text{formula}_n$$

This overall formula is distributed over clauses 7 to 11. Its outline is as follows:

... IsInterchangeSet$^{2.1}(doc)$ ...
and ... IsGenericDocumentDescription$^{2.2}(gdoc)$ ...

(see clause 7)

and ... (*cont*)DescribesContPortOf$^{2.153}(b)$

and ... IsDocumentProfile$^{4.1}(cst)$ ...

(see clause 8)

and ... IsCIEColour$^{4.96}(v)$ ...

and ... IsDefaultableCharacterContentArchitectureAttribute$^{6.1}(att)$ ...

(see clause 9)

and ... IsGraphicCharacter$^{6.113}(v)$ ...

and ... IsDefaultableRasterGraphicsContentArchitectureAttribute$^{7.1}(att)$ ...

(see clause 10)

and ... IsInterleavingFormatValue$^{7.32}(v)$ ...
... IsDefaultableGeometricGraphicsContentArchitectureAttribute$^{8.1}(att)$ ...

(see clause 11)

and ... IsRegisteredEdgeType$^{8.61}(v)$ ...

The complete part of each section of the overall formula is given in the indicated clauses. The overall formula has been split according to the individual parts of ISO/IEC 8613.

In the present context, each of these formulae is called a "definition" and identified through a unique reference number. A definition defines either a concept used in the narrative part of ISO/IEC 8613 or a concept which has a subsidiary function in the network of definitions in that it has been separated and "encapsulated" to render other definitions more readable.

The definitions are grouped into several clauses and subclauses. For example, within the formal specifications of the document structures the definitions relating to the sets of constituents are contained in 7.1, those relating to constituents in 7.2 and those relating to attributes in 7.3. In addition, concepts which are not used in the formal specification but are used in the text of ISO/IEC 8613-2 are defined in 7.5. The definitions relating to the document profile are given in clause 8, those relating to the character content architectures in clause 9, etc.

The "factorization" of definitions is only for the convenience of authors and readers; it does not in any way impair the formal rigidity of the approach.

Variables occurring in the definitions are always bound by universal ($\forall$) or existential ($\exists$) quantifiers. Therefore, once a value has been chosen for a variable it has to be retained throughout the scope of the quantifier wherever the variable appears.

All predicates apart from those pertaining to the specification language are defined using the same format. For unary predicates the format is:

$\forall$ *variable* (*predicate-symbol*(*variable*) *iff formula*)

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

A similar format is applied for n–ary predicates. There, the variables have been placed around the predicate symbol in a "natural" way, e.g. (*id*)IsIdContIn(*b*) is expected to be read as "*id* is an identifier of a content portion in the basic object *b*".

A "predicate" in first-order predicate logic is a propositional pattern where some places are left free for the insertion of individual entities. An example is the pattern "...is greater than ...". Once the free places have been occupied by entities, the pattern becomes a proposition for which the "truth value" can be evaluated. The proposition "5 is greater than 2" evaluates to True, whereas the propositions "3 is greater than 5" and "a Mercedes is greater than SC 18" evaluate to False. Note that a proposition may never evaluate to an undefined truth value.

A unary predicate is a pattern with only one free parameter place. Its definition is evaluated by substituting an individual entity for the parameter. This yields a proposition on the left-hand side of the *iff* and transforms the whole expression in parentheses into a proposition.

A uniform format has also been adopted for the definition of most operators. It relies on the conditional-term of the specification language:

$\forall$ *variable* (*operator-symbol*(*variable*) = IF *formula* THEN *term* ELSE UNDEF)

An "operator" is a term pattern where some places are left free for the insertion of individual entities. An example is the pattern "...plus ...". Once the free places have been occupied by entities, the pattern becomes a term, i.e. an expression which evaluates to or denotes an individual entity. The term "5 plus 2" denotes the entity 7, whereas the term "WG 3 plus SC 18" denotes the entity which is represented by UNDEF in the formal specification language. Therefore, a term evaluates to "undefined", if the operator is not defined for the inserted entities.

It should be noted that the formal specifications cannot be applied *directly* to the Open Document Interchange Format (ISO/IEC 8613-5) and the reader should not necessarily expect to see any direct correspondences. The formal specifications are based on the textual description of the concepts of ISO/IEC 8613 and do not necessarily reflect the ODIF encoding of ODA documents. Those clauses in ISO/IEC 8613 which specify the ODIF encoding usually impose additional rules for the make-up of the interchanged data stream. These additional rules are considered outside of the scope of the formal specifications: The ODIF encoding specifies these additional rules formally and there is therefore no need for repeating these additional rules in FODA.

Furthermore, although defaultable attributes or parameters are not required to be present in the interchanged data stream they are considered to be present for the formal specifications since their value, unlike that of non-mandatory attributes, must be derivable by the defaulting mechanism defined in ISO/IEC 8613-2, 9.1.2.4. The concept of defaultable attributes or parameters is formally covered by the predicate IsPlaceholder (see clause 6).

Throughout this formal specification the predicate "IsAtom(v)" is used whenever v is an entity defined in an International Standard other than ISO/IEC 8613. For example,

1 $\forall v$
2 ($_0$IsISO6937MSCharacter(v) *iff*
3 IsAtom(v)$_0$)

is used to indicate that the entity v, a character from the minimum subrepertoire of ISO 6937-2, is defined in ISO 6937. The definition of this entity is outside the scope of ISO/IEC 8613.

Where advantageous, *imported definitions* may be used in lieu of IsAtom(v). An imported definition is a function or a predicate whose semantics are defined by an International Standard other than ISO/IEC 8613. Within the scope of this part of ISO/IEC 8613 the semantics for imported definitions are expressed only by semiformal descriptions with references to the appropriate International Standards. The formal specification of these definitions is outside the scope of this part of ISO/IEC 8613.

The following hints are intended to help the reader understand the formal specifications. Note that there is no reading or evaluation sequence in a formula apart from that which results from applying the precedence rules defined for logical symbols and operator symbols (along with parentheses). The overall formulae are *not* intended to imply a sequence of definitions and need *not* be read in the order shown. Rather the order is intended to support a systematic approach to the specifications. The same applies for any definition which in turn consists of formulae.

Superscripts are used to indicate the number of the definition of a predicate symbol or operator symbol (e.g., IsISO6937MSCharacter$^{1.21}$(v)). A superscript does not appear if the symbol is part of the specification language

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

(i.e., is a predefined symbol) or is the one currently under definition. These superscripts (reference numbers) are not part of the specification language but are only for the convenience of the readers so they can quickly look up the definition of a referenced symbol.

Furthermore, line numbers and subscripted parentheses are used to structure the formulae and to establish cross references between the semi-formal descriptions and the formal definitions (see above). Again, these line numbers and subscripts are not part of the formal specifications but are only for the convenience of the readers.

Each definition of the formal specification is introduced by a so-called semi-formal description. These semi-formal descriptions give an explanation of the definitions to which they belong in English, making an extensive use of the terms of the specification method. The semi-formal descriptions do not belong to the formal specification itself, however, they are intended to help understand the formal definitions.

In the semi-formal descriptions, references to line numbers are put in parentheses, e.g., (8) or (11,12) or (4-10), whereas references to expressions between subscripted parentheses are prefixed by "expr.", e.g. (expr. 5).

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

6 Commonly used definitions

This clause contains those definitions which are not specific to the formal specification of the document structures, the document profile or one of the content architectures only.

Semiformal Description 1.1

Predicate "is a non-empty collection"

An entity *col* is a non-empty collection (set of constructs) if it is a collection which is not empty.

Definition 1.1

1 $\forall\, col$
2 $(_0\,\mathrm{IsNeCol}(col)$ *iff*
3 $\quad \mathrm{IsCol}(col)$ *and* $col \neq [\,]_0)$

Semiformal Description 1.2

Predicate "is a non-empty nomination"

An entity *nom* is a non-empty nomination (mapping of names onto constructs) if it is a nomination which is not empty.

Definition 1.2

1 $\forall\, nom$
2 $(_0\,\mathrm{IsNeNom}(nom)$ *iff*
3 $\quad \mathrm{IsNom}(nom)$ *and* $nom \neq [:]_0)$

Semiformal Description 1.3

Predicate "is a non-empty catenation"

An entity *cat* is a non-empty catenation (sequence of constructs) if it is a catenation which is not empty.

Definition 1.3

1 $\forall\, cat$
2 $(_0\,\mathrm{IsNeCat}(cat)$ *iff*
3 $\quad \mathrm{IsCat}(cat)$ *and* $cat \neq [\rightarrow]_0)$

Semiformal Description 1.4

Predicate "is an empty collection"

An entity *col* is an empty collection if it contains no components.

Definition 1.4

1 $\forall\, col$
2 $(_0\,\mathrm{IsEmptyCol}(col)$ *iff*
3 $\quad col = [\,]_0)$

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 1.5

Predicate "is an empty nomination"

An entity *nom* is an empty nomination if it contains no components.

Definition 1.5

1 $\forall\, nom$
2 $(_0$IsEmptyNom(nom) *iff*
3 $nom = [\,:\,]_0)$

Semiformal Description 1.6

Predicate "is an empty catenation"

An entity *cat* is an empty catenation if it contains no components.

Definition 1.6

1 $\forall\, cat$
2 $(_0$IsEmptyCat(cat) *iff*
3 $cat = [\,\rightarrow\,]_0)$

Semiformal Description 1.7

Predicate "is a non-negative integer"

An entity *int* is a non-negative integer if it is a natural number or zero.

Definition 1.7

1 $\forall\, int$
2 $(_0$IsNnInt(int) *iff*
3 IsNat(int) *or* $int = 0_0)$

Semiformal Description 1.8

Predicate "is a pair of non-negative integers"

A pair of non-negative integers is a catenation of two non-negative integers.

Definition 1.8

1 $\forall\, q$
2 $(_0$IsPairOfNnInt(q) *iff*
3 $\exists\, l, r$
4 $(_1\, q = [\,\rightarrow l \rightarrow r \rightarrow\,]$ *and* IsNnInt$^{1.7}(l)$ *and* IsNnInt$^{1.7}(r)_1)_0)$

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 1.9

Predicate "is a pair of positive integers"

A pair of positive integers is a catenation of two positive integers.

Definition 1.9

```
1   ∀q
2   (₀IsPairOfPosInt(q) iff
3     ∃l, r
4     (₁q = [ → l → r → ] and IsNat(l) and IsNat(r)₁)₀)
```

Semiformal Description 1.10

Predicate "is an octet string"

An octet string is an atomic construct in the formal specification.

Definition 1.10

```
1   ∀v
2   (₀IsOctetString(v) iff
3     IsAtom(v)₀)
```

Semiformal Description 1.11

Function "position from the front end"

PF returns the position number of the indicated component, the count beginning with the first component and ending at and including the considered component. The operand p is required to denote a set of exactly one spot immediately inward of a catenation.

Definition 1.11

```
1   ∀p
2   (₀PF(p) =
3     IF IsSingle(p) and IsCat(C p•)
4     THEN LENGTH^1.16 (F p) + 1
5     ELSE UNDEF₀)
```

Semiformal Description 1.12

Function "position from the rear end"

PR returns the position number of the indicated component, the count beginning with the last component and ending at and including the considered component. The operand p is required to denote a set of exactly one spot immediately inward of a catenation.

Definition 1.12

```
1   ∀p
2   (₀PR(p) =
3     IF IsSingle(p) and IsCat(C p•)
4     THEN LENGTH^1.16 (R p) + 1
5     ELSE UNDEF₀)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 1.13

Function "first component"

HEAD returns the first component of a non-empty catenation *cat*.

Definition 1.13

```
1  ∀ cat
2  (0 HEAD(cat) =
3    IF IsNeCat1.3(cat)
4    THEN C (^cat • <xs || PF1.11 xs = 1>)
5    ELSE UNDEF0)
```

Semiformal Description 1.14

Function "last component"

LASTC returns the last component of a catenation *cat*.

Definition 1.14

```
1  ∀ cat
2  (0 LASTC (cat) =
3    IF IsNeCat1.3(cat)
4    THEN C (^cat • <xs || PR1.12 xs = 1>)
5    ELSE UNDEF0)
```

Semiformal Description 1.15

Function "tail component"

TAIL returns the rear part of a non-empty catenation *cat* where the first component has been removed.

Definition 1.15

```
1  ∀ cat
2  (0 TAIL(cat) =
3    IF IsNeCat1.3(cat)
4    THEN R (^cat • <xs || PF1.11 xs = 1>)
5    ELSE UNDEF0)
```

Semiformal Description 1.16

Function "length of a catenation"

LENGTH returns the number of component spots of the catenation *cat*.

Definition 1.16

```
1  ∀ cat
2  (0 LENGTH (cat) =
3    IF IsCat(cat)
4    THEN CARD (^cat •)
5    ELSE UNDEF0)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Semiformal Description 1.17

Function "collection of component constructs"

COLC returns the collection of the component constructs at the spots of the (possibly empty) spotset *ss*.

Definition 1.17

1 $\forall ss$
2 $({}_0\text{COLC}\ (ss) =$
3 IF IsSpotset(ss)
4 THEN $[g \mid \exists p \hat{\in} ss\ (g = \text{C}\ p)]$
5 ELSE $\text{UNDEF}_0)$

Semiformal Description 1.18

Function "name set of a nomination"

NAMS returns the name set of the given nomination *n*, i.e. the collection of names of the components of the nomination.

Definition 1.18

1 $\forall n$
2 $({}_0\text{NAMS}(n) =$
3 IF IsNom(n)
4 THEN $[m \mid \exists p \hat{\in} \hat{}n \bullet (m = \text{N}\ p)]$
5 ELSE $\text{UNDEF}_0)$

Semiformal Description 1.19

Predicate "is the placeholder for any attribute value"

An entity *v* is the placeholder for the value of a defaultable attribute if it is the atom with this particular interpretation.

NOTE — The term 'placeholder for any attribute value' is not an ISO/IEC 8613 attribute value but a construct which is introduced to achieve consistency throughout the formal specifications and to distinguish between mandatory and defaultable attributes or parameters. The actual value is dependent on the respective attribute and the defaulting rules and will be determined during the processing of a document (editing process, layout process, imaging process) whenever a value for a defaultable attribute or parameter is needed.

Definition 1.19

1 $\forall v$
2 $({}_0\text{IsPlaceholder}(v)$ *iff*
3 $v =$ 'placeholder for any attribute value'$_0)$

Semiformal Description 1.20

Predicate "is an ISO 6937 minimal subrepertoire string"

An ISO 6937 minimal subrepertoire string is a catenation of ISO 6937 minimal subrepertoire characters.

Definition 1.20

1 $\forall v$
2 $({}_0\text{IsISO6937MSString}(v)$ *iff*
3 $\text{IsNeCat}^{1.3}(v)$ *and* $\forall m \hat{\in} \hat{}v \bullet (\text{IsISO6937MSCharacter}^{1.21}(\text{C}\ m))_0)$

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 1.21

Predicate "is an ISO 6937 minimal subrepertoire character"

An ISO 6937 minimal subrepertoire character is an element of the minimum subrepertoire of ISO 6937-2 and is considered an atomic construct in the formal specification.

Definition 1.21

1 $\forall z$
2 $(_{o}$ IsISO6937MSCharacter(z) *iff*
3 IsAtom(z)$_{o}$)

Semiformal Description 1.22

Predicate "is a sequence of different non-negative integers"

A sequence of different non-negative integers is a catenation of non-negative integers, where no integer occurs more than once.

Definition 1.22

1 $\forall q$
2 $(_{o}$ IsSeqOfDiffNnInt(q) *iff*
3 IsNeCat$^{1.3}$(q) *and*
4 $\forall m \in \hat{}q \bullet$ (IsNnInt$^{1.7}$(C m)) *and*
5 CARD (COLC$^{1.17}$ ($\hat{}q \bullet$)) = CARD ($\hat{}q \bullet$)$_{o}$)

Semiformal Description 1.23

Predicate "is a sequence of non-negative integers"

A sequence of non-negative integers is a catenation of non-negative integers, where an integer may occur more than once.

Definition 1.23

1 $\forall q$
2 $(_{o}$ IsSeqOfNnInt(q) *iff*
3 IsNeCat$^{1.3}$(q) *and*
4 $\forall m \in \hat{}q \bullet$ (IsNnInt$^{1.7}$(C m))$_{o}$)

Semiformal Description 1.24

Predicate "is a non-negative real number"

A non-negative real number is a real number whose value is greater than or equal to zero.

Definition 1.24

1 $\forall v$
2 $(_{o}$ IsNnReal(v) *iff*
3 IsReal(v) *and* $v \geq 0.0_{o}$)

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 1.25

Predicate "is a positive real number"

A positive real number is a real number whose value is greater than zero.

Definition 1.25

1 $\forall v$
2 $(_{o}\text{IsPosReal}(v)$ *iff*
3 $\text{IsReal}(v)$ *and* $v > 0.0_{o})$

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

7 Formal specification of the document structures

Within the semiformal descriptions there is usually a reference to one or more clause numbers. Within this clause 7 these clause numbers refer to the respective clauses in ISO/IEC 8613-2, unless explicitly otherwise specified.

What follows is the outline of the formula which specifies the ODA Document Structures, according to the specifications in ISO/IEC 8613-2. The dots indicate formal text fragments which have been left out for the sake of readability.

The full formula can be obtained by replacing each line (apart from the *and*) with the definition which is referenced by the superscript of the predicate symbol or operator symbol, respectively. Thereby the variables used in the definition of the predicate have to be replaced by those appearing in the outline (if they are different). In general, mnemonic names for variables have been chosen. Formally this is insignificant, since a definition with, for example, the variable *slay* must work for any value of *slay*, not only for a correct specific layout description. That is, the definition yields True for a particular value of *slay*, if and only if this value is a correct specific layout description, otherwise the definition yields False. Note that a definition is a formula, hence it may never yield an undefined result, whatever value has been inserted for the variable.

-.sets of constituents: .-

... IsInterchangeSet$^{2.1}(doc)$ ...

and ... IsGenericDocumentDescription$^{2.2}(gdoc)$ ...
and ... IsProtectedDocumentDescription$^{2.3}(pdoc)$ ...
and ... IsDocumentDescription$^{2.4}(doc)$ ...

and ... IsProcessable$^{2.5}(doc)$ ...
and ... IsFormattedProcessable$^{2.6}(doc)$ ...
and ... IsFormatted$^{2.7}(doc)$ ...

and ... IsDocumentBody$^{2.8}(doby)$ ...

and ... IsGenericDescription$^{2.9}(g)$ ...
and ... IsCompleteGenericDescription$^{2.10}(g)$ ...
and ... IsGenericLogicalDescription$^{2.11}(glog)$ ...
and ... IsGenericLayoutDescription$^{2.12}(glay)$ ...

and ... IsSpecificDescription$^{2.13}(s)$ ...
and ... IsSpecificLogicalDescription$^{2.14}(slog)$ ...
and ... IsSpecificLayoutDescription$^{2.15}(slay)$ ...

and ... IsLayoutStyleSet$^{2.16}(laystys)$ ...
and ... IsPresentationStyleSet$^{2.17}(prestys)$ ...
and ... IsProtectedPart$^{2.18}(ppart)$ ...

-.constituents: .-

and ... IsRootClassDescription$^{2.19}(cst)$ ...

and ... IsLogicalRootClassDescription$^{2.20}(cst)$ ...
and ... IsCompositeLogicalObjectClassDescription$^{2.21}(cst)$ ...
and ... IsBasicLogicalObjectClassDescription$^{2.22}(cst)$ ...
and ... IsLogicalObjectClassDescription$^{2.23}(cst)$ ...

and ... IsLayoutRootClassDescription$^{2.24}(cst)$ ...
and ... IsPageSetClassDescription$^{2.25}(cst)$ ...
and ... IsCompositePageClassDescription$^{2.26}(cst)$ ...
and ... IsBasicPageClassDescription$^{2.27}(cst)$ ...
and ... IsFrameClassDescription$^{2.28}(cst)$ ...
and ... IsBlockClassDescription$^{2.29}(cst)$ ...
and ... IsLayoutObjectClassDescription$^{2.30}(cst)$ ...

and ... IsCompositeObjectClassDescription$^{2.31}(cst)$ ...
and ... IsCompositeLayoutObjectClassDescription$^{2.32}(cst)$ ...
and ... IsBasicLayoutObjectClassDescription$^{2.33}(cst)$ ...
and ... IsBasicObjectClassDescription$^{2.34}(cst)$ ...

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

© ISO/IEC

and ... IsObjectClassDescription$^{2.35}(cst)$ ...

and ... IsRootDescription$^{2.36}(cst)$ ...

and ... IsLogicalRootDescription$^{2.37}(cst)$ ...
and ... IsCompositeLogicalObjectDescription$^{2.38}(cst)$ ...
and ... IsBasicLogicalObjectDescription$^{2.39}(cst)$ ...
and ... IsLogicalObjectDescription$^{2.40}(cst)$ ...

and ... IsLayoutRootDescription$^{2.41}(cst)$ ...
and ... IsPageSetDescription$^{2.42}(cst)$ ...
and ... IsCompositePageDescription$^{2.43}(cst)$ ...
and ... IsBasicPageDescription$^{2.44}(cst)$ ...
and ... IsPageDescription$^{2.45}(cst)$ ...
and ... IsFrameDescription$^{2.46}(cst)$ ...
and ... IsBlockDescription$^{2.47}(cst)$ ...
and ... IsLayoutObjectDescription$^{2.48}(cst)$ ...

and ... IsCompositeObjectDescription$^{2.49}(cst)$ ...
and ... IsBasicObjectDescription$^{2.50}(cst)$ ...
and ... IsCompositeLayoutObjectDescription$^{2.51}(cst)$ ...
and ... IsBasicLayoutObjectDescription$^{2.52}(cst)$ ...
and ... IsObjectDescription$^{2.53}(cst)$ ...

and ... IsContentPortionDescription$^{2.54}(cst)$ ...

and ... IsLayoutStyle$^{2.55}(cst)$ ...
and ... IsPresentationStyle$^{2.56}(cst)$ ...
and ... IsSealedDocumentProfileDescription$^{2.57}(cst)$ ...
and ... IsEncipheredDocumentProfileDescription$^{2.58}(cst)$ ...
and ... IsPreEncipheredDocumentBodyPartDescription$^{2.59}(cst)$ ...
and ... IsPostEncipheredDocumentBodyPartDescription$^{2.60}(cst)$ ...

-.attributes: .-

and ... IsAttributeSet$^{2.61}(as)$ ...
and ... IsProfileAttributeSetPart2$^{2.62}(as)$ ...

and ... IsBindingsValueExpression$^{2.63}(v)$ ...
and ... IsStringExpression$^{2.64}(v)$ ...
and ... IsAtomicStringExpression$^{2.65}(v)$ ...
and ... IsStringFunction$^{2.66}(v)$ ...
and ... IsNumericExpression$^{2.67}(v)$ ...
and ... IsNumericFunction$^{2.68}(v)$ ...
and ... IsObjectOrObjectClassIdExpression$^{2.69}(v)$ ...
and ... IsObjectOrObjectClassSelectionFunction$^{2.70}(v)$ ...
and ... IsBindingReferenceExpression$^{2.71}(v)$ ...
and ... IsBindingReference$^{2.72}(v)$ ...
and ... IsBindingName$^{2.73}(v)$ ...
and ... IsBindingSelectionFunction$^{2.74}(v)$ ...
and ... IsCurrentInstanceFunction$^{2.75}(v)$ ...
and ... IsObjectTypeValue$^{2.76}(v)$ ...
and ... IsLogicalObjectId$^{2.77}(v)$ ...
and ... IsLayoutObjectId$^{2.78}(v)$ ...
and ... IsObjectId$^{2.79}(v)$ ...
and ... IsLogicalObjectClassId$^{2.80}(v)$ ...
and ... IsLayoutObjectClassId$^{2.81}(v)$ ...
and ... IsObjectClassId$^{2.82}(v)$ ...
and ... IsSeqOfObjectClassId$^{2.83}(v)$ ...
and ... IsConstructionExpression$^{2.84}(v)$ ...
and ... IsConstructionType$^{2.85}(v)$ ...
and ... IsConstructionTerm$^{2.86}(v)$ ...

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

and ... IsContentArchitectureClassValue$^{2.87}(v)$ ...
and ... IsPresentationAttributesValue$^{2.88}(v)$ ...
and ... IsBindingsValue$^{2.89}(v)$ ...
and ... IsDefaultValueListsValue$^{2.90}(v)$ ...
and ... IsPositionValue$^{2.91}(v)$ ...
and ... IsDimensionsValue$^{2.92}(v)$ ...
and ... IsBorderValue$^{2.93}(v)$ ...
and ... IsBalanceValue$^{2.94}(v)$ ...
and ... IsLayoutPathValue$^{2.95}(v)$ ...
and ... IsTransparencyValue$^{2.96}(v)$ ...
and ... IsColourValue$^{2.97}(v)$ ...
and ... IsMediumTypeValue$^{2.98}(v)$ ...
and ... IsProtectionValue$^{2.99}(v)$ ...
and ... IsLayoutStyleId$^{2.100}(v)$ ...
and ... IsBlockAlignmentValue$^{2.101}(v)$ ...
and ... IsConcatenationValue$^{2.102}(v)$ ...
and ... IsFillOrderValue$^{2.103}(v)$ ...
and ... IsLayoutObjectReference$^{2.104}(v)$ ...
and ... IsLayoutCatName$^{2.105}(v)$ ...
and ... IsSetOfLayoutCatNames$^{2.106}(v)$ ...
and ... IsOffsetValue$^{2.107}(v)$ ...
and ... IsSameLayoutObjectValue$^{2.108}(v)$ ...
and ... IsSeparationValue$^{2.109}(v)$ ...
and ... IsPresentationStyleId$^{2.110}(v)$ ...
and ... IsLayoutContentPortionId$^{2.111}(v)$ ...
and ... IsLogicalContentPortionId$^{2.112}(v)$ ...
and ... IsTypeOfCodingValue$^{2.113}(v)$ ...
and ... IsContentInformationValue$^{2.114}(v)$ ...
and ... IsAlternativeCharacterSetsString$^{2.115}(v)$ ...
and ... IsCodingAttributesValue$^{2.116}(v)$ ...
and ... IsCharSeq$^{2.117}(v)$ ...
and ... IsCharacter$^{2.118}(z)$ ...
and ... IsEncipheredValue$^{2.119}(v)$ ...
and ... IsSealedValue$^{2.120}(v)$ ...
and ... IsSealedDocumentProfileInformationValue$^{2.121}(v)$ ...
and ... IsProtectedPartId$^{2.122}(v)$ ...
and ... IsEncipheredBodyPartId$^{2.123}(v)$ ...
and ... IsSetOfSealIds$^{2.124}(v)$ ...
and ... IsSealId$^{2.125}(v)$ ...
and ... IsColourExpression$^{2.126}(v)$ ...
and ... IsIndexedColourExpression$^{2.127}(v)$ ...
and ... IsDirectColourExpression$^{2.128}(v)$ ...
and ... IsColourSpecificationValue$^{2.129}(v)$ ...
and ... IsColourToleranceValue$^{2.130}(v)$ ...
and ... IsColourTable$^{2.131}(v)$ ...
and ... IsSetOfColourTableEntries$^{2\ 132}(v)$ ...
and ... IsColourTableEntry$^{2.133}(v)$ ...
and ... IsFloatabilityRangeValue$^{2.134}(v)$ ...

-.subsidiary definitions: -

and ... (s)IsDescribedBy$^{2.135}(g)$ ...
and ... (s)IsCompletelyDescribedBy$^{2.136}(g)$ ...
and ... IsContentConnected$^{2.137}(csts)$ ...
and ... IsContentComplete$^{2.138}(csts)$ ...
and ... IsClassImSubComplete$^{2.139}(g)$ ...
and ... IsImSubComplete$^{2.140}(s)$ ...

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

and ... (*slog*)IsContainedIn$^{2.141}$(*slay*) ...
and ... (*id*)IsClassIdImSubIn$^{2.142}$(*gco*) ...
and ... (*id*)IsIdImSubIn$^{2.143}$(*sco*) ...
and ... (*id*)IsIdContIn$^{2.144}$(*b*) ...
and ... (*idq*)IsSeqClassIdImSubIn$^{2.145}$(*gco*) ...
and ... (*idq*)CoveredBy$^{2.146}$(*r*) ...
and ... (*id*)OccursIn$^{2.147}$(*m*) ...
and ... (*go*)IsInitialGenericIn$^{2.148}$(*gos*) ...
and ... (*so*)IsInitialSpecificIn$^{2.149}$(*sos*) ...
and ... (*lgo*)DescribesClassImSubOf$^{2.150}$(*hgo*) ...
and ... (*lgo*)DescribesClassSubOf$^{2.151}$(*hgo*)In(*g*) ...
and ... (*lso*)DescribesImSubOf$^{2.152}$(*hso*) ...
and ... (*cont*)DescribesContPortOf$^{2.153}$(*b*)

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

In principle a given interchange set may be verified to conform to ISO/IEC 8613 by establishing

$\mathcal{AS}$ *impl* IsInterchangeSet$^{2.1}$(-.**this-term**.-)

where $\mathcal{AS}$ stands for the conjugation of all axioms and definitions and the argument of IsInterchangeSet stands for a formal expression (a term) which denotes the particular interchange set in question. If the extended overall formula yields True, the document description conforms to ISO/IEC 8613, if False, it does not. An undefined result is impossible.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

7.1 Sets of constituents

Semiformal Description 2.1

Predicate "is an interchange set" (clause 6.3)

An entity *doc* is an interchange set if it is a generic document description, a document description or a protected document description.

Definition 2.1

1 $\forall\, doc$
2 $(_0\, \text{IsInterchangeSet}(doc)$ *iff*
3 $\text{IsGenericDocumentDescription}^{2.2}(doc)$ *or* $\text{IsDocumentDescription}^{2.4}(doc)$
4 *or* $\text{IsProtectedDocumentDescription}^{2.3}(doc)_0)$

Semiformal Description 2.2

Predicate "is a generic document description" (clause 6.3.10)

A generic document description consists of a document profile and a document body (4-6). The document body consists of the union of four entities *glog*, *glay*, *laystys* and *prestys* (7). If the layout style set is non-empty, the generic logical structure is non-empty (8). Either a generic logical or a generic layout description (or both) must always be present (9). The document profile and the document body satisfy the constraints imposed by the content architectures (10-12).

Definition 2.2

1 $\forall\, gdoc$
2 $(_0\, \text{IsGenericDocumentDescription}(gdoc)$ *iff*
3 $\exists\, prof, glog, glay, laystys, prestys, doby$
4 $(_1\, gdoc = [prof] \cup doby$ *and*
5 $\text{IsDocumentProfile}^{4.1}(prof)$ *and*
6 $\text{IsDocumentBody}^{2.8}(doby)$ *and*
7 $doby = glog \cup glay \cup laystys \cup prestys$ *and*
8 $(_2\, \text{IsNeCol}^{1.1}(laystys)$ *impl* $\text{IsNeCol}^{1.1}(glog)\ _2)$ *and*
9 $(_3\, \text{IsNeCol}^{1.1}(glog)$ *or* $\text{IsNeCol}^{1.1}(glay)\ _3)$ *and*
10 $\text{SatisfiesCharacterContentArchitectureConstraints}^{6.4}(prof, doby)$ *and*
11 $\text{SatisfiesRasterGraphicsContentArchitectureConstraints}^{7.4}(prof, doby)$ *and*
12 $\text{SatisfiesGeometricGraphicsContentArchitectureConstraints}^{8.4}(prof, doby)_1)_0)$

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Semiformal Description 2.3

Predicate "is a protected document description" (clause 6.6)

A protected document description *pdoc* contains a non-empty subset *ppart* which is a protected part.

NOTE — Only the existence of a (non-empty) protected part in a protected document description is formally specified. After deciphering the protected part and creating the objects contained in the protected part, the interchange set is either a generic document description or a document description.

Definition 2.3

$\forall\, pdoc$
$(_0\, \text{IsProtectedDocumentDescription}(pdoc)$ *iff*
  $\exists\, upart, ppart$
  $(_1\, pdoc = upart \cup ppart$ *and*
    $\text{IsNeCol}^{1.1}(ppart)$ *and* $\text{IsProtectedPart}^{2.18}(ppart)_1)_0)$

Semiformal Description 2.4

Predicate "is a document description" (clause 6.3)

An entity *doc* is a document description if it is a document profile *prof* (5) or if it is an entity which is processable, formatted processable or formatted (5–6), according to the value of the document profile attribute 'document architecture class' (7–13), with a 'resource document' specified in the profile if any 'resource' is specified in the document (14–15).

Definition 2.4

$\forall\, doc$
$(_0\, \text{IsDocumentDescription}(doc)$ *iff*
  $\exists\, prof$
  $(_1\, \text{IsDocumentProfile}^{4.1}(prof)$ *and*
    $(_2\, doc = [prof]$ *or* $\text{IsProcessable}^{2.5}(doc)$ *or*
      $\text{IsFormattedProcessable}^{2.6}(doc)$ *or* $\text{IsFormatted}^{2.7}(doc)\ _2)$ *and*
    $doc \neq [prof]$ *impl*
    $(_3(_4\, C\ \hat{}prof \bullet$ 'document architecture class' = 'processable' *iff*
      $\text{IsProcessable}^{2.5}(doc)_4)$ *and*
      $(_5\, C\ \hat{}prof \bullet$ 'document architecture class' = 'formatted processable' *iff*
        $\text{IsFormattedProcessable}^{2.6}(doc)_5)$ *and*
      $(_6\, C\ \hat{}prof \bullet$ 'document architecture class' = 'formatted' *iff*
        $\text{IsFormatted}^{2.7}(doc)_6)$ *and*
      $\forall\, cst \in doc$
      $(_7$ 'resource' $\in \text{NAMS}^{1.18}(cst)$ *impl* 'resource document' $\in \text{NAMS}^{1.18}(prof)\ _7)_3)_1)_0)$

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 2.5

Predicate "is processable" (clauses 6.3.9, 6.3.13)

An entity *doc* is processable if it is a collection of constituents consisting of a document profile *prof* along with a document body *doby* (4,5) which includes (possibly empty) constituent sets *glog*, *slog*, *glay*, *laystys* and *prestys* (6). Attributes in the document profile indicate the presence of a specific logical description (which must always be present) and of (optionally) a generic logical description, generic layout description, layout style set and presentation style set (7-15). If the value of the profile attribute 'generic logical structure' is 'complete generator set', the specific logical description *slog* is completely described by the generic logical description *glog* (16,17). If the profile attribute 'generic layout structure' is specified, its value is 'complete generator set' and the generic layout description *glay* is a complete generic description (18-20). The profile attribute 'specific layout structure' is not specified (21). The profile attribute 'document architecture class' has the value 'processable' (22). The document profile and the document body satisfy the constraints imposed by the content architectures (23-25).

Definition 2.5

1 $\forall\, doc$
2 $(_0\,\mathrm{IsProcessable}(doc)$ *iff*
3 $\exists\, prof, doby, glog, slog, glay, laystys, prestys$
4 $(_1\, doc = [prof] \cup doby$ *and*
5 $\mathrm{IsDocumentProfile}^{4.1}(prof)$ *and* $\mathrm{IsDocumentBody}^{2.8}(doby)$ *and*
6 $doby = glog \cup slog \cup glay \cup laystys \cup prestys$ *and*
7 'specific logical structure' $\in \mathrm{NAMS}^{1.18}(prof)$ *and* $\mathrm{IsSpecificLogicalDescription}^{2.14}(slog)$ *and*
8 $(_2$ 'generic logical structure' $\in \mathrm{NAMS}^{1.18}(prof)$ *iff*
9 $(_3\,\mathrm{IsNeCol}^{1.1}(glog)$ *and* $\mathrm{IsGenericLogicalDescription}^{2.11}(glog)_3)_2)$ *and*
10 $(_4$ 'generic layout structure' $\in \mathrm{NAMS}^{1.18}(prof)$ *iff*
11 $(_5\,\mathrm{IsNeCol}^{1.1}(glay)$ *and* $\mathrm{IsGenericLayoutDescription}^{2.12}(glay)_5)_4)$ *and*
12 $(_6$ 'layout styles' $\in \mathrm{NAMS}^{1.18}(prof)$ *iff*
13 $(_7\,\mathrm{IsNeCol}^{1.1}(laystys)$ *and* $\mathrm{IsLayoutStyleSet}^{2.16}(laystys)_7)_6)$ *and*
14 $(_8$ 'presentation styles' $\in \mathrm{NAMS}^{1.18}(prof)$ *iff*
15 $(_9\,\mathrm{IsNeCol}^{1.1}(prestys)$ *and* $\mathrm{IsPresentationStyleSet}^{2.17}(prestys)_9)_8)$ *and*
16 $(_{10}\,\mathrm{C}\ \hat{}prof \bullet$ 'generic logical structure' = 'complete generator set' *impl*
17 $(slog)\mathrm{IsCompletelyDescribedBy}^{2.136}(glog)_{10})$ *and*
18 $(_{11}$ 'generic layout structure' $\in \mathrm{NAMS}^{1.18}(prof)$ *impl*
19 $(_{12}\,\mathrm{C}\ \hat{}prof \bullet$ 'generic layout structure' = 'complete generator set' *and*
20 $\mathrm{IsCompleteGenericDescription}^{2.10}(glay)_{12})_{11})$ *and*
21 'specific layout structure' $\notin \mathrm{NAMS}^{1.18}(prof)$ *and*
22 $\mathrm{C}\ \hat{}prof \bullet$ 'document architecture class' = 'processable' *and*
23 $\mathrm{SatisfiesCharacterContentArchitectureConstraints}^{6.4}(prof, doby)$ *and*
24 $\mathrm{SatisfiesRasterGraphicsContentArchitectureConstraints}^{7.4}(prof, doby)$ *and*
25 $\mathrm{SatisfiesGeometricGraphicsContentArchitectureConstraints}^{8.4}(prof, doby)_1)_0)$

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 2.6

Predicate "is formatted processable" (clauses 6.3.9, 6.3.13)

An entity *doc* is formatted processable if it is a collection of constituents consisting of a document profile *prof* along with a document body *doby* (4,5) which includes (possibly empty) constituent sets *glog*, *slog*, *glay*, *slay*, *laystys* and *prestys* (6). Attributes in the document profile indicate the presence of a specific logical description, the specific layout description, the generic layout description (these three sets are always present) and of (optionally) a generic logical description, layout style set and presentation style set (7-15). If the value of the profile attribute 'generic logical structure' is 'complete generator set', the specific logical description *slog* is completely described by the generic logical description *glog* (16,17). If the value of the profile attribute 'generic layout structure' is 'complete generator set', the specific layout description *slay* is completely described by the generic layout description *glay* (18,19). The specific logical description is contained in the specific layout description (20). The profile attribute 'document architecture class' has the value 'formatted processable' (21). The document profile and the document body satisfy the constraints imposed by the content architectures (22-24).

Definition 2.6

1 $\forall\, doc$
2 $(_0$ IsFormattedProcessable(doc) *iff*
3 $\exists\, prof, doby, glog, slog, glay, slay, laystys, prestys$
4 $(_1\, doc = [prof] \cup doby$ *and*
5 IsDocumentProfile$^{4.1}(prof)$ *and* IsDocumentBody$^{2.8}(doby)$ *and*
6 $doby = glog \cup slog \cup glay \cup slay \cup laystys \cup prestys$ *and*
7 'specific logical structure' $\in$ NAMS$^{1.18}(prof)$ *and* IsSpecificLogicalDescription$^{2.14}(slog)$ *and*
8 'specific layout structure' $\in$ NAMS$^{1.18}(prof)$ *and* IsSpecificLayoutDescription$^{2.15}(slay)$ *and*
9 'generic layout structure' $\in$ NAMS$^{1.18}(prof)$ *and* IsGenericLayoutDescription$^{2.12}(glay)$ *and*
10 $(_2$ 'generic logical structure' $\in$ NAMS$^{1.18}(prof)$ *iff*
11 $(_3$ IsNeCol$^{1.1}(glog)$ *and* IsGenericLogicalDescription$^{2.11}(glog)_3)_2)$ *and*
12 $(_4$ 'layout styles' $\in$ NAMS$^{1.18}(prof)$ *iff*
13 $(_5$ IsNeCol$^{1.1}(laystys)$ *and* IsLayoutStyleSet$^{2.16}(laystys)_5)_4)$ *and*
14 $(_6$ 'presentation styles' $\in$ NAMS$^{1.18}(prof)$ *iff*
15 $(_7$ IsNeCol$^{1.1}(prestys)$ *and* IsPresentationStyleSet$^{2.17}(prestys)_7)_6)$ *and*
16 $(_8$ C $\hat{}prof$ • 'generic logical structure' = 'complete generator set' *impl*
17 $(slog)$IsCompletelyDescribedBy$^{2.136}(glog)_8)$ *and*
18 $(_9$ C $\hat{}prof$ • 'generic layout structure' = 'complete generator set' *impl*
19 $(slay)$IsCompletelyDescribedBy$^{2.136}(glay)_9)$ *and*
20 $(slog)$IsContainedIn$^{2.141}(slay)$ *and*
21 C $\hat{}prof$ • 'document architecture class' = 'formatted processable' *and*
22 SatisfiesCharacterContentArchitectureConstraints$^{6.4}(prof, doby)$ *and*
23 SatisfiesRasterGraphicsContentArchitectureConstraints$^{7.4}(prof, doby)$ *and*
24 SatisfiesGeometricGraphicsContentArchitectureConstraints$^{8.4}(prof, doby)_1)_0)$

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 2.7

Predicate "is formatted" (clauses 6.3.9, 6.3.13)

An entity *doc* is formatted if it is a collection of constituents consisting of a document profile *prof* along with a document body *doby* (4,5) which includes (possibly empty) constituent sets *slay*, *glay* and *prestys* (6). Attributes in the document profile indicate the presence of a specific layout description (which must always be present) and of (optionally) a generic layout description and presentation style set (7-11). The value of the profile attribute 'generic layout structure', if specified, is 'partial generator set' or 'factor set' (12,13). The profile attributes 'specific logical structure', 'generic logical structure' and 'layout styles' are not specified (14-16). The profile attribute 'document architecture class' has the value 'formatted' (17). The document profile and the document body satisfy the constraints imposed by the content architectures (18-20).

Definition 2.7

1 $\forall\, doc$
2 $(_0\,\mathrm{IsFormatted}(doc)$ *iff*
3 $\exists\, prof, doby, glay, slay, prestys$
4 $(_1\, doc = [prof] \cup doby$ *and*
5 $\mathrm{IsDocumentProfile}^{4.1}(prof)$ *and* $\mathrm{IsDocumentBody}^{2.8}(doby)$ *and*
6 $doby = glay \cup slay \cup prestys$ *and*
7 'specific layout structure' $\in \mathrm{NAMS}^{1.18}(prof)$ *and* $\mathrm{IsSpecificLayoutDescription}^{2.15}(slay)$ *and*
8 $(_2$ 'generic layout structure' $\in \mathrm{NAMS}^{1.18}(prof)$ *iff*
9 $(_3\,\mathrm{IsNeCol}^{1.1}(glay)$ *and* $\mathrm{IsGenericLayoutDescription}^{2.12}(glay)_3)_2)$ *and*
10 $(_4$ 'presentation styles' $\in \mathrm{NAMS}^{1.18}(prof)$ *iff*
11 $(_5\,\mathrm{IsNeCol}^{1.1}(prestys)$ *and* $\mathrm{IsPresentationStyleSet}^{2.17}(prestys)_5)_4)$ *and*
12 $(_6$ 'generic layout structure' $\in \mathrm{NAMS}^{1.18}(prof)$ *impl*
13 $C\ \hat{}\ prof \bullet$ 'generic layout structure' $\in$ ['partial generator set'; 'factor set'] $_6)$ *and*
14 'specific logical structure' $\notin \mathrm{NAMS}^{1.18}(prof)$ *and*
15 'generic logical structure' $\notin \mathrm{NAMS}^{1.18}(prof)$ *and*
16 'layout styles' $\notin \mathrm{NAMS}^{1.18}(prof)$ *and*
17 $C\ \hat{}\ prof \bullet$ 'document architecture class' = 'formatted' *and*
18 $\mathrm{SatisfiesCharacterContentArchitectureConstraints}^{6.4}(prof, doby)$ *and*
19 $\mathrm{SatisfiesRasterGraphicsContentArchitectureConstraints}^{7.4}(prof, doby)$ *and*
20 $\mathrm{SatisfiesGeometricGraphicsContentArchitectureConstraints}^{8.4}(prof, doby)_1)_0)$

Semiformal Description 2.8

Predicate "is a document body" (clause 6.3.14)

An entity *doby* is a document body if it is a collection of (possibly empty) constituents *glog*, *slog*, *glay*, *slay*, *laystys* and *prestys* (3,4). If *glog* is not empty it is a generic logical description (5). If *slog* is not empty it is a specific logical description (6). If *glay* is not empty it is a generic layout description (7). If *slay* is not empty it is a specific layout description (8). If *laystys* is not empty it is a layout style set (9). If *prestys* is not empty it is a presentation style set (10). If a generic description and a corresponding specific description are present in a document body, the specific one is described by the generic one (11, 12), i.e. the specific description is an instance of the class described by the generic description. All references to layout object classes from attributes of layout styles are satisfied (17-19, 24-27), i.e., each referenced layout object class exists in the generic layout structure. All references to layout categories from attributes of layout styles are satisfied (17, 20-22, 28, 29), i.e., each referenced layout category is specified by some layout style. All references to layout styles from attributes of layout styles are satisfied (22, 23), i.e., each referenced layout style exists in the layout style set. All references from *glay* to *glog* by means of 'logical source' are satisfied (31, 32). All objects or object classes which are referenced by the attribute 'balance' from the specific or generic layout structure, respectively, are present (33-38). All references to layout categories by means of the attribute 'permitted categories' are satisfied (39-41). If the attributes 'imaging order' and 'subordinates' are both specified their values must specify the same set of constituents (42-43).

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Definition 2.8

$\forall\, doby$
$(_0 \text{IsDocumentBody}(doby)\ \underline{iff}$
$\exists\, glog, slog, glay, slay, laystys, prestys$
$(_1\ doby = glog \cup slog \cup glay \cup slay \cup laystys \cup prestys\ \underline{and}$
$(_2 \text{IsNeCol}^{1.1}(glog)\ \underline{impl}\ \text{IsGenericLogicalDescription}^{2.11}(glog)_2)\ \underline{and}$
$(_3 \text{IsNeCol}^{1.1}(slog)\ \underline{impl}\ \text{IsSpecificLogicalDescription}^{2.14}(slog)_3)\ \underline{and}$
$(_4 \text{IsNeCol}^{1.1}(glay)\ \underline{impl}\ \text{IsGenericLayoutDescription}^{2.12}(glay)_4)\ \underline{and}$
$(_5 \text{IsNeCol}^{1.1}(slay)\ \underline{impl}\ \text{IsSpecificLayoutDescription}^{2.15}(slay)_5)\ \underline{and}$
$(_6 \text{IsNeCol}^{1.1}(laystys)\ \underline{impl}\ \text{IsLayoutStyleSet}^{2.16}(laystys)_6)\ \underline{and}$
$(_7 \text{IsNeCol}^{1.1}(prestys)\ \underline{impl}\ \text{IsPresentationStyleSet}^{2.17}(prestys)_7)\ \underline{and}$
$(_8 (_9 \text{IsNeCol}^{1.1}(slog)\ \underline{and}\ \text{IsNeCol}^{1.1}(glog)_9)\ \underline{impl}\ (slog)\text{IsDescribedBy}^{2.135}(glog)_8)\ \underline{and}$
$(_{10} (_{11} \text{IsNeCol}^{1.1}(slay)\ \underline{and}\ \text{IsNeCol}^{1.1}(glay)_{11})\ \underline{impl}\ (slay)\text{IsDescribedBy}^{2.135}(glay)_{10})\ \underline{and}$
$\forall\, cst \in (glog \cup slog)\ (_{12} \forall\, x \in \hat{}cst \bullet\ (\text{N}\ x = \text{'layout style'})\ \underline{impl}$
$(_{13} \exists\, ls \in laystys\ (\text{C}\ \hat{}ls \bullet \text{'layout style identifier'} = \text{C}\ x)_{13})_{12})\ \underline{and}$
$\forall\, cst \in (glog \cup slog \cup glay \cup slay)\ (_{14} \forall\, x \in \hat{}cst \bullet\ (\text{N}\ x = \text{'presentation style'})\ \underline{impl}$
$(_{15} \exists\, ps \in prestys\ (\text{C}\ \hat{}ps \bullet \text{'presentation style identifier'} = \text{C}\ x)_{15})_{14})\ \underline{and}$
$\forall\, cst \in laystys\ (_{16} \forall\, y \hat{\in} \hat{}cst \bullet$
$(_{17} (_{18} \text{IsLayoutObjectClassId}^{2.81}(\text{C}\ y)\ \underline{impl}$
$(_{19} \exists\, go \in glay\ (\text{C}\ \hat{}go \bullet \text{'object class identifier'} = \text{C}\ y)_{19})_{18})\ \underline{and}$
$(_{20} \text{IsLayoutCatName}^{2.105}(\text{C}\ y)\ \underline{impl}$
$(_{21} \exists\, ls \in laystys\ (\text{C}\ \hat{}ls \bullet \text{'layout category'} = \text{C}\ y)_{21})_{20})\ \underline{and}$
$(_{22} \text{IsLayoutStyleId}^{2.100}(\text{C}\ y)\ \underline{impl}$
$(_{23} \exists\, ls \in laystys\ (\text{C}\ \hat{}ls \bullet \text{'layout style identifier'} = \text{C}\ y)_{23})_{22})\ \underline{and}$
$\forall\, l, r$
$(_{24} (_{25} (_{26} \text{N}\ y = \text{'same layout object'}\ \underline{and}$
$\text{C}\ y = [\text{'logical object'} : l; \text{'layout object'} : r]\ \underline{and}\ \text{IsLayoutObjectClassId}^{2.81}(r)_{26})\ \underline{impl}$
$(_{27} \exists\, go \in glay\ (\text{C}\ \hat{}go \bullet \text{'object class identifier'} = r)_{27})_{25})\ \underline{and}$
$(_{28} (_{29} \text{N}\ y = \text{'same layout object'}\ \underline{and}$
$\text{C}\ y = [\text{'logical object'} : l; \text{'layout object'} : r]\ \underline{and}\ \text{IsLayoutCatName}^{2.105}(r)_{29})\ \underline{impl}$
$(_{30} \exists\, ls \in laystys\ (\text{C}\ \hat{}ls \bullet \text{'layout category'} = r)_{30})_{28})_{24})_{17})_{16})\ \underline{and}$
$\forall\, a \hat{\in} \hat{}glay \bullet \bullet \text{'logical source'}$
$(_{31} \exists\, go \in glog\ (\text{C}\ \hat{}go \bullet \text{'object class identifier'} = \text{C}\ a)_{31})\ \underline{and}$
$\forall\, a \hat{\in} \hat{}slay \bullet \bullet \text{'balance'}$
$(_{32} \text{C}\ a \neq \text{'null'}\ \underline{impl}$
$\forall\, b \hat{\in} \hat{}(\text{C}\ a) \bullet (_{33} \exists\, so \in slay\ (\text{C}\ \hat{}so \bullet \text{'object identifier'} = b)_{33})_{32})\ \underline{and}$
$\forall\, a \hat{\in} \hat{}glay \bullet \bullet \text{'balance'}$
$(_{34} \text{C}\ a \neq \text{'null'}\ \underline{impl}$
$\forall\, b \hat{\in} \hat{}(\text{C}\ a) \bullet (_{35} \exists\, go \in glay\ (\text{C}\ \hat{}go \bullet \text{'object class identifier'} = b)_{35})_{34})\ \underline{and}$
$\forall\, a \in \text{C}\ \hat{}slay \bullet \bullet \text{'permitted categories'}$
$(_{36} a \neq \text{'null'}\ \underline{impl}$
$\exists\, ls \in laystys\ (_{37} \text{C}\ \hat{}ls \bullet \text{'layout category'} = a_{37})_{36})\ \underline{and}$
$\forall\, cst \in slay$
$(_{38} ([\text{'imaging order'}; \text{'subordinates'}] \subset \text{NAMS}^{1.18}(cst))\ \underline{impl}$
$[x \mid x \hat{\in} \hat{}cst \bullet \text{'imaging order'}] = [y \mid y \hat{\in} \hat{}cst \bullet \text{'subordinates'}]_{38})_1)_0)$

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 2.9

Predicate "is a generic description" (clause 6.10)

An entity g is a generic description, if it is a set of object class descriptions (5,6) and optional content portion descriptions (10,11). It is content connected and is complete with respect to the content portion descriptions and the object class descriptions (13-15). A generic description has not more than one root class description (12). The values of the attributes 'object class identifier' are unique within the set g (7-9).

Definition 2.9

1 $\forall g$
2 $(_0$ IsGenericDescription(g) *iff*
3 $\exists ocs, conts$
4 $(_1\ g = ocs \cup conts$ *and*
5 IsNeCol$^{1.1}(ocs)$ *and*
6 $\forall cst \in ocs$ (IsObjectClassDescription$^{2.35}(cst)$) *and*
7 $\forall x, y \in ocs$
8 $(_2\ x \neq y$ *impl*
9 C ^x • 'object class identifier' $\neq$ C ^y • 'object class identifier'$_2$) *and*
10 IsCol($conts$) *and*
11 $\forall cst \in conts$ (IsContentPortionDescription$^{2.54}(cst))_1$) *and*
12 CARD $[cst \mid cst \in g$ *and* IsRootClassDescription$^{2.19}(cst)] \leq 1$ *and*
13 IsContentConnected$^{2.137}(g)$ *and*
14 IsContentComplete$^{2.138}(g)$ *and*
15 IsClassImSubComplete$^{2.139}(g)_0$)

Semiformal Description 2.10

Predicate "is a complete generic description" (clauses 6.3.4, 7.5)

An entity g is a complete generic description if it is a generic description with one or more initial generic constituents (object class descriptions) igs (5). Exactly one of them (r) is a logical or layout root class description (6,8) and refers to at least one subordinate object class description (7). In addition, every non-initial object class description is directly or indirectly referred to from exactly one initial generic constituent (9–13). Thus, a complete generic description is a set of connected, directed, single-rooted graphs (via the attribute 'generator for subordinates').

Definition 2.10

1 $\forall g$
2 $(_0$ IsCompleteGenericDescription(g) *iff*
3 IsGenericDescription$^{2.9}(g)$ *and*
4 $\exists igs, r$
5 $(_1\ igs = [cst \mid (cst)$IsInitialGenericIn$^{2.148}(g)]$ *and* $r \in igs$ *and*
6 IsRootClassDescription$^{2.19}(r)$ *and*
7 'generator for subordinates' $\in$ NAMS$^{1.18}(r)$ *and*
8 *not* $\exists ig \in igs$ ($ig \neq r$ *and* IsRootClassDescription$^{2.19}(ig)$) *and*
9 $\forall cst \in g$
10 $(_2$ *not* ($cst \in igs$ *or* IsContentPortionDescription$^{2.54}(cst)$) *impl*
11 $\exists ig \in igs$
12 $(_3(cst)$DescribesClassSubOf$^{2.151}(ig)$In(g)] *and*
13 $\forall h \in igs(_4\ h = ig$ *or not* (cst)DescribesClassSubOf$^{2.151}(h)$In(g) $_4)_3)_2)_1)_0$)

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 2.11

Predicate "is a generic logical description" (clauses 6.3.4, 7.5)

An entity *glog* is a generic logical description if it is a generic description containing only logical object class descriptions and content portion descriptions (4-6). For a content portion description the attribute 'content identifier logical' is present and its value is a generic logical identifier.

Definition 2.11

1 $\forall\, glog$
2 $({}_0\text{IsGenericLogicalDescription}(glog)$ *iff*
3 $\text{IsGenericDescription}^{2.9}(glog)$ *and*
4 $\forall\, cst \in glog$
5 $({}_1\text{IsLogicalObjectClassDescription}^{2.23}(cst)$ *or*
6 $({}_2\text{IsContentPortionDescription}^{2.54}(cst)$ *and*
7 'content identifier logical' $\in \text{NAMS}^{1.18}(cst)$ *and*
8 $\text{IsLogicalObjectClassId}^{2.80}(\text{C}\ \hat{}cst \bullet \text{'content identifier logical'})_2)_1)_0)$

Semiformal Description 2.12

Predicate "is a generic layout description" (clause 7.5.7)

An entity *glay* is a generic layout description if it is a generic description containing only layout object class descriptions and content portion descriptions (4-6). For a content portion description the attribute 'content identifier layout' is present and its value is a generic layout identifier.

Definition 2.12

1 $\forall\, glay$
2 $({}_0\text{IsGenericLayoutDescription}(glay)$ *iff*
3 $\text{IsGenericDescription}^{2.9}(glay)$ *and*
4 $\forall\, cst \in glay$
5 $({}_1\text{IsLayoutObjectClassDescription}^{2.30}(cst)$ *or*
6 $({}_2\text{IsContentPortionDescription}^{2.54}(cst)$ *and*
7 'content identifier layout' $\in \text{NAMS}^{1.18}(cst)$ *and*
8 $\text{IsLayoutObjectClassId}^{2.81}(\text{C}\ \hat{}cst \bullet \text{'content identifier layout'})_2)_1)_0)$

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 2.13

Predicate "is a specific description" (clause 7.1)

An entity s is a specific description if it is a set of object descriptions (5,6) and optional content portion descriptions (7,8). It is content connected and is complete with respect to the content portion descriptions and the object descriptions (10-12). A specific description has exactly one root description (9) which is the only initial specific constituent (object description) in s (13-15). Thus the object description r is the root of a tree which is established by the attributes 'subordinates' and 'content portions' and contains all constituents of s as nodes.

Definition 2.13

1 $\forall s$
2 $(_0$ IsSpecificDescription(s) *iff*
3 $\exists os, conts$
4 $(_1\ s = os \cup conts$ *and*
5 IsNeCol$^{1.1}(os)$ *and*
6 $\forall cst \in os$ (IsObjectDescription$^{2.53}(cst)$) *and*
7 IsCol($conts$) *and*
8 $\forall cst \in conts$ (IsContentPortionDescription$^{2.54}(cst))_1)$ *and*
9 CARD $[cst \mid cst \in s$ *and* IsRootDescription$^{2.36}(cst)] = 1$ *and*
10 IsContentConnected$^{2.137}(s)$ *and*
11 IsContentComplete$^{2.138}(s)$ *and*
12 IsImSubComplete$^{2.140}(s)$ *and*
13 $\exists r$
14 $(_2\ [r] = [cst \mid (cst)$IsInitialSpecificIn$^{2.149}(s)]$ *and*
15 IsRootDescription$^{2.36}(r)_2)_0)$

Semiformal Description 2.14

Predicate "is a specific logical description" (clauses 7.2)

An entity *slog* is a specific logical description if it is a specific description containing only logical object descriptions and content portion descriptions (4-6). For a content portion description the attribute 'content identifier logical' is present and its value is a specific logical identifier.

Definition 2.14

1 $\forall slog$
2 $(_0$ IsSpecificLogicalDescription($slog$) *iff*
3 IsSpecificDescription$^{2.13}(slog)$ *and*
4 $\forall cst \in slog$
5 $(_1$ IsLogicalObjectDescription$^{2.40}(cst)$ *or*
6 $(_2$ IsContentPortionDescription$^{2.54}(cst)$ *and*
7 'content identifier logical' $\in$ NAMS$^{1.18}(cst)$ *and*
8 IsLogicalObjectId$^{2.77}$(C $\hat{}\ cst$ • 'content identifier logical')$_2)_1)_0)$

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 2.15

Predicate "is a specific layout description" (clauses 7.3)

An entity *slay* is a specific layout description if it is a specific description containing only layout object descriptions and content portion descriptions (4-6). For a content portion description the attribute 'content identifier layout' is present and its value is a specific layout identifier (7–8). The immediate subordinates of the layout root or a page set are page sets and/or pages (9–11). The immediate subordinates of a composite page are frames and/or blocks (12–15). The immediate subordinates of a frame are frames and/or blocks (16–19). If blocks are immediate subordinates of a constituent, the attribute 'balance' must not be specified for this constituent (15, 19), otherwise it must be present or it is defaulted.

Definition 2.15

1 $\forall\, slay$
2 $(_0$IsSpecificLayoutDescription$(slay)$ *iff*
3 IsSpecificDescription$^{2.13}(slay)$ *and*
4 $\forall\, cst \in slay$
5 $(_1 (_2$IsLayoutObjectDescription$^{2.48}(cst)$ *or*
6 $(_3$IsContentPortionDescription$^{2.54}(cst)$ *and*
7 'content identifier layout' $\in$ NAMS$^{1.18}(cst)$ *and*
8 IsLayoutObjectId$^{2.78}$(C ^cst • 'content identifier layout') $_3)_2)$ *and*
9 $(_4 (_5$IsLayoutRootDescription$^{2.41}(cst)$ *or* IsPageSetDescription$^{2.42}(cst)$ $_5)$ *impl*
10 $\forall\, so \in$ IMSUB$^{2.154}(cst, slay)$
11 (IsPageSetDescription$^{2.42}(so)$ *or* IsPageDescription$^{2.45}(so)$) $_4)$ *and*
12 $(_6$IsCompositePageDescription$^{2.43}(cst)$ *impl*
13 $\forall\, so \in$ IMSUB$^{2.154}(cst, slay)$
14 $(_7$(IsFrameDescription$^{2.46}(so)$ *and* 'balance' $\in$ NAMS$^{1.18}(cst)$) *or*
15 (IsBlockDescription$^{2.47}(so)$ *and* 'balance' $\notin$ NAMS$^{1.18}(cst)$) $_7)_6)$ *and*
16 $(_8$IsFrameDescription$^{2.46}(cst)$ *impl*
17 $\forall\, so \in$ IMSUB$^{2.154}(cst, slay)$
18 $(_9$(IsFrameDescription$^{2.46}(so)$ *and* 'balance' $\in$ NAMS$^{1.18}(cst)$) *or*
19 (IsBlockDescription$^{2.47}(so)$ *and* 'balance' $\notin$ NAMS$^{1.18}(cst)$) $_9)_8)_1)_0)$

Semiformal Description 2.16

Predicate "is a layout style set" (clause 6.3.5)

An entity *laystys* is a layout style set if it is a non-empty collection of layout styles (3,4) whose identifiers are unique within that collection (5-7).

Definition 2.16

1 $\forall\, laystys$
2 $(_0$IsLayoutStyleSet$(laystys)$ *iff*
3 IsNeCol$^{1.1}(laystys)$ *and*
4 $\forall\, cst \in laystys$ (IsLayoutStyle$^{2.55}(cst)$) *and*
5 $\forall\, x, y \in laystys$
6 $(_1\, x \neq y$ *impl*
7 C ^x • 'layout style identifier' $\neq$ C ^y • 'layout style identifier' $_1)_0)$

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 2.17

Predicate "is a presentation style set" (clause 6.3.5)

An entity *prestys* is a presentation style set if it is a non-empty collection of presentation styles (3,4) whose identifiers are unique within that collection (5-7).

Definition 2.17

1 $\forall\, prestys$
2 $(_0$ IsPresentationStyleSet($prestys$) *iff*
3 IsNeCol$^{1.1}$($prestys$) *and*
4 $\forall\, cst \in prestys$ (IsPresentationStyle$^{2.56}$(cst)) *and*
5 $\forall\, x, y \in prestys$
6 $(_1\, x \neq y$ *impl*
7 C ^x • 'presentation style identifier' $\neq$ C ^y • 'presentation style identifier' $_1)_0)$

7.2 Constituents

In constituents, the values of defaultable attributes need not be specified explicitly — as opposed to mandatory or non-mandatory attributes. In these cases the formal specification provides a placeholder for the value which is either specified elsewhere in the document description or specified in the International Standard as the default value (5.1.2.4). Such a "dummy value" is formally defined as satisfying the predicate IsPlaceholder$^{1.19}$.

Semiformal Description 2.18

Predicate "is a protected part" (clause 6.6)

A protected part consists of sealed document profile descriptions, enciphered document profile descriptions, pre-enciphered document body part descriptions and post-enciphered document body part descriptions.

Definition 2.18

1 $\forall\, ppart$
2 $(_0$ IsProtectedPart($ppart$) *iff*
3 $\forall\, cst \in ppart$
4 $(_1$ IsSealedDocumentProfileDescription$^{2.57}$(cst) *or*
5 IsEncipheredDocumentProfileDescription$^{2.58}$(cst) *or*
6 IsPreEncipheredDocumentBodyPartDescription$^{2.59}$(cst) *or*
7 IsPostEncipheredDocumentBodyPartDescription$^{2.60}$(cst)$_1)_0)$

Semiformal Description 2.19

Predicate "is a root class description" (clauses 7.5.6, 7.5.7)

A root class description is either a logical root class description or a layout root class description.

Definition 2.19

1 $\forall\, cst$
2 $(_0$ IsRootClassDescription(cst) *iff*
3 IsLogicalRootClassDescription$^{2.20}$(cst) *or* IsLayoutRootClassDescription$^{2.24}$(cst)$_0)$

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 2.20

Predicate "is a logical root class description" (clause 7.5.6)

A logical root class description is an attribute set (nomination) which may contain the attributes 'application comments', 'bindings', 'default value lists', 'generator for subordinates', 'layout style', 'object class identifier', 'object type', 'protection', 'resource', 'sealed', 'user-readable comments' and 'user-visible name'. The attributes 'object class identifier' and 'object type' are both required and have 2 and 'document logical root' as their respective values (13-14). None of the attribute values is defaultable (12).

Definition 2.20

```
1   ∀ cst
2   (0 IsLogicalRootClassDescription(cst) iff
3     IsAttributeSet2.61(cst) and
4     NAMS1.18(cst) ⊇ ['object class identifier';     'object type'] and
5     NAMS1.18(cst) ⊆ ['application comments';        'bindings';
6                      'default value lists';         'generator for subordinates';
7                      'layout style';                'object class identifier';
8                      'object type';                 'protection';
9                      'resource';                    'sealed';
10                     'user-readable comments';      'user-visible name'] and
11    ∀ a ∈̂ cst •
12    (1 not IsPlaceholder1.19(C a) and
13      (2 N a = 'object class identifier' impl C a = [ → 2 → ] 2) and
14      (3 N a = 'object type' impl C a = 'document logical root'3)1) and
15    not (cst)DescribesClassImSubOf2.150(cst)0)
```

Semiformal Description 2.21

Predicate "is a composite logical object class description" (clause 7.5.6)

A composite logical object class description is a logical root class description or an attribute set (nomination) which may contain the attributes 'application comments', 'bindings', 'default value lists', 'generator for subordinates', 'layout style', 'object class identifier', 'object type', 'protection', 'resource', 'sealed', 'user-readable comments' and 'user-visible name'. The attributes 'object class identifier' and 'object type' are both required and have a logical object class identifier and a 'composite logical object' as their respective values (14-15). None of the attribute values is defaultable (13).

Definition 2.21

```
1   ∀ cst
2   (0 IsCompositeLogicalObjectClassDescription(cst) iff
3     IsLogicalRootClassDescription2.20(cst) or
4     (1 IsAttributeSet2.61(cst) and
5       NAMS1.18(cst) ⊇ ['object class identifier';     'object type'] and
6       NAMS1.18(cst) ⊆ ['application comments';        'bindings';
7                        'default value lists';         'generator for subordinates';
8                        'layout style';                'object class identifier';
9                        'object type';                 'protection';
10                       'resource';                    'sealed';
11                       'user-readable comments';      'user-visible name'] and
12      ∀ a ∈̂ cst •
13      (2 not IsPlaceholder1.19(C a) and
14        (3 N a = 'object class identifier' impl IsLogicalObjectClassId2.80(C a) 3) and
15        (4 N a = 'object type' impl C a = 'composite logical object'4)2)1)0)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 2.22

Predicate "is a basic logical object class description" (clause 7.5.6)

A basic logical object class description is an attribute set (nomination) which may contain the attributes 'application comments', 'bindings', 'content architecture class', 'content generator', 'content portions', 'enciphered', 'layout style', 'object class identifier', 'object type', 'presentation style', 'protection', 'resource', 'sealed', 'user-readable comments' and 'user-visible name'. The attributes 'object class identifier' and 'object type' are both required and have a logical object class identifier and a 'basic logical object' as their respective values (15-16). None of the attribute values is defaultable (14).

Definition 2.22

1 $\forall\, cst$
2 $({}_{0}$ IsBasicLogicalObjectClassDescription(cst) *iff*
3 IsAttributeSet$^{2.61}(cst)$ *and*
4 NAMS$^{1.18}(cst) \supseteq$ ['object class identifier'; 'object type'] *and*
5 NAMS$^{1.18}(cst) \subseteq$ ['application comments'; 'bindings';
6 'content architecture class'; 'content generator';
7 'content portions'; 'enciphered';
8 'layout style'; 'object class identifier';
9 'object type'; 'presentation style';
10 'protection'; 'resource';
11 'sealed'; 'user-readable comments';
12 'user-visible name'] *and*
13 $\forall\, a \in \hat{}cst \bullet$
14 $({}_{1}$ *not* IsPlaceholder$^{1.19}$(C a) *and*
15 $({}_{2}$N a = 'object class identifier' *impl* IsLogicalObjectClassId$^{2.80}$(C a) $_{2}$) *and*
16 $({}_{3}$N a = 'object type' *impl* C a = 'basic logical object'$_{3})_{1})_{0}$)

Semiformal Description 2.23

Predicate "is a logical object class description" (clause 7.5.6)

A logical object class description is either a basic logical object class description or a composite logical object class description.

Definition 2.23

1 $\forall\, cst$
2 $({}_{0}$ IsLogicalObjectClassDescription(cst) *iff*
3 IsBasicLogicalObjectClassDescription$^{2.22}(cst)$ *or*
4 IsCompositeLogicalObjectClassDescription$^{2.21}(cst)_{0}$)

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 2.24

Predicate "is a layout root class description" (clause 7.5.7)

A layout root class description is an attribute set (nomination) which may contain the attributes 'application comments', 'balance', 'bindings', 'default value lists', 'generator for subordinates', 'object class identifier', 'object type', 'resource', 'sealed', 'user-readable comments' and 'user-visible name'. The attributes 'object class identifier' and 'object type' are both required and have 0 and 'document layout root' as their respective values (17–18). None of the attribute values is defaultable (12). The value of 'balance' is either 'null' or a sequence of immediately subordinate object class identifiers. In the latter case, the attribute 'generator for subordinates' must be specified (13–16).

NOTE — The required correspondence between these object class identifiers and the value of 'generator for subordinates' (5.4.2.1) can only be checked during the layout process.

Definition 2.24

1 $\forall\, cst$
2 $(_0$IsLayoutRootClassDescription(cst) *iff*
3 IsAttributeSet$^{2.61}$(cst) *and*
4 NAMS$^{1.18}$(cst) $\supseteq$ ['object class identifier'; 'object type'] *and*
5 NAMS$^{1.18}$(cst) $\subseteq$ ['application comments'; 'balance';
6 'bindings'; 'default value lists';
7 'generator for subordinates'; 'object class identifier';
8 'object type'; 'resource';
9 'sealed'; 'user-readable comments';
10 'user-visible name'] *and*
11 $\forall\, a \in \hat{}\, cst$ •
12 $(_1$ *not* IsPlaceholder$^{1.19}$(C a) *and*
13 $(_2$N a = 'balance' *impl*
14 $(_3$C a = 'null' *or*
15 $(_4$ 'generator for subordinates' $\in$ NAMS$^{1.18}$(cst) *and*
16 $\forall\, so \in$ C $\hat{}\, cst$ • 'balance' • ((C so)IsClassIdImSubIn$^{2.142}$(cst)) $_4)_3)_2)$ *and*
17 $(_5$N a = 'object class identifier' *impl* C a = [$\rightarrow$ 0 $\rightarrow$] $_5)$ *and*
18 $(_6$N a = 'object type' *impl* C a = 'document layout root' $_6)_1)$ *and*
19 *not* (cst)DescribesClassImSubOf$^{2.150}$(cst)$_0$)

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 2.25

Predicate "is a page set class description" (clause 7.5.7)

A page set class description is an attribute set (nomination) which may contain the attributes 'application comments', 'balance', 'bindings', 'default value lists', 'generator for subordinates', 'layout stream categories', 'layout stream sub-categories', 'object class identifier', 'object type', 'resource', 'sealed', 'user-readable comments' and 'user-visible name'. The attributes 'object class identifier' and 'object type' are both required and have a layout object class identifier and 'page set' as their respective values (18, 19). None of the attribute values is defaultable (13). The value of 'balance' is either 'null' or a sequence of immediately subordinate object class identifiers. In the latter case, the attribute 'generator for subordinates' must be specified (14–17).

NOTE — The required correspondence between these object class identifiers and the value of 'generator for subordinates' (5.4.2.1) can only be checked during the layout process.

Definition 2.25

```
1   ∀ cst
2   (0 IsPageSetClassDescription(cst) iff
3     IsAttributeSet^2.61(cst) and
4     NAMS^1.18(cst) ⊇ ['object class identifier';          'object type'] and
5     NAMS^1.18(cst) ⊆ ['application comments';             'balance';
6                       'bindings';                         'default value lists';
7                       'generator for subordinates';       'layout stream categories';
8                       'layout stream sub-categories';     'object class identifier';
9                       'object type';                      'resource';
10                      'sealed';                           'user-readable comments';
11                      'user-visible name'] and
12    ∀ a ∈ ^cst •
13    (1 not IsPlaceholder^1.19(C a) and
14      (2 N a = 'balance' impl
15        (3 C a = 'null' or
16          (4 'generator for subordinates' ∈ NAMS^1.18(cst) and
17            ∀ so ∈̂ C ^cst • 'balance' • ((C so)IsClassIdImSubIn^2.142(cst)) 4)3)2) and
18      (5 N a = 'object class identifier' impl IsLayoutObjectClassId^2.81(C a) 5) and
19      (6 N a = 'object type' impl C a = 'page set' 6)1)0)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Semiformal Description 2.26

Predicate "is a composite page class description" (clause 7.5.7)

A composite page class description is an attribute set (nomination) which may contain the attributes 'application comments', 'balance', 'bindings', 'colour', 'colour of layout object', 'default value lists', 'dimensions', 'generator for subordinates', 'layout stream categories', 'layout stream sub-categories', 'medium type', 'object class identifier', 'object type', 'page position', 'resource', 'sealed', 'transparency', 'user-readable comments' and 'user-visible name'. The attributes 'object class identifier' and 'object type' are both required and have a layout object class identifier and a 'composite or basic page' as their respective values (23, 24). None of the attribute values is defaultable (16). The value of 'balance' is either 'null' or a sequence of immediately subordinate object class identifiers. In the latter case, the attribute 'generator for subordinates' must be specified (17–20). The sub-parameters 'rule A', 'rule B' and 'variable page height' may not be specified for the attribute 'dimensions' (21, 22).

NOTE — The required correspondence between these object class identifiers and the value of 'generator for subordinates' (5.4.2.1) can only be checked during the layout process.

Definition 2.26

```
1   ∀ cst
2   (0 IsCompositePageClassDescription(cst) iff
3     IsAttributeSet^2.61(cst) and
4     NAMS^1.18(cst) ⊇ ['object class identifier';     'object type'] and
5     NAMS^1.18(cst) ⊆ ['application comments';     'balance';
6                       'bindings';                 'colour';
7                       'colour of layout object';  'default value lists';
8                       'dimensions';               'generator for subordinates';
9                       'layout stream categories'; 'layout stream sub-categories';
10                      'medium type';              'object class identifier';
11                      'object type';              'page position';
12                      'resource';                 'sealed';
13                      'transparency';             'user-readable comments';
14                      'user-visible name'] and
15    ∀ a ∈ ^cst •
16    (1 not IsPlaceholder^1.19(C a) and
17      (2 N a = 'balance' impl
18        (3 C a = 'null' or
19          (4 'generator for subordinates' ∈ NAMS^1.18(cst) and
20            ∀ so ∈̂ C ^cst • 'balance' • ((C so)IsClassIdImSubIn^2.142(cst)) 4)3)2) and
21      (5 N a = 'dimensions' impl
22        (6 ∀ b ∈̂ a • • (7 N b ∉ ['rule A'; 'rule B'; 'variable page height'] 7)6)5) and
23      (8 N a = 'object class identifier' impl IsLayoutObjectClassId^2.81(C a) 8) and
24      (9 N a = 'object type' impl C a = 'composite or basic page' 9)1)0)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 2.27

Predicate "is a basic page class description" (clause 7.5.7)

A basic page class description is an attribute set (nomination) which may contain the attributes 'application comments', 'bindings', 'colour', 'colour of layout object', 'content architecture class', 'content background colour', 'content colour table', 'content foreground colour', 'content generator', 'content portions', 'dimensions', 'enciphered', 'layout stream categories', 'layout stream sub-categories', 'medium type', 'object class identifier', 'object colour table', 'object type', 'page position', 'presentation attributes', 'presentation style', 'resource', 'sealed', 'transparency', 'user-readable comments' and 'user-visible name'. The attributes 'object class identifier' and 'object type' are both required and have a layout object class identifier and a 'composite or basic page' as their respective values (22, 23). None of the attribute values is defaultable (19). The sub-parameters 'rule A' and 'rule B' may not be specified for the attribute 'dimensions' (20, 21).

Definition 2.27

```
1   ∀ cst
2   (0 IsBasicPageClassDescription(cst) iff
3     IsAttributeSet^2.61(cst) and
4     NAMS^1.18(cst) ⊇ ['object class identifier';      'object type'] and
5     NAMS^1.18(cst) ⊆ ['application comments';         'bindings';
6                       'colour';                       'colour of layout object';
7                       'content architecture class';   'content background colour';
8                       'content colour table';         'content foreground colour';
9                       'content generator';            'content portions';
10                      'dimensions';                   'enciphered';
11                      'layout stream categories';     'layout stream sub-categories';
12                      'medium type';                  'object class identifier';
13                      'object colour table';          'object type';
14                      'page position';                'presentation attributes';
15                      'presentation style';           'resource';
16                      'sealed';                       'transparency';
17                      'user-readable comments';       'user-visible name'] and
18    ∀ a ∈ ^cst •
19    (1 not IsPlaceholder^1.19(C a) and
20      (2 N a = 'dimensions' impl
21        (3 ∀ b ∈ a • • (4 N b ∉ ['rule A'; 'rule B'] 4)3)2) and
22      (5 N a = 'object class identifier' impl IsLayoutObjectClassId^2.81(C a) 5) and
23      (6 N a = 'object type' impl C a = 'composite or basic page' 6)1)0)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 2.28

Predicate "is a frame class description" (clause 7.5.7)

A frame class description is an attribute set (nomination) which may contain the attributes 'application comments', 'balance', 'bindings', 'border', 'colour', 'colour of layout object', 'default value lists', 'dimensions', 'generator for subordinates', 'layout path', 'layout stream categories', 'layout stream sub-categories', 'logical source', 'object class identifier', 'object colour table', 'object type', 'permitted categories', 'position', 'resource', 'sealed', 'transparency', 'user-readable comments' and 'user-visible name'. The attributes 'object class identifier' and 'object type' are both required and have a layout object class identifier and 'frame' as their respective values (25, 26). None of the attribute values is defaultable (18). The value of 'balance' is either 'null' or a sequence of immediately subordinate object class identifiers (19–22). In the latter case, the attribute 'generator for subordinates' must be specified (21, 22). The sub-parameter 'variable page height' may not be specified for the attribute 'dimensions' (23, 24).

NOTE: The required correspondence between these object class identifiers and the value of 'generator for subordinates' (5.4.2.1) can only be checked during the layout process.

Definition 2.28

```
1   ∀ cst
2   (0 IsFrameClassDescription(cst) iff
3     IsAttributeSet^2.61(cst) and
4     NAMS^1.18(cst) ⊇ ['object class identifier';      'object type'] and
5     NAMS^1.18(cst) ⊆ ['application comments';        'balance';
6                       'bindings';                    'border';
7                       'colour';                      'colour of layout object';
8                       'default value lists';         'dimensions';
9                       'generator for subordinates';  'layout path';
10                      'layout stream categories';    'layout stream sub-categories';
11                      'logical source';              'object class identifier';
12                      'object colour table';         'object type';
13                      'permitted categories';        'position';
14                      'resource';                    'sealed';
15                      'transparency';                'user-readable comments';
16                      'user-visible name'] and
17  ∀ a ∈̂ ^cst •
18  (1 not IsPlaceholder^1.19(C a) and
19    (2 N a = 'balance' impl
20      (3 C a = 'null' or
21        (4 'generator for subordinates' ∈ NAMS^1.18(cst) and
22          ∀ so ∈̂ C ^cst • 'balance' • ((C so)IsClassIdImSubIn^2.142(cst)) 4)3)2) and
23    (5 N a = 'dimensions' impl
24      (6 ∀ b ∈̂ a • • (7 N b ≠ 'variable page height' 7)6)5) and
25    (8 N a = 'object class identifier' impl IsLayoutObjectClassId^2.81(C a) 8) and
26    (9 N a = 'object type' impl C a = 'frame' 9)1)0)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 2.29

Predicate "is a block class description" (clause 7.5.7)

A block class description is an attribute set (nomination) which may contain the attributes 'application comments', 'bindings', 'border', 'colour', 'colour of layout object', 'content architecture class', 'content background colour', 'content colour table', 'content foreground colour', 'content generator', 'content portions', 'dimensions', 'enciphered', 'layout stream categories', 'layout stream sub-categories', 'object class identifier', 'object colour table', 'object type', 'position', 'presentation attributes', 'presentation style', 'resource', 'sealed', 'transparency', 'user-readable comments' and 'user-visible name'. The attributes 'object class identifier' and 'object type' are both required and have a layout object class identifier and 'block' as their respective values (24, 25). None of the attribute values is defaultable (19). The sub-parameters 'rule A', 'rule B' and 'variable page height' may not be specified for the attribute 'dimensions' (20, 21). The parameter 'variable position' may not be specified for the attribute 'position' (22, 23).

Definition 2.29

1 $\forall\, cst$
2 $(_0$IsBlockClassDescription(cst) *iff*
3 IsAttributeSet$^{2.61}(cst)$ *and*
4 NAMS$^{1.18}(cst) \supseteq$ ['object class identifier'; 'object type'] *and*
5 NAMS$^{1.18}(cst) \subseteq$ ['application comments'; 'bindings';
6 'border'; 'colour';
7 'colour of layout object'; 'content architecture class';
8 'content background colour'; 'content colour table';
9 'content foreground colour'; 'content generator';
10 'content portions'; 'dimensions';
11 'enciphered'; 'layout stream categories';
12 'layout stream sub-categories'; 'object class identifier';
13 'object colour table'; 'object type';
14 'position'; 'presentation attributes';
15 'presentation style'; 'resource';
16 'sealed'; 'transparency';
17 'user-readable comments'; 'user-visible name'] *and*
18 $\forall\, a \in \hat{}\, cst \bullet$
19 $(_1$ *not* IsPlaceholder$^{1.19}$(C a) *and*
20 $(_2$N a = 'dimensions' *impl*
21 $(_3 \forall\, b \in a \bullet\bullet\ (_4$N $b \notin$ ['rule A'; 'rule B'; 'variable page height'] $_4)_3)_2)$ *and*
22 $(_5$N a = 'position' *impl*
23 $(_6 \forall\, b \in a \bullet\ (_7$N $b \neq$ 'variable position' $_7)_6)_5)$ *and*
24 $(_8$N a = 'object class identifier' *impl* IsLayoutObjectClassId$^{2.81}$(C a) $_8)$ *and*
25 $(_9$N a = 'object type' *impl* C a = 'block' $_9)_1)_0)$

Semiformal Description 2.30

Predicate "is a layout object class description" (clause 7.5.7)

A layout object class description is either a basic layout object class description or composite layout object class description.

Definition 2.30

1 $\forall\, cst$
2 $(_0$IsLayoutObjectClassDescription(cst) *iff*
3 IsBasicLayoutObjectClassDescription$^{2.33}(cst)$ *or*
4 IsCompositeLayoutObjectClassDescription$^{2.32}(cst)_0)$

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 2.31

Predicate "is a composite object class description" (clauses 7.5.6, 7.5.7)

A composite object class description is either a composite logical object class description or a composite layout object class description.

Definition 2.31

1 $\forall cst$
2 ($_0$IsCompositeObjectClassDescription(cst) *iff*
3 IsCompositeLogicalObjectClassDescription$^{2.21}$(cst) *or*
4 IsCompositeLayoutObjectClassDescription$^{2.32}$(cst)$_0$)

Semiformal Description 2.32

Predicate "is a composite layout object class description" (clause 7.5.7)

A composite layout object class description is a layout root class description or a page set class description or a composite page class description or a frame class description.

Definition 2.32

1 $\forall cst$
2 ($_0$IsCompositeLayoutObjectClassDescription(cst) *iff*
3 IsLayoutRootClassDescription$^{2.24}$(cst) *or*
4 IsPageSetClassDescription$^{2.25}$(cst) *or*
5 IsCompositePageClassDescription$^{2.26}$(cst) *or*
6 IsFrameClassDescription$^{2.28}$(cst)$_0$)

Semiformal Description 2.33

Predicate "is a basic layout object class description" (clause 7.5.7)

A basic layout object class description is either a basic page class description or a block class description.

Definition 2.33

1 $\forall cst$
2 ($_0$IsBasicLayoutObjectClassDescription(cst) *iff*
3 IsBasicPageClassDescription$^{2.27}$(cst) *or*
4 IsBlockClassDescription$^{2.29}$(cst)$_0$)

Semiformal Description 2.34

Predicate "is a basic object class description" (clauses 7.5.6, 7.5.7)

A basic object class description is either a basic logical object class description or a basic layout object class description.

Definition 2.34

1 $\forall cst$
2 ($_0$IsBasicObjectClassDescription(cst) *iff*
3 IsBasicLogicalObjectClassDescription$^{2.22}$(cst) *or*
4 IsBasicLayoutObjectClassDescription$^{2.33}$(cst)$_0$)

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 2.35

Predicate "is an object class description" (clauses 7.5.6, 7.5.7)

An object class description is either a logical object class description or a layout object class description. The "binding value" is a collection of catenations with two components where the second one is a bindings value expression.

Definition 2.35

1 $\forall\, cst$
2 $({}_0$IsObjectClassDescription(cst) *iff*
3 $({}_1$ IsLogicalObjectClassDescription$^{2.23}(cst)$ *or*
4 IsLayoutObjectClassDescription$^{2.30}(cst)_1$) *and*
5 $\forall\, a \in \hat{}\, cst \bullet$
6 $({}_2$N a = 'bindings' *impl*
7 $({}_3$IsNeCol$^{1.1}$(C a) *and*
8 $\forall\, b \in$ (C a) $\exists\, l, r$
9 $({}_4 b = [\rightarrow l \rightarrow r \rightarrow]$ *and* IsBindingsValueExpression$^{2.63}(r)\ {}_4)_3)_2)_0)$

Semiformal Description 2.36

Predicate "is a root description" (clauses 7.5.6, 7.5.7)

A root description is either a logical root description or a layout root description.

Definition 2.36

1 $\forall\, cst$
2 $({}_0$IsRootDescription(cst) *iff*
3 IsLogicalRootDescription$^{2.37}(cst)$ *or* IsLayoutRootDescription$^{2.41}(cst)_0)$

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 2.37

Predicate "is a logical root description" (clause 7.2.1)

A logical root description is an attribute set (nomination) which contains the attributes 'application comments', 'bindings', 'object identifier', 'object type', 'protection', 'subordinates', 'user-readable comments', 'user-visible name' and, optionally, the attributes 'enciphered', 'layout style', 'default value lists', 'object class' and 'sealed'. The attributes 'object class' and 'object identifier' have a logical object class identifier and 3 as their respective values (16, 17). The attribute 'object type' has 'document logical root' as its value (18, 19). If the value of the attribute 'object type' is defaulted the attribute 'object class' must be present (20).

Definition 2.37

```
1   ∀ cst
2   (0 IsLogicalRootDescription(cst) iff
3     IsAttributeSet^2.61(cst) and
4     NAMS^1.18(cst) ⊇ ['application comments';     'bindings';
5                       'object identifier';       'object type';
6                       'protection';              'subordinates';
7                       'user-readable comments';  'user-visible name'] and
8     NAMS^1.18(cst) ⊆ ['application comments';     'bindings';
9                       'default value lists';     'enciphered';
10                      'layout style';            'object class';
11                      'object identifier';       'object type';
12                      'protection';              'sealed';
13                      'subordinates';            'user-readable comments';
14                      'user-visible name'] and
15    ∀ a ∈̂ ^cst •
16    (1 (2 N a = 'object class' impl IsLogicalObjectClassId^2.80(C a) 2) and
17      (3 N a = 'object identifier' impl C a = [ → 3 → ] 3) and
18      (4 N a = 'object type' impl
19        (5 C a = 'document logical root' or IsPlaceholder^1.19(C a) 5)4)1) and
20    (6 IsPlaceholder^1.19(C ^cst • 'object type') impl 'object class' ∈ NAMS^1.18(cst)6)0)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 2.38

Predicate "is a composite logical object description" (clauses 7.2.1, 7.2.2)

A composite logical object description is a composite logical root description or an attribute set (nomination) which contains the attributes 'application comments', 'bindings', 'object type', 'object identifier', 'protection', 'subordinates' 'user-readable comments', 'user-visible name' and, optionally, the attributes 'enciphered', 'layout style', 'default value lists', 'object class' and 'sealed'. The attributes 'object class' and 'object identifier' have a logical object class identifier and a logical object identifier as their respective values (17, 18). The attribute 'object type' has 'composite logical object' as its value (19, 20). If the value of the attribute 'object type' is defaulted the attribute 'object class' must be present (21).

Definition 2.38

```
1   ∀ cst
2   (0 IsCompositeLogicalObjectDescription(cst) iff
3     IsLogicalRootDescription^2.37(cst) or
4     (1 IsAttributeSet^2.61(cst) and
5       NAMS^1.18(cst) ⊇ ['application comments';    'bindings';
6                         'object identifier';      'object type';
7                         'protection';             'subordinates';
8                         'user-readable comments'; 'user-visible name'] and
9       NAMS^1.18(cst) ⊆ ['application comments';    'bindings';
10                        'default value lists';    'enciphered';
11                        'layout style';           'object class';
12                        'object identifier';      'object type';
13                        'protection';             'sealed';
14                        'subordinates';           'user-readable comments';
15                        'user-visible name'] and
16      ∀ a ∈ ^cst •
17      (2 (3 N a = 'object class' impl IsLogicalObjectClassId^2.80(C a) 3) and
18        (4 N a = 'object identifier' impl IsLogicalObjectId^2.77(C a) 4) and
19        (5 N a = 'object type' impl
20          (6 C a = 'composite logical object' or IsPlaceholder^1.19(C a) 6)5)2) and
21      (7 IsPlaceholder^1.19(C ^cst • 'object type') impl 'object class' ∈ NAMS^1.18(cst)7)1)0)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Semiformal Description 2.39

Predicate "is a basic logical object description" (clause 7.2.3)

A basic logical object description is an attribute set (nomination) which contains the attributes 'application comments', 'bindings', 'content architecture class', 'object identifier', 'object type', 'protection', 'user-readable comments', 'user-visible name' and, optionally, the attributes 'alternative', 'enciphered', 'layout style', 'content portions', 'content generator', 'object class', 'presentation style', 'primary' and 'sealed'. The attributes 'object class' and 'object identifier' have a logical object class identifier and a logical object identifier as their respective values (18, 19). The attribute 'object type' has 'basic logical object' as its value (20, 21). At least one of the attributes 'object class', 'content portions' and 'content generator' must be specified for this constituent (22). If the value of the attribute 'object type' is defaulted the attribute 'object class' must be present (23).

Definition 2.39

1 $\forall\, cst$
2 $({}_{0}$IsBasicLogicalObjectDescription(cst) *iff*
3 IsAttributeSet$^{2.61}(cst)$ *and*
4 NAMS$^{1.18}(cst) \supseteq$ ['application comments'; 'bindings';
5 'content architecture class'; 'object identifier';
6 'object type'; 'protection';
7 'user-readable comments'; 'user-visible name'] *and*
8 NAMS$^{1.18}(cst) \subseteq$ ['alternative'; 'application comments';
9 'bindings'; 'content architecture class';
10 'content generator'; 'content portions';
11 'enciphered'; 'layout style';
12 'object class'; 'object type';
13 'object identifier'; 'presentation style';
14 'primary'; 'protection';
15 'sealed'; 'user-readable comments';
16 'user-visible name'] *and*
17 $\forall\, a \in \hat{}cst \bullet$
18 $({}_{1}({}_{2}$N a = 'object class' *impl* IsLogicalObjectClassId$^{2.80}$(C a) $_{2}$) *and*
19 $({}_{3}$N a = 'object identifier' *impl* IsLogicalObjectId$^{2.77}$(C a) $_{3}$) *and*
20 $({}_{4}$N a = 'object type' *impl*
21 $({}_{5}$C a = 'basic logical object' *or* IsPlaceholder$^{1.19}$(C a) $_{5})_{4})_{1}$) *and*
22 $({}_{6}$NAMS$^{1.18}(cst) \cap$ ['content portions'; 'object class'; 'content generator'] $\neq$ [ ]$_{6}$) *and*
23 $({}_{7}$IsPlaceholder$^{1.19}$(C $\hat{}cst \bullet$ 'object type') *impl* 'object class' $\in$ NAMS$^{1.18}(cst)_{7})_{0}$)

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 2.40

Predicate "is a logical object description" (clauses 7.2.2, 7.2.3)

A logical object description is either a basic logical object description or a composite logical object description. The "binding value", if specified, is a collection of catenations with two components where the second one is a bindings value expression.

Definition 2.40

```
1   ∀ cst
2   (0 IsLogicalObjectDescription(cst) iff
3     (1 IsBasicLogicalObjectDescription2.39(cst) or
4       IsCompositeLogicalObjectDescription2.38(cst)1) and
5     ∀ a ∈ ^cst •
6     (2 (3 N a = 'bindings' and not IsPlaceholder1.19(C a) 3) impl
7       (4 IsNeCol1.1(C a) and
8         ∀ b ∈ (C a) ∃ l, r
9         (5 b = [ → l → r → ] and IsBindingsValueExpression2.63(r) 5)4)2)0)
```

Semiformal Description 2.41

Predicate "is a layout root description" (clause 7.3.1.1)

A layout root description is an attribute set (nomination) which contains the attributes 'application comments', 'balance', 'bindings', 'object identifier', 'object type', 'subordinates', 'user-readable comments', 'user-visible name' and, optionally, the attributes 'default value lists', 'enciphered', 'object class' and 'sealed'. The attributes 'object class' and 'object identifier' have a layout object class identifier and 1 as their respective values (18, 19). The attribute 'object type' has 'document layout root' as its value (20, 21). If the value of the attribute 'object type' is defaulted the attribute 'object class' must be present (22). The value of the attribute 'balance' is either 'null' or a sequence of immediately subordinate object identifiers (15–17).

Definition 2.41

```
1   ∀ cst
2   (0 IsLayoutRootDescription(cst) iff
3     IsAttributeSet2.61(cst) and
4     NAMS1.18(cst) ⊇ ['application comments';     'balance';
5                      'bindings';                 'object identifier';
6                      'object type';              'subordinates';
7                      'user-readable comments';   'user-visible name'] and
8     NAMS1.18(cst) ⊆ ['application comments';     'balance';
9                      'bindings';                 'default value lists';
10                     'enciphered';               'object type';
11                     'object class';             'object identifier';
12                     'sealed';                   'subordinates';
13                     'user-readable comments';   'user-visible name'] and
14    ∀ a ∈ ^cst •
15    (1 (2 N a = 'balance' impl
16        (3 IsPlaceholder1.19(C a) or C a = 'null' or
17          ∀ so ∈ C ^cst • 'balance' • ((C so)IsIdImSubIn2.143(cst)) 3)2) and
18      (4 N a = 'object class' impl IsLayoutObjectClassId2.81(C a) 4) and
19      (5 N a = 'object identifier' impl C a = [ → 1 → ] 5) and
20      (6 N a = 'object type' impl
21        (7 C a = 'document layout root' or IsPlaceholder1.19(C a) 7)6)1) and
22    (8 IsPlaceholder1.19(C ^cst • 'object type') impl 'object class' ∈ NAMS1.18(cst)8)0)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 2.42

Predicate "is a page set description" (clause 7.3.1.2)

A page set description is an attribute set (nomination) which contains the attributes 'application comments', 'balance', 'bindings', 'layout stream categories', 'layout stream sub-categories', 'object identifier', 'object type', 'subordinates', 'user-readable comments', 'user-visible name' and, optionally, the attributes 'default value lists', 'enciphered', 'object class' and 'sealed'. The attributes 'object class' and 'object identifier' have a layout object class identifier and a layout object identifier as their respective values (20, 21). The attribute 'object type' has 'page set' as its value (22, 23). If the value of the attribute 'object type' is defaulted the attribute 'object class' must be present (24). The value of the attribute 'balance' is either 'null' or a sequence of immediately subordinate object identifiers (17-19).

Definition 2.42

```
1   ∀ cst
2   (0 IsPageSetDescription(cst) iff
3     IsAttributeSet2.61(cst) and
4     NAMS1.18(cst) ⊇ ['application comments';        'balance';
5                      'bindings';                    'layout stream categories';
6                      'layout stream sub-categories'; 'object identifier';
7                      'object type';                 'subordinates';
8                      'user-readable comments';      'user-visible name'] and
9     NAMS1.18(cst) ⊆ ['application comments';        'balance';
10                     'bindings';                    'default value lists';
11                     'enciphered';                  'layout stream categories';
12                     'layout stream sub-categories'; 'object class';
13                     'object identifier';           'object type';
14                     'sealed';                      'subordinates';
15                     'user-readable comments';      'user-visible name'] and
16    ∀ a ∈ ^cst •
17    (1 (2 N a = 'balance' impl
18        (3 IsPlaceholder1.19(C a) or C a = 'null' or
19          ∀ so ∈ C ^cst • 'balance' • ((C so)IsIdImSubIn2.143(cst)) 3)2) and
20      (4 N a = 'object class' impl IsLayoutObjectClassId2.81(C a) 4) and
21      (5 N a = 'object identifier' impl IsLayoutObjectId2.78(C a) 5) and
22      (6 N a = 'object type' impl
23        (7 C a = 'page set' or IsPlaceholder1.19(C a) 7)6)1) and
24    (8 IsPlaceholder1.19(C ^cst • 'object type') impl 'object class' ∈ NAMS1.18(cst)8)0)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Semiformal Description 2.43

Predicate "is a composite page description" (clause 7.3.1.3)

A composite page description is an attribute set (nomination) which contains the attributes 'application comments', 'bindings', 'colour', 'colour of layout object', 'dimensions', 'layout stream categories', 'layout stream sub-categories', 'medium type', 'object colour table', 'object identifier', 'object type', 'page position', 'subordinates', 'transparency', 'user-readable comments', 'user-visible name' and, optionally, the attributes 'balance', 'default value lists', 'enciphered', 'imaging order', 'object class' and 'sealed'. The attributes 'object class' and 'object identifier' have a layout object class identifier and a layout object identifier as their respective values (34, 35). The attribute 'object type' has 'composite or basic page' as its value (36, 37). If the value of the attribute 'object type' is defaulted the attribute 'object class' must be present (38). The value of the attribute 'balance' is either 'null' or a sequence of immediately subordinate object identifiers (24–27). The value of the attribute 'imaging order' is a sequence of immediately subordinate object identifiers (31–33). The sub-parameters 'rule A', 'rule B' and 'variable page height' may not be specified for the attribute 'dimensions' (28–30).

Definition 2.43

```
1   ∀ cst
2   (0 IsCompositePageDescription(cst) iff
3     IsAttributeSet^2.61(cst) and
4     NAMS^1.18(cst) ⊇ ['application comments';        'bindings';
5                       'colour';                      'colour of layout object';
6                       'dimensions';                  'layout stream categories';
7                       'layout stream sub-categories'; 'medium type';
8                       'object colour table';         'object identifier';
9                       'object type';                 'page position';
10                      'subordinates';                'transparency';
11                      'user-readable comments';      'user-visible name'] and
12    NAMS^1.18(cst) ⊆ ['application comments';        'balance';
13                      'bindings';                    'colour';
14                      'colour of layout object';     'default value lists';
15                      'dimensions';                  'enciphered';
16                      'imaging order';               'layout stream categories';
17                      'layout stream sub-categories'; 'medium type';
18                      'object colour table';         'object type';
19                      'object class';                'object identifier';
20                      'page position';               'sealed';
21                      'subordinates';                'transparency';
22                      'user-readable comments';      'user-visible name'] and
23    ∀ a ∈ ^cst •
24    (1 (2 N a = 'balance' impl
25         (3 IsPlaceholder^1.19(C a) or C a = 'null' or
26           ∀ so ∈ a •
27           ∀ so ∈ C ^cst • 'balance' • (4 (C so)IsIdImSubIn^2.143(cst) 4)3)2) and
28      (5 N a = 'dimensions' impl (6 IsPlaceholder^1.19(C a) or
29         (7 ∀ b ∈ a • •
30           (8 N b ∉ ['rule A'; 'rule B'; 'variable page height'] 8)7)6)5) and
31      (9 N a = 'imaging order' impl
32         ∀ so ∈ a •
33         ((C ^cst • 'object identifier' // [ → C so → ])IsIdImSubIn^2.143(cst)) 9) and
34      (10 N a = 'object class' impl IsLayoutObjectClassId^2.81(C a) 10) and
35      (11 N a = 'object identifier' impl IsLayoutObjectId^2.78(C a) 11) and
36      (12 N a = 'object type' impl
37         (13 C a = 'composite or basic page' or IsPlaceholder^1.19(C a) 13)12)1) and
38    (14 IsPlaceholder^1.19(C ^cst • 'object type') impl 'object class' ∈ NAMS^1.18(cst)14)0)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 2.44

Predicate "is a basic page description" (clause 7.3.1.3)

A basic page description is an attribute set (nomination) which contains the attributes 'application comments', 'bindings', 'colour', 'colour of layout object', 'content architecture class', 'content background colour', 'content colour table', 'content foreground colour', 'dimensions', 'layout stream categories', 'layout stream sub-categories', 'medium type', 'object colour table', 'object identifier', 'object type', 'page position', 'presentation attributes', 'transparency', 'user-readable comments', 'user-visible name' and, optionally, the attributes 'alternative', 'content portions', 'enciphered', 'presentation style', 'primary', 'object class' and 'sealed'. The attributes 'object class' and 'object identifier' have a layout object class identifier and a layout object identifier as their respective values (31, 32). The attribute 'object type' has 'composite or basic page' as its value (33, 34). At least one of the attributes 'object class' and 'content portions' must be specified for this constituent (35). If the value of the attribute 'object type' is defaulted the attribute 'object class' must be present (36). The sub-parameters 'rule A' and 'rule B' may not be specified for the attribute 'dimensions' (29, 30).

Definition 2.44

1 $\forall\, cst$
2 $({}_{0}\,\text{IsBasicPageDescription}(cst)$ *iff*
3 $\text{IsAttributeSet}^{2.61}(cst)$ *and*
4 $\text{NAMS}^{1.18}(cst) \supseteq$ ['application comments'; 'bindings';
5 'colour'; 'colour of layout object';
6 'content architecture class'; 'content background colour';
7 'content colour table'; 'content foreground colour';
8 'dimensions'; 'layout stream categories';
9 'layout stream sub-categories'; 'medium type';
10 'object colour table'; 'object identifier';
11 'object type'; 'page position';
12 'presentation attributes'; 'transparency';
13 'user-readable comments'; 'user-visible name'] *and*
14 $\text{NAMS}^{1.18}(cst) \subseteq$ ['alternative'; 'application comments';
15 'bindings'; 'colour';
16 'colour of layout object'; 'content architecture class';
17 'content background colour'; 'content colour table';
18 'content foreground colour'; 'content portions';
19 'dimensions'; 'enciphered';
20 'layout stream categories'; 'layout stream sub-categories';
21 'medium type'; 'object class';
22 'object colour table'; 'object identifier';
23 'object type'; 'page position';
24 'presentation attributes'; 'presentation style';
25 'primary'; 'sealed';
26 'transparency'; 'user-readable comments';
27 'user-visible name'] *and*
28 $\forall\, a \in \hat{}\,cst \bullet$
29 $({}_{1}({}_{2}\text{N}\ a = \text{'dimensions'}$ *impl* $({}_{3}\,\text{IsPlaceholder}^{1.19}(\text{C}\ a)$ *or*
30 $({}_{4}\forall\, b \in a \bullet\bullet\ ({}_{5}\text{N}\ b \notin \text{['rule A'; 'rule B']}\ {}_{5})_{4})_{3})_{2})$ *and*
31 $({}_{6}\text{N}\ a = \text{'object class'}$ *impl* $\text{IsLayoutObjectClassId}^{2.81}(\text{C}\ a)\ {}_{6})$ *and*
32 $({}_{7}\text{N}\ a = \text{'object identifier'}$ *impl* $\text{IsLayoutObjectId}^{2.78}(\text{C}\ a)\ {}_{7})$ *and*
33 $({}_{8}\text{N}\ a = \text{'object type'}$ *impl*
34 $({}_{9}\text{C}\ a = \text{'composite or basic page'}$ *or* $\text{IsPlaceholder}^{1.19}(\text{C}\ a)\ {}_{9})_{8})_{1})$ *and*
35 $({}_{10}\,\text{NAMS}^{1.18}(cst) \cap \text{['content portions'; 'object class']} \neq [\,]_{10})$ *and*
36 $({}_{11}\,\text{IsPlaceholder}^{1.19}(\text{C}\ \hat{}\,cst \bullet \text{'object type'})$ *impl* $\text{'object class'} \in \text{NAMS}^{1.18}(cst)_{11})_{0})$

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 2.45

Predicate "is a page description" (clause 7.3.1.3)

A page description is either a composite page description or a basic page description.

Definition 2.45

1 $\forall cst$
2 $({}_{0}\text{IsPageDescription}(cst)$ *iff*
3 $\text{IsCompositePageDescription}^{2.43}(cst)$ *or* $\text{IsBasicPageDescription}^{2.44}(cst)_{0})$

Semiformal Description 2.46

Predicate "is a frame description" (clause 7.3.1.4)

A frame description is an attribute set (nomination) which contains the attributes 'application comments', 'bindings', 'border', 'colour', 'colour of layout object', 'dimensions', 'layout path', 'layout stream categories', 'layout stream sub-categories', 'object colour table', 'object identifier', 'object type', 'permitted categories', 'position', 'subordinates', 'transparency', 'user-readable comments', 'user-visible name' and, optionally, the attributes 'balance', 'default value lists', 'enciphered', 'imaging order', 'object class' and 'sealed'. The attributes 'object class' and 'object identifier' have a layout object class identifier and a layout object identifier as their respective values (35, 36). The attribute 'object type' has 'frame' as its value (37, 38). If the value of the attribute 'object type' is defaulted the attribute 'object class' must be present (41). The value of the attribute 'balance' is either 'null' or a sequence of immediately subordinate object identifiers (26–28). The value of the attribute 'imaging order' is a sequence of immediately subordinate object identifiers (32–34). The sub-parameters 'rule A', 'rule B' and 'variable page height' may not be specified for the attribute 'dimensions' (29–31). The parameter 'variable position' may not be specified for the attribute 'position' (39, 40).

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Definition 2.46

```
1   ∀ cst
2   (0 IsFrameDescription(cst) iff
3     IsAttributeSet2.61(cst) and
4     NAMS1.18(cst) ⊇ ['application comments';        'bindings';
5                      'border';                      'colour';
6                      'colour of layout object';     'dimensions';
7                      'layout path';                 'layout stream categories';
8                      'layout stream sub-categories'; 'object colour table';
9                      'object identifier';           'object type';
10                     'permitted categories';        'position';
11                     'subordinates';                'transparency';
12                     'user-readable comments';      'user-visible name'] and
13    NAMS1.18(cst) ⊆ ['application comments';        'balance';
14                     'bindings';                    'border';
15                     'colour';                      'colour of layout object';
16                     'default value lists';         'dimensions';
17                     'enciphered';                  'imaging order';
18                     'layout path';                 'layout stream categories';
19                     'layout stream sub-categories'; 'object class';
20                     'object colour table';         'object identifier';
21                     'object type';                 'permitted categories';
22                     'position';                    'sealed';
23                     'subordinates';                'transparency';
24                     'user-readable comments';      'user-visible name'] and
25    ∀ a ∈ ^cst •
26    (1(2 N a = 'balance' impl
27        (3 IsPlaceholder1.19(C a) or C a = 'null' or
28          ∀ so ∈ C ^cst • 'balance' • ((C so)IsIdImSubIn2.143(cst)) 3)2) and
29      (4 N a = 'dimensions' impl (5 IsPlaceholder1.19(C a) or
30        (6 ∀ b ∈ a • •
31          (7 N b ∉ ['rule A'; 'rule B'; 'variable page height'] 7)6)5)4) and
32      (8 N a = 'imaging order' impl
33        ∀ so ∈ a •
34        (9 (C ^cst • 'object identifier' // [ → C so → ])IsIdImSubIn2.143(cst) 9)8) and
35      (10 N a = 'object class' impl IsLayoutObjectClassId2.81(C a) 10) and
36      (11 N a = 'object identifier' impl IsLayoutObjectId2.78(C a) 11) and
37      (12 N a = 'object type' impl
38        (13 C a = 'frame' or IsPlaceholder1.19(C a) 13)12) and
39      (14 N a = 'position' impl
40        (15 ∀ b ∈ a • (16 N b ≠ 'variable position' 16)15)14)1) and
41    (17 IsPlaceholder1.19(C ^cst • 'object type') impl 'object class' ∈ NAMS1.18(cst)17)0)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 2.47

Predicate "is a block description" (clause 7.3.1.5)

A block description is an attribute set (nomination) which contains the attributes 'application comments', 'bindings', 'border', 'colour', 'colour of layout object', 'content architecture class', 'content background colour', 'content colour table', 'content foreground colour', 'dimensions', 'layout stream categories', 'layout stream sub-categories', 'object colour table', 'object identifier', 'object type', 'position', 'transparency', 'user-readable comments', 'user-visible name' and, optionally, the attributes 'alternative', 'content portions', 'enciphered', 'object class', 'presentation attributes', 'presentation style', 'primary' and 'sealed'. The attributes 'object class' and 'object identifier' have a layout object class identifier and a layout object identifier as their respective values (32, 33). The attribute 'object type' has 'block' as its value (34, 35). At least one of the attributes 'object class' and 'content portions' must be specified for this constituent (38). If the value of the attribute 'object type' is defaulted the attribute 'object class' must be present (39). The sub-parameters 'rule A', 'rule B' and 'variable page height' may not be specified for the attribute 'dimensions' (28–31). The parameter 'variable position' may not be specified for the attribute 'position' (36, 37).

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Definition 2.47

```
1   ∀ cst
2   (0 IsBlockDescription(cst) iff
3     IsAttributeSet^2.61(cst) and
4     NAMS^1.18(cst) ⊇ ['application comments';          'bindings';
5                       'border';                       'colour';
6                       'colour of layout object';      'content architecture class';
7                       'content background colour';    'content colour table';
8                       'content foreground colour';    'dimensions';
9                       'layout stream categories';     'layout stream sub-categories';
10                      'object colour table';          'object identifier';
11                      'object type';                  'position';
12                      'transparency';                 'user-readable comments';
13                      'user-visible name'     ł
14    NAMS^1.18(cst) ⊆ ['alternative';                  'application comments';
15                      'bindings';                     'border';
16                      'colour';                       'colour of layout object';
17                      'content architecture class';   'content background colour';
18                      'content colour table';         'content foreground colour';
19                      'content portions';             'dimensions';
20                      'enciphered';                   'layout stream categories';
21                      'layout stream sub-categories'; 'object class';
22                      'object colour table';          'object identifier';
23                      'object type';                  'position';
24                      'presentation attributes';      'presentation style';
25                      'primary';                      'sealed';
26                      'transparency';                 'user-readable comments';
27                      'user-visible name'] and
28    ∀ a ∈̂ ^cst •
29    (1 (2 N a = 'dimensions' impl (3 IsPlaceholder^1.19(C a) or
30          (4 ∀ b ∈ a • •
31            (5 N b ∉ ['rule A'; 'rule B'; 'variable page height'] 5)4)3)2) and
32      (6 N a = 'object class' impl IsLayoutObjectClassId^2.81(C a) 6) and
33      (7 N a = 'object identifier' impl IsLayoutObjectId^2.78(C a) 7) and
34      (8 N a = 'object type' impl
35        (9 C a = 'block' or IsPlaceholder^1.19(C a) 9)8) and
36      (10 N a = 'position' impl
37        (11 ∀ b ∈̂ a • (12 N b ≠ 'variable position' 12)11)10)1) and
38    (13 NAMS^1.18(cst) ∩ ['content portions'; 'object class'] ≠ [ ]13) and
39    (14 IsPlaceholder^1.19(C ^cst • 'object type') impl 'object class' ∈ NAMS^1.18(cst)14)0)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 2.48

Predicate "is a layout object description" (clause 9.3.5.4)

A layout object description is either a basic layout object description or a composite layout object description. The "binding value", if specified, is a collection of catenations with two components where the second one is either an octet string (string literal) or an integer (numeric literal) or an object identifier.

Definition 2.48

```
1   ∀ cst
2   (0 IsLayoutObjectDescription(cst) iff
3     (1 IsBasicLayoutObjectDescription^2.52(cst) or
4       IsCompositeLayoutObjectDescription^2.51(cst)1) and
5     ∀ a ∈ ~cst •
6     (2 (3 N a = 'bindings' and not IsPlaceholder^1.19(C a) 3) impl
7       (4 IsNeCol^1.1(C a) and
8         ∀ b ∈ (C a) ∃ l, r
9         (5 b = [ → l → r → ] and
10        (6 IsOctetString^1.10(r) or IsInt(r) or IsObjectId^2.79(r) 6)5)4)2)0)
```

Semiformal Description 2.49

Predicate "is a composite object description" (clauses 7.2, 7.3)

A composite object description is either composite logical object description or a composite layout object description.

Definition 2.49

```
1   ∀ cst
2   (0 IsCompositeObjectDescription(cst) iff
3     IsCompositeLogicalObjectDescription^2.38(cst) or
4     IsCompositeLayoutObjectDescription^2.51(cst)0)
```

Semiformal Description 2.50

Predicate "is a basic object description" (clauses 7.2, 7.3)

A basic object description is either basic logical object description or a basic layout object description.

Definition 2.50

```
1   ∀ cst
2   (0 IsBasicObjectDescription(cst) iff
3     IsBasicLogicalObjectDescription^2.39(cst) or
4     IsBasicLayoutObjectDescription^2.52(cst)0)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 2.51

Predicate "is a composite layout object description" (clause 7.3)

A composite layout object description is a layout root description or a page set description or a composite page description or a frame description.

Definition 2.51

1 $\forall cst$
2 $(_0$ IsCompositeLayoutObjectDescription(cst) *iff*
3 IsLayoutRootDescription$^{2.41}(cst)$ *or*
4 IsPageSetDescription$^{2.42}(cst)$ *or*
5 IsCompositePageDescription$^{2.43}(cst)$ *or*
6 IsFrameDescription$^{2.46}(cst)_0$)

Semiformal Description 2.52

Predicate "is a basic layout object description" (clause 7.3)

A basic layout object description is either a basic page description or a block description.

Definition 2.52

1 $\forall cst$
2 $(_0$ IsBasicLayoutObjectDescription(cst) *iff*
3 IsBasicPageDescription$^{2.44}(cst)$ *or*
4 IsBlockDescription$^{2.47}(cst)_0$)

Semiformal Description 2.53

Predicate "is an object description" (clauses 7.2, 7.3)

An object description is either a logical object description or a layout object description.

Definition 2.53

1 $\forall cst$
2 $(_0$ IsObjectDescription(cst) *iff*
3 IsLogicalObjectDescription$^{2.40}(cst)$ *or*
4 IsLayoutObjectDescription$^{2.48}(cst)_0$)

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 2.54

Predicate "is a content portion description" (clause 7.1.3)

A content portion description is an attribute set (nomination) which contains the attributes 'application comments', 'type of coding', either one or both of the attributes 'content identifier logical' and 'content identifier layout' and, optionally, the attributes 'alternative representation', 'coding attributes' and 'content information'. A content portion description is a character content portion description, a raster graphics content portion description or a geometric graphics content portion description (10-12).

Definition 2.54

1 $\forall\, cont$
2 $({}_{0}$IsContentPortionDescription$(cont)$ *iff*
3 IsAttributeSet$^{2.61}(cont)$ *and*
4 NAMS$^{1.18}(cont) \cap$ ['content identifier layout'; 'content identifier logical'] $\neq$ [] *and*
5 NAMS$^{1.18}(cont) \supseteq$ ['application comments'; 'type of coding'] *and*
6 NAMS$^{1.18}(cont) \subseteq$ ['application comments'; 'alternative representation';
7 'coding attributes'; 'content identifier layout';
8 'content identifier logical'; 'content information';
9 'type of coding'] *and*
10 $({}_{1}$IsCharacterContentPortionDescription$^{6.11}(cont)$ *or*
11 IsRasterGraphicsContentPortionDescription$^{7.7}(cont)$ *or*
12 IsGeometricGraphicsContentPortionDescription$^{8.5}(cont)$ $)_{0})$

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Semiformal Description 2.55

Predicate "is a layout style" (clause 6.3.5)

A layout style is an attribute set (nomination) which contains the attribute 'layout style identifier' and, optionally, the attributes 'application comments', 'block alignment', 'concatenation', 'derived from', 'fill order', 'floatability range', 'indivisibility', 'layout category', 'layout object class', 'logical stream category', 'logical stream sub-category', 'new layout object', 'offset', 'same layout object', 'sealed', 'separation', 'synchronization', 'user-readable comments' and 'user-visible name'. The attributes 'layout object class' and 'layout style identifier' have a layout object class identifier and a layout style identifier as their respective values (17, 18). The value of the attribute 'derived from' is a layout style identifier (19). None of the attribute values is defaultable (16).

Definition 2.55

1 $\forall\, cst$
2 $({}_{0}$IsLayoutStyle(cst) *iff*
3 IsAttributeSet$^{2.61}(cst)$ *and*
4 NAMS$^{1.18}(cst) \supseteq$ ['layout style identifier'] *and*
5 NAMS$^{1.18}(cst) \subseteq$ ['application comments'; 'block alignment';
6 'concatenation'; 'derived from';
7 'fill order'; 'floatability range';
8 'indivisibility'; 'layout category';
9 'layout object class'; 'layout style identifier';
10 'logical stream category'; 'logical stream sub-category';
11 'new layout object'; 'offset';
12 'same layout object'; 'sealed';
13 'separation'; 'synchronization';
14 'user-readable comments'; 'user-visible name'] *and*
15 $\forall\, a \in \hat{}\, cst \bullet$
16 $({}_{1}$ *not* IsPlaceholder$^{1.19}$(C a) *and*
17 $({}_{2}$N a = 'layout object class' *impl* IsLayoutObjectClassId$^{2.81}$(C a) $_{2}$) *and*
18 $({}_{3}$N a = 'layout style identifier' *impl* IsLayoutStyleId$^{2.100}$(C a) $_{3}$) *and*
19 $({}_{4}$N a = 'derived from' *impl* IsLayoutStyleId$^{2.100}$(C a) $_{4})_{1})_{0})$

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Semiformal Description 2.56

Predicate "is a presentation style" (clause 6.3.5)

A presentation style is an attribute set (nomination) which contains the attribute 'presentation style identifier' and, optionally, the attributes 'border', 'colour', 'colour of layout object', 'content background colour', 'content colour table', 'content foreground colour', 'derived from', 'object colour table', 'presentation attributes', 'sealed', 'transparency' 'user-readable comments' and 'user-visible name'. The attribute 'presentation style identifier' has a presentation style identifier as its value (14). The attribute 'derived from' has a presentation style identifier as its values (15). None of the attribute values is defaultable (13).

Definition 2.56

1 $\forall\, cst$
2 $(_0$IsPresentationStyle(cst) *iff*
3 IsAttributeSet$^{2.61}(cst)$ *and*
4 NAMS$^{1.18}(cst) \supseteq$ ['presentation style identifier'] *and*
5 NAMS$^{1.18}(cst) \subseteq$ ['border'; 'colour';
6 'colour of layout object'; 'content background colour';
7 'content colour table'; 'content foreground colour';
8 'derived from'; 'object colour table';
9 'presentation attributes'; 'presentation style identifier';
10 'sealed'; 'transparency';
11 'user-readable comments'; 'user-visible name'] *and*
12 $\forall\, a \in \hat{}\, cst \bullet$
13 $(_1$ *not* IsPlaceholder$^{1.19}$(C a) *and*
14 $(_2$N a = 'presentation style identifier' *impl* IsPresentationStyleId$^{2.110}$(C $a)_2$) *and*
15 $(_3$N a = 'derived from' *impl* IsPresentationStyleId$^{2.110}$(C $a)_3)_1)_0$)

Semiformal Description 2.57

Predicate "is a sealed document profile description" (clause 6.6)

A sealed document profile description is an attribute set (nomination) which contains the attribute 'protected part identifier' and, optionally, the attribute 'sealed document profile information'. The attribute 'protected part identifier' has a protected part identifier as its value whose first integer is 6 (8, 9). None of the attribute values is defaultable (7).

Definition 2.57

1 $\forall\, cst$
2 $(_0$IsSealedDocumentProfileDescription(cst) *iff*
3 IsAttributeSet$^{2.61}(cst)$ *and*
4 NAMS$^{1.18}(cst) \supseteq$ ['protected part identifier'] *and*
5 NAMS$^{1.18}(cst) \subseteq$ ['protected part identifier'; 'sealed document profile information'] *and*
6 $\forall\, a \in \hat{}\, cst \bullet$
7 $(_1$ *not* IsPlaceholder$^{1.19}$(C a) *and*
8 $(_2$N a = 'protected part identifier' *impl*
9 $(_3$IsProtectedPartId$^{2.122}$(C a) *and* HEAD$^{1.13}$(C a) = $6_3)_2)_1)_0$)

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 2.58

Predicate "is an enciphered document profile description" (clause 6.6)

An enciphered document profile description is an attribute set (nomination) which contains the attribute 'protected part identifier' and, optionally, the attribute 'enciphered information'. The attribute 'protected part identifier' has a protected part identifier as its value whose first integer is 7 (8, 9). None of the attribute values is defaultable (7).

Definition 2.58

1 $\forall\, cst$
2 $(_0$IsEncipheredDocumentProfileDescription(cst) *iff*
3 IsAttributeSet$^{2.61}(cst)$ *and*
4 NAMS$^{1.18}(cst) \supseteq$ ['protected part identifier'] *and*
5 NAMS$^{1.18}(cst) \subseteq$ ['protected part identifier'; 'enciphered information'] *and*
6 $\forall\, a \in \hat{}cst \bullet$
7 $(_1$ *not* IsPlaceholder$^{1.19}$(C a) *and*
8 $(_2$N a = 'protected part identifier' *impl*
9 $(_3$IsProtectedPartId$^{2.122}$(C a) *and* HEAD$^{1.13}$(C a) = $7_3)_2)_1)_0)$

Semiformal Description 2.59

Predicate "is a pre-enciphered document body part description" (clause 6.6)

A pre-enciphered document body part description is an attribute set (nomination) which contains the attribute 'protected part identifier' and, optionally, the attribute 'enciphered information'. The attribute 'protected part identifier' has a protected part identifier as its value whose first integer is 8 (8, 9). None of the attribute values is defaultable (7).

Definition 2.59

1 $\forall\, cst$
2 $(_0$IsPreEncipheredDocumentBodyPartDescription(cst) *iff*
3 IsAttributeSet$^{2.61}(cst)$ *and*
4 NAMS$^{1.18}(cst) \supseteq$ ['protected part identifier'] *and*
5 NAMS$^{1.18}(cst) \subseteq$ ['protected part identifier'; 'enciphered information'] *and*
6 $\forall\, a \in \hat{}cst \bullet$
7 $(_1$ *not* IsPlaceholder$^{1.19}$(C a) *and*
8 $(_2$N a = 'protected part identifier' *impl*
9 $(_3$IsProtectedPartId$^{2.122}$(C a) *and* HEAD$^{1.13}$(C a) = $8_3)_2)_1)_0)$

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 2.60

Predicate is a post-enciphered document body part description" (clause 6.6)

A post-enciphered document body part description is an attribute set (nomination) which contains the attribute 'protected part identifier' and, optionally, the attribute 'enciphered information'. The attribute 'protected part identifier' has a protected part identifier as its value whose first integer is 9 (8, 9). None of the attribute values is defaultable (7).

Definition 2.60

1 $\forall\, cst$
2 $(_0$IsPostEncipheredDocumentBodyPartDescription(cst) *iff*
3 IsAttributeSet$^{2.61}$(cst) *and*
4 NAMS$^{1.18}$(cst) $\supseteq$ ['protected part identifier'] *and*
5 NAMS$^{1.18}$(cst) $\subseteq$ ['protected part identifier'; 'enciphered information'] *and*
6 $\forall\, a \in \hat{}cst$ •
7 $(_1$ *not* IsPlaceholder$^{1.19}$(C a) *and*
8 $(_2$N a = 'protected part identifier' *impl*
9 $(_3$IsProtectedPartId$^{2.122}$(C a) *and* HEAD$^{1.13}$(C a) = $9_3)_2)_1)_0)$

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

7.3 Attributes

Semiformal Description 2.61

Predicate "is an attribute set" (clause 9)

In this definition the attribute names are linked with the ranges of the attribute values.

Definition 2.61

```
1   ∀ as
2   (0 IsAttributeSet(as) iff
3     (1 IsNeNom1.2(as) and
4       ∀ a ∈̂ as •
5       (2 (3 N a = 'alternative' impl
6           IsObjectId2.79(C a) 3) and
7         (4 N a = 'alternative representation' impl
8           IsAlternativeCharacterSetsString2.115(C a) 4) and
9         (5 N a = 'application comments' impl
10          IsOctetString1.10(C a) 5) and
11        (6 N a = 'balance' impl
12          (7 IsPlaceholder1.19(C a) or IsBalanceValue2.94(C a) 7)6) and
13        (8 N a = 'bindings' impl
14          (9 IsPlaceholder1.19(C a) or IsBindingsValue2.89(C a) 9)8) and
15        (10 N a = 'block alignment' impl
16          (11 IsPlaceholder1.19(C a) or IsBlockAlignmentValue2.101(C a) 11)10) and
17        (12 N a = 'border' impl
18          (13 IsPlaceholder1.19(C a) or IsBorderValue2.93(C a) 13)12) and
19        (14 N a = 'coding attributes' impl
20          (15 IsPlaceholder1.19(C a) or IsCodingAttributesValue2.116(C a) 15)14) and
21        (16 N a = 'colour' impl
22          (17 IsPlaceholder1.19(C a) or IsColourValue2.97(C a) 17)16) and
23        (18 N a = 'colour of layout object' impl
24          (19 IsPlaceholder1.19(C a) or IsColourExpression2.126(C a) 19)18) and
25        (20 N a = 'concatenation' impl
26          (21 IsPlaceholder1.19(C a) or IsConcatenationValue2.102(C a) 21)20) and
27        (22 N a = 'content architecture class' impl
28          (23 IsPlaceholder1.19(C a) or IsContentArchitectureClassValue2.87(C a) 23)22) and
29        (24 N a = 'content background colour' impl
30          (25 IsPlaceholder1.19(C a) or IsColourExpression2.126(C a) or
31            C a = 'content background transparency' 25)24) and
32        (26 N a = 'content colour table' impl
33          (27 IsPlaceholder1.19(C a) or IsColourTable2.131(C a) 27)26) and
34        (28 N a = 'content foreground colour' impl
35          (29 IsPlaceholder1.19(C a) or IsColourExpression2.126(C a) or
36            C a ∈ ['content foreground transparency'; 'implementation-defined'] 29)28) and
37        (30 N a = 'content generator' impl
38          IsStringExpression2.64(C a) 30) and
39        (31 N a = 'content identifier layout' impl
40          IsLayoutContentPortionId2.111(C a) 31) and
41        (32 N a = 'content identifier logical' impl
42          IsLogicalContentPortionId2.112(C a) 32) and
43        (33 N a = 'content information' impl
44          IsContentInformationValue2.114(C a) 33) and
45        (34 N a = 'content portions' impl
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
46          IsSeqOfDiffNnInt^1.22(C a) 34) and
47        (35 N a = 'default value lists' impl
48          IsDefaultValueListsValue^2.90(C a) 35) and
49        (36 N a = 'derived from' impl
50          (37 IsLayoutStyleId^2.100(C a) or IsPresentationStyleId^2.110(C a) 37)36) and
51        (38 N a = 'dimensions' impl
52          (39 IsPlaceholder^1.19(C a) or IsDimensionsValue^2.92(C a) 39)38) and
53        (40 N a = 'enciphered' impl
54          IsEncipheredValue^2.119(C a) 40) and
55        (41 N a = 'enciphered information' impl
56          IsOctetString^1.10(C a) 41) and
57        (42 N a = 'fill order' impl
58          (43 IsPlaceholder^1.19(C a) or IsFillOrderValue^2.103(C a) 43)42) and
59        (44 N a = 'floatability range' impl
60          (45 IsPlaceholder^1.19(C a) or IsFloatabilityRangeValue^2.134(C a) 45)44) and
61        (46 N a = 'generator for subordinates' impl
62          IsConstructionExpression^2.84(C a) 46) and
63        (47 N a = 'imaging order' impl
64          IsSeqOfDiffNnInt^1.22(C a) 47) and
65        (48 N a = 'indivisibility' impl
66          (49 IsPlaceholder^1.19(C a) or IsLayoutObjectReference^2.104(C a) 49)48) and
67        (50 N a = 'layout category' impl
68          (51 IsPlaceholder^1.19(C a) or C a = 'null' or IsLayoutCatName^2.105(C a) 51)50) and
69        (52 N a = 'layout object class' impl
70          (53 IsPlaceholder^1.19(C a) or C a = 'null' or
71            IsLayoutObjectClassId^2.81(C a) 53)52) and
72        (54 N a = 'layout path' impl
73          (55 IsPlaceholder^1.19(C a) or IsLayoutPathValue^2.95(C a) 55)54) and
74        (56 N a = 'layout stream categories' impl
75          (57 IsPlaceholder^1.19(C a) or C a = 'null' or IsSetOfLayoutCatNames^2.106(C a) 57)56) and
76        (58 N a = 'layout stream sub-categories' impl
77          (59 IsPlaceholder^1.19(C a) or C a = 'null' or IsSetOfLayoutCatNames^2.106(C a) 59)58) and
78        (60 N a = 'layout style' impl
79          IsLayoutStyleId^2.100(C a) 60) and
80        (61 N a = 'layout style identifier' impl
81          IsLayoutStyleId^2.100(C a) 61) and
82        (62 N a = 'logical source' impl
83          IsLogicalObjectClassId^2.80(C a) 62) and
84        (63 N a = 'logical stream category' impl
85          (64 IsPlaceholder^1.19(C a) or C a = 'null' or IsLayoutCatName^2.105(C a) 64)63) and
86        (65 N a = 'logical stream sub-category' impl
87          (66 IsPlaceholder^1.19(C a) or C a = 'null' or IsLayoutCatName^2.105(C a) 66)65) and
88        (67 N a = 'medium type' impl
89          (68 IsPlaceholder^1.19(C a) or IsMediumTypeValue^2.98(C a) 68)67) and
90        (69 N a = 'new layout object' impl
91          (70 IsPlaceholder^1.19(C a) or IsLayoutObjectReference^2.104(C a) 70)69) and
92        (71 N a = 'object class identifier' impl
93          IsObjectClassId^2.82(C a) 71) and
94        (72 N a = 'object class' impl
95          IsObjectClassId^2.82(C a) 72) and
96        (73 N a = 'object colour table' impl
97          (74 IsPlaceholder^1.19(C a) or IsColourTable^2.131(C a) 74)73) and
98        (75 N a = 'object identifier' impl
99          IsObjectId^2.79(C a) 75) and
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
100     (76 N a = 'object type' impl
101       (77 IsPlaceholder1.19(C a) or IsObjectTypeValue2.76(C a) 77)76) and
102     (78 N a = 'offset' impl
103       (79 IsPlaceholder1.19(C a) or IsOffsetValue2.107(C a) 79)78) and
104     (80 N a = 'page position' impl
105       (81 IsPlaceholder1.19(C a) or IsPairOfNnInt1.8(C a) 81)80) and
106     (82 N a = 'permitted categories' impl
107       (83 IsPlaceholder1.19(C a) or C a = 'null' or
108         IsSetOfLayoutCatNames2.106(C a) 83)82) and
109     (84 N a = 'position' impl
110       (85 IsPlaceholder1.19(C a) or IsPositionValue2.91(C a) 85)84) and
111     (86 N a = 'presentation attributes' impl
112       (87 IsPlaceholder1.19(C a) or IsPresentationAttributesValue2.88(C a) 87)86) and
113     (88 N a = 'presentation style' impl
114       IsPresentationStyleId2.110(C a) 88) and
115     (89 N a = 'presentation style identifier' impl
116       IsPresentationStyleId2.110(C a) 89) and
117     (90 N a = 'primary' impl
118       IsObjectId2.79(C a) 90) and
119     (91 N a = 'protected part identifier' impl
120       IsProtectedPartId2.122(C a) 91) and
121     (92 N a = 'protection' impl
122       (93 IsPlaceholder1.19(C a) or IsProtectionValue2.99(C a) 93)92) and
123     (94 N a = 'resource' impl
124       IsISO6937MSString1.20(C a)94) and
125     (95 N a = 'same layout object' impl
126       (96 IsPlaceholder1.19(C a) or IsSameLayoutObjectValue2.108(C a) 96)95) and
127     (97 N a = 'sealed' impl
128       IsSealedValue2.120(C a) 97) and
129     (98 N a = 'sealed document profile information' impl
130       IsSealedDocumentProfileInformationValue2.121(C a) 98) and
131     (99 N a = 'separation' impl
132       (100 IsPlaceholder1.19(C a) or IsSeparationValue2.109(C a) 100)99) and
133     (101 N a = 'subordinates' impl
134       IsSeqOfDiffNnInt1.22(C a) 101) and
135     (102 N a = 'synchronization' impl
136       (103 IsPlaceholder1.19(C a) or C a = 'null' or IsLogicalObjectId2.77(C a) or
137         IsObjectOrObjectClassIdExpression2.69(C a) 103)102) and
138     (104 N a = 'transparency' impl
139       (105 IsPlaceholder1.19(C a) or IsTransparencyValue2.96(C a) 105)104) and
140     (106 N a = 'type of coding' impl
141       IsTypeOfCodingValue2.113(C a) 106) and
142     (107 N a = 'user-readable comments' impl
143       (108 IsPlaceholder1.19(C a) or IsCharSeq2.117(C a) 108)107) and
144     (109 N a = 'user-visible name' impl
145       (110 IsPlaceholder1.19(C a) or IsCharSeq2.117(C a) 110)109)2)1)0)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 2.62

Predicate "is a profile attribute set of Part 2"

(The formal specification of the attribute set for the document profile has only been carried out here, as far as it is necessary for the formal specification of the document structures described in ISO/IEC 8613-2.)

A profile attribute set is a non-empty nomination with the indicated value ranges for the specified attributes.

Definition 2.62

```
1   ∀ as
2   (0 IsProfileAttributeSetPart2(as) iff
3     IsNeNom1.2(as) and
4     ∀ a ∈ ^as •
5     (1 (N a ∈ ['generic layout structure'; 'generic logical structure'] impl
6       C a ∈ ['partial generator set'; 'complete generator set'; 'factor set']) and
7      (N a = 'specific layout structure' impl C a = 'present') and
8      (N a = 'specific logical structure' impl C a = 'present') and
9      (N a = 'layout styles' impl C a = 'present') and
10     (N a = 'presentation styles' impl C a = 'present') and
11     (N a = 'document architecture class' impl
12      C a ∈ ['formatted'; 'formatted processable'; 'processable']) 1)0)
```

Semiformal Description 2.63

Predicate "is a bindings value expression" (clause 9.1.3)

A bindings value expression is a string expression or a numeric expression or an object or object class identifier expression.

Definition 2.63

```
1   ∀ v
2   (0 IsBindingsValueExpression(v) iff
3     IsStringExpression2.64(v) or
4     IsNumericExpression2.67(v) or
5     IsObjectOrObjectClassIdExpression2.69(v) 0)
```

Semiformal Description 2.64

Predicate "is a string expression" (clause 9.1.3.1)

A string expression is a sequence of one or more substring expressions.

Definition 2.64

```
1   ∀ v
2   (0 IsStringExpression(v) iff
3     (1 IsNeCat1.3(v) and
4     ∀ m ∈ ^v • (IsAtomicStringExpression2.65(C m))1)0)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 2.65

Predicate "is an atomic string expression" (clause 9.1.3.1)

An atomic string expression is an octet string (string literal) or a binding reference or a nomination, where the name is a string function and the component is a numeric expression.

Definition 2.65

```
1  ∀ v
2  (0 IsAtomicStringExpression(v) iff
3    IsOctetString^1.10(v) or IsBindingReference^2.72(v) or
4    ∃ n, c
5    (1 v = [n : c] and
6      IsStringFunction^2.66(n) and IsNumericExpression^2.67(c)1)0)
```

Semiformal Description 2.66

Predicate "is a string function" (clause 9.1.3.1)

A string function is one of the following: 'MAKE-STRING', 'UPPER-ALPHA', 'LOWER-ALPHA', 'UPPER-ROMAN', 'LOWER-ROMAN'.

Definition 2.66

```
1  ∀ v
2  (0 IsStringFunction(v) iff
3    v ∈ ['MAKE-STRING'; 'UPPER-ALPHA'; 'LOWER-ALPHA'; 'UPPER-ROMAN';
4          'LOWER-ROMAN']0)
```

Semiformal Description 2.67

Predicate "is a numeric expression" (clause 9.1.3.2)

A numeric expression is an integer (numeric literal) or a binding reference or a nomination, where the name is a numeric function and the component is a numeric expression (if the numeric function is 'INCREMENT' or 'DECREMENT') or an object identifier or an object or object class identifier expression (if the numeric function is 'ORDINAL').

Definition 2.67

```
1  ∀ v
2  (0 IsNumericExpression(v) iff
3    IsInt(v) or IsBindingReference^2.72(v) or
4    ∃ n, c
5    (1 v = [n : c] and IsNumericFunction^2.68(n) and
6      (2 n ∈ ['INCREMENT'; 'DECREMENT'] impl
7        IsNumericExpression^2.67(c)2) and
8      (3 n = 'ORDINAL' impl
9        (4 IsObjectId^2.79(c) or IsObjectOrObjectClassIdExpression^2.69(c) 4)3)1)0)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 2.68

Predicate "is a numeric function" (clause 9.1.3.2)

A numeric function is one of the following: 'INCREMENT', 'DECREMENT', 'ORDINAL'.

Definition 2.68

1 $\forall v$
2 $(_0$IsNumericFunction(v) *iff*
3 $v \in$ ['INCREMENT'; 'DECREMENT'; 'ORDINAL']$_0$)

Semiformal Description 2.69

Predicate "is an object or object class identifier expression"(clause 9.1.3.3)

An object or object class identifier expression is eithe the selection function 'CURRENT-OBJECT' or a nomination where the name is an object or object class selection function (except 'CURRENT-OBJECT') and the component is an object or object class identifier expression (if the selection function is 'SUPERIOR-OBJECT' or 'PRECEDING-OBJECT'), or a current instance function (if the selection function is 'CURRENT-INSTANCE'). (This is a recursive definition.)

Definition 2.69

1 $\forall v$
2 $(_0$IsObjectOrObjectClassIdExpression(v) *iff*
3 $v =$ 'CURRENT-OBJECT' *or*
4 $\exists n, c$
5 $(_1 v = [n : c]$ *and*
6 IsObjectOrObjectClassSelectionFunction$^{2.70}(n)$ *and*
7 $n \neq$ 'CURRENT-OBJECT' *and*
8 $(_2 n \in$ ['SUPERIOR-OBJECT'; 'PRECEDING-OBJECT'] *impl*
9 IsObjectOrObjectClassIdExpression$^{2.69}(c)_2$) *and*
10 $(_3 n =$ 'CURRENT-INSTANCE' *impl* IsCurrentInstanceFunction$^{2.75}(c)_3)_1)_0$)

Semiformal Description 2.70

Predicate "is an object or object class selection function" (clause 9.1.3.3)

An object or object class selection function is one of the following: 'CURRENT-OBJECT', 'CURRENT-INSTANCE', 'SUPERIOR-OBJECT', 'PRECEDING-OBJECT'.

Definition 2.70

1 $\forall v$
2 $(_0$IsObjectOrObjectClassSelectionFunction(v) *iff*
3 $v \in$ ['CURRENT-OBJECT'; 'CURRENT-INSTANCE'; 'SUPERIOR-OBJECT';
4 'PRECEDING-OBJECT']$_0$)

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 2.71

Predicate "is a binding reference expression" (clause 9.1.3.4)

A binding reference expression is an object identifier or an object class identifier or a binding selection function.

Definition 2.71

1 $\forall v$
2 $(_0$IsBindingReferenceExpression(v) *iff*
3 IsObjectId$^{2.79}(v)$ *or* IsObjectClassId$^{2.82}(v)$ *or*
4 IsBindingSelectionFunction$^{2.74}(v)_0)$

Semiformal Description 2.72

Predicate "is a binding reference" (clause 9.1.3.4)

A binding reference is a pair where the first component is a binding reference expression and the second one is a binding identifier.

Definition 2.72

1 $\forall v$
2 $(_0$IsBindingReference(v) *iff*
3 $\exists l, r$
4 $(_1 v = [\rightarrow l \rightarrow r \rightarrow]$ *and*
5 IsBindingReferenceExpression$^{2.71}(l)$ *and* IsBindingName$^{2.73}(r)_1)_0)$

Semiformal Description 2.73

Predicate "is a binding name" (clause 9.1.3.4)

A binding name is a string of characters from the minimum subrepertoire of ISO 6937-2.

Definition 2.73

1 $\forall v$
2 $(_0$IsBindingName(v) *iff*
3 IsISO6937MSString$^{1.20}(v)_0)$

Semiformal Description 2.74

Predicate "is a binding selection function" (clause 9.1.3.4)

A binding selection function is either 'CURRENT-OBJECT' or a nomination where the name is 'SUPERIOR' or 'PRECEDING' (in this case the component is an object or object class identifier expression) or the name is 'CURRENT-INSTANCE' (in this case the component is a current instance function).

Definition 2.74

1 $\forall v$
2 $(_0$IsBindingSelectionFunction(v) *iff*
3 $v =$ 'CURRENT-OBJECT' *or*
4 $\exists f$
5 $(_1 (_2 v =$ ['SUPERIOR' : $f]$ *and* IsObjectOrObjectClassIdExpression$^{2.69}(f)_2)$ *or*
6 $(_3 v =$ ['PRECEDING' : $f]$ *and* IsObjectOrObjectClassIdExpression$^{2.69}(f)_3)$ *or*
7 $(_4 v =$ ['CURRENT-INSTANCE' : $f]$ *and* IsCurrentInstanceFunction$^{2.75}(f)_4)_1)_0)$

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 2.75

Predicate "is a current instance function" (clauses 9.1.3.5)

A current instance function is a pair where the first component is either an object class identifier or an object type value, and the second one is an object identifier or an object class identifier or an object or object class identifier expression.

Definition 2.75

1 $\forall v$
2 $({}_0$IsCurrentInstanceFunction(v) *iff*
3 $\exists l, r$
4 $({}_1 v = [\rightarrow l \rightarrow r \rightarrow]$ *and*
5 $({}_2$IsObjectClassId$^{2.82}(l)$ *or* IsObjectTypeValue$^{2.76}(l)_2)$ *and*
6 $({}_3$IsObjectId$^{2.79}(r)$ *or* IsObjectClassId$^{2.82}(r)$ *or*
7 IsObjectOrObjectClassIdExpression$^{2.69}(r)_3)_1)_0)$

Semiformal Description 2.76

Predicate "is an object type value" (clause 9.3.1.1)

An object type value is one of the following: 'basic logical object', 'block', 'composite logical object', 'document layout root', 'document logical root', 'frame', 'composite or basic page', 'page set'.

Definition 2.76

1 $\forall v$
2 $({}_0$IsObjectTypeValue(v) *iff*
3 $v \in$ ['basic logical object'; 'block'; 'composite logical object';
4 'document layout root'; 'document logical root'; 'frame';
5 'composite or basic page'; 'page set'$]_0)$

Semiformal Description 2.77

Predicate "is a logical object identifier" (clause 9.3.1.2)

A logical object identifier is a sequence of non-negative integers where the first integer is 3.

Definition 2.77

1 $\forall v$
2 $({}_0$IsLogicalObjectId(v) *iff*
3 IsSeqOfNnInt$^{1.23}(v)$ *and* HEAD$^{1.13}(v) = 3_0)$

Semiformal Description 2.78

Predicate "is a layout object identifier" (clause 9.3.1.2)

A layout object identifier is a sequence of non-negative integers where the first integer is 1.

Definition 2.78

1 $\forall v$
2 $({}_0$IsLayoutObjectId(v) *iff*
3 IsSeqOfNnInt$^{1.23}(v)$ *and* HEAD$^{1.13}(v) = 1_0)$

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 2.79

Predicate "is an object identifier" (clause 9.3.1.2)

An object identifier is either a logical object identifier or a layout object identifier.

Definition 2.79

1 $\forall v$
2 $({}_0\text{IsObjectId}(v)$ *iff*
3 $\text{IsLogicalObjectId}^{2.77}(v)$ *or* $\text{IsLayoutObjectId}^{2.78}(v)_0)$

Semiformal Description 2.80

Predicate "is a logical object class identifier" (clause 9.3.1.3)

A logical object class identifier is a sequence of non-negative integers where the first integer is 2.

Definition 2.80

1 $\forall v$
2 $({}_0\text{IsLogicalObjectClassId}(v)$ *iff*
3 $\text{IsSeqOfNnInt}^{1.23}(v)$ *and* $\text{HEAD}^{1.13}(v) = 2_0)$

Semiformal Description 2.81

Predicate "is a layout object class identifier" (clause 9.3.1.3)

A layout object class identifier is a sequence of non-negative integers where the first integer is 0.

Definition 2.81

1 $\forall v$
2 $({}_0\text{IsLayoutObjectClassId}(v)$ *iff*
3 $\text{IsSeqOfNnInt}^{1.23}(v)$ *and* $\text{HEAD}^{1.13}(v) = 0_0)$

Semiformal Description 2.82

Predicate "is an object class identifier" (clause 9.3.1.3)

An object class identifier is either a logical object class identifier or a layout object class identifier.

Definition 2.82

1 $\forall v$
2 $({}_0\text{IsObjectClassId}(v)$ *iff*
3 $\text{IsLogicalObjectClassId}^{2.80}(v)$ *or* $\text{IsLayoutObjectClassId}^{2.81}(v)_0)$

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 2.83

Predicate "is a sequence of object class identifiers"

Each element of a sequence of object class identifiers is an object class identifier.

Definition 2.83

```
1  ∀ v
2  (0 IsSeqOfObjectClassId(v) iff
3    IsCat(v) and ∀ m ∈̂ v • (IsObjectClassId^2.82(C m))0)
```

Semiformal Description 2.84

Predicate "is a construction expression" (clause 9.3.2.1)

A construction expression is either a construction type or a construction term.

Definition 2.84

```
1  ∀ m
2  (0 IsConstructionExpression(m) iff
3    IsConstructionType^2.85(m) or IsConstructionTerm^2.86(m)0)
```

Semiformal Description 2.85

Predicate "is a construction type" (clause 9.3.2.1)

A construction type is a sequence of two elements where the first one is 'sequence', 'aggregate' or 'choice' and the second one is a non-empty sequence, whose elements are construction terms.

Definition 2.85

```
1  ∀ m
2  (0 IsConstructionType(m) iff
3    ∃ l, q
4    (1 m = [ → l → q → ] and
5      l ∈ ['sequence'; 'aggregate'; 'choice'] and
6      IsNeCat^1.3(q) and ∀ t ∈̂ q • (IsConstructionTerm^2.86(C t)) 1)0)
```

Semiformal Description 2.86

Predicate "is a construction term" (clause 9.3.2.1)

A construction term is either a sequence with one element where the element is an object class identifier or a construction type, or a sequence of two elements where the first one is 'req', 'opt', 'rep' or 'optrep' and the second one is either an object class identifier or a construction type.

Definition 2.86

```
1  ∀ m
2  (0 IsConstructionTerm(m) iff
3    (1 ∃ q (2 m = [ → q → ] and (3 IsObjectClassId^2.82(q) or IsConstructionType^2.85(q)3) 2) or
4      ∃ l, q
5      (4 m = [ → l → q → ] and
6        l ∈ ['req'; 'opt'; 'rep'; 'optrep'] and
7        (IsObjectClassId^2.82(q) or IsConstructionType^2.85(q)) 4)1)0)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 2.87

Predicate "is a content architecture class value" (clause 9.3.4)

The value of the attribute 'content architecture class' is a character content architecture class value, a raster graphics content architecture class value or a geometric graphics content architecture class value.

Definition 2.87

1 $\forall v$
2 $(_0$ IsContentArchitectureClassValue(v) *iff*
3 $(_1$ IsCharacterContentArchitectureClassValue$^{6.26}(v)$ *or*
4 IsRasterGraphicsContentArchitectureClassValue$^{7.19}(v)$ *or*
5 IsGeometricGraphicsContentArchitectureClassValue$^{8.51}(v)$ $_1)_0)$

Semiformal Description 2.88

Predicate "is a presentation attributes value" (clause 9.4.4)

The value of the attribute 'presentation attributes' is a collection **where** each element is a character content presentation attribute, a raster graphics content presentation attribute or geometric graphics content presentation attribute.

Definition 2.88

1 $\forall v$
2 $(_0$ IsPresentationAttributesValue(v) *iff*
3 IsCol(v) *and*
4 $\forall a \in v$
5 $(_1$ IsCharacterContentPresentationAttribute$^{6.12}(a)$ *or*
6 IsRasterGraphicsContentPresentationAttribute$^{7.8}(a)$ *or*
7 IsGeometricGraphicsContentPresentationAttribute$^{8.6}(a)$ $_1)_0)$

Semiformal Description 2.89

Predicate "is a bindings value" (clause 9.3.5.4)

The value of the attribute 'bindings' is a set (3) where each element of the set is a catenation of two components (5.6). The first component is an ISO 6937 character string (6) which is unique within the set (9, 10). The second component is an attribute value expression, an integer, an octet string or an object identifier (7, 8).

Definition 2.89

1 $\forall v$
2 $(_0$ IsBindingsValue(v) *iff*
3 IsNeCol$^{1.1}(v)$ *and*
4 $\forall a \in v \; \exists f, l$
5 $(_1 a = [\rightarrow f \rightarrow l \rightarrow]$ *and*
6 IsISO6937MSString$^{1.20}(f)$ *and*
7 $(_2$ IsBindingsValueExpression$^{2.63}(l)$ *or* IsOctetString$^{1.10}(l)$ *or*
8 IsInt(l) *or* IsObjectId$^{2.79}(l)$ $_2)_1)$ *and*
9 $\forall p, q \in v$
10 $(_3 p \neq q$ *impl* HEAD$^{1.13}(p) \neq$ HEAD$^{1.13}(q)_3)_0)$

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 2.90

Predicate "is a default value lists value" (clause 9.3.5.5)

A value of the attribute 'default value lists' is a non-empty nomination where the name is 'composite logical object', 'basic logical object' 'composite or basic page', 'frame' or 'block' and the component is an attribute set, whose name set depends on the name. For 'block' the component can also be a set of attribute sets.

Definition 2.90

```
1   ∀ v
2   (0 IsDefaultValueListsValue(v) iff
3     IsNeNom^1.2(v) and
4     NAMS^1.18(v) ⊆ ['composite logical object'; 'basic logical object';
5                     'composite or basic page'; 'frame'; 'block'] and
6     ∀ a ∈ ˆv •
7     (1 (2 N a = 'composite logical object' impl
8         (3 IsAttributeSet^2.61(C a) and
9           NAMS^1.18(C a) ⊆ ['layout style'; 'protection'; 'sealed']3)2) and
10      (4 N a = 'basic logical object' impl
11        (5 IsAttributeSet^2.61(C a) and
12          NAMS^1.18(C a) ⊆ ['content architecture class'; 'layout style';
13                            'presentation style'; 'protection'; 'sealed']5)4) and
14      (6 N a = 'composite or basic page' impl
15        (7 IsAttributeSet^2.61(C a) and
16          NAMS^1.18(C a) ⊆ ['colour'; 'colour of layout object'; 'content architecture class'; 'dimensions';
17                            'medium type'; 'page position'; 'presentation attributes'; 'presentation style';
18                            'sealed'; 'transparency']7)6) and
19      (8 N a = 'frame' impl
20        (9 IsAttributeSet^2.61(C a) and
21          NAMS^1.18(C a) ⊆ ['colour'; 'colour of layout object'; 'border'; 'dimensions'; 'layout path';
22                            'object colour table'; 'permitted categories'; 'position'; 'sealed';
23                            'transparency']9)8) and
24      (10 N a = 'block' impl
25        (11 (12 IsAttributeSet^2.61(C a) and
26              NAMS^1.18(C a) ⊆ ['border'; 'colour'; 'content architecture class'; 'content background colour';
27                                'content colour table'; 'content foreground colour'; 'dimensions';
28                                'object colour table'; 'position'; 'presentation attributes'; 'presentation style';
29                                'sealed'; 'transparency']12) or
30          (13 IsCol(C a) and ∀ as ∈ (C a)
31            (14 IsAttributeSet^2.61(as) and
32              NAMS^1.18(as) ⊆ ['border'; 'colour'; 'content architecture class'; 'content background colour';
33                                'content colour table'; 'content foreground colour'; 'dimensions';
34                                'object colour table'; 'position'; 'presentation attributes'; 'presentation style';
35                                'sealed'; 'transparency']14)13)11)10)1)0)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Semiformal Description 2.91

Predicate "is a position value" (clause 9.4.1.1)

The value of the attribute 'position' is a nomination where the name is either 'fixed position' or 'variable position' and the component is a nomination again (4–7, 10, 11). For the name 'fixed position' the names are 'horizontal position' or 'vertical position' and the components are non-negative integers (6–9). For 'variable position' the names are 'offset', 'separation', 'alignment' and 'fill order' (10–12). For 'offset' the component is a nomination, where the names are 'leading offset', 'trailing offset', 'left-hand offset' and 'right-hand offset' and the components are non-negative integers (14–19). For 'separation' the component is also a nomination, where the names are 'leading-edge', 'trailing-edge' and 'centre separation' and the components are non-negative integers (20–25). For 'alignment' the component is either 'right-hand aligned', 'centred' or 'left-hand aligned' (26-28). For 'fill order' the component is either 'normal order' or 'reverse order' (29-31). The parameters and sub-parameters are independently defaultable (9, 15, 19, 21, 27, 30).

Definition 2.91

```
1   ∀ v
2   (0 IsPositionValue(v) iff
3     ∃ n, c
4     (1 v = [n : c] and
5     (2 n ∈ ['fixed position'; 'variable position'] and
6       (3 n = 'fixed position' impl
7         (4 IsNom(c) and
8           NAMS^1.18(c) = ['horizontal position'; 'vertical position'] and
9           ∀ a ∈̂ c • (IsPlaceholder^1.19(C a) or IsNnInt^1.7(C a)) 4)3) and
10      (5 n = 'variable position' impl
11        (6 IsNom(c) and
12          NAMS^1.18(c) = ['offset'; 'separation'; 'alignment'; 'fill order'] and
13          ∀ a ∈̂ c •
14          (7 (8 N a = 'offset' impl
15              (9 IsPlaceholder^1.19(C a) or
16                (10 IsNeNom^1.2(C a) and
17                  NAMS^1.18(C a) = ['leading offset'; 'trailing offset'; 'left-hand offset'
18                                    'right-hand offset'] and
19                  ∀ b ∈ a • (IsPlaceholder^1.19(C b) or IsNnInt^1.7(C b)) 10)9)8) and
20            (11 N a = 'separation' impl
21              (12 IsPlaceholder^1.19(C a) or
22              (13 IsNeNom^1.2(C a) and
23                NAMS^1.18(C a) = ['leading-edge'; 'trailing-edge';
24                                  'centre separation'] and
25                ∀ b ∈ a • (IsPlaceholder^1.19(C b) or IsNnInt^1.7(C b)) 13)12)11) and
26            (14 N a = 'alignment' impl
27              (15 IsPlaceholder^1.19(C a) or
28                C a ∈ ['right-hand aligned'; 'centred'; 'left-hand aligned'] 15)14) and
29            (16 N a = 'fill order' impl
30              (17 IsPlaceholder^1.19(C a) or
31                C a ∈ ['normal order'; 'reverse order'] 17)16)7)6)5)2)1)0)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 2.92

Predicate "is a dimensions value" (clause 9.4.1.2)

The value of the attribute 'dimensions' is a nomination of two components where the name is either 'horizontal dimension' or 'vertical dimension'. The component is a nomination again, where the name is 'maximum size' (the corresponding component ist 'applies'), 'fixed dimension' (the corresponding component is then a positive integer) or 'variable page height' (the corresponding component is any integer) or 'rule A' or 'rule B'. In the last two cases the component is once again a nomination where the name is 'minimum dimension' or 'maximum dimension' and the component is a positive integer. The parameters are independently defaultable. The sub-parameter 'variable page height' is not permitted for the parameter 'horizontal dimension'.

Definition 2.92

1 $\forall v$
2 $(_0$IsDimensionsValue(v) *iff*
3 IsNeNom$^{1.2}(v)$ *and*
4 NAMS$^{1.18}(v)$ = ['horizontal dimension'; 'vertical dimension'] *and*
5 $\forall a \in \hat{}v \bullet$
6 $(_1$IsPlaceholder$^{1.19}$(C a) *or*
7 $(_2 \exists n, c$
8 $(_3$C a = $[n : c]$ *and*
9 $n \in$ ['fixed dimension'; 'maximum size'; 'variable page height'; 'rule A'; 'rule B'] *and*
10 $(_4 (_5 n$ = 'fixed dimension' *impl* IsNat$(c)_5$) *and*
11 $(_6 n$ = 'maximum size' *impl* c = 'applies'$_6$) *and*
12 $(_7 n$ = 'variable page height' *impl* IsInt$(c)_7$) *and*
13 $(_8 n \in$ ['rule A'; 'rule B'] *impl*
14 $(_9$IsNom(c) *and*
15 NAMS$^{1.18}(c) \subseteq$ ['minimum dimension'; 'maximum dimension'] *and*
16 $\forall b \in \hat{}c \bullet$ (IsNat(C b)) $_9)_8)_4)$ *and*
17 N a = 'horizontal dimension' *impl* $n \neq$ 'variable page height'$_3)_2)_1)_0)$

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Semiformal Description 2.93

Predicate "is a border value" (clause 9.4.1.3)

The value of the attribute 'border' is a nomination of up to four components where the names are 'left-hand-edge', 'right-hand-edge', 'leading-edge' or 'trailing-edge'. The component is either 'null' or a nomination of up to four components, where the names are 'border line colour', 'border line width', 'border line type' or 'border freespace width'. The components with the name 'border line width' or 'border freespace width' are non-negative integers. The component with the name 'border line colour' is either 'implementation-defined' or a colour expression. The component with the name 'border line type' is one of the following: 'solid', 'dashed', 'dot', 'dash-dot', 'dash-dot-dot', 'invisible'.

Definition 2.93

```
1   ∀ v
2   (0 IsBorderValue(v) iff
3     IsNom(v) and
4     NAMS1.18(v) ⊆ ['left-hand-edge'; 'right-hand-edge'; 'trailing-edge';
5                    'leading-edge'] and
6     ∀ a ∈̂ ^v •
7     (1 C a = 'null' or
8       (2 IsNom(C a) and
9         NAMS1.18(C a) ⊆ ['border line colour'; 'border line width'; 'border line type';
10                         'border freespace width']) and
11        ∀ b ∈̂ a •
12        (3 (4 N b ∈ ['border line width'; 'border freespace width'] impl
13            IsNnInt1.7(C b)4) and
14          (5 N b = 'border line colour' impl
15            (6 C b = 'implementation-defined' or IsColourExpression2.126(C a) 6)5) and
16          (7 N b = 'border line type' impl
17            C b ∈ ['solid'; 'dashed'; 'dot'; 'dash-dot'; 'dash-dot-dot'; 'invisible']7)3)2)1)0)
```

Semiformal Description 2.94

Predicate "is a balance value" (clause 9.4.2.1)

The value of the attribute 'balance' is either 'null' or a sequence of more than one identifiers of immediately subordinate object descriptions (in the case of object descriptions) or object class descriptions (in the case of object class descriptions).

Definition 2.94

```
1   ∀ v
2   (0 IsBalanceValue(v) iff
3     v = 'null' or
4     (1 IsNeCat1.3(v) and LENGTH1.16 (v) > 1 and
5       (2 ∀ a ∈̂ ^v • (IsLayoutObjectId2.78(C a) ) or
6         ∀ a ∈̂ ^v • (IsLayoutObjectClassId2.81(C a) )2)1)0)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 2.95

Predicate "is a layout path value" (clause 9.4.2.2)

A layout path value is one of the following: '0°', '90°', '180°', '270°'.

Definition 2.95

1 $\forall v$
2 ($_{o}$IsLayoutPathValue(v) *iff*
3 $v \in$ ['0°'; '90°'; '180°'; '270°']$_{o}$)

Semiformal Description 2.96

Predicate "is a transparency value" (clause 9.4.3.2)

The value of the attribute 'transparency' is either 'transparent' or 'opaque'.

Definition 2.96

1 $\forall v$
2 ($_{o}$IsTransparencyValue(v) *iff*
3 $v \in$ ['transparent'; 'opaque']$_{o}$)

Semiformal Description 2.97

Predicate "is a colour value" (clause 9.4.3.3)

The value of the attribute 'colour' is either 'colour of media' or 'coloured'.

Definition 2.97

1 $\forall v$
2 ($_{o}$IsColourValue(v) *iff*
3 $v \in$ ['colour of media'; 'coloured']$_{o}$)

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 2.98

Predicate "is a medium type value" (clause 9.4.3.5)

The value of the attribute 'medium type' is a nomination of three components. For the name 'nominal page size' the component is a pair of positive integers, the first one specifying the horizontal, the second one the vertical dimensions of the page. For the name 'side of sheet' the component is 'recto', 'verso' or 'unspecified'. For the name 'colour of medium' the component is either a colour expression or 'unspecified'. The parameters are independently defaultable.

Definition 2.98

```
1   ∀ v
2   (0 IsMediumTypeValue(v) iff
3     IsNeNom1.2(v) and
4     NAMS1.18(v) = ['colour of medium';      inal page size'; 'side of sheet'] and
5     ∀ a ∈ ^v •
6     (1 (2 N a = 'nominal page size' impl
7         (3 IsPlaceholder1.19(C a) or IsPairOfPosInt1.9(C a)3)2)  and
8       (4 N a = 'side of sheet' impl
9         (5 IsPlaceholder1.19(C a) or C a ∈ ['recto'; 'verso'; 'unspecified']5)4) and
10      (6 N a = 'colour of medium' impl
11        (7 IsPlaceholder1.19(C a) or IsColourExpression2.126(C a) or C a = 'unspecified'7)6)1)0)
```

Semiformal Description 2.99

Predicate "is a protection value" (clause 9.5.1)

A protection value is either 'protected' or 'unprotected'.

Definition 2.99

```
1   ∀ v
2   (0 IsProtectionValue(v) iff
3     v ∈ ['protected'; 'unprotected']0)
```

Semiformal Description 2.100

Predicate "is a layout style identifier" (clause 9.6.1)

A layout style identifier is a pair of two non-negative integers where the first integer is 4.

Definition 2.100

```
1   ∀ v
2   (0 IsLayoutStyleId(v) iff
3     IsPairOfNnInt1.8(v) and HEAD1.13(v) = 40)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 2.101

Predicate "is a block alignment value" (clause 9.7.2)

A block alignment value is one of the following: 'right-hand aligned', 'left-hand aligned', 'centred', 'null'.

Definition 2.101

1 $\forall v$
2 ($_0$IsBlockAlignmentValue(v) *iff*
3 $v \in$ ['right-hand aligned'; 'left-hand aligned'; 'centred'; 'null']$_0$)

Semiformal Description 2.102

Predicate "is a concatenation value" (clause 9.7.3)

The value of the attribute 'concatenation' is either 'concatenated' or 'non-concatenated'.

Definition 2.102

1 $\forall v$
2 ($_0$IsConcatenationValue(v) *iff*
3 $v \in$ ['concatenated'; 'non-concatenated']$_0$)

Semiformal Description 2.103

Predicate "is a fill order value" (clause 9.7.4)

The value of the attribute 'fill order' is either 'normal order' or 'reverse order'.

Definition 2.103

1 $\forall v$
2 ($_0$IsFillOrderValue(v) *iff*
3 $v \in$ ['normal order'; 'reverse order']$_0$)

Semiformal Description 2.104

Predicate "is a layout object reference" (clauses 9.7.8)

A layout object reference is 'object type page' or 'null' or a layout category name or a layout object class identifier.

Definition 2.104

1 $\forall v$
2 ($_0$IsLayoutObjectReference(v) *iff*
3 $v \in$ ['object type page'; 'null'] *or*
4 IsLayoutCatName$^{2.105}$(v) *or* IsLayoutObjectClassId$^{2.81}$(v) $_0$)

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 2.105

Predicate "is a layout category name" (clause 9.7.7)

A layout category name is either an empty string (denoting the implicit stream root category) or a string of characters from the minimum subrepertoire of ISO 6937-2.

Definition 2.105

```
1  ∀ v
2  (₀IsLayoutCatName(v) iff
3    IsEmptyCat^1.6(v) or IsISO6937MSString^1.20(v)₀)
```

Semiformal Description 2.106

Predicate "is a set of layout category names"

Each element of a set of layout category names is a layout category name.

Definition 2.106

```
1  ∀ v
2  (₀IsSetOfLayoutCatNames(v) iff
3    IsCol(v) and
4    ∀ cst ∈ v (IsLayoutCatName^2.105(cst))₀)
```

Semiformal Description 2.107

Predicate "is an offset value" (clause 9.7.12)

The value of the attribute 'offset' is a nomination of four components with the names 'leading offset', 'trailing offset', 'left-hand offset' and 'right-hand offset'. The components are non–negative numbers. The parameters are independently defaultable.

Definition 2.107

```
1  ∀ v
2  (₀IsOffsetValue(v) iff
3    IsNeNom^1.2(v) and
4    NAMS^1.18(v) = ['leading offset'; 'trailing offset'; 'left-hand offset';
5                    'right-hand offset'] and
6    ∀ a ∈̂ v • (IsPlaceholder^1.19(C a) or IsNnInt^1.7(C a))₀)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 2.108

Predicate "is a same layout object value" (clause 9.7.13)

The value of the attribute 'same layout object' is a nomination of up to two components. For the name 'logical object' the component is 'null', a logical object identifier or an object or object class identifier expression. If the component is not 'null' the name 'layout object' must be specified with a value different from 'null'. For the name 'layout object' the component is 'null', 'object type page', a layout object class identifier or a layout category name. If the component is not 'null' the name 'logical object' must be specified with a value different from 'null'.

Definition 2.108

```
1   ∀ v
2   (0 IsSameLayoutObjectValue(v) iff
3     IsNeNom1.2(v) and
4     NAMS1.18(v) ⊆ ['layout object'; 'logical object'] and
5     ∀ a ∈ ^v •
6     (1 N a = 'logical object' impl
7       (2(3 C a = 'null' or IsLogicalObjectId2.77(C a) or IsObjectOrObjectClassIdExpression2.69(C a)3) and
8         C a ≠ 'null' impl
9         (4 'layout object' ∈ NAMS1.18(v) and (C ∈ ^'layout object' •) ≠ 'null'4)2) and
10      N a = 'layout object' impl
11      (5(6 C a ∈ ['null'; 'object type page'] or IsLayoutObjectClassId2.81(C a) or IsLayoutCatName2.105(C a)6) and
12        C a ≠ 'null' impl
13        (7 'logical object' ∈ NAMS1.18(v) and (C ∈ ^'logical object' •) ≠ 'null'7)5)1)0)
```

Semiformal Description 2.109

Predicate "is a separation value" (clause 9.7.14)

The value of the attribute 'separation' is a nomination, i.e., a set of pairs (*name*, *value*). The *name* is 'leading-edge', 'trailing-edge' or 'centre separation'; the *value* is a non-negative integer. The parameters are independently defaultable.

Definition 2.109

```
1   ∀ v
2   (0 IsSeparationValue(v) iff
3     IsNeNom1.2(v) and
4     NAMS1.18(v) = ['leading-edge'; 'trailing-edge'; 'centre separation'] and
5     ∀ a ∈ ^v •
6     (1 IsPlaceholder1.19(C a) or IsNnInt1.7(C a)1)0)
```

Semiformal Description 2.110

Predicate "is a presentation style identifier" (clause 9.8.1)

A presentation style identifier is a pair of non-negative integers where the first integer is 5.

Definition 2.110

```
1   ∀ v
2   (0 IsPresentationStyleId(v) iff
3     IsPairOfNnInt1.8(v) and HEAD1.13(v) = 50)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 2.111

Predicate "is a layout content portion identifier" (clause 9.9.1)

A layout content portion identifier is either a layout object identifier or a layout object class identifier whose length is greater than 1.

Definition 2.111

1 $\forall v$
2 $({}_{0}$IsLayoutContentPortionId(v) *iff*
3 IsLayoutObjectId$^{2.78}(v)$ *or* IsLayoutObjectClassId$^{2.81}(v)$ *and*
4 LENGTH$^{1.16}$ $(v) > 1_{0})$

Semiformal Description 2.112

Predicate "is a logical content portion identifier" (clause 9.9.1)

A logical content portion identifier is either a logical object identifier or a logical object class identifier whose length is greater than 1.

Definition 2.112

1 $\forall v$
2 $({}_{0}$IsLogicalContentPortionId(v) *iff*
3 IsLogicalObjectId$^{2.77}(v)$ *or* IsLogicalObjectClassId$^{2.80}(v)$ *and*
4 LENGTH$^{1.16}$ $(v) > 1_{0})$

Semiformal Description 2.113

Predicate "is a type of coding value" (clause 9.9.2)

The value of the attribute **'type of coding'** is a character content type of coding value, a raster graphics content type of coding value or a geometric graphics content type of coding value.

Definition 2.113

1 $\forall v$
2 $({}_{0}$IsTypeOfCodingValue(v) *iff*
3 $({}_{1}$IsCharacterContentTypeOfCodingValue$^{6.20}(v)$ *or*
4 IsRasterGraphicsContentTypeOfCodingValue$^{7.20}(v)$ *or*
5 IsGeometricGraphicsContentTypeOfCodingValue$^{8.52}(v)$ $_{1})_{0})$

Semiformal Description 2.114

Predicate "is a content information value" (clause 9.9.3.1)

The value of the attribute **'content information'** is a character content information value, a raster graphics content information value or a geometric graphics content information value.

Definition 2.114

1 $\forall v$
2 $({}_{0}$IsContentInformationValue(v) *iff*
3 $({}_{1}$IsCharacterContentInformationValue$^{6.111}(v)$ *or*
4 IsRasterGraphicsContentInformationValue$^{7.25}(v)$ *or*
5 IsGeometricGraphicsContentInformationValue$^{8.53}(v)$ $_{1})_{0})$

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 2.115

Predicate "is an alternative character sets string" (clause 9.9.3.2)

The value of the attribute 'alternative representation' is an atomic construct, i.e., the internal structure of the value is not considered in the formal specification of the document structure. The permitted values of this attribute depend on the value of the attribute 'alternative representation character sets' in the document profile.

Definition 2.115

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

Semiformal Description 2.116

Predicate "is a coding attributes value" (clause 9.9.4)

The value of the attribute 'coding attributes' is a collection where each element is a character content coding attribute, a raster graphics content coding attribute or geometric graphics content coding attribute.

Definition 2.116

1 $\forall v$
2 $(_0$IsCodingAttributesValue(v) *iff*
3 IsCol(v) *and*
4 $\forall a \in v$
5 $(_1$IsCharacterContentCodingAttribute$^{6.13}(a)$ *or*
6 IsRasterGraphicsContentCodingAttribute$^{7.9}(a)$ *or*
7 IsGeometricGraphicsContentCodingAttribute$^{8.7}(a)$ $_1)_0$)

Semiformal Description 2.117

Predicate "is a character sequence"

A character sequence is a catenation of characters.

Definition 2.117

1 $\forall v$
2 $(_0$IsCharSeq(v) *iff*
3 IsCat(v) *and* $\forall m \in \hat{}v \bullet$ (IsCharacter$^{2.118}$(C m))$_0$)

Semiformal Description 2.118

Predicate "is a character"

A character is an element of a defined character set and is considered an atomic construct in the formal specification.

Definition 2.118

1 $\forall z$
2 $(_0$IsCharacter(z) *iff*
3 IsAtom$(z)_0$)

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 2.119

Predicate "is an enciphered value" (clause 9.3.6.1)

The value of the attribute **'enciphered'** is a nomination of up to two components. For the name **'enciphered subordinates'** (which must always be present) the component is a sequence of different non-negative integers or **'all'** or **'none'**. For the name **'protected part identifier'** the component is the identifier of an enciphered document body part.

Definition 2.119

```
1   ∀ v
2   (0 IsEncipheredValue(v) iff
3     IsNeNom1.2(v) and
4     NAMS1.18(v) ⊆ ['enciphered subordinates'; 'protected part identifier'] and
5     NAMS1.18(v) ⊇ ['enciphered subordinates'] and
6     ∀ a ∈̂ ^v •
7     (1 (2 N a = 'enciphered subordinates' impl
8         (3 IsSeqOfDiffNnInt1.22(C a) or C a ∈ ['all'; 'none']3)2) and
9       (4 N a = 'protected part identifier' impl IsEncipheredBodyPartId2.123(C a)4)1)0)
```

Semiformal Description 2.120

Predicate "is a sealed value" (clause 9.3.6.2)

The value of the attribute **'sealed'** is a nomination of up to two components. For the name **'seal identifiers'** the component is a set of seal identifiers. For the name **'sealed status'** (which must always be present) the component is either **'no'** or **'yes'**.

Definition 2.120

```
1   ∀ v
2   (0 IsSealedValue(v) iff
3     IsNeNom1.2(v) and
4     NAMS1.18(v) ⊆ ['seal identifiers'; 'sealed status'] and
5     NAMS1.18(v) ⊇ ['sealed status'] and
6     ∀ a ∈̂ ^v •
7     (1 (2 N a = 'seal identifiers' impl IsSetOfSealIds2.124(C a)2) and
8       (3 N a = 'sealed status' impl
9         (4 IsPlaceholder1.19(C a) or C a ∈ ['no'; 'yes']4)3)1)0)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 2.121

Predicate "is a sealed document profile information value" (clause 9.10.2)

A sealed document profile information value is a non-empty set of attribute names of attributes permitted in the document profile.

Definition 2.121

$\forall v$
$(_0$IsSealedDocumentProfileInformationValue(v) *iff*
  IsNeCol$^{1.1}(v)$ *and*
  $\forall a \in v$
  $a \in$ ['abstract'; 'access rights';
    'additional information'; 'alternative feature sets';
    'alternative representation character sets'; 'authorization';
    'authors'; 'block alignments';
    'borders'; 'coding attributes';
    'colour characteristics'; 'colour of layout objects';
    'colours'; 'colour spaces list';
    'comments character sets'; 'content architecture classes';
    'content background colours'; 'content colour tables';
    'content foreground colours'; 'copyright';
    'creation date and time'; 'distribution list';
    'document application profile'; 'document application profile defaults';
    'document architecture class'; 'document date and time';
    'document reference'; 'document size';
    'document type'; 'enciphered document profiles';
    'enciphered profiles'; 'expiry date and time';
    'external-document class'; 'fill orders';
    'fonts list'; 'generic layout structure';
    'generic logical structure'; 'interchange format class';
    'keywords'; 'languages';
    'layout paths'; 'layout styles';
    'local file references'; 'local filing date and time';
    'medium types'; 'number of objects per page';
    'number of pages'; 'object colour tables';
    'ODA security label'; 'ODA version';
    'organizations'; 'owners';
    'page dimensions'; 'page positions';
    'post-enciphered body parts'; 'post-enciphered document body parts';
    'post-sealed document body parts'; 'pre-enciphered body parts';
    'pre-enciphered document body parts'; 'preparers';
    'pre-sealed document body parts'; 'presentation features';
    'presentation styles'; 'profile character sets';
    'protections'; 'purge date and time';
    'references to other documents'; 'release date and time';
    'resource document'; 'resources';
    'revision history'; 'sealed document profiles';
    'sealed profiles'; 'security classification';
    'specific layout structure'; 'specific logical structure';
    'start date and time'; 'status';
    'subject'; 'superseded documents';
    'title'; 'transparencies';
    'types of coding'; 'unit scaling';
    'user-specific codes']$_0$)

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 2.122

Predicate "is a protected part identifier" (clause 9.10.1)

A protected part identifier is a pair of non-negative integers where the first integer is 6, 7, 8 or 9.

Definition 2.122

1 $\forall v$
2 $({}_{0}\,\text{IsProtectedPartId}(v)$ *iff*
3 $\quad \text{IsPairOfNnInt}^{1.8}(v)$ *and* $\text{HEAD}^{1.13}(v) \in [6;7;8;9]_{0})$

Semiformal Description 2.123

Predicate "is an enciphered body part identifier" (clause 9.10.1)

An enciphered body part identifier is a protected part identifier where the first integer is 8 or 9.

Definition 2.123

1 $\forall v$
2 $({}_{0}\,\text{IsEncipheredBodyPartId}(v)$ *iff*
3 $\quad \text{IsProtectedPartId}^{2.122}(v)$ *and* $\text{HEAD}^{1.13}(v) \in [8;9]_{0})$

Semiformal Description 2.124

Predicate "is a set of seal identifiers" (clause 9.3.6.2)

A set of seal identifiers is a set whose elements are seal identifiers.

Definition 2.124

1 $\forall v$
2 $({}_{0}\,\text{IsSetOfSealIds}(v)$ *iff*
3 $\quad \text{IsCol}(v)$ *and*
4 $\quad \forall a \in v\ (\text{IsSealId}^{2.125}(a))\ {}_{0})$

Semiformal Description 2.125

Predicate "is a seal identifier" (clause 9.3.6.2)

A seal identifier is an integer.

Definition 2.125

1 $\forall v$
2 $({}_{0}\,\text{IsSealId}(v)$ *iff*
3 $\quad \text{IsInt}(v)_{0})$

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 2.126

Predicate "is a colour expression" (clause 9.1.4.1)

A colour expression is either a direct colour expression or an indexed colour expresion.

Definition 2.126

```
1  ∀ v
2  (0 IsColourExpression(v) iff
3    IsDirectColourExpression^2.128(v) or IsIndexedColourExpression^2.127(v)0)
```

Semiformal Description 2.127

Predicate "is an indexed colour expression" (clause 9.1.4)

An indexed colour expression is a nomination with the two components 'colour access mode' and 'indexed'. For the name 'colour access mode' the value is 'indexed'. For the name 'index' the component is a non-negative integer. The parameter 'index' is defaultable.

Definition 2.127

```
1  ∀ v
2  (0 IsIndexedColourExpression(v) iff
3    IsNeNom^1.2(v) and
4    NAMS^1.18(v) = ['colour access mode'; 'index'] and
5    ∀ a ∈ ^v •
6    (1 (2 N a = 'colour access mode' impl C a = 'indexed'2) and
7      (3 N a = 'index' impl
8        (4 IsPlaceholder^1.19(C a) or IsNnInt^1.7(C a)4)3)1)0)
```

Semiformal Description 2.128

Predicate "is a direct colour expression" (clause 9.1.4)

An direct colour expression is nomination with four components. For the name 'colour access mode' the component is 'direct'. For the name 'colour space id' the component is a non-negative integer. For the name 'colour specification' the component is a colour specification value. For the name 'colour tolerance' the component is a colour tolerance value. All parameters are defaultable.

Definition 2.128

```
1  ∀ v
2  (0 IsDirectColourExpression(v) iff
3    IsNeNom^1 2(v) and
4    NAMS^1.18(v) = ['colour access mode'; 'colour space id'; 'colour specification'; 'colour tolerance'] and
5    ∀ a ∈ ^v •
6    (1 (2 N a = 'colour access mode' impl
7        (3 IsPlaceholder^1.19(C a) or C a = 'direct'3) and
8      (4 N a = 'colour space id' impl
9        (5 IsPlaceholder^1.19(C a) or IsNnInt^1.7(C a)5)4) and
10     (6 N a = 'colour specification' impl
11       (7 IsPlaceholder^1.19(C a) or IsColourSpecificationValue^2.129(C a)7)6) and
12     (8 N a = 'colour tolerance' impl
13       (9 IsPlaceholder^1.19(C a) or IsColourToleranceValue^2.130(C a)9)8)2)1)0)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 2.129

Predicate "is a colour specification value" (clause 9.1.4)

A colour specification value is a catenation of three or four components which are real numbers or integers.

Definition 2.129

```
1   ∀ v
2   (0 IsColourSpecificationValue(v) iff
3     IsCat(v) and LENGTH1.16 (v) ≥ 3 and LENGTH1.16 (v) ≤ 4 and
4     ∀ b ∈ v (1 IsReal(b) or IsInt(b)1)0)
```

Semiformal Description 2.130

Predicate "is a colour tolerance value" (clause 9.1.4)

A colour tolerance value is a nomination with one component. If the name is 'unspecified tolerance' the component is 'infinite'. (This value is defaultable.) If the name is 'specified tolerance' the component is a nomination again with the names 'tolerance value' (the component is a positive integer or real number) and 'tolerance space' (the component is either 'CIELAB' or 'CIELUV').

Definition 2.130

```
1   ∀ v
2   (0 IsColourToleranceValue(v) iff
3     ∃ n, a
4     (1 v = [n : a] and n ∈ ['unspecified tolerance'; 'specified tolerance'] and
5       (2 (3 n = 'unspecified tolerance' and IsPlaceholder1.19(C a) or a = 'infinite'3) or
6         (4 n = 'specified tolerance' and
7           ∃ b, c
8           (5 a = ['tolerance value' : b; 'tolerance space' : c] and
9             (6 IsNat(b) or IsPosReal1.25(b) 6) and
10            c ∈ ['CIELAB'; 'CIELUV']5)4)2)1)0)
```

Semiformal Description 2.131

Predicate "is a colour table" (clause 9.1.4)

A colour table is a nomination with two components. For the name 'colour space id' the component is a non-negative integer. For the name 'colour table entries' the component is a set of colour table entries. The parameter 'colour space id' is defaultable.

Definition 2.131

```
1   ∀ v
2   (0 IsColourTable(v) iff
3     IsNeNom1.2(v) and
4     NAMS1.18(v) = ['colour space id'; 'colour table entries'] and
5     ∀ a ∈ ^v •
6     (1 (2 N a = 'colour space id' impl
7         (3 IsPlaceholder1.19(C a) or IsNnInt1.7(C a) 3) and
8       (4 N a = 'colour table entries' impl IsSetOfColourTableEntries2.132(C a)4)2)1)0)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 2.132

Predicate "is a set of colour table entries" (clause 9.1.4)

A set of colour table entries is a non-empty set whose elements are colour table entries.

Definition 2.132

```
1   ∀ v
2   (0 IsSetOfColourTableEntries(v) iff
3     IsNeCol1 1(v) and
4     ∀ a ∈ v (IsColourTableEntry2.133(a)) 0)
```

Semiformal Description 2.133

Predicate "is a colour table entry" (clause 9.1.4)

An colour table entry is nomination with three components. For the name 'colour specification' the component is a colour specification value. For the name 'colour tolerance' the component is a colour tolerance value. For the name 'index' the component is non-negative integer. All parameters are defaultable.

Definition 2.133

```
1   ∀ v
2   (0 IsColourTableEntry(v) iff
3     IsNeNom1.2(v) and
4     NAMS1.18(v) = ['colour specification'; 'colour tolerance'; 'index'] and
5     ∀ a ∈ ^v •
6     (1 (2 N a = 'colour specification' impl
7         (3 IsPlaceholder1.19(C a) or IsColourSpecificationValue2.129(C a)3) and
8       (4 N a = 'colour tolerance' impl
9         (5 IsPlaceholder1.19(C a) or IsColourToleranceValue2.130(C a)5)4)2) and
10      (6 N a = 'index' impl
11        (7 IsPlaceholder1.19(C a) or IsNnInt1.7(C a)7)6)1)0)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 2.134

Predicate "is a floatability range value" (clause 9.7.5)

A floatability range value is a nomination where the names are 'forward limit' and 'backward limit'. Their components are nominations again where the names are 'layout object' and 'logical object'. For the name 'logical object' the component is 'null', a logical object identifier or an object or object class identifier expression. For the name 'layout object' the component is 'null', 'object type page', a layout object class identifier or a layout category name.

Definition 2.134

```
1   ∀ v
2   (0 IsFloatabilityRangeValue(v) iff
3     IsNeNom1.2(v) and
4     NAMS1.18(v) = ['forward limit'; 'backward limit'] and
5     ∀ a ∈̃ ~v •
6     (1 IsNeNom1.2(C a) and
7       NAMS1.18(C a) ⊆ ['layout object'; 'logical object'] and
8       ∀ b ∈̃ ^(C a) •
9       (2 N b = 'logical object' impl
10        (3 C b = 'null' or IsLogicalObjectId2.77(C b) or IsObjectOrObjectClassIdExpression2.69(C b)3) and
11        N b = 'layout object' impl
12        (4 C b ∈ ['null'; 'object type page'] or IsLayoutObjectClassId2.81(C b) or
13          IsLayoutCatName2.105(C b)4)2)1)0)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

7.4 Subsidiary definitions

This is the list of all definitions for predicates and functions (operators) which have been factorized from 7.2–7.3 for the sake of readability.

7.4.1 Predicates

Semiformal Description 2.135

Predicate "is described by"

An entity s is described by an entity g if s is a specific description and g is a corresponding generic description (3-5) and if each object description so in s which has an attribute 'object class' (6,7) refers to an object class description go in g (8-10) with the same value of 'object type' as specified in so (11-12). If so is a basic logical object description which has no attribute 'content portions' (13) (and therefore refers to a basic logical object class description go) the content portions connected with go through 'content portions' or the content obtained from go through 'content generator' (14) are regarded as the content of so. If so is a basic layout object description which has no attribute 'content portions' (15) (and therefore refers to a basic layout object class description go) the content portions connected with go through 'content portions' (16) are regarded as the content of so. (If so is a basic object description which has an attribute 'content portions', any specific content portion descriptions are allowed.)

Definition 2.135

1 $\forall s, g$
2 $(_0(s)\text{IsDescribedBy}(g)$ *iff*
3 $(_1(_2\text{IsSpecificLogicalDescription}^{2.14}(s)$ *and* $\text{IsGenericLogicalDescription}^{2.11}(g)_2)$
4 *or*
5 $(_3\text{IsSpecificLayoutDescription}^{2.15}(s)$ *and* $\text{IsGenericLayoutDescription}^{2.12}(g)_3)_1)$ *and*
6 $\forall so \in s$
7 $(_4\text{IsObjectDescription}^{2.53}(so)$ *and* 'object class' $\in \text{NAMS}^{1.18}(so)$ *impl*
8 $\exists go \in g$
9 $(_5\text{IsObjectClassDescription}^{2.35}(go)$ *and*
10 $C\,\hat{}so\bullet$'object class' $= C\,\hat{}go\bullet$'object class identifier' *and*
11 $(_6$ 'object type' $\in \text{NAMS}^{1.18}(so)$ *impl*
12 $C\,\hat{}so\bullet$'object type' $= C\,\hat{}go\bullet$'object type'$_6)$ *and*
13 $(_7\ (_8\text{IsBasicLogicalObjectDescription}^{2.39}(so)$ *and* 'content portions' $\notin \text{NAMS}^{1.18}(so)_8)$ *impl*
14 $(_9$ 'content portions' $\in \text{NAMS}^{1.18}(go)$ *or* 'content generator' $\in \text{NAMS}^{1.18}(go)_9)_7)$ *and*
15 $(_{10}(_{11}\text{IsBasicLayoutObjectDescription}^{2.52}(so)$ *and* 'content portions' $\notin \text{NAMS}^{1.18}(so)_{11})$ *impl*
16 'content portions' $\in \text{NAMS}^{1.18}(go)_{10})_5)_4)_0)$

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Semiformal Description 2.136

Predicate "is completely described by"

An entity *s* is completely described by an entity *g* if *s* is described by *g* which is a complete generic description (4) and if each object description *so* in *s* (5,6) refers through its attribute 'object class' (7) to an object class description *go* in *g* (8-10), which has an attribute 'generator for subordinates' if it is a composite object class description (11,12). If *go* has the attribute 'content generator' specified, *go* and/or *so* have the attribute 'content portions' specified (17, 18). If *so* is a composite object description (14) and the associated *go* has an attribute 'generator for subordinates' (15), *go* specifies through its attribute 'generator for subordinates' a sequence of or a set of sequences of (subordinate) object class identifiers. Among them is that sequence of object class identifiers which is obtained from the object descriptions immediately subordinate to *so* (16).

Definition 2.136

1 $\forall s, g$
2 $({}_0(s)\text{IsCompletelyDescribedBy}(g)$ *iff*
3 $(s)\text{IsDescribedBy}^{2.135}(g)$ *and*
4 $\text{IsCompleteGenericDescription}^{2.10}(g)$ *and*
5 $\forall so \in s$
6 $({}_1\text{IsObjectDescription}^{2.53}(so)$ *impl*
7 $({}_2$ *not* $\text{IsBlockDescription}^{2.47}(so)$ *impl* 'object class' $\in \text{NAMS}^{1.18}(so)_2)$ *and*
8 'object class' $\in \text{NAMS}^{1.18}(so)$ *impl*
9 $\exists go \in g$
10 $({}_3\text{IsObjectClassDescription}^{2.35}(go)$ *and*
11 C ^so• 'object class' = C ^go• 'object class identifier' *and*
12 $({}_4\text{IsCompositeObjectClassDescription}^{2.31}(go)$ *impl*
13 'generator for subordinates' $\in \text{NAMS}^{1.18}(go)_4)$ *and*
14 $({}_5\text{IsCompositeObjectDescription}^{2.49}(so)$ *and*
15 'generator for subordinates' $\in \text{NAMS}^{1.18}(go)$ *impl*
16 $({}_6\text{SEQCLASSIDIMSUB}^{2.156}(s, so, \text{C } \hat{}so\bullet \text{'subordinates'})\text{IsSeqClassIdImSubIn}^{2.145}(go)_8)_5)$ *and*
17 $({}_7$ 'content generator' $\in \text{NAMS}^{1.18}(go)$ *impl*
18 $({}_8$ 'content portions' $\in \text{NAMS}^{1.18}(go)$ *or* 'content portions' $\in \text{NAMS}^{1.18}(so)_8)_7)_3)_1)_0)$

Semiformal Description 2.137

Predicate "is content connected"

An entity *csts* is content connected if it is a collection (3) and if each content portion description *cont* in it (4,5) is referenced at least from one basic object description or basic object class description *cst* (6).

Definition 2.137

1 $\forall csts$
2 $({}_0\text{IsContentConnected}(csts)$ *iff*
3 $\text{IsCol}(csts)$ *and*
4 $\forall cont \in csts$
5 $({}_1\text{IsContentPortionDescription}^{2.54}(cont)$ *impl*
6 $\exists cst \in csts((cont)\text{DescribesContPortOf}^{2.153}(cst))_1)_0)$

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 2.138

Predicate "is content complete"

An entity *csts* is content complete if it is a collection (3) and if for all basic object descriptions or basic object class descriptions *cst* in it (4–6) each reference out of the attribute 'content portions' (6–8) is satisfied (9–15).

Definition 2.138

```
1   ∀ csts
2   (0 IsContentComplete(csts) iff
3     IsCol(csts) and
4     ∀ cst ∈ csts
5     (1 (2 (3 IsBasicObjectDescription^2.50(cst) or
6           IsBasicObjectClassDescription^2.34(cst)3) and
7       'content portions' ∈ NAMS^1.18(cst)2) impl
8       ∀ r ∈ ^cst • 'content portions' •
9       (4 ∃ cont ∈ csts
10        (5 IsContentPortionDescription^2.54(cont) and
11          (6 C ^cont • 'content identifier logical' =
12            C ^cst •<'object identifier', 'object class identifier'>//[ → C r → ]
13            or
14            C ^cont • 'content identifier layout' =
15            C ^cst •<'object identifier', 'object class identifier'>//[ → C r → ]6)5)4)1)0)
```

Semiformal Description 2.139

Predicate "is class-immediate-subordinates complete"

An entity *g* is complete with respect to the object class descriptions for immediate subordinates if it is a collection (3) and if for all composite object class descriptions *cst* in it (4,5) each reference *r* out of the value *e* of the attribute 'generator for subordinates' is satisfied (7–13).

Definition 2.139

```
1   ∀ g
2   (0 IsClassImSubComplete(g) iff
3     IsCol(g) and
4     ∀ cst ∈ g
5     (1 (2 IsCompositeObjectClassDescription^2.31(cst) and
6         'generator for subordinates' ∈ NAMS^1.18(cst)2) impl
7       (3 ∃ e
8         (4 e = C ^cst • 'generator for subordinates' and
9           IsConstructionExpression^2.84(e) and
10          ∀ r ∈ [id | (id)OccursIn^2.147(e)]
11          (5 ∃ sub ∈ g
12            (6 IsObjectClassDescription^2.35(sub) and
13              C ^sub • 'object class identifier' = r6)5)4)3)1)0)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 2.140

Predicate "is immediate-subordinates complete"

An entity s is complete with respect to the object descriptions for immediate subordinates if it is a collection (3) and if for all composite object descriptions *cst* in it (4,5) each reference r out of the value of the attribute 'subordinates' (6) is satisfied (7-10).

Definition 2.140

1 $\forall s$
2 $({}_0$IsImSubComplete(s) *iff*
3 IsCol(s) *and*
4 $\forall cst \in s$
5 $({}_1$IsCompositeObjectDescription$^{2.49}(cst)$ *impl*
6 $\forall r \hat{\in}$ ^*cst*• 'subordinates' •
7 $({}_2 \exists sub \in s$
8 $({}_3$IsObjectDescription$^{2.53}(sub)$ *and*
9 C ^*sub*• 'object identifier' =
10 C ^*cst*• 'object identifier'//[→ C r →]$_3)_2)_1)_0$)

Semiformal Description 2.141

Predicate "is contained in"

An entity *slog* is contained in an entity *slay* if *slay* is a specific layout description, *slog* is a specific logical description and the set of all content portion descriptions of *slog* is a subset of all content portion descriptions of *slay*.

Definition 2.141

1 $\forall slay, slog$
2 $({}_0(slog)$IsContainedIn($slay$) *iff*
3 IsSpecificLayoutDescription$^{2.15}(slay)$ *and*
4 IsSpecificLogicalDescription$^{2.14}(slog)$ *and*
5 ALLCONT$^{2.155}(slog) \subseteq$ ALLCONT$^{2.155}(slay)_0$)

Semiformal Description 2.142

Predicate "is an object class identifier for immediate subordinates"

An entity *id* is an object class identifier for immediate subordinates of the entity *gco*, if *gco* has an attribute 'generator for subordinates', whose value e is a "construction expression". The predicate OccursIn is True, if the object class identifier *id* occurs in the "construction expression" e.

Definition 2.142

1 $\forall id, gco$
2 $({}_0(id)$IsClassIdImSubIn(gco) *iff*
3 IsAttributeSet$^{2.61}(gco)$ *and*
4 $\exists a \hat{\in}$ ^*gco*• $\exists e$
5 $({}_1$N a = 'generator for subordinates' *and* e = C a *and*
6 IsConstructionExpression$^{2.84}(e)$ *and* (id)OccursIn$^{2.147}(e)$ $_1)_0$)

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 2.143

Predicate "is an object identifier of an immediate subordinate"

An entity *id* is an object identifier of an immediate subordinate object of the entity *sco* if it is referred to (via 'subordinates') from the entity *sco*.

Definition 2.143

1 $\forall\, id, sco$
2 $({}_0(id)\text{IsIdImSubIn}(sco)$ *iff*
3 $\quad \text{IsAttributeSet}^{2.61}(sco)$ *and*
4 $\quad \exists\, n \in \hat{}sco \bullet$ 'subordinates' $\bullet$
5 $\quad (id = \text{C } \hat{}sco \bullet$ 'object identifier' $//[\rightarrow \text{C } n \rightarrow])_0)$

Semiformal Description 2.144

Predicate "is an identifier of a content portion"

An entity *id* is an identifier of a content portion within the entity *b*, if it is referred to (via 'content portions') from the entity *b*, i.e., if *id* is the identifier of *b* extended with one of the numbers drawn from the value of 'content portions'.

Definition 2.144

1 $\forall\, id, b$
2 $({}_0(id)\text{IsIdContIn}(b)$ *iff*
3 $\quad \text{IsAttributeSet}^{2.61}(b)$ *and*
4 $\quad \exists\, n \in \hat{}b \bullet$ 'content portions' $\bullet$
5 $\quad ({}_1 id = \text{C } \hat{}b \bullet <$'object identifier', 'object class identifier'$>//[\rightarrow \text{C } n \rightarrow]_1)_0)$

Semiformal Description 2.145

Predicate "is a sequence of object class identifiers for immediate subordinates"

An entity *idq* is a sequence of object class identifiers for immediate subordinates of the entity *gco*, if *gco* has an attribute 'generator for subordinates' (5,6), whose value *e* is a "construction expression". The sequence *idq* is one of the sequences defined by *e*, i.e., is covered by *e*.

Definition 2.145

1 $\forall\, idq, gco$
2 $({}_0(idq)\text{IsSeqClassIdImSubIn}(gco)$ *iff*
3 $\quad \text{IsAttributeSet}^{2.61}(gco)$ *and*
4 $\quad \exists\, a \in \hat{}gco \bullet\ \exists\, e$
5 $\quad ({}_1(\text{N } a =$ 'generator for subordinates' *and* $e = \text{C } a)$ *and*
6 $\quad\quad \text{IsConstructionExpression}^{2.84}(e)$ *and* $(idq)\text{CoveredBy}^{2.146}(e)\ {}_1)_0)$

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Semiformal Description 2.146

Predicate "is covered by"

A sequence *idq* of object class identifiers is covered by the construction expression *e* (which defines a possibly infinite set of such sequences), if *idq* is one of the sequences of object class identifiers defined by *e*. In general, *e* is the value of the attribute **'generator for subordinates'** of an object class description, i.e., a "construction expression".

Definition 2.146

1 $\forall\, idq, e$
2 $(_0(idq)\text{CoveredBy}(e)$ *iff*
3 $\quad \text{IsSeqOfObjectClassId}^{2.83}(idq)$ *and*
4 $\quad \text{IsConstructionExpression}^{2.84}(e)_0)$

Semiformal Description 2.147

Predicate "occurs in"

An object class identifier *id* occurs in a construction expression *m* if it appears at least once in the construction expression *m*. In general, *m* is the value of the attribute **'generator for subordinates'** of an object class description.

Definition 2.147

1 $\forall\, id, m$
2 $(_0(id)\text{OccursIn}(m)$ *iff*
3 $\quad (_1\, \text{IsConstructionExpression}^{2.84}(m)$ *and*
4 $\quad\quad id \in \text{COLC}^{1.17}\ \hat{}m \downarrow <xs \parallel \text{IsNnInt}^{1.7}(\text{C}\ xs)>^{\bullet}{}_1)_0)$

Semiformal Description 2.148

Predicate "is an initial generic constituent in"

An entity *cst* is an initial generic constituent in an entity *g* if *g* is a collection (3) and *cst* is an object class description in it (3,4) and if there is no entity *k* in *g* from which a reference via **'generator for subordinates'** is made to *cst* (5). Note that a self-reference ($k = cst$, i.e. from the object class description *cst* to itself) is not permitted for initial generic constituents.

Definition 2.148

1 $\forall\, cst, g$
2 $(_0(cst)\text{IsInitialGenericIn}(g)$ *iff*
3 $\quad \text{IsCol}(g)$ *and* $cst \in g$ *and*
4 $\quad \text{IsObjectClassDescription}^{2.35}(cst)$ *and*
5 $\quad$ *not* $\exists\, k \in g((cst)\text{DescribesClassImSubOf}^{2.150}(k))_0)$

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 2.149

Predicate "is initial specific constituent in"

An entity *cst* is an initial specific constituent in an entity *s* if *s* is a collection (3) and *cst* is an object description in it (3,4) and if there is no other entity *k* in *s* from which a reference via **'subordinates'** is made to *cst* (5). Note that a self-reference (from the object description *cst* to itself) is not possible in a specific description.

Definition 2.149

1 $\forall\, cst, s$
2 $(_0(cst)\text{IsInitialSpecificIn}(s)$ *iff*
3 $\quad \text{IsCol}(s)$ *and* $cst \in s$ *and*
4 $\quad \text{IsObjectDescription}^{2.53}(cst)$ *and*
5 $\quad$ *not* $\exists\, k \in s((cst)\text{DescribesImSubOf}^{2.152}(k))_0)$

Semiformal Description 2.150

Predicate "describes a class of immediate subordinates"

The "lower" object class description *lgo* describes a class of objects which are immediate subordinates of instances of the class described by the "higher" object class description *hgo*, if the **'object class identifier'** of *lgo* is referred to from *hgo* via **'generator for subordinates'**.

Definition 2.150

1 $\forall\, lgo, hgo$
2 $(_0(lgo)\text{DescribesClassImSubOf}(hgo)$ *iff*
3 $\quad (C\ \hat{}\ lgo \bullet$ **'object class identifier'**$)\text{IsClassIdImSubIn}^{2.142}(hgo)_0)$

Semiformal Description 2.151

Predicate "describes a class of subordinates"

An entity *lgo* ("lower" object class description) describes a class of objects which are (not necessarily immediate) subordinates of instances of the class described by the entity *hgo* ("higher" object class description) within the entity *g*, if *g* is a generic description containing *lgo* and *hgo* (3) and if *lgo* is referred to from *hgo* via **'generator for subordinates'** (4) or has an "intermediate" object class description *i* immediately superior to it, which in turn describes a class of objects subordinate to instances of the object class described by *hgo* (5,6). The recursion ends when the intermediate object class description is immediately subordinate to *hgo* (4).

Definition 2.151

1 $\forall\, lgo, hgo, g$
2 $(_0(lgo)\text{DescribesClassSubOf}(hgo)\text{In}(g)$ *iff*
3 $\quad \text{IsGenericDescription}^{2.9}(g)$ *and* $lgo \in g$ *and* $hgo \in g$ *and*
4 $\quad (_1(lgo)\text{DescribesClassImSubOf}^{2.150}(hgo)$ *or*
5 $\quad\quad \exists\, i \in g$
6 $\quad\quad ((lgo)\text{DescribesClassImSubOf}^{2.150}(i)$ *and* $(i)\text{DescribesClassSubOf}^{2.151}(hgo)\text{In}(g)_1)_0)$

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Semiformal Description 2.152

Predicate "describes an immediate subordinate"

The "lower" object description *lso* describes an object which is an immediate subordinate of the object described by the "higher" object description *hso*, if the 'object identifier' of *lso* is referred to from *hso* via 'subordinates'.

Definition 2.152

1 $\forall\, lso, hso$
2 $(_0(lso)\text{DescribesImSubOf}(hso)$ *iff*
3 $\quad(_1\text{IsObjectDescription}^{2.53}(lso)$ *and*
4 $\quad\quad\text{IsCompositeObjectDescription}^{2.49}(hso)$ *and*
5 $\quad\quad\exists\, n \in \hat{}\,hso \bullet$ 'subordinates' $\bullet$
6 $\quad\quad(\text{C}\ \hat{}\,lso \bullet$ 'object identifier' $= \text{C}\ \hat{}\,hso \bullet$ 'object identifier' $//[\rightarrow \text{C}\ n \rightarrow])_1)_0)$

Semiformal Description 2.153

Predicate "describes a content portion"

An entity *cont* describes a content portion of the basic object described by *b* or of a basic object, which is an instance of the object class described by *b*, if *cont* is a content portion description and either the 'content identifier logical' or the 'content identifier layout' is referred to from *b* via 'content portions'.

Definition 2.153

1 $\forall\, cont, b$
2 $(_0(cont)\text{DescribesContPortOf}(b)$ *iff*
3 $\quad(_1\text{IsContentPortionDescription}^{2.54}(cont)$ *and*
4 $\quad\quad(\text{IsBasicObjectDescription}^{2.50}(b)$ *or* $\text{IsBasicObjectClassDescription}^{2.34}(b))$ *and*
5 $\quad\quad(_2(\text{C}\ \hat{}\,cont \bullet$ 'content identifier logical')$\text{IsIdContIn}^{2.144}(b)$ *or*
6 $\quad\quad\quad(\text{C}\ \hat{}\,cont \bullet$ 'content identifier layout')$\text{IsIdContIn}^{2.144}(b)_2)_1)_0)$

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

7.4.2 Functions

Semiformal Description 2.154

Function "set of immediate subordinates"

The function IMSUB is defined, if *so* is an attribute set with at least the attributes 'object identifier' and 'subordinates', and if for each component spot *m* of the value of 'subordinates' the collection *sos* contains a nomination *cst* whose component 'object identifier' is equal to the object identifier of *so* with the component at *m* having been appended (i.e., if all subordinate object descriptions are within *sos*).

IMSUB returns the collection of all specific object descriptions within the collection *sos* whose identifiers, without the last number, are equal to the object identifier of the specific object description *so* and whose last number of their identifiers is among the numbers of 'subordinates' in the attribute set *so*.

Definition 2.154

```
1   ∀ so, sos
2   (0IMSUB(so, sos) =
3     IF IsAttributeSet^2.61(so) and
4        ['object identifier'; 'subordinates'] ⊆ NAMS^1.18(so) and
5        ∀ m ∈̂ (^so • 'subordinates' •) ∃ cst ∈ sos
6        (1 C ^cst • 'object identifier' = C ^so • 'object identifier'//[ → C m → ]1)
7     THEN
8     COLC^1.17 ^sos •<2 xs ||F xs • 'object identifier' •<ys || PR^1.12 ys = 1> =
9                             C ^so • 'object identifier' and
10                            C xs • 'object identifier' •<ys || PR^1.12 ys = 1> ∈
11                            COLC^1.17 ^so • 'subordinates' • 2>
12    ELSE UNDEF0)
```

Semiformal Description 2.155

Function "all content portions"

ALLCONT returns the collection of all content portion descriptions contained in the collection *csts*.

Definition 2.155

```
1   ∀ csts
2   (0 ALLCONT(csts) =
3     IF IsCol(csts)
4     THEN [cst | cst ∈ csts and IsContentPortionDescription^2.54(cst)]
5     ELSE UNDEF0)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Semiformal Description 2.156

Function "sequence of class identifiers of immediate subordinates"

The function SEQCLASSIDIMSUB is defined, if *so* is an attribute set (3) with at least the attributes 'object identifier' and 'subordinates' (4), if *sub* is equal to a catenation which extends from some component to the last component of the value of 'subordinates' (5), and if all subordinate object descriptions of *so* are in *sos* (6) and have the attribute 'object class' (7).

SEQCLASSIDIMSUB returns the sequence of object class identifiers drawn from those immediate subordinate object descriptions which are referred to using the numbers of *sub* along with the object identifier of *so*. The object class identifiers in the obtained sequence appear in the same order as their comprising object descriptions according to the catenation *sub*. The sequence is obtained through recursive application of the function: a sequence of one object class identifier is formed using the first component of *sub* (9-11) and is concatenated with the result of SEQCLASSIDIMSUB applied to the remaining rearpart of *sub* (14) or with the empty catenation (13) if the recursion is to end.

Definition 2.156

```
1   ∀ sos, so, sub
2   (0SEQCLASSIDIMSUB(sos, so, sub) =
3     IF IsAttributeSet^2.61(so) and
4        ['object identifier'; 'subordinates'] ⊆ NAMS^1.18(so) and
5        ∃ m ∈̇ (^so • 'subordinates' •) (sub = [ → C m → ]//R m) and
6        IMSUB^2.154(so, sos) ⊆ sos and
7        ∀ cst ∈ IMSUB^2.154(so, sos) ('object class' ∈ NAMS^1.18(cst))
8     THEN
9     [ →
10    C ^sos • <1 xs ||C xs • 'object identifier' =
11                  C ^so • 'object identifier'//[ → HEAD^1.13(sub) → ] 1> • 'object class' → ] //
12    IF LENGTH^1.16 sub = 1
13    THEN [ → ]
14    ELSE SEQCLASSIDIMSUB^2.156(sos, so, TAIL^1.15(sub))
15    ELSE UNDEF0)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

7.5 Additional terminological definitions

Semiformal Description 2.157

Predicate "is a constituent"

A constituent is a document profile or an object class description or an object description or a content portion description or a layout style or a presentation style.

Definition 2.157

1 $\forall\, cst$
2 $({}_{0}\text{IsConstituent}(cst)$ *iff*
3 $\text{IsDocumentProfile}^{4.1}(cst)$ *or*
4 $\text{IsObjectClassDescription}^{2.35}(cst)$ *or*
5 $\text{IsObjectDescription}^{2.53}(cst)$ *or*
6 $\text{IsContentPortionDescription}^{2.54}(cst)$ *or*
7 $\text{IsLayoutStyle}^{2.55}(cst)$ *or*
8 $\text{IsPresentationStyle}^{2.56}(cst)_{0})$

Semiformal Description 2.158

Predicate "is a page class description" (clause 7.5.7)

A page class description is either a composite page class description or a basic page class description.

Definition 2.158

1 $\forall\, cst$
2 $({}_{0}\text{IsPageClassDescription}(cst)$ *iff*
3 $\text{IsCompositePageClassDescription}^{2.26}(cst)$ *or* $\text{IsBasicPageClassDescription}^{2.27}(cst)_{0})$

Semiformal Description 2.159

Predicate "is a logical content portion description"

A logical content portion description is a content portion description where the attribute 'content identifier logical' is specified.

Definition 2.159

1 $\forall\, logcont$
2 $({}_{0}\text{IsLogicalContentPortionDescription}(logcont)$ *iff*
3 $\text{IsContentPortionDescription}^{2.54}(logcont)$ *and* 'content identifier logical' $\in \text{NAMS}^{1.18}(logcont)_{0})$

Semiformal Description 2.160

Predicate "is a layout content portion description"

A layout content portion description is a content portion description where the attribute 'content identifier layout' is specified.

Definition 2.160

1 $\forall\, laycont$
2 $({}_{0}\text{IsLayoutContentPortionDescription}(laycont)$ *iff*
3 $\text{IsContentPortionDescription}^{2.54}(laycont)$ *and* 'content identifier layout' $\in \text{NAMS}^{1.18}(laycont)_{0})$

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 2.161

Predicate "is a logical identifier"

A logical identifier is a sequence of non-negative integers where the first integer is 2 or 3.

Definition 2.161

1 $\forall v$
2 $(_0$IsLogicalId(v) *iff*
3 IsSeqOfNnInt$^{1.23}(v)$ *and*
4 $(_1$ HEAD$^{1.13}(v) = 2$ *or* HEAD$^{1.13}(v) = 3_1)_0)$

Semiformal Description 2.162

Predicate "is a layout identifier"

A layout identifier is a sequence of non-negative integers where the first integer is 0 or 1.

Definition 2.162

1 $\forall v$
2 $(_0$IsLayoutId(v) *iff*
3 IsSeqOfNnInt$^{1.23}(v)$ *and*
4 $(_1$ HEAD$^{1.13}(v) = 0$ *or* HEAD$^{1.13}(v) = 1_1)_0)$

Semiformal Description 2.163

Predicate "is a specific identifier" (clause 9.3.1.2)

A specific identifier is either a specific logical identifier or a specific layout identifier.

Definition 2.163

1 $\forall v$
2 $(_0$IsSpecificId(v) *iff*
3 IsLogicalObjectId$^{2.77}(v)$ *or* IsLayoutObjectId$^{2.78}(v)_0)$

Semiformal Description 2.164

Predicate "is a generic identifier" (clause 9.3.1.3)

A generic identifier is either a generic logical identifier or a generic layout identifier.

Definition 2.164

1 $\forall v$
2 $(_0$IsGenericId(v) *iff*
3 IsLogicalObjectClassId$^{2.80}(v)$ *or* IsLayoutObjectClassId$^{2.81}(v)_0)$

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 2.165

Predicate "is a factor set" (clause 7.5.2)

An entity g is a factor set if it is a generic description which consists only of initial generic constitutents, i.e., no object class description has the attribute 'generator for subordinates'.

Definition 2.165

1 $\forall g$
2 $({}_{o}\text{IsFactorSet}(g)$ *iff*
3 $\text{IsGenericDescription}^{2.9}(g)$ *and*
4 $\forall cst \in g$ (*not* 'generator for subordinates' $\in \text{NAMS}^{1.18}(cst))_{o})$

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

SatisfiesCharacterContentArchitectureConstraints 2.2, 2.4, 2.5, 2.6, 2.17
SatisfiesGeometricGraphicsContentArchitectureConstraints 2.2, 2.4, 2.5, 2.6, 2.19
SatisfiesRasterGraphicsContentArchitectureConstraints 2.2, 2.4, 2.5, 2.6, 2.18
separation 2.60, 2.62, 2.98
SEQCLASSIDIMSUB 2.135, 2.156, 2.156
sequence 2.86
side of sheet 2.105
solid 2.100
specific layout structure 2.4, 2.5, 2.6, 2.20, 2.63
specific logical structure 2.4, 2.5, 2.6, 2.20, 2.63
subordinates 2.7, 2.39, 2.40, 2.43, 2.44, 2.45, 2.48, 2.62, 2.135, 2.140, 2.143, 2.152, 2.154, 2.156
SUPERIOR-OBJECT 2.70, 2.71
SUPERIOR 2.75
synchronization 2.60, 2.62
TAIL 1.15, 2.156
trailing edge 2.98, 2.100, 2.116
trailing offset 2.98, 2.114
transparency 2.28, 2.29, 2.30, 2.31, 2.45, 2.46, 2.48, 2.49, 2.61, 2.62, 2.97
transparent 2.103
type of coding 2.56, 2.62
UNDEF 1.11, 1.12, 1.13, 1.14, 1.15, 1.16, 1.17, 1.18, 2.154, 2.155, 2.156, 2.168, 2.169
unprotected 2.106
unspecified 2.105
UPPER-ALPHA 2.67
UPPER-ROMAN 2.67
user-readable comments 2.22, 2.23, 2.24, 2.26, 2.27, 2.28, 2.29, 2.30, 2.31, 2.39, 2.40, 2.41, 2.43, 2.44, 2.45, 2.46, 2.48, 2.49, 2.60, 2.61, 2.62
user-visible name 2.22, 2.23, 2.24, 2.26, 2.27, 2.28, 2.29, 2.30, 2.31, 2.39, 2.40, 2.41, 2.43, 2.44, 2.45, 2.46, 2.48, 2.49, 2.60, 2.61, 2.62
variable page height 2.28, 2.30, 2.31, 2.45, 2.48, 2.49, 2.99
variable position 2.98
verso 2.105
vertical dimension 2.99
vertical position 2.98
white 2.104

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

and ... IsSetOfPersonalNames$^{4.46}(v)$ ...
and ... IsPersonalName$^{4.47}(v)$ ...
and ... IsAlternativeFeaturesSetsValue$^{4.48}(v)$ ...
and ... IsASN1ObjectIdentifierSet$^{4.49}(v)$ ...
and ... IsEncipheredDocumentProfilesValue$^{4.50}(v)$ ...
and ... IsPrivilegedRecipientInformationValue$^{4.51}(v)$ ...
and ... IsKeyInformationValue$^{4.52}(v)$ ...
and ... IsMethodInformationValue$^{4.53}(v)$ ...
and ... IsKeyAdditionalInformationValue$^{4.54}(v)$ ...
and ... IsKeyMethodValue$^{4.55}(v)$ ...
and ... IsEncipheredDocumentProfileId$^{4.56}(v)$ ...
and ... IsODASecurityLabelValue$^{4.57}(v)$ ...
and ... IsPreEncipheredDocumentBodyPartsValue$^{4.58}(v)$ ...
and ... IsPreEncipheredDocumentBodyPartId$^{4.59}(v)$ ...
and ... IsPostEncipheredDocumentBodyPartsValue$^{4.60}(v)$ ...
and ... IsPostEncipheredDocumentBodyPartId$^{4.61}(v)$ ...
and ... IsSealedDocumentBodyPartsValue$^{4.62}(v)$ ...
and ... IsDocumentSealValue$^{4.63}(v)$ ...
and ... IsSeqOfConstituentIds$^{4.64}(v)$ ...
and ... IsConstituentId$^{4.65}(v)$ ...
and ... IsSealMethodValue$^{4.66}(v)$ ...
and ... IsSealedInformationValue$^{4.67}(v)$ ...
and ... IsLocationValue$^{4.68}(v)$ ...
and ... IsSealedDocumentProfilesValue$^{4.69}(v)$ ...
and ... IsSealedDocumentProfileId$^{4.70}(v)$ ...
and ...IsColourOfLayoutObjectsValue$^{4.71}(v)$ ...
and ...IsContentBackgroundColoursValue$^{4.72}(v)$ ...
and ...IsContentForegroundColoursValue$^{4.73}(v)$ ...
and ...IsColourTablesValue$^{4.74}(v)$ ...
and ...IsColourCharacteristicsValue$^{4.75}(v)$ ...
and ...IsColourModesPresentValue$^{4.76}(v)$ ...
and ...IsColourSpacesPresentValue$^{4.77}(v)$ ...
and ...IsColourCalibrationTypeValue$^{4.78}(v)$ ...
and ...IsColourSpaceTypeValue$^{4.79}(v)$ ...
and ...IsColourSpacesListValue$^{4.80}(v)$ ...
and ...IsColourSpaceDefinition$^{4.81}(v)$ ...
and ...IsCalibrationDataValue$^{4.82}(v)$ ...
and ...IsCIECalibrationData$^{4.83}(v)$ ...
and ...IsRGBCalibrationData$^{4.84}(v)$ ...
and ...IsCMYKCalibrationData$^{4.85}(v)$ ...
and ...IsReferenceWhiteValue$^{4.86}(v)$ ...
and ...IsColourLookupTable$^{4.87}(v)$ ...
and ...IsColourTableList$^{4.88}(v)$ ...
and ...IsColourTableCorrection$^{4.89}(v)$ ...
and ...IsGridSpecificationValue$^{4.90}(v)$ ...
and ...IsGridSpecification$^{4.91}(v)$ ...
and ...Is3x3Matrix$^{4.92}(v)$ ...
and ...IsColourDataScalingValue$^{4.93}(v)$ ...
and ...IsScaleAndOffsetValue$^{4.94}(v)$ ...
and ...IsCMYKColour$^{4.95}(v)$ ...
and ...IsCIEColour$^{4.96}(v)$ ...

NOTE — Other predicates or operators which are used here, but are defined in other clauses, are not listed here.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

8.1 The Document Profile

Semiformal Description 4.1

Predicate "is a document profile"

A document profile is a set of profile attributes. The attributes 'content architecture classes', 'document architecture class', 'document reference', 'interchange format class' and 'ODA version' are mandatory, the other ones are optional.

If the value of the attribute 'document architecture class' is 'processable', and the attributes 'generic layout structure' and 'specific logical structure', but not 'specific layout structure' are specified, the value of the attribute 'generic layout structure' is 'complete generator set'. If the value of the attribute 'document architecture class' is 'formatted processable', and the attributes 'generic layout structure', 'specific logical structure' and 'specific layout structure' are specified, the value of the attribute 'generic layout structure' is 'complete generator set'. If the value of the attribute 'document architecture class' is 'formatted', and the attributes 'generic layout structure' and 'specific layout structure', but not 'specific logical structure' and 'generic logical structure' are specified, the value of the attribute 'generic layout structure' is either 'partial generator set' or 'factor set'.

If the attribute 'document application profile defaults' is specified, the attribute 'document application profile' must also be specified.

Definition 4.1

$\forall\, cst$
$({}_{o}\mathrm{IsDocumentProfile}(cst)$ *iff*
  $\mathrm{IsProfileAttributeSet}^{4.2}(cst)$ *and*
  $\mathrm{NAMS}^{1.18}(cst) \supseteq$ ['content architecture classes'; 'document architecture class';
    'document reference'; 'interchange format class';
    'ODA version'] *and*
  $\mathrm{NAMS}^{1.18}(cst) \subseteq$ ['abstract'; 'access rights';
    'additional information'; 'alternative feature sets';
    'alternative representation character sets'; 'authorization';
    'authors'; 'block alignments';
    'borders'; 'coding attributes';
    'colour characteristics'; 'colour of layout objects';
    'colours'; 'colour spaces list';
    'comments character sets'; 'content architecture classes';
    'content background colours'; 'content colour tables';
    'content foreground colours'; 'copyright';
    'creation date and time'; 'distribution list';
    'document application profile'; 'document application profile defaults';
    'document architecture class'; 'document date and time';
    'document reference'; 'document size';
    'document type'; 'enciphered document profiles';
    'enciphered profiles'; 'expiry date and time';
    'external-document class'; 'fill orders';
    'fonts list'; 'generic layout structure';
    'generic logical structure'; 'interchange format class';
    'keywords'; 'languages';
    'layout paths'; 'layout styles';
    'local file references'; 'local filing date and time';
    'medium types'; 'number of objects per page';
    'number of pages'; 'object colour tables';
    'ODA security label'; 'ODA version';
    'organizations'; 'owners';
    'page dimensions'; 'page positions';
    'post-enciphered body parts'; 'post-enciphered document body parts';

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
35                    'post-sealed document body parts';      'pre-enciphered body parts';
36                    'pre-enciphered document body parts';   'preparers';
37                    'pre-sealed document body parts';       'presentation features';
38                    'presentation styles';                  'profile character sets';
39                    'protections';                          'purge date and time';
40                    'references to other documents';        'release date and time';
41                    'resource document';                    'resources';
42                    'revision history';                     'sealed document profiles';
43                    'sealed profiles';                      'security classification';
44                    'specific layout structure';            'specific logical structure';
45                    'start date and time';                  'status';
46                    'subject';                              'superseded documents';
47                    'title';                                'transparencies';
48                    'types of coding';                      'unit scaling';
49                    'user-specific codes'] and
50  (1(2C ^prof • 'document architecture class' = 'processable' and
51      'specific logical structure' ∈ NAMS^1.18(prof) and
52      'specific layout structure' ∉ NAMS^1.18(prof) and
53      (3 'generic layout structure' ∈ NAMS^1.18(prof) impl
54        C ^prof • 'generic layout structure' = 'complete generator set' 3)2) or
55    (4C ^prof • 'document architecture class' = 'formatted processable' and
56      'specific logical structure' ∈ NAMS^1.18(prof) and
57      'specific layout structure' ∈ NAMS^1.18(prof) and
58      'generic layout structure' ∈ NAMS^1.18(prof) and
59      C ^prof • 'generic layout structure' = 'complete generator set' 4) or
60    (5C ^prof • 'document architecture class' = 'formatted' and
61      'specific logical structure' ∉ NAMS^1 18(prof) and
62      'generic logical structure' ∉ NAMS^1.18(prof) and
63      'specific layout structure' ∈ NAMS^1.18(prof) and
64      (6 'generic layout structure' ∈ NAMS^1.18(prof) impl
65        C ^prof • 'generic layout structure' ∈ ['partial generator set';'factor set'] 6)5)1) and
66  (7 'document application profile defaults' ∈ NAMS^1.18(cst) impl
67    'document application profile' ∈ NAMS^1.18(cst) 7)0)
```

8.2 Attributes of the document profile

Semiformal Description 4.2

Predicate "is a profile attribute set" (clause 7, table B.1)

In this definition the attribute names are linked with the ranges of the attribute values.

Definition 4.2

```
1   ∀as
2   (0IsProfileAttributeSet(as) iff
3     (1 IsNeNom^1.2(as) and
4       ∀a ∈ ^as •
5       (2 N a = 'abstract' impl IsProfCharSeq^4.4(C a) 2) and
6       (3 N a = 'access rights' impl IsSetOfProfCharSeq^4.3(C a) 3) and
7       (4 N a = 'additional information' impl IsAdditionalInformationValue^4.42(C a) 4) and
8       (5 N a = 'alternative feature sets' impl IsAlternativeFeaturesSetsValue^4.48(C a) 5) and
9       (6 N a = 'alternative representation character sets' impl IsCharacterSetsValue^6.23(C a) 6) and
10      (7 N a = 'authorization' impl IsAuthorizationValue^4.45(C a) 7) and
11      (8 N a = 'authors' impl IsAuthorsValue^4.36(C a) 8) and
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
12      (9 N a = 'block alignments' impl IsBlockAlignmentsValue4.17(C a) 9) and
13      (10 N a = 'borders' impl IsBordersValue4.21(C a) 10) and
14      (11 N a = 'coding attributes' impl IsProfileCodingAttributesValue4.24(C a) 11) and
15      (12 N a = 'colour characteristics' impl IsColourCharacteristicsValue4.75(C a) 12) and
16      (13 N a = 'colour of layout objects' impl IsColourOfLayoutObjectsValue4.71(C a) 13) and
17      (14 N a = 'colours' impl IsColoursValue4.20(C a) 14) and
18      (15 N a = 'colour spaces list' impl IsColourSpacesListValue4.80(C a) 15) and
19      (16 N a = 'comments character sets' impl IsCharacterSetsValue6.23(C a) 16) and
20      (17 N a = 'content architecture classes' impl IsContentArchitectureClassesValue4.11(C a) 17) and
21      (18 N a = 'content background colours' impl IsContentBackgroundColoursValue4.72(C a) 18) and
22      (19 N a = 'content colour tables' impl IsColourTablesValue4.74(C a) 19) and
23      (20 N a = 'content foreground colours' impl IsContentForegroundColoursValue4.73(C a) 20) and
24      (21 N a = 'copyright' impl IsCopyrightValue4.37(C a) 21) and
25      (22 N a = 'creation date and time' impl IsDateAndTimeValue4.31(C a) 22) and
26      (23 N a = 'distribution list' impl IsDistributionListValue4.41(C a) 23) and
27      (24 N a = 'document application profile' impl IsDocumentApplicationProfileValue4.9(C a) 24) and
28      (25 N a = 'document application profile defaults' impl
29        IsDocumentApplicationProfileDefaultsValue4.10(C a) 25) and
30      (26 N a = 'document architecture class' impl
31        C a ∈ ['formatted'; 'processable'; 'formatted processable'] 26) and
32      (27 N a = 'document date and time' impl IsDateAndTimeValue4.31(C a) 27) and
33      (28 N a = 'document reference' impl IsExternalReferenceValue4.6(C a) 28) and
34      (29 N a = 'document size' impl IsNat(C a) 29) and
35      (30 N a = 'document type' impl IsProfCharSeq4.4(C a) 30) and
36      (31 N a = 'enciphered document profiles' impl IsEncipheredDocumentProfilesValue4.50(C a) 31) and
37      (32 N a = 'enciphered profiles' impl C a = 'present' 32) and
38      (33 N a = 'expiry date and time' impl IsDateAndTimeValue4.31(C a) 33) and
39      (34 N a = 'external-document class' impl IsExternalReferenceValue4.6(C a) 34) and
40      (35 N a = 'fill orders' impl IsFillOrdersValue4.18(C a) 35) and
41      (36 N a = 'fonts list' impl IsFontsListValue4.26(C a) 36) and
42      (37 N a = 'generic layout structure' impl
43        C a ∈ ['factor set'; 'partial generator set'; 'complete generator set'] 37) and
44      (38 N a = 'generic logical structure' impl
45        C a ∈ ['factor set'; 'partial generator set'; 'complete generator set'] 38) and
46      (39 N a = 'interchange format class' impl C a ∈ ['A'; 'B'] 39) and
47      (40 N a = 'keywords' impl IsSetOfProfCharSeq4.3(C a) 40) and
48      (41 N a = 'languages' impl IsSetOfProfCharSeq4.3(C a) 41) and
49      (42 N a = 'layout paths' impl IsLayoutPathsValue4.15(C a) 42) and
50      (43 N a = 'layout styles' impl C a = 'present' 43) and
51      (44 N a = 'local file references' impl IsLocalFileReferenceValue4.44(C a) 44) and
52      (45 N a = 'local filing date and time' impl IsDateAndTimeSeq4.39(C a) 45) and
53      (46 N a = 'medium types' impl IsMediumTypesValue4.14(C a) 46) and
54      (47 N a = 'number of objects per page' impl IsNat(C a) 47) and
55      (48 N a = 'number of pages' impl IsNnInt1.7(C a) 48) and
56      (49 N a = 'object colour tables' impl IsColourTablesValue4.74(C a) 49) and
57      (50 N a = 'ODA security label' impl IsODASecurityLabelValue4.57(C a) 50) and
58      (51 N a = 'ODA version' impl IsODAVersionValue4.12(C a) 51) and
59      (52 N a = 'organizations' impl IsSetOfProfCharSeq4.3(C a) 52) and
60      (53 N a = 'owners' impl IsOwnersValue4.35(C a) 53) and
61      (54 N a = 'page dimensions' impl IsPageDimensionsValue4.13(C a) 54) and
62      (55 N a = 'page positions' impl IsPagePositionsValue4.22(C a) 55) and
63      (56 N a = 'post-enciphered body parts' impl C a = 'present' 56) and
64      (57 N a = 'post-enciphered document body parts' impl
65        IsPostEncipheredDocumentBodyPartsValue4.60(C a) 57) and
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
66      (58 N a = 'post-sealed document body parts' impl IsSealedDocumentBodyPartsValue4.62(C a) 58) and
67      (59 N a = 'pre-enciphered body parts' impl C a = 'present' 59) and
68      (60 N a = 'pre-enciphered document body parts' impl
69        IsPreEncipheredDocumentBodyPartsValue4.58(C a) 60) and
70      (61 N a = 'preparers' impl IsPreparersValue4.34(C a) 61) and
71      (62 N a = 'pre-sealed document body parts' impl IsSealedDocumentBodyPartsValue4.62(C a) 62) and
72      (63 N a = 'presentation features' impl IsPresentationFeaturesValue4.25(C a) 63) and
73      (64 N a = 'presentation styles' impl C a = 'present' 64) and
74      (65 N a = 'profile character sets' impl IsCharacterSetsValue6.23(C a) 65) and
75      (66 N a = 'protections' impl IsProtectionsValue4.16(C a) 66) and
76      (67 N a = 'purge date and time' impl IsDateAndTimeValue4.31(C a) 67) and
77      (68 N a = 'references to other documents' impl IsSetOfExternalReferences4.43(C a) 68) and
78      (69 N a = 'release date and time' impl IsDateAndTimeValue4.31(C a) 69) and
79      (70 N a = 'resource document' impl IsExternalReferenceValue4.6(C a) 70) and
80      (71 N a = 'resources' impl IsResourcesValue4.8(C a) 71) and
81      (72 N a = 'revision history' impl IsRevisionHistoryValue4.32(C a) 72) and
82      (73 N a = 'sealed document profiles' impl IsSealedDocumentProfilesValue4.69(C a) 73) and
83      (74 N a = 'sealed profiles' impl C a = 'present' 74) and
84      (75 N a = 'security classification' impl IsProfCharSeq4.4(C a) 75) and
85      (76 N a = 'specific layout structure' impl C a = 'present' 76) and
86      (77 N a = 'specific logical structure' impl C a = 'present' 77) and
87      (78 N a = 'start date and time' impl IsDateAndTimeValue4.31(C a) 78) and
88      (79 N a = 'status' impl IsProfCharSeq4.4(C a) 79) and
89      (80 N a = 'subject' impl IsProfCharSeq4.4(C a) 80) and
90      (81 N a = 'superseded documents' impl IsSetOfExternalReferences4.43(C a) 81) and
91      (82 N a = 'title' impl IsProfCharSeq4.4(C a) 82) and
92      (83 N a = 'transparencies' impl IsTransparenciesValue4.19(C a) 83) and
93      (84 N a = 'types of coding' impl IsTypesOfCodingValue4.23(C a) 84) and
94      (85 N a = 'unit scaling' impl IsPairOfPosInt1.9(C a) 85) and
95      (86 N a = 'user-specific codes' impl IsSetOfProfCharSeq4.3(C a) 86)1)0)
```

Semiformal Description 4.3

Predicate "is a set of profile character sequences"

A set of profile character sequences is a set where each element is a profile character sequence.

Definition 4.3

```
1   ∀ v
2   (0 IsSetOfProfCharSeq(v) iff
3     IsNeCol1.1(v) and ∀ m ∈ v (IsProfCharSeq4.4(m)) 0)
```

Semiformal Description 4.4

Predicate "is a profile character sequence" (e.g., clause 7.2.12)

A profile character sequence is a catenation of characters from the document profile character set.

Definition 4.4

```
1   ∀ v
2   (0 IsProfCharSeq(v) iff
3     IsNeCat1.3(v) and ∀ m ∈̂ ~v • (IsProfileCharacter4.5(C m)) 0)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 4.5

Predicate "is a profile character" (e.g., clause 7.2.12)

A profile character is considered an atomic construct in the formal specification.

Definition 4.5

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

Semiformal Description 4.6

Predicate "is an external reference value" (e.g., clause 7.2.11)

An external reference value is either an ASN.1 object identifier or a sequence of characters from the document profile character set.

Definition 4.6

1 $\forall v$
2 $(_0$ IsExternalReferenceValue(v) *iff*
3 $(_1$ IsASN1ObjectIdentifier$^{4.7}(v)$ *or* IsProfCharSeq$^{4.4}(v)_1)_0)$

Semiformal Description 4.7

Predicate "is an ASN.1 object identifier" (e.g., clause 7.2.11)

An ASN.1 object identifier is considered an atomic construct in the formal specification.

Definition 4.7

1 $\forall v$
2 $(_0$ IsASN1ObjectIdentifier(v) *iff*
3 IsAtom$(v)_0)$

Semiformal Description 4.8

Predicate "is a resources value" (clause 7.2.13)

The value of the attribute **'resources'** is a set where each element consists of a pair, the first part being a sequence of characters from the minimum subrepertoire of ISO 6937-2, the second part being an object class identifier.

Definition 4.8

1 $\forall v$
2 $(_0$ IsResourcesValue(v) *iff*
3 IsNeCol$^{1.1}(v)$ *and*
4 $\forall a \in v \; \exists l, r$
5 $(_1 a = [\rightarrow l \rightarrow r \rightarrow]$ *and*
6 IsISO6937MSString$^{1.20}(l)$ *and* IsObjectClassId$^{2.82}(r)_1)_0)$

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 4.9

Predicate "is a document application profile value" (clause 7.3.1)

The value of the attribute 'document application profile' is either an ASN.1 object identifier or the integer 2.

Definition 4.9

1 $\forall v$
2 $({}_0$IsDocumentApplicationProfileValue(v) *iff*
3 $({}_1$IsASN1ObjectIdentifier$^{4.7}(v)$ *or* $v = 2_1)_0)$

Semiformal Description 4.10

Predicate "is a document application profile defaults value" (clause 7.3.2)

A document application profile defaults value is either a non-empty nomination, where the names are 'block alignment', 'border', 'colour', 'colour of layout object', 'content architecture class', 'content background colour', 'content colour table', 'content foreground colour', 'dimensions', 'layout path', 'medium type', 'page position', 'object colour table', 'transparency', and/or 'type of coding' and the component is a value of the corresponding attribute as defined in ISO/IEC 8613-2, or it is a defaultable attribute from the character content architecture, the raster graphics architecture or the geometric graphics architecture.

Definition 4.10

1 $\forall v$
2 $({}_0$IsDocumentApplicationProfileDefaultsValue(v) *iff*
3 $({}_1({}_2$IsNeNom$^{1.2}(v)$ *and*
4 NAMS$^{1.18}(v) \subseteq$ ['block alignment'; 'border';
5 'colour'; 'colour of layout object';
6 'content architecture class'; 'content background colour';
7 'content colour table'; 'content foreground colour';
8 'dimensions'; 'layout path';
9 'medium type'; 'page position';
10 'object colour table'; 'transparency';
11 'type of coding'] *and*
12 $\forall a \in \hat{}v \bullet$
13 $({}_3$N a = 'block alignment' *impl* IsBlockAlignmentValue$^{2.101}$(C a) $_3$) *and*
14 $({}_4$N a = 'border' *impl* IsBorderValue$^{2.93}$(C a) $_4$) *and*
15 $({}_5$N a = 'colour' *impl* IsColourValue$^{2.97}$(C a) $_5$) *and*
16 $({}_6$N a = 'colour of layout object' *impl* IsColourExpression$^{2.126}$(C a) $_6$) *and*
17 $({}_7$N a = 'content architecture class' *impl* IsContentArchitectureClassValue$^{2.87}$(C a) $_7$) *and*
18 $({}_8$N a = 'content background colour' *impl* IsColourExpression$^{2.126}$(C a) $_8$) *and*
19 $({}_9$N a = 'content colour table' *impl* IsColourTable$^{2.131}$(C a) $_9$) *and*
20 $({}_{10}$N a = 'content foreground colour' *impl* IsColourExpression$^{2.126}$(C a) $_{10}$) *and*
21 $({}_{11}$N a = 'dimensions' *impl* IsDimensionsValue$^{2.92}$(C a) $_{11}$) *and*
22 $({}_{12}$N a = 'layout path' *impl* IsLayoutPathValue$^{2.95}$(C a) $_{12}$) *and*
23 $({}_{13}$N a = 'medium type' *impl* IsMediumTypeValue$^{2.98}$(C a) $_{13}$) *and*
24 $({}_{14}$N a = 'object colour table' *impl* IsColourTable$^{2.131}$(C a) $_{14}$) *and*
25 $({}_{15}$N a = 'page position' *impl* IsPairOfNnInt$^{1.8}$(C a) $_{15}$) *and*
26 $({}_{16}$N a = 'transparency' *impl* IsTransparencyValue$^{2.96}$(C a) $_{16}$) *and*
27 $({}_{17}$N a = 'type of coding' *impl* IsTypeOfCodingValue$^{2.113}$(C a) $_{17})_2$) *or*
28 IsDefaultableCharacterContentArchitectureAttribute$^{6.1}(v)$ *or*
29 IsDefaultableRasterGraphicsContentArchitectureAttribute$^{7.1}(v)$ *or*
30 IsDefaultableGeometricGraphicsContentArchitectureAttribute$^{8.1}(v)_1)_0)$

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Semiformal Description 4.11

Predicate "is a content architecture classes value" (clause 7.3.4)

The value of the attribute 'content architecture classes' is a non-empty set where each element is a content architecture class value.

Definition 4.11

1 $\forall v$
2 $(_0$ IsContentArchitectureClassesValue(v) *iff*
3 IsNeCol$^{1.1}(v)$ *and*
4 $\forall a \in v$ (IsContentArchitectureClassValue$^{2.87}(a)$) $_0)$

Semiformal Description 4.12

Predicate "is an ODA version value" (clause 7.3.6)

An ODA version value is a catenation of two elements. The first element is a sequence of characters from the document profile character set, the second element is a date value in accordance with ISO 8601.

Definition 4.12

1 $\forall v$
2 $(_0$ IsODAVersionValue(v) *iff*
3 $\exists l, r$
4 $(_1 v = [\rightarrow l \rightarrow r \rightarrow]$ *and*
5 IsProfCharSeq$^{4.4}(l)$ *and* IsDateValue$^{4.40}(r)$ $_1)_0)$

Semiformal Description 4.13

Predicate "is a page dimensions value" (clause 7.3.8.4.1)

The value of the attribute 'page dimensions' is a set of pairs of positive integers.

Definition 4.13

1 $\forall v$
2 $(_0$ IsPageDimensionsValue(v) *iff*
3 IsNeCol$^{1.1}(v)$ *and*
4 $\forall a \in v$ (IsPairOfPosInt$^{1.9}(a)$) $_0)$

Semiformal Description 4.14

Predicate "is a medium types value" (clause 7.3.8.4.2)

The value of the attribute 'medium types' is a set where each element is a medium type value.

Definition 4.14

1 $\forall v$
2 $(_0$ IsMediumTypesValue(v) *iff*
3 IsNeCol$^{1.1}(v)$ *and*
4 $\forall a \in v$ (IsMediumTypeValue$^{2.98}(a)$) $_0)$

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 4.15

Predicate "is a layout paths value" (clause 7.3.8.4.3)

The value of the attribute **'layout paths'** is a set where each element is a layout path value.

Definition 4.15

1 $\forall v$
2 ($_{o}$IsLayoutPathsValue(v) *iff*
3 IsNeCol$^{1.1}$(v) *and*
4 $\forall a \in v$ (IsLayoutPathValue$^{2.95}$(a)) $_{o}$)

Semiformal Description 4.16

Predicate "is a protections value" (clause 7.3.8.4.4)

The value of the attribute **'protections'** is a set where each element is a protection value.

Definition 4.16

1 $\forall v$
2 ($_{o}$IsProtectionsValue(v) *iff*
3 IsNeCol$^{1.1}$(v) *and*
4 $\forall a \in v$ (IsProtectionValue$^{2.99}$(a)) $_{o}$)

Semiformal Description 4.17

Predicate "is a block alignments value" (clause 7.3.8.4.5)

The value of the attribute **'block alignments'** is a set where each element is a block alignment value.

Definition 4.17

1 $\forall v$
2 ($_{o}$IsBlockAlignmentsValue(v) *iff*
3 IsNeCol$^{1.1}$(v) *and*
4 $\forall a \in v$ (IsBlockAlignmentValue$^{2.101}$(a)) $_{o}$)

Semiformal Description 4.18

Predicate "is a fill orders value" (clause 7.3.8.4.6)

The value of the attribute **'fill orders'** is a set where each element is a fill order value.

Definition 4.18

1 $\forall v$
2 ($_{o}$IsFillOrdersValue(v) *iff*
3 IsNeCol$^{1.1}$(v) *and*
4 $\forall a \in v$ (IsFillOrderValue$^{2.103}$(a)) $_{o}$)

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 4.19

Predicate "is a transparencies value" (clause 7.3.8.4.7)

The value of the attribute **'transparencies'** is a set where each element is a transparency value.

Definition 4.19

1 $\forall v$
2 ($_{o}$IsTransparenciesValue(v) *iff*
3 IsNeCol$^{1.1}$(v) *and*
4 $\forall a \in v$ (IsTransparencyValue$^{2.96}$(a)) $_{o}$)

Semiformal Description 4.20

Predicate "is a colours value" (clause 7.3.8.4.8)

The value of the attribute **'colours'** is a set where each element is a colour value.

Definition 4.20

1 $\forall v$
2 ($_{o}$IsColoursValue(v) *iff*
3 IsNeCol$^{1.1}$(v) *and*
4 $\forall a \in v$ (IsColourValue$^{2.97}$(a)) $_{o}$)

Semiformal Description 4.21

Predicate "is a borders value" (clause 7.3.8.4.14)

The value of the attribute **'borders'** is a set where each element is a border value.

Definition 4.21

1 $\forall v$
2 ($_{o}$IsBordersValue(v) *iff*
3 IsNeCol$^{1.1}$(v) *and*
4 $\forall a \in v$ (IsBorderValue$^{2.93}$(a)) $_{o}$)

Semiformal Description 4.22

Predicate "is a page positions value" (clause 7.3.8.4.15)

The value of the attribute **'page positions'** is a set where each element is a pair of non-negative integers.

Definition 4.22

1 $\forall v$
2 ($_{o}$IsPagePositionsValue(v) *iff*
3 IsNeCol$^{1.1}$(v) *and*
4 $\forall a \in v$ (IsPairOfNnInt$^{1.8}$(a)) $_{o}$)

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 4.23

Predicate "is a types of coding value" (clause 7.3.8.4.16)

The value of the attribute 'types of coding' is a set where each element is a type of coding value.

Definition 4.23

1 $\forall v$
2 $(_0$ IsTypesOfCodingValue(v) *iff*
3 IsNeCol$^{1.1}(v)$ *and*
4 $\forall a \in v$ (IsTypeOfCodingValue$^{2.113}(a)$) $_0)$

Semiformal Description 4.24

Predicate "is a profile coding attributes value" (clause 7.3.8.5)

The value of the attribute 'coding attributes' is a nomination where the name is 'character coding attributes', 'raster-graphics coding attributes' and/or 'geometric-graphics coding attributes'. The corresponding component is a character coding specification, raster graphics coding specification or geometric graphics coding specification.

Definition 4.24

1 $\forall v$
2 $(_0$ IsProfileCodingAttributesValue(v) *iff*
3 IsNeNom$^{1.2}(v)$ *and*
4 NAMS$^{1.18}(v) \subseteq$ ['character coding attributes'; 'raster-graphics coding attributes';
5 'geometric-graphics coding attributes'] *and*
6 $\forall b \in \hat{}v \bullet$
7 $(_1 (_2$ N b = 'character coding attributes' *impl*
8 IsCharacterCodingSpecification$^{6.2}$(C b) $_2)$ *and*
9 $(_3$ N b = 'raster-graphics coding attributes' *impl*
10 IsRasterGraphicsCodingSpecification$^{7.2}$(C b) $_3)$ *and*
11 $(_4$ N b = 'geometric-graphics coding attributes' *impl*
12 IsGeometricGraphicsCodingSpecification$^{8.2}$(C b) $_4)_1)_0)$

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 4.25

Predicate "is a presentation features value" (clause 7.3.8.6)

The value of the attribute 'presentation features' is a nomination where the name is 'character presentation features', 'raster-graphics presentation features' and/or 'geometric-graphics presentation features'. The corresponding component is a character presentation feature, a raster graphics presentation feature or a geometric graphics presentation feature.

Definition 4.25

```
1   ∀ v
2   (0 IsPresentationFeaturesValue(v) iff
3     IsNeNom1.2(v) and
4     NAMS1.18(v) ⊆ ['character presentation features'; 'raster-graphics presentation features';
5                    'geometric-graphics presentation features'] and
6     ∀ b ∈̂ ^v •
7     (1 (2 N b = 'character presentation features' impl IsCharacterPresentationFeature6.3(C b) 2) and
8       (3 N b = 'raster-graphics presentation features' impl
9         IsRasterGraphicsPresentationFeature7.3(C b) 3) and
10      (4 N b = 'geometric-graphics presentation features' impl
11        IsGeometricGraphicsPresentationFeature8.3(C b) 4)1)0)
```

Semiformal Description 4.26

Predicate "is a fonts list value" (clause 7.3.10.2)

The value of the attribute 'fonts list' is a set of one or more pairs, where the first element of the pair is an integer and the second element is a font reference.

Definition 4.26

```
1   ∀ v
2   (0 IsFontsListValue(v) iff
3     IsNeCol1.1(v) and
4     ∀ b ∈ v ∃ l, r
5     (1 b = [ → l → r → ] and
6       IsInt(l) and IsFontReference4.27(r) 1)0)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 4.27

Predicate "is a font reference" (clause 7.3.10.2)

A font reference is a non-empty nomination where the name is 'reference properties', 'user-readable comments' and/or 'user-visible name'. At least the name 'reference properties' must be specified. For the name 'reference properties' the component is a set of reference properties. For the names 'user-readable comments' and 'user-visible name' the components are character sequences.

Definition 4.27

```
1  ∀ v
2  (0 IsFontReference(v) iff
3    IsNeNom1.2(v) and
4    NAMS1.18(v) ⊇ ['reference properties'; 'user-readable comments'; 'user-visible name'] and
5    'reference properties' ∈ NAMS1.18(v) and
6    ∀ a ∈̂ ^v •
7    (1 (2 N a = 'reference properties' impl IsSetOfReferenceProperties4.28(C a)2)
8      (3 N a = 'user-readable comments' impl IsCharSeq2.117(C a)3)
9      (4 N a = 'user-visible name' impl IsCharSeq2.117(C a)4) 1)0)
```

Semiformal Description 4.28

Predicate "is a set of reference properties" (clause 7.3.10.2)

A set of reference properties is a non-empty collection whose elements are reference properties values.

Definition 4.28

```
1  ∀ v
2  (0 IsSetOfReferenceProperties(v) iff
3    IsNeCol1.1(v) and
4    ∀ a ∈ v IsReferencePropertiesValue4.29(a)0)
```

Semiformal Description 4.29

Predicate "is a reference properties value" (clause 7.3.10.2)

A reference properties value is a non-empty nomination where the name is 'precedence number', 'properties' and/or 'user-readable comments'. At least the name 'properties' must be specified. For the name 'precedence number' the component is a non-negative integer. For the name 'properties' the value is an ISO 9541, Part2, attribute set. For the name 'user-readable comments' the component is a character sequences.

Definition 4.29

```
1  ∀ v
2  (0 IsReferencePropertiesValue(v) iff
3    IsNeNom1.2(v) and
4    NAMS1.18(v) ⊇ ['precedence number'; 'properties'; 'user-readable comments'] and
5    'properties' ∈ NAMS1.18(v) and
6    ∀ a ∈̂ ^v •
7    (1 (2 N a = 'precedence number' impl IsNnInt1.7(C a)2)
8      (3 N a = 'properties' impl IsISO9541Part2AttributeSet4.30(C a)3)
9      (4 N a = 'user-readable comments' impl IsCharSeq2.117(C a)4) 1)0)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 4.30

Predicate "is an ISO 9541, Part 2, attribute set" (clause 7.3.10.2)

An ISO 9541, Part 2, attribute set is considered an atomic construct in the formal specification of the document profile.

Definition 4.30

1 $\forall v$
2 $(_{o}$ IsISO9541Part2AttributeSet(v) *iff*
3 IsAtom(v)$_{o}$)

Semiformal Description 4.31

Predicate "is a date and time value" (e.g., clause 7.4.2.1)

A date and time value is a date character string and, optionally, a time of day character string, in accordance with ISO 8601. It is considered an atomic construct in the formal specification.

Definition 4.31

1 $\forall v$
2 $(_{o}$ IsDateAndTimeValue(v) *iff*
3 IsAtom(v)$_{o}$)

Semiformal Description 4.32

Predicate "is a revision history value" (clause 7.4.2.8)

The value of the attribute 'revision history' is a non-empty sequence of revision values.

Definition 4.32

1 $\forall v$
2 $(_{o}$ IsRevisionHistoryValue(v) *iff*
3 IsNeCat$^{1.3}$(v) *and*
4 $\forall a \hat{\in} \hat{} v \bullet$ (IsRevisionValue$^{4.33}$(C a)) $_{o}$)

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 4.33

Predicate "is an revision value" (e.g., clause 7.4.2.8)

A revision value is a nomination where the name is 'revision date and time' (the corresponding component is a data and time value), 'version number' (the component is an integer), 'version reference' (the component is an external reference value), 'user comments' (the component is a sequence of characters from the document profile character set) or 'revisers'. For the last case, the component is a nomination again, where the (nomination) name is 'names' (the component is a set of personal names), 'position' and/or 'organization' (the corresponding components are sequences of characters from the document profile character set). At least one of these three sub-parameters must be specified.

Definition 4.33

```
1   ∀ v
2   (0 IsRevisionValue(v) iff
3     IsNom(v) and
4     NAMS1.18(v) ⊆ ['revisers'; 'revision date and time'; 'user comments';
5                    'version number'; 'version reference'] and
6     ∀ a ∈ ^v •
7     (1 (2 N a = 'revisers' impl
8         (3 IsNeNom1.2(C a) and
9           NAMS1.18(C a) ⊆ ['names'; 'position'; 'organization'] and
10          ∀ b ∈ ^(C a) •
11          (4 (5 N b = 'names' impl IsSetOfPersonalNames4.46(C b) 5) and
12            (6 N b ∈ ['position'; 'organization'] impl IsProfCharSeq4.4(C b) 6)4)3)2) and
13      (7 N a = 'revision date and time' impl IsDateAndTimeValue4.31(C a) 7) and
14      (8 N a = 'user comments' impl IsProfCharSeq4.4(C a) 8) and
15      (9 N a = 'version number' impl IsProfCharSeq4.4(C a) 9) and
16      (10 N a = 'version reference' impl IsExternalReferenceValue4.6(C a) 10)1)0)
```

Semiformal Description 4.34

Predicate "is a preparers value" (clause 7.4.3.2)

The value of the attribute 'preparers' is a set of nominations. For each nomination, the (nomination) name is 'personal name of preparer' (the corresponding component is a personal name) and/or 'preparers organization' (the corresponding component is a sequence of characters from the document profile character set). At least one of these parameters must be specified.

Definition 4.34

```
1   ∀ v
2   (0 IsPreparersValue(v) iff
3     IsNeCol1.1(v) and
4     ∀ b ∈ v
5     (1 IsNeNom1.2(b) and
6       NAMS1.18(b) ⊆ ['personal name of preparer'; 'preparers organization'] and
7       ∀ a ∈̂ ^b •
8       (2 (3 N a = 'personal name of preparer' impl IsPersonalName4.47(C a) 3) and
9         (4 N a = 'preparers organization' impl IsProfCharSeq4.4(C a) 4)2)1)0)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 4.35

Predicate "is an owners value" (clause 7.4.3.3)

The value of the attribute 'owners' is a set of nominations. For each nomination, the (nomination) name is 'personal name of owner' (the corresponding component is a personal name) and/or 'owners organization' (the corresponding component is a sequence of characters from the document profile character set). At least one of these parameters must be specified.

Definition 4.35

1 $\forall v$
2 $(_0$IsOwnersValue(v) *iff*
3 IsNeCol$^{1.1}(v)$ *and*
4 $\forall b \in v$
5 $(_1$IsNeNom$^{1.2}(b)$ *and*
6 NAMS$^{1.18}(b) \subseteq$ ['personal name of owner'; 'owners organization'] *and*
7 $\forall a \hat{\in} \hat{}b \bullet$
8 $(_2(_3$N a = 'personal name of owner' *impl* IsPersonalName$^{4.47}$(C a) $_3$) *and*
9 $(_4$N a = 'owners organization' *impl* IsProfCharSeq$^{4.4}$(C a) $_4)_2)_1)_0$)

Semiformal Description 4.36

Predicate "is an authors value" (clause 7.4.3.4)

The value of the attribute 'authors' is a set of nominations. For each nomination, the (nomination) name is 'personal name of author' (the corresponding component is a personal name) and/or 'authors organization' (the corresponding component is a sequence of characters from the document profile character set). At least one of these parameters must be specified.

Definition 4.36

1 $\forall v$
2 $(_0$IsAuthorsValue(v) *iff*
3 IsNeCol$^{1.1}(v)$ *and*
4 $\forall b \in v$
5 $(_1$IsNeNom$^{1.2}(b)$ *and*
6 NAMS$^{1.18}(b) \subseteq$ ['personal name of author'; 'authors organization'] *and*
7 $\forall a \hat{\in} \hat{}b \bullet$
8 $(_2(_3$N a = 'personal name of author' *impl* IsPersonalName$^{4.47}$(C a) $_3$) *and*
9 $(_4$N a = 'authors organization' *impl* IsProfCharSeq$^{4.4}$(C a) $_4)_2)_1)_0$)

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 4.37

Predicate "is a copyright value" (clause 7.4.4.1)

The value of the attribute 'copyright' is a set of nominations. For each nomination, the name is 'copyright information' (the corresponding component is a set of sequences of characters from the document profile character set) and/or 'copyright dates' (the corresponding component is a date value).

Definition 4.37

```
1   ∀ v
2   (0 IsCopyrightValue(v) iff
3     IsNeCol1.1(v) and
4     ∀ b ∈ v
5     (1 IsNeNom1.2(b) and
6       NAMS1.18(b) ⊆ ['copyright information'; 'copyright dates'] and
7       ∀ a ∈ ^b •
8       (2 (3 N a = 'copyright information' impl IsSetOfProfCharSeq4.3(C a) 3) and
9         (4 N a = 'copyright dates' impl IsSetOfDatesValue4.38(C a) 4)2)1)0)
```

Semiformal Description 4.38

Predicate "is a set of dates value" (clause 7.4.4.1)

A set of dates value is a set where each element of the set is a date value.

Definition 4.38

```
1   ∀ v
2   (0 IsSetOfDatesValue(v) iff
3     IsNeCol1.1(v) and
4     ∀ b ∈ v (IsDateValue4.40(b)) 0)
```

Semiformal Description 4.39

Predicate "is a date and time sequence" (clause 7.4.2)

A date and time sequence is a catenation where each element is a date and time value.

Definition 4.39

```
1   ∀ v
2   (0 IsDateAndTimeSeq(v) iff
3     IsNeCat1.3(v) and
4     ∀ b ∈̇ ^v • (IsDateAndTimeValue4.31(C b)) 0)
```

Semiformal Description 4.40

Predicate "is a date value" (clause 7.4.2)

A date value is a date character string in accordance with ISO 8601. It is considered an atomic construct in the formal specification.

Definition 4.40

```
1   ∀ v
2   (0 IsDateValue(v) iff
3     IsAtom(v)0)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 4.41

Predicate "is a distribution list value" (clause 7.4.4.4)

The value of the attribute 'distribution list' is a set of nominations. For each nomination, the (nomination) name is 'personal name of recipient' (the corresponding component is a personal name) and/or 'recipients organization' (the corresponding component is a sequence of characters from the document profile character set). Both parameters are optional.

Definition 4.41

```
1  ∀ v
2  (0 IsDistributionListValue(v) iff
3    IsNeCol^1.1(v) and
4    ∀ b ∈ v
5    (1 IsNeNom^1.2(b) and
6      NAMS^1.18(b) ⊆ ['personal name of recipient'; 'recipients organization'] and
7      ∀ a ∈̂ ^b •
8      (2 (3 N a = 'personal name of recipient' impl IsPersonalName^4.47(C a) 3) and
9        (4 N a = 'recipients organization' impl IsProfCharSeq^4.4(C a) 4)2)1)0)
```

Semiformal Description 4.42

Predicate "is an additional information value" (clause 7.4.4.5)

The value of the attribute 'additional information' can have any value. It is considered an atomic construct in the formal specification.

Definition 4.42

```
1  ∀ v
2  (0 IsAdditionalInformationValue(v) iff
3    IsAtom(v)0)
```

Semiformal Description 4.43

Predicate "is a set of external references" (clause 7.4.5.1)

A set of external references is a set where each element of the set is an external reference.

Definition 4.43

```
1  ∀ v
2  (0 IsSetOfExternalReferences(v) iff
3    IsNeCol^1.1(v) and
4    ∀ b ∈ v (IsExternalReferenceValue^4.6(b)) 0)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 4.44

Predicate "is a local file reference value" (clause 7.4.6)

The value of the attribute 'local file references' is a set of nominations. For each nomination, the (nomination) name is 'file name', 'location of the document' and/or 'user comments'. Each component is a sequence of characters from the document profile character set.

Definition 4.44

```
1   ∀ v
2   (0 IsLocalFileReferenceValue(v) iff
3     IsNeCol^1.1(v) and
4     ∀ b ∈ v
5     (1 IsNeNom^1.2(b) and
6       NAMS^1.18(b) ⊆ ['file name'; 'location of the document'; 'user comments'] and
7       ∀ a ∈̂ b • (IsProfCharSeq^4.4(C a)) 1)0)
```

Semiformal Description 4.45

Predicate "is an authorization value" (clause 7.4.8.1)

An authorization value is either a personal name or a sequence of characters from the document profile character set.

Definition 4.45

```
1   ∀ v
2   (0 IsAuthorizationValue(v) iff
3     (1 IsPersonalName^4.47(v) or IsProfCharSeq^4.4(v) 1)0)
```

Semiformal Description 4.46

Predicate "is a set of personal names"

A set of personal names consists of elements which are personal names.

Definition 4.46

```
1   ∀ v
2   (0 IsSetOfPersonalNames(v) iff
3     IsNeCol^1.1(v) and
4     ∀ a ∈ v (IsPersonalName^4.47(a)) 0)
```

Semiformal Description 4.47

Predicate "is a personal name" (Annex A of ISO/IEC 8613-4)

A personal name is a nomination where the (nomination) name is 'surname' (must be present), 'givenname', 'initials', 'title' (optional). The components are sequences of characters from the document profile character set.

Definition 4.47

```
1   ∀ v
2   (0 IsPersonalName(v) iff
3     IsNeNom^1.2(v) and 'surname' ∈ NAMS^1.18(v) and
4     NAMS^1.18(v) ⊆ ['surname'; 'givenname'; 'initials'; 'title'] and
5     ∀ a ∈ ̂v • (IsProfCharSeq^4.4(C a)) 0)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 4.48

Predicate "is an alternative features sets value" (clause 7.3.7)

An alternative features sets value is a non-empty sets whose elements are ASN.1 object identifier sets.

Definition 4.48

```
1  ∀ v
2  (0 IsAlternativeFeaturesSetsValue(v) iff
3    IsNeCol^1.1(v) and
4    ∀ a ∈ v IsASN1ObjectIdentifierSet^4.49(a)0)
```

Semiformal Description 4.49

Predicate "is an ASN.1 object identifier set" (clause 7.3.7)

An ASN.1 object identifier set is a non-empty sets whose elements are ASN.1 object identifier.

Definition 4.49

```
1  ∀ v
2  (0 IsASN1ObjectIdentifierSet(v) iff
3    IsNeCol^1.1(v) and
4    ∀ a ∈ v IsASN1ObjectIdentifier^4.7(a)0)
```

Semiformal Description 4.50

Predicate "is an enciphered document profiles value" (clause 7.5.6)

An enciphered document profiles value is a non-empty nomination with the names 'privileged recipient information' and 'protected document part identifier'. For the name 'privileged recipient information' the value is a privileged recipient information value. For the name 'protected document part identifier' the value is an enciphered document profile identifier.

Definition 4.50

```
1  ∀ v
2  (0 IsEncipheredDocumentProfilesValue(v) iff
3    IsNeNom^1.2(v) and
4    NAMS^1.18(v) = ['privileged recipient information'; 'protected document part identifier'] and
5    ∀ a ∈̂ ^v •
6    (1 (2 N a = 'privileged recipient information' impl IsPrivilegedRecipientInformationValue^4.51(C a) 2)
7      (3 N a = 'protected document part identifier' impl IsEncipheredDocumentProfileId^4.56(C a) 3)1)0)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Semiformal Description 4.51

Predicate "is a privileged recipient information value" (clause 7.5.5)

A privileged recipient information value is a non-empty nomination with the names 'key information', 'method information' and 'privileged recipients'. For the name 'key information' the value is a key information value. For the name 'method information' the value is a method information value. For the name 'privileged recipients' the value is a set of personal names.

Definition 4.51

1 $\forall v$
2 $({}_0$IsPrivilegedRecipientInformationValue(v) *iff*
3 IsNeNom$^{1.2}(v)$ *and*
4 NAMS$^{1.18}(v)$ = ['key information'; 'method information'; 'privileged recipients'] *and*
5 $\forall a \hat{\in} \hat{}v \bullet$
6 $({}_1({}_2$N a = 'key information' *impl* IsKeyInformationValue$^{4.52}$(C a) $_2$)
7 $({}_3$N a = 'method information' *impl* IsMethodInformationValue$^{4.53}$(C a) $_3$)
8 $({}_4$N a = 'privileged recipients' *impl* IsSetOfPersonalNames$^{4.46}$(C a) $_4)_1)_0)$

Semiformal Description 4.52

Predicate "is a key information value" (clause 7.5.5)

A key information value is a non-empty nomination with the names 'additional information' and 'key method'. For the name 'additional information' the value is a key additional information value. For the name 'key method' the value is a key method value.

Definition 4.52

1 $\forall v$
2 $({}_0$IsKeyInformationValue(v) *iff*
3 IsNeNom$^{1.2}(v)$ *and*
4 NAMS$^{1.18}(v)$ = ['additional information'; 'key method'] *and*
5 $\forall a \hat{\in} \hat{}v \bullet$
6 $({}_1({}_2$N a = 'additional information' *impl* IsKeyAdditionalInformationValue$^{4.54}$(C a) $_2$)
7 $({}_3$N a = 'key method' *impl* IsKeyMethodValue$^{4.55}$(C a) $_3)_1)_0)$

Semiformal Description 4.53

Predicate "is a method information value" (clause 7.5.5)

A method information value is a sequence of two elements where the first one is an ASN.1 object identifier and the second one a character sequence.

Definition 4.53

1 $\forall v$
2 $({}_0$IsMethodInformationValue(v) *iff*
3 $\exists a, b$
4 $({}_1 v = [\rightarrow a \rightarrow b \rightarrow]$ *and*
5 IsASN1ObjectIdentifier$^{4.7}(a)$ *and* IsCharSeq$^{2.117}(b)$ $_1)_0)$

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 4.54

Predicate "is a key additional information value" (clause 7.5.5)

A key additional information value is a sequence of two elements where the first one is a character sequence and the second one an octet string.

Definition 4.54

1 $\forall v$
2 $(_0$IsKeyAdditionalInformationValue(v) *iff*
3 $\exists a, b$
4 $(_1 v = [\rightarrow a \rightarrow b \rightarrow]$ *and*
5 IsCharSeq$^{2.117}(a)$ *and* IsOctetString$^{1.10}(b)$ $_1)_0)$

Semiformal Description 4.55

Predicate "is a key method value" (clause 7.5.5)

A key method value is a sequence of two elements where the first one is an ASN.1 object identifier and the second one a character sequence.

Definition 4.55

1 $\forall v$
2 $(_0$IsKeyMethodValue(v) *iff*
3 $\exists a, b$
4 $(_1 v = [\rightarrow a \rightarrow b \rightarrow]$ *and*
5 IsASN1ObjectIdentifier$^{4.7}(a)$ *and* IsCharSeq$^{2.117}(b)$ $_1)_0)$

Semiformal Description 4.56

Predicate "is an enciphered document profile identifier" (clause 7.5.6)

An enciphered document profile identifier is a sequence of two non-negative integers where the first one is 7.

Definition 4.56

1 $\forall v$
2 $(_0$IsEncipheredDocumentProfileId(v) *iff*
3 IsPairOfNnInt$^{1.8}(v)$ *and* HEAD$^{1.13}(v) = 7_0)$

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 4.57

Predicate "is an ODA security label value" (clause 7.5.2)

An ODA security label value is a non-empty nomination with the names 'ODA label data' and 'ODA label text'. For the name 'ODA label data' the value is an octet string. For the name 'ODA label text' the value is a character sequence.

Definition 4.57

```
1  ∀ v
2  (0 IsODASecurityLabelValue(v) iff
3    IsNeNom1.2(v) and
4    NAMS1.18(v) = ['ODA label data'; 'ODA label text'] and
5    ∀ a ∈̂ v •
6    (1 (2 N a = 'ODA label data' impl IsOctetString1.10(C a) 2)
7      (3 N a = 'ODA label text' impl IsCharSeq2.117(C a) 3)1)0)
```

Semiformal Description 4.58

Predicate "is a pre-enciphered document body parts value" (clause 7.5.7)

A pre-enciphered document body parts value is a non-empty nomination with the names 'privileged recipient information' and 'protected document part identifier'. For the name 'privileged recipient information' the value is a privileged recipient information value. For the name 'protected document part identifier' the value is a pre-enciphered document body part identifier.

Definition 4.58

```
1  ∀ v
2  (0 IsPreEncipheredDocumentBodyPartsValue(v) iff
3    IsNeNom1.2(v) and
4    NAMS1.18(v) = ['privileged recipient information'; 'protected document part identifier'] and
5    ∀ a ∈̂ v •
6    (1 (2 N a = 'privileged recipient information' impl IsPrivilegedRecipientInformationValue4.51(C a) 2)
7      (3 N a = 'protected document part identifier' impl IsPreEncipheredDocumentBodyPartId4.59(C a) 3)1)0)
```

Semiformal Description 4.59

Predicate "is a pre-enciphered document body part identifier" (clause 7.5.7)

A pre-enciphered document body part identifier is a sequence of two non-negative integers where the first one is 8.

Definition 4.59

```
1  ∀ v
2  (0 IsPreEncipheredDocumentBodyPartId(v) iff
3    IsPairOfNnInt1.8(v) and HEAD1.13(v) = 80)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Semiformal Description 4.60

Predicate "is a post-enciphered document body parts value" (clause 7.5.8)

A post-enciphered document body parts value is a non-empty nomination with the names 'privileged recipient information' and 'protected document part identifier'. For the name 'privileged recipient information' the value is a privileged recipient information value. For the name 'protected document part identifier' the value is a post-enciphered document body part identifier.

Definition 4.60

```
1   ∀ v
2   (0 IsPostEncipheredDocumentBodyPartsValue(v) iff
3     IsNeNom1.2(v) and
4     NAMS1.18(v) = ['privileged recipient information'; 'protected document part identifier'] and
5     ∀ a ∈ ^v •
6     (1 (2 N a = 'privileged recipient information' impl IsPrivilegedRecipientInformationValue4.51(C a) 2)
7       (3 N a = 'protected document part identifier' impl IsPostEncipheredDocumentBodyPartId4.61(C a) 3)1)0)
```

Semiformal Description 4.61

Predicate "is a post-enciphered document body part identifier" (clause 7.5.8)

A post-enciphered document body part identifier is a sequence of two non-negative integers where the first one is 9.

Definition 4.61

```
1   ∀ v
2   (0 IsPostEncipheredDocumentBodyPartId(v) iff
3     IsPairOfNnInt1.8(v) and HEAD1.13(v) = 90)
```

Semiformal Description 4.62

Predicate "is a sealed document body parts value" (clauses 7.5.4 and 7.5.5)

A sealed document body parts value is a nomination with the names 'document body part seal', 'privileged recipients', 'sealed constituents' and 'seal identifier'. For the name 'document body part seal' the value is a document seal value. For the name 'privileged recipients' the value is a set of personal names. For the name 'sealed constituents' the value is a sequence of constituent identifiers. For the name 'seal identifier' the value is an integer.

Definition 4.62

```
1   ∀ v
2   (0 IsSealedDocumentBodyPartsValue(v) iff
3     IsNeNom1.2(v) and
4     NAMS1.18(v) = ['document body part seal'; 'privileged recipients';
5                    'sealed constituents'; 'seal identifier'] and
6     ∀ a ∈ ^v •
7     (1 (2 N a = 'document body part seal' impl IsDocumentSealValue4.63(C a) 2)
8       (3 N a = 'privileged recipients' impl IsSetOfPersonalNames4.46(C a) 3)
9       (4 N a = 'sealed constituents' impl IsSeqOfConstituentIds4.64(C a) 4)
10      (5 N a = 'seal identifier' impl IsInt(C a) 5)1)0)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 4.63

Predicate "is a document seal value" (clauses 7.5.3, 7.5.4 and 7.5.5)

A document seal value is a nomination with the names 'seal', 'seal method' and 'sealed information'. For the name 'seal' the value is an octet string. For the name 'seal method' the value is a seal method value. For the name 'sealed information' the value is a sealed information value.

Definition 4.63

```
1  ∀ v
2  (0 IsDocumentSealValue(v) iff
3    IsNeNom1.2(v) and
4    NAMS1.18(v) = ['seal'; 'seal method'; 'sealed information'] and
5    ∀ a ∈ ^v •
6    (1 (2 N a = 'seal' impl IsOctetString1.10(C a) 2)
7      (3 N a = 'seal method' impl IsSealMethodValue4.66(C a) 3)
8      (4 N a = 'sealed information' impl IsSealedInformationValue4.67(C a) 4)1)0)
```

Semiformal Description 4.64

Predicate "is a sequence of constituent identifiers" (clauses 7.5.4 and 7.5.5)

A sequence of constituent identifiers is a non-empty catenation whose components are constituent identifiers.

Definition 4.64

```
1  ∀ v
2  (0 IsSeqOfConstituentIds(v) iff
3    IsNeCat1.3(v) and
4    ∀ a ∈ ^v • IsConstituentId4.65(C a) 0)
```

Semiformal Description 4.65

Predicate "is a constituent identifier" (clauses 7.5.4 and 7.5.5)

A constituent identifier is an object identifier, object class identifier, layout style identifier or presentation style identifier.

Definition 4.65

```
1  ∀ v
2  (0 IsConstituentId(v) iff
3    IsObjectId2.79(v) or IsObjectClassId2.82(v) or IsLayoutStyleId2.100(v) or IsPresentationStyleId2.110(v) 0)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Semiformal Description 4.66

Predicate "is a seal method value" (clauses 7.5.4 and 7.5.5)

A seal method value is a nomination with the names 'fingerprint information', 'fingerprint method', 'sealing information' and 'sealing method'. For the names 'fingerprint information' and 'sealing information' the value is a key information value. For the names 'fingerprint method' and 'sealing method' the value is a method information value.

Definition 4.66

```
1   ∀v
2   (0 IsSealMethodValue(v) iff
3     IsNeNom1.2(v) and
4     NAMS1.18(v) = ['fingerprint information'; 'fingerprint method';
5                    'sealing information'; 'sealing method'] and
6     ∀a ∈̂ v •
7     (1 (2 N a = 'fingerprint information' impl IsKeyInformationValue4.52(C a) 2)
8       (3 N a = 'fingerprint method' impl IsMethodInformationValue4.53(C a) 3)
9       (4 N a = 'sealing information' impl IsKeyInformationValue4.52(C a) 4)
10      (5 N a = 'sealing method' impl IsMethodInformationValue4.53(C a) 5)1)0)
```

Semiformal Description 4.67

Predicate "is a sealed information value" (clauses 7.5.4 and 7.5.5)

A sealed information value is a nomination with the names 'fingerprint', 'location', 'seal originator' and 'time'. For the name 'fingerprint' the value is an octet string. For the name 'location' the value is a location value. For the name 'seal originator' the value is a personal name. For the name 'time' the value is a date and time value.

Definition 4.67

```
1   ∀v
2   (0 IsSealedInformationValue(v) iff
3     IsNeNom1.2(v) and
4     NAMS1.18(v) = ['fingerprint'; 'location'; 'seal originator'; 'time'] and
5     ∀a ∈̂ v •
6     (1 (2 N a = 'fingerprint' impl IsOctetString1.10(C a) 2)
7       (3 N a = 'location' impl IsLocationValue4.68(C a) 3)
8       (4 N a = 'seal originator' impl IsPersonalName4.47(C a) 4)
9       (5 N a = 'time' impl IsDateAndTimeValue4.31(C a) 5)1)0)
```

Semiformal Description 4.68

Predicate "is a location value" (clauses 7.5.4 and 7.5.5)

A location value is a sequence of two elements where the first one is an ASN.1 object identifier and the second one a character sequence.

Definition 4.68

```
1   ∀v
2   (0 IsLocationValue(v) iff
3     ∃a, b
4     (1 v = [→ a → b →] and
5       IsASN1ObjectIdentifier4.7(a) and IsCharSeq2.117(b) 1)0)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 4.69

Predicate "is a sealed document profiles value" (clauses 7.5.3)

A sealed document profiles value is a nomination with the names 'document profile seal', 'privileged recipients' and 'sealed document profile identifier'. For the name 'document profile seal' the value is a document seal value. For the name 'privileged recipients' the value is a set of personal names. For the name 'sealed document profile identifier' the value is a sealed document profile identifier.

Definition 4.69

1 $\forall v$
2 $(_0$IsSealedDocumentProfilesValue(v) *iff*
3 IsNeNom$^{1.2}(v)$ *and*
4 NAMS$^{1.18}(v)$ = ['document profile seal'; 'privileged recipients'; 'sealed document profile identifier'] *and*
5 $\forall a \hat{\in} \hat{}v \bullet$
6 $(_1(_2$N a = 'document profile seal' *impl* IsDocumentSealValue$^{4.63}$(C a) $_2)$
7 $(_3$N a = 'privileged recipients' *impl* IsSetOfPersonalNames$^{4.46}$(C a) $_3)$
8 $(_4$N a = 'sealed document profile identifier' *impl* IsSealedDocumentProfileId$^{4.70}$(C a) $_4)_1)_0)$

Semiformal Description 4.70

Predicate "is an sealed document profile identifier" (clause 7.5.3)

An sealed document profile identifier is a sequence of two non-negative integers where the first one is 6.

Definition 4.70

1 $\forall v$
2 $(_0$IsSealedDocumentProfileId(v) *iff*
3 IsPairOfNnInt$^{1.8}(v)$ *and* HEAD$^{1.13}(v) = 6_0)$

Semiformal Description 4.71

Predicate "is a colour of layout objects value" (clause 7.3.8.4.9)

A colour of layout objects value is a non-empty set whose elements are colour expressions.

Definition 4.71

1 $\forall v$
2 $(_0$IsColourOfLayoutObjectsValue(v) *iff*
3 IsNeCol$^{1.1}(v)$ *and*
4 $\forall a \in v\ (_1$IsColourExpression$^{2.126}(a)_1)_0)$

Semiformal Description 4.72

Predicate "is a content background colours value" (clause 7.3.8.4.11)

A content background colours value is a non-empty set whose elements are colour expressions.

Definition 4.72

1 $\forall v$
2 $(_0$IsContentBackgroundColoursValue(v) *iff*
3 IsNeCol$^{1.1}(v)$ *and*
4 $\forall a \in v\ (_1$IsColourExpression$^{2.126}(a)_1)_0)$

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 4.73

Predicate "is a content foreground colours value" (clause 7.3.8.4.12)

A content foreground colours value is a non-empty set whose elements are colour expressions.

Definition 4.73

1 $\forall v$
2 $(_0$IsContentForegroundColoursValue(v) *iff*
3 IsNeCol$^{1.1}(v)$ *and*
4 $\forall a \in v\ (_1$IsColourExpression$^{2.126}(a)_1)_0)$

Semiformal Description 4.74

Predicate "is a colour tables value" (clauses 7.3.8.4.10 and 7.3.8.4.13)

A colour tables value is a non-empty set whose elements are colour tables.

Definition 4.74

1 $\forall v$
2 $(_0$IsColourTablesValue(v) *iff*
3 IsNeCol$^{1.1}(v)$ *and*
4 $\forall a \in v\ (_1$IsColourTable$^{2.131}(a)_1)_0)$

Semiformal Description 4.75

Predicate "is a colour characteristics value" (clause 7.3.10.3)

A colour characteristics value is a nomination with the names 'colour modes present', and, optionally, 'colour spaces present', 'maximum CMY(K) grid size', 'maximum number of colour table entries', 'maximum number of look-up table entries' and 'minimum colour tolerance'. For the name 'colour modes present' the value is a colour modes present value. For the name 'colour spaces present' the value is a colour spaces present value. For the names 'maximum CMY(K) grid size', 'maximum number of colour table entries' and 'maximum number of look-up table entries' the value is positive integer. For the name 'minimum colour tolerance' the value is a colour tolerance value.

Definition 4.75

1 $\forall v$
2 $(_0$IsColourCharacteristicsValue(v) *iff*
3 IsNeNom$^{1.2}(v)$ *and*
4 NAMS$^{1.18}(v) \subseteq$ ['colour modes present'; 'colour spaces present';
5 'maximum CMY(K) grid size'; 'maximum number of colour table entries';
6 'maximum number of look-up table entries'; 'minimum colour tolerance'] *and*
7 NAMS$^{1.18}(v) \supseteq$ ['colour modes present'] *and*
8 $\forall a \in \hat{}v \bullet$
9 $(_1(_2$N a = 'colour modes present' *impl* IsColourModesPresentValue$^{4.76}$(C a) $_2)$ *and*
10 $(_3$N a = 'colour spaces present' *impl* IsColourSpacesPresentValue$^{4.77}$(C a) $_3)$ *and*
11 $(_4$N a = 'maximum CMY(K) grid size' *impl* IsNat(C a) $_4)$ *and*
12 $(_5$N a = 'maximum number of colour table entries' *impl* IsNat(C a) $_5)$ *and*
13 $(_6$N a = 'maximum number of look-up table entries' *impl* IsNat(C a) $_6)$ *and*
14 $(_7$N a = 'minimum colour tolerance' *impl* IsColourToleranceValue$^{2.130}$(C a) $_7)_1)_0)$

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 4.76

Predicate "is a colour modes present value" (clause 7.3.10.3)

A colour modes present value is a non-empty set of elements which are either 'indexed', 'direct' or 'both'.

Definition 4.76

```
1  ∀ v
2  (0 IsColourModesPresentValue(v) iff
3    IsNeCol1.1(v) and
4    ∀ a ∈ v (1 a ∈ ['direct'; 'indexed'; 'both']1) 0)
```

Semiformal Description 4.77

Predicate "is a colour spaces present value" (clause 7.3.10.3)

A colour spaces present value is nomination with the names 'colour space type' and, optionally, 'colour calibration type'. For the name 'colour space type' the component is a colour space type value. For the name 'colour calibration type' the component is a colour calibration type value.

Definition 4.77

```
1  ∀ v
2  (0 IsColourSpacesPresentValue(v) iff
3    IsNeCol1.1(v) and
4    ∀ a ∈ v
5    (1 IsNeNom1.2(a) and
6      NAMS1.18(a) ⊆ ['colour calibration type'; 'colour space type'] and
7      NAMS1.18(a) ⊇ ['colour space type'] and
8    ∀ b ∈̂ v •
9    (2(3 N b = 'colour calibration type' impl IsColourCalibrationTypeValue4.78(C b) 3) and
10     (4 N b = 'colour space type' impl IsColourSpaceTypeValue4.79(C b) 4)2)1)0)
```

Semiformal Description 4.78

Predicate "is a colour calibration type value" (clause 7.3.10.3)

A colour calibration type value is 'no calibration', 'matrices', 'look-up table' or 'matrices and look-up table'.

Definition 4.78

```
1  ∀ v
2  (0 IsColourCalibrationTypeValue(v) iff
3    v ∈ ['no calibration'; 'matrices'; 'look-up table'; 'matrices and look-up table']0)
```

Semiformal Description 4.79

Predicate "is a colour space type value" (clause 7.3.10.3)

A colour space type value is 'RGB', 'CMYK', 'CMY', 'CIELUV' or 'CIELAB'.

Definition 4.79

```
1  ∀ v
2  (0 IsColourSpaceTypeValue(v) iff
3    v ∈ ['RGB'; 'CMYK'; 'CMY'; 'CIELUV'; 'CIELAB']0)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 4.80

Predicate "is a colour spaces list value" (clause 7.3.10.4)

A colour spaces list value is a non-empty set whose elements are colour space definitions.

Definition 4.80

1 $\forall v$
2 $(_0$IsColourSpacesListValue(v) *iff*
3 IsNeCol$^{1.1}(v)$ *and*
4 $\forall a \in v$ $(_1$IsColourSpaceDefinition$^{4.81}$(C a) $_1)_0)$

Semiformal Description 4.81

Predicate "is a colour space definition" (clause 7.3.10.4)

A colour space definition is a nomination with the names 'colour space id' and 'colour space type' and, optionally, 'calibration data', 'colour data scaling' and 'colour space name'. For the name 'calibration data' the component is a calibration data value. For the name 'colour data scaling' the component is a colour data scaling value. For the name 'colour space id' the component is a non-negative integer. For the name 'colour space name' the component is a string of characters from the document profile character set. For the name 'colour space type' the component is a colour space type value.

Definition 4.81

1 $\forall v$
2 $(_0$IsColourSpaceDefinition(v) *iff*
3 IsNeNom$^{1.2}(v)$ *and*
4 NAMS$^{1.18}(v) \subseteq$ ['calibration data'; 'colour data scaling'; 'colour space id';
5 'colour space name'; 'colour space type'] *and*
6 NAMS$^{1.18}(v) \supseteq$ ['colour space id'; 'colour space type'] *and*
7 $\forall a \hat{\in} \hat{}v \bullet$
8 $(_1(_2$N a = 'calibration data' *impl* IsCalibrationDataValue$^{4.82}$(C a) $_2)$ *and*
9 $(_3$N a = 'colour data scaling' *impl* IsColourDataScalingValue$^{4.93}$(C a) $_3)$ *and*
10 $(_4$N a = 'colour space id' *impl* IsNnInt$^{1.7}$(C a) $_4)$ *and*
11 $(_5$N a = 'colour space name' *impl* IsProfCharSeq$^{4.4}$(C a) $_5)$ *and*
12 $(_6$N a = 'colour space type' *impl* IsColourSpaceTypeValue$^{4.79}$(C a) $_6)_1)_0)$

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Semiformal Description 4.82

Predicate "is a calibration data value" (clause 7.3.10.4)

A calibration data value is a nomination with exactly one construct. For the names 'CIELUV' or 'CIELAB' the component is CIE calibration data. For the name 'RGB' or 'CIELAB' the component is RGB calibration data. For the names 'CMY' or 'CMYK' the component is CMYK calibration data.

Definition 4.82

```
1  ∀ v
2  (0 IsCalibrationDataValue(v) iff
3    IsNom(v) and CARD (^v•) = 1 and
4    NAMS1.18(v) ⊆ ['CIELAB';'CIELUV';'CMY';'CMYK';'RGB'] and
5    ∀ a ∈̂ ^v•
6    (1 (2 N a ∈ ['CIELAB';'CIELUV'] impl IsCIECalibrationData4.83(C a) 2) and
7      (3 N a ∈ ['CMY';'CMYK'] impl IsCMYKCalibrationData4.85(C a) 3) and
8      (4 N a = 'RGB' impl IsRGBCalibrationData4.84(C a) 4)1)0)
```

Semiformal Description 4.83

Predicate "is CIE calibration data" (clause 7.3.10.4)

CIE calibration data is a nomination with the name 'reference white' whose component is a reference white value.

Definition 4.83

```
1  ∀ v
2  (0 IsCIECalibrationData(v) iff
3    IsNeNom1.2(v) and
4    NAMS1.18(v) = ['reference white'] and
5    ∀ a ∈̂ ^v•
6    (1 N a = 'reference white' impl IsReferenceWhiteValue4.86(C a) 1)0)
```

Semiformal Description 4.84

Predicate "is RGB calibration data" (clause 7.3.10.4)

RGB calibration data is a nomination with the names 'reference white' and, optionally, 'colour look-up table', 'matrix1', 'matrix2' and 'reference white'. For the name 'colour look-up table' the component is a colour look-up table. For the names 'matrix1' and 'matrix2' the component is a 3 by 3 matrix. For the name 'reference white' the component is a reference white value.

Definition 4.84

```
1  ∀ v
2  (0 IsRGBCalibrationData(v) iff
3    IsNeNom1.2(v) and
4    NAMS1.18(v) ⊆ ['colour look-up table';'matrix1';'matrix2';'reference white'] and
5    NAMS1.18(v) ⊇ ['reference white'] and
6    ∀ a ∈̂ ^v•
7    (1 (2 N a = 'colour look-up table' impl IsColourLookupTable4.87(C a) 2) and
8      (3 N a ∈ ['matrix1';'matrix2'] impl Is3x3Matrix4.92.(C a) 3) and
9      (4 N a = 'reference white' impl IsReferenceWhiteValue4.86(C a) 4)1)0)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 4.85

Predicate "is CMYK calibration data" (clause 7.3.10.4)

CMYK calibration data is a nomination with the names 'reference white' and, optionally, 'comments' and 'grid specification'. For the name 'comments' the component is a string of characters from the document profile character set. For the name 'grid specification' the component is a grid specification value. For the name 'reference white' the component is a reference white value.

Definition 4.85

```
1  ∀v
2  (0 IsCMYKCalibrationData(v) iff
3    IsNeNom1.2(v) and
4    NAMS1.18(v) ⊆ ['comments'; 'grid specification'; 'reference white'] and
5    NAMS1.18(v) ⊇ ['reference white'] and
6    ∀a ∈̂ ^v •
7    (1 (2 N a = 'comments' impl IsProfCharSeq4.4(C a) 2) and
8      (3 N a = 'grid specification' impl IsGridSpecificationValue4.90(C a) 3) and
9      (4 N a = 'reference white' impl IsReferenceWhiteValue4.86(C a) 4)1)0)
```

Semiformal Description 4.86

Predicate "is a reference white value" (clause 7.3.10.4)

A reference white value is a nomination with the names 'Xn value', 'Yn value' and 'Zn value' whose components are integers or real numbers.

Definition 4.86

```
1  ∀v
2  (0 IsReferenceWhiteValue(v) iff
3    IsNeNom1.2(v) and
4    NAMS1.18(v) = ['Xn value'; 'Yn value'; 'Zn value'] and
5    ∀a ∈̂ ^v •
6    (1 IsInt(C a) or IsReal(C a) 1)0)
```

Semiformal Description 4.87

Predicate "is a colour look-up table" (clause 7.3.10.4)

A colour look-up table is a nomination with the names 'colour table', 'm', 'n' and 'number of entries'. For the names 'm', 'n' and 'number of entries' the component is an integer. For the name 'colour table' the component is a colour table list.

Definition 4.87

```
1  ∀v
2  (0 IsColourLookupTable(v) iff
3    IsNeNom1.2(v) and
4    NAMS1.18(v) = ['colour table'; 'm'; 'n'; 'number of entries'] and
5    ∀a ∈̂ ^v •
6    (1 (2 N a ∈ ['m'; 'n'; 'number of entries'] impl IsInt(C a)2) and
7      (3 N a = 'colour table' impl IsColourTableList4.88(C a)3)1)0)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 4.88

Predicate "is a colour table list" (clause 7.3.10.4)

A colour table list is a set whose elements are colour table corrections.

Definition 4.88

1 $\forall v$
2 $({}_0\text{IsColourTableList}(v)$ *iff*
3 $\text{IsNeCol}^{1.1}(v)$ *and*
4 $\forall a \in v\ ({}_1\text{IsColourTableCorrection}^{4.89}(a)\ {}_1)_0)$

Semiformal Description 4.89

Predicate "is a colour table correction" (clause 7.3.10.4)

A colour table correction is a nomination with the names 'index', 'R value', 'G value' and 'B value'. For the name 'index' the component is a non-negative integer. For the names 'R value', 'G value' and 'B value' the component is an integer or a real number.

Definition 4.89

1 $\forall v$
2 $({}_0\text{IsColourTableCorrection}(v)$ *iff*
3 $\text{IsNeNom}^{1.2}(v)$ *and*
4 $\text{NAMS}^{1.18}(v) = [\text{'index'}; \text{'R value'}; \text{'G value'}; \text{'B value'}]$ *and*
5 $\forall a \hat{\in} \hat{}v \bullet$
6 $({}_1({}_2\text{N}\ a = \text{'index'}$ *impl* $\text{IsNnInt}^{1.7}(\text{C}\ a)\ {}_2)$ *and*
7 $({}_3\text{N}\ a \in [\text{'R value'}; \text{'G value'}; \text{'B value'}]$ *impl* $({}_4\text{IsInt}(\text{C}\ a)$ *or* $\text{IsReal}(\text{C}\ a)\ {}_4)_3)_1)_0)$

Semiformal Description 4.90

Predicate "is a grid specification value" (clause 7.3.10.4)

A grid specification value is a set whose entries are grid specifications.

Definition 4.90

1 $\forall v$
2 $({}_0\text{IsGridSpecificationValue}(v)$ *iff*
3 $\text{IsNeCol}^{1.1}(v)$ *and*
4 $\forall a \in v\ ({}_1\text{IsGridSpecification}^{4.91}(a)\ {}_1)_0)$

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Semiformal Description 4.91

Predicate "is a grid specification" (clause 7.3.10.4)

A grid specification is a nomination with the two names 'grid location' and 'grid value'. For the name 'grid location' the component is a CMYK colour. For the name 'grid value' the component is a CIE colour.

Definition 4.91

```
1  ∀v
2  (0 IsGridSpecification(v) iff
3    IsNeNom1.2(v) and
4    NAMS1.18(v) = ['grid location'; 'grid value'] and
5    ∀a ∈̂ ^v •
6    (1 (2 N a = 'grid location' impl IsCMYKColour4.95(C a) 2) and
7      (3 N a = 'grid value' impl IsCIEColour4.96(C a) 3)1)0)
```

Semiformal Description 4.92

Predicate "is a 3 by 3 matrix" (clause 7.3.10.4)

A 3 by 3 matrix is a sequence of three elements which in turn are sequences of three integers or real numbers.

Definition 4.92

```
1  ∀v
2  (0 Is3x3Matrix(v) iff
3    IsCat(v) and CARD (^v •) = 3 and
4    ∀a ∈̂ ^v •
5    (1 IsCat(a) and CARD (^a •) = 3 and
6      ∀b ∈̂ ^a • (2 IsInt(b) or IsReal(b)2)1)0)
```

Semiformal Description 4.93

Predicate "is a colour data scaling value" (clause 7.3.10.4)

A colour data scaling value is a nomination with the names 'first component', 'second component', 'third component' and, optionally, 'fourth component'. The components are scale and offset values.

Definition 4.93

```
1  ∀v
2  (0 IsColourDataScalingValue(v) iff
3    IsNeNom1.2(v) and
4    NAMS1.18(v) ⊆ ['first component'; 'second component'; 'third component'; 'fourth component'] and
5    NAMS1.18(v) ⊇ ['first component'; 'second component'; 'third component'] and
6    ∀a ∈̂ ^v • (1 IsScaleAndOffsetValue4.94(C a)1)0)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 4.94

Predicate "is a scale and offset value" (clause 7.3.10.4)

A scale and offset value is a nomination with the names 'colour offset' and 'colour scale' whose components are integers or real numbers.

Definition 4.94

1 $\forall v$
2 $(_0$IsScaleAndOffsetValue(v) *iff*
3 IsNeNom$^{1.2}(v)$ *and*
4 NAMS$^{1.18}(v)$ = ['colour offset'; 'colour scale'] *and*
5 $\forall a \in \hat{}v \bullet$ $(_1$IsInt(C a) *or* IsReal(C $a)_1)_0$)

Semiformal Description 4.95

Predicate "is a CMYK colour" (clause 7.3.10.4)

A CMYK colour is a nomination with the names 'C value', 'M value', 'Y value' and, optionally, 'K value' whose components are integers or real numbers.

Definition 4.95

1 $\forall v$
2 $(_0$IsCMYKColour(v) *iff*
3 IsNeNom$^{1.2}(v)$ *and*
4 NAMS$^{1.18}(v) \subseteq$ ['C value'; 'M value'; 'Y value'; 'K value'] *and*
5 NAMS$^{1.18}(v) \supseteq$ ['C value'; 'M value'; 'Y value'] *and*
6 $\forall a \in \hat{}v \bullet$
7 $(_1$IsInt(C a) *or* IsReal(C $a)_1)_0$)

Semiformal Description 4.96

Predicate "is a CIE colour" (clause 7.3.10.4)

A CIE colour is a nomination with the names 'X value', 'Y value' and 'Z value' whose components are integers or real numbers.

Definition 4.96

1 $\forall v$
2 $(_0$IsCIEColour(v) *iff*
3 IsNeNom$^{1.2}(v)$ *and*
4 NAMS$^{1.18}(v)$ = ['X value'; 'Y value'; 'Z value'] *and*
5 $\forall a \in \hat{}v \bullet$
6 $(_1$IsInt(C a) *or* IsReal(C $a)_1)_0$)

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

9 Formal specification of the character content architectures

This clause gives a formal specification of the character content architectures as described in ISO/IEC 8613-6.

Clause 9.1 provides the interface to the document profile - by giving a formal specification of those attributes which may be included in the document profile attribute 'document application profile defaults'.

Clauses 9.2 and 9.3 provide the interface to the document architecture by giving a formal specification of character content portion presentation attributes; and any associated content portion attributes applicable to the character content portions.

Clause 9.4 provides a description of the elements of the character content information. In particular, the following general rules on the syntactic and semantic usage of control functions are formally specified:

1. GCC-0 shall be followed by 2 or more graphic characters (see ISO/IEC 8613-6, clause 13.1.2).

   NOTE: ISO/IEC 8613-6, clause 13.1.2 states implicitly that 2 or more graphic characters can follow a GCC-0 but only the *2 graphic characters* immediately following the GCC-0 are concatenated (if they are not split by a line break which would introduce control function characters).

2. GCC-1 shall be followed by 2 or more graphic characters followed by GCC-2 (see ISO/IEC 8613-6, clause 13.1.2).

   NOTE: ISO/IEC 8613-6, clause 13.1.2 states implicitly that at least 2, but possibly more, graphic characters can follow a GCC-1 but only the *graphic characters* immediately following the GCC-1 and which are terminated by a GCC-2 are concatenated (if they are not split by a line break which would introduce control function characters).

3. LF is only permitted: (a) at the beginning of a character string, (b) after a CR and (c) after a LF (see ISO/IEC 8613-6, clause 13.1.4).

4. The number of PLUs shall be complemented exactly by PLDs before any LF is encountered and at any position in the string the following condition is applicable; $|\#\mathsf{PLU} - \#\mathsf{PLD}| \leq 1$ (see ISO/IEC 8613-6, clauses 13.1.5 and 13.1.6).

   NOTE: For rules 4. and 6., the # means *number of*. For rule 4. the $| x - y |$ represents the mathematical modulus.

5. SRS-1 shall be followed by one or more characters except CR LF and followed by SRS-0 (see ISO/IEC 8613-6, clause 13.1.11).

6. The #PLU = #PLD between SRS-1 and SRS-0 (see ISO/IEC 8613-6, clause 13.1.11).

7. The effects of VPB and VPR shall be cancelled (see 8. for details). within any SRS-1 and SRS-0 (see ISO/IEC 8613-6, clause 13.1.11).

8. The effects of all VPB-n_is and VPR-m_js shall be cancelled (i.e., $\sum_{i=1}^{p}$ VPB-n_i = $\sum_{j=1}^{q}$ VPR-m_j where $\{i, j, p, q, n_i, m_j \in \mathbf{N}\}$) before the control functions CR LF are encountered (see ISO/IEC 8613-6, clauses 13.1.15 and 13.1.16).

9. PTX-1 shall be followed by a character string which shall be followed by PTX-3 followed by a character string followed by PTX-0 (see ISO/IEC 8613-6, clause 13.3.3).

10. SOS is followed by a character string followed by ST (NOTE: Nesting is not forbidden). (See ISO/IEC 8613-6, clause 13.4.2).

Any time a clause number is specified in the semi-formal descriptions this refers to a clause number in ISO/IEC 8613-6.

What follows is the outline of that part of the overall formula (see clause 5) which specifies the character content architectures.

... IsDefaultableCharacterContentArchitectureAttribute$^{6.1}(att)$ ...
and ... IsCharacterCodingSpecification$^{6.2}(v)$ ...
and ... IsCharacterPresentationFeature$^{6.3}(v)$ ...
and ... SatisfiesCharacterContentArchitectureConstraints$^{6.4}(prof, doby)$ ...
and ... IsProcessableCharacterDescription$^{6.5}(cont)$ ...
and ... IsProcessableCharacterContent$^{6.6}(cont)$ ...
and ... IsFormattedCharacterDescription$^{6.7}(cont)$ ...

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

and ... IsFormattedCharacterContent$^{6.8}(cont)$ ...
and ... IsFormattedProcessableCharacterDescription$^{6.9}(cont)$ ...
and ... IsFormattedProcessableCharacterContent$^{6.10}(cont)$ ...
and ... IsCharacterContentPortionDescription$^{6.11}(cont)$ ...
and ... IsCharacterContentPresentationAttribute$^{6.12}(att)$ ...
and ... IsCharacterContentCodingAttribute$^{6.13}(att)$ ...
and ... IsCharacterContentPortionAttributeSet$^{6.14}(as)$ ...
and ... IsAlignmentValue$^{6.15}(v)$ ...
and ... IsCharacterFontsValue$^{6.16}(v)$ ...
and ... IsCharacterOrientationValue$^{6.17}(v)$ ...
and ... IsFormattingIndicatorValue$^{6.18}(v)$ ...
and ... IsCharacterPathValue$^{6.19}(v)$ ...
and ... IsCharacterContentTypeOfCodingValue$^{6.20}(v)$ ...
and ... IsCharacterSpacingValue$^{6.21}(v)$ ...
and ... IsFirstLineOffsetValue$^{6.22}(v)$ ...
and ... IsCharacterSetsValue$^{6.23}(v)$ ...
and ... IsGraphicCharacterSubrepertoireValue$^{6.24}(v)$ ...
and ... IsISO6937SubrepertoireId$^{6.25}(v)$ ...
and ... IsCharacterContentArchitectureClassValue$^{6.26}(v)$ ...
and ... IsGraphicRenditionValue$^{6.27}(v)$ ...
and ... IsWeightValue$^{6.28}(v)$ ...
and ... IsPostureValue$^{6.29}(v)$ ...
and ... IsUnderliningValue$^{6.30}(v)$ ...
and ... IsBlinkingValue$^{6.31}(v)$ ...
and ... IsImageInversionValue$^{6.32}(v)$ ...
and ... IsCrossingOutValue$^{6.33}(v)$ ...
and ... IsIndentationValue$^{6.34}(v)$ ...
and ... IsCharacterInitialOffsetValue$^{6.35}(v)$ ...
and ... IsKerningOffsetValue$^{6.36}(v)$ ...
and ... IsOrphanSizeValue$^{6.37}(v)$ ...
and ... IsCodeExtensionAnnouncersValue$^{6.38}(v)$ ...
and ... IsPairwiseKerningValue$^{6.39}(v)$ ...
and ... IsWidowSizeValue$^{6.40}(v)$ ...
and ... IsProportionalLineSpacingValue$^{6.41}(v)$ ...
and ... IsLineLayoutTableValue$^{6.42}(v)$ ...
and ... IsTabReferenceString$^{6.43}(v)$ ...
and ... IsLineSpacingValue$^{6.44}(v)$ ...
and ... IsCharacterLineProgressionValue$^{6.45}(v)$ ...
and ... IsItemizationValue$^{6.46}(v)$ ...
and ... ContainsFormattedElements$^{6.47}(v)$ ...
and ... ContainsProcessableElements$^{6.48}(v)$ ...
and ... ContainsFormattedProcessableElements$^{6.49}(v)$ ...
and ... IsValidFormattedContentSyntax$^{6.50}(v)$ ...
and ... IsValidProcessableContentSyntax$^{6.51}(v)$ ...
and ... IsValidFormattedProcessableContentSyntax$^{6.52}(v)$ ...
and ... IsFormattedContentElement$^{6.53}(v)$ ...
and ... IsProcessableContentElement$^{6.54}(v)$ ...
and ... IsFormattedProcessableContentElement$^{6.55}(v)$ ...
and ... IsCommonContentElement$^{6.56}(v)$ ...
and ... IsValidFormattedContentSemantics$^{6.57}(v)$ ...
and ... IsValidProcessableContentSemantics$^{6.58}(v)$ ...
and ... IsValidFormattedProcessableContentSemantics$^{6.59}(v)$ ...
and ... IsValidCommonContentSemantics$^{6.60}(v)$ ...
and ... IsValidLogicalControlSemantics$^{6.61}(v)$ ...
and ... IsValidDelimiterControlSemantics$^{6.62}(v)$ ...
and ... IsValidSharedControlSemantics$^{6.63}(v)$ ...

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

and ... $\text{IsValidGCC0}^{6.64}(v)$ ...
and ... $\text{IsValidGCC12}^{6.65}(v)$ ...
and ... $\text{IsValidGCC12End}^{6.66}(v)$ ...
and ... $\text{IsValidLF}^{6.67}(v)$ ...
and ... $\text{IsValidNextLF0}^{6.68}(v)$ ...
and ... $\text{IsValidNextLF1}^{6.69}(v)$ ...
and ... $\text{IsValidPLUD}^{6.70}(v)$ ...
and ... $\text{IsIncompletePLU}^{6.71}(v)$ ...
and ... $\text{IsIncompletePLD}^{6.72}(v)$ ...
and ... $\text{IsValidSRS}^{6.73}(v)$ ...
and ... $\text{IsValidSRSEnd}^{6.74}(v)$ ...
and ... $\text{IsValidSRSNest}^{6.75}(v, v_1)$ ...
and ... $\text{IsValidPLUDInSRS}^{6.76}(v)$ ...
and ... $\text{IsValidPLUDInSRSEnd}^{6.77}(v)$ ...
and ... $\text{IsIncompletePLUInSRS}^{6.78}(v)$ ...
and ... $\text{IsIncompletePLDInSRS}^{6.79}(v)$ ...
and ... $\text{IsValidVP}^{6.80}(v)$ ...
and ... $\text{IsValidVPSum}^{6.81}(v, n)$ ...
and ... $\text{IsValidPTX}^{6.82}(v)$ ...
and ... $\text{IsValidPTXNext}^{6.83}(v)$ ...
and ... $\text{IsValidPTXEnd}^{6.84}(v)$ ...
and ... $\text{IsValidSOS}^{6.85}(v)$ ...
and ... $\text{IsValidSOSEnd}^{6.86}(v)$ ...
and ... $\text{IsValidSOSNest}^{6.87}(v, v_1)$ ...
and ... $\text{IsValidVPInSRS}^{6.88}(v)$ ...
and ... $\text{IsValidVPSumInSRS}^{6.89}(v, n)$ ...
and ... $\text{IsValidVPSumInSRSEnd}^{6.90}(v)$ ...
and ... $\text{IsCodeExtensionControlFn}^{6.91}(v)$ ...
and ... $\text{IsISO2022String}^{6.92}(v)$ ...
and ... $\text{IsISO2022Character}^{6.93}(v)$ ...
and ... $\text{IsISO6429String}^{6.94}(v)$ ...
and ... $\text{IsISO6429Character}^{6.95}(v)$ ...
and ... $\text{IsSharedControlFn}^{6.96}(v)$ ...
and ... $\text{IsParameterlessSharedControlFn}^{6.97}(v)$ ...
and ... $\text{IsParameterizedSharedControlFnName}^{6.98}(v)$ ...
and ... $\text{IsParameterizedSharedControlFn}^{6.99}(n, p)$ ...
and ... $\text{IsLayoutControlFn}^{6.100}(v)$ ...
and ... $\text{IsParameterlessLayoutControlFn}^{6.101}(v)$ ...
and ... $\text{IsParameterizedLayoutControlFnName}^{6.102}(v)$ ...
and ... $\text{IsParameterizedLayoutControlFn}^{6.103}(n, p)$ ...
and ... $\text{IsLogicalControlFn}^{6.104}(v)$ ...
and ... $\text{IsParameterlessLogicalControlFn}^{6.105}(v)$ ...
and ... $\text{IsParameterizedLogicalControlFnName}^{6.106}(v)$ ...
and ... $\text{IsParameterizedLogicalControlFn}^{6.107}(n, p)$ ...
and ... $\text{IsDelimiterControlFn}^{6.108}(v)$ ...
and ... $\text{IsSpaceControlFn}^{6.109}(v)$ ...
and ... $\text{IsEscapeControlFn}^{6.110}(v)$ ...
and ... $\text{IsCharacterContentInformationValue}^{6.111}(v)$ ...
and ... $\text{IsGraphicCharacterString}^{6.112}(v)$ ...
and ... $\text{IsGraphicCharacter}^{6.113}(v)$ ...

NOTE — Other predicates or operators which are used here, but are defined in other clauses, are not listed here.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9.1 Interface to the Document Profile

Semiformal Description 6.1

Predicate "is a profile defaultable character content architecture attribute" (clause 9)

A profile defaultable character content architecture attribute corresponds to the set of character presentation attributes.

Definition 6.1

1 $\forall\, att$
2 $(_0$ IsDefaultableCharacterContentArchitectureAttribute(att) *iff*
3 IsCharacterContentPresentationAttribute$^{6.12}(att)_0$)

Semiformal Description 6.2

Predicate "is a profile character coding specification" (clause 10.1)

A character coding specification is a nomination where each element is a character content coding attribute.

Definition 6.2

1 $\forall\, v$
2 $(_0$ IsCharacterCodingSpecification(v) *iff*
3 IsNom(v) *and*
4 $\forall\, b \hat{\in} \hat{}\, v \bullet$ (IsCharacterContentCodingAttribute$^{6.13}$(C b))$_0$)

Semiformal Description 6.3

Predicate "is a profile character presentation feature" (clause 9)

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

Definition 6.3

1 $\forall\, v$
2 $(_0$ IsCharacterPresentationFeature(v) *iff*
3 IsNom(v) *and*
4 $\forall\, b \hat{\in} \hat{}\, v \bullet$ (IsCharacterContentPresentationAttribute$^{6.12}$(C b))$_0$)

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

9.2 Interface to the Document Architecture

Semiformal Description 6.4

Predicate "satisfies character content architecture contraints" (clauses 6.8 and 17)

A document profile *prof* and a document body *doby* satisfy the constraints imposed by the character content architecture if the following holds for all constituents and associated content portions:

If the attribute 'content architecture class' of the constituent has the value '2 8 2 6 1' then the content portion contains processable character content (6), the attributes 'formatting indicator' and 'initial offset' are not specified for this content (7) and the value '2 8 2 6 1' is an element of the document profile attribute 'content architecture classes' (8).

If the attribute 'content architecture class' of the constituent has the value '2 8 2 6 0' then the content portion contains formatted character content (11), the attributes 'widow size', 'orphan size', 'proportional line spacing' and 'indentation' are not specified for this content (12,13) and the value '2 8 2 6 0' is an element of the document profile attribute 'content architecture classes' (14).

If the attribute 'content architecture class' has the value '2 8 2 6 2' then content portion contains formatted processable character content (17) and the value '2 8 2 6 2' is an element of the document profile attribute 'content architecture classes' (18).

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

Definition 6.4

1 $\forall prof, doby$
2 $(_0$SatisfiesCharacterContentArchitectureConstraints$(prof, doby)$ *iff*
3 $\forall cst, cont \in doby$
4 $(_1 (_2 (_3 (cont)$DescribesContPortOf$^{2.153}(cst)$ *and*
5 C ^cst • 'content architecture class' = '2 8 2 6 1'$_3$) *impl*
6 $(_4$IsProcessableCharacterDescription$^{6.5}(cont)$ *and*
7 IsEmptyCol$^{1.4}$(['formatting indicator'; 'initial offset'] ∩ NAMS$^{1.18}(cst)$) *and*
8 '2 8 2 6 1' ∈ C ^$prof$ • 'content architecture classes'$_4$)$_2$) *and*
9 $(_5 (_6 (cont)$DescribesContPortOf$^{2.153}(cst)$ *and*
10 C ^cst • 'content architecture class' = '2 8 2 6 0'$_6$) *impl*
11 $(_7$IsFormattedCharacterDescription$^{6.7}(cont)$ *and*
12 IsEmptyCol$^{1.4}$(['widow size'; 'orphan size';
13 'proportional line spacing'; 'indentation'] ∩ NAMS$^{1.18}(cst)$) *and*
14 '2 8 2 6 0' ∈ C ^$prof$ • 'content architecture classes'$_7$)$_5$) *and*
15 $(_8 (_9 (cont)$DescribesContPortOf$^{2.153}(cst)$ *and*
16 C ^cst • 'content architecture class' = '2 8 2 6 2'$_9$) *impl*
17 $(_{10}$IsFormattedProcessableCharacterDescription$^{6.9}(cont)$ *and*
18 '2 8 2 6 2' ∈ C ^$prof$ • 'content architecture classes'$_{10}$)$_8$)$_1$)$_0$)

Semiformal Description 6.5

Predicate "is processable character description" (clause 17)

Content is processable if it is a character content portion description and the character content is processable.

Definition 6.5

1 $\forall cont$
2 $(_0$IsProcessableCharacterDescription$(cont)$ *iff*
3 IsCharacterContentPortionDescription$^{6.11}(cont)$ *and*
4 IsProcessableCharacterContent$^{6.6}(cont)_0$)

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 6.6

Predicate "is processable character content" (clause 17.2 and table 6)

A processable character content portion is a catenation containing a processable character string.

Definition 6.6

1 $\forall\, cont$
2 $({}_{0}$IsProcessableCharacterContent($cont$) *iff*
3 $\forall\, a \in \hat{}cont \bullet$
4 (N a = 'content information' *impl* ContainsProcessableElements$^{6.48}$(C a))$_{0}$)

Semiformal Description 6.7

Predicate "is formatted character description" (clause 17)

Content is formatted if it is a character content portion description and the character content is formatted.

Definition 6.7

1 $\forall\, cont$
2 $({}_{0}$IsFormattedCharacterDescription($cont$) *iff*
3 IsCharacterContentPortionDescription$^{6.11}$($cont$) *impl*
4 IsFormattedCharacterContent$^{6.8}$($cont$)$_{0}$)

Semiformal Description 6.8

Predicate "is formatted character content" (clause 17.1 and table 6)

A formatted character content portion is a catenation containing a formatted character string.

Definition 6.8

1 $\forall\, cont$
2 $({}_{0}$IsFormattedCharacterContent($cont$) *iff*
3 $\forall\, a \in \hat{}cont \bullet$
4 (N a = 'content information' *impl* ContainsFormattedElements$^{6.47}$(C a))$_{0}$)

Semiformal Description 6.9

Predicate "is formatted processable character description" (clause 17)

Content is formatted processable if it is a character content portion description and the character content is formatted processable.

Definition 6.9

1 $\forall\, cont$
2 $({}_{0}$IsFormattedProcessableCharacterDescription($cont$) *iff*
3 IsCharacterContentPortionDescription$^{6.11}$($cont$) *impl*
4 IsFormattedProcessableCharacterContent$^{6.10}$($cont$)$_{0}$)

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 6.10

Predicate "is formatted processable character content" (clause 17.3 and table 6)

A formatted processable character content is a catenation containing a formatted processable character string.

Definition 6.10

```
1   ∀ cont
2   (0 IsFormattedProcessableCharacterContent(cont) iff
3     ∀ a ∈̂ ^cont •
4     (N a = 'content information' impl ContainsFormattedProcessableElements^6.49(C a))0)
```

Semiformal Description 6.11

Predicate "is character content portion description"

A character content portion description is a set of character content portion attributes.

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

Definition 6.11

```
1   ∀ cont
2   (0 IsCharacterContentPortionDescription(cont) iff
3     IsCharacterContentPortionAttributeSet^6.14(cont)0)
```

Semiformal Description 6.12

Predicate "is character content presentation attribute" (clauses 6.3 and 9)

A character content architecture presentation attribute is one of the attributes 'alignment', 'character fonts', 'character orientation', 'character path', 'character spacing', 'code extension announcers', 'first line offset', 'formatting indicator', 'graphic character sets', 'graphic character subrepertoire', 'graphic rendition', 'indentation', 'initial offset', 'itemization', 'kerning offset', 'line layout table', 'line spacing', 'line progression', 'orphan size', 'pairwise kerning', 'proportional line spacing' or 'widow size' with corresponding value(s).

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

Definition 6.12

```
1   ∀ att
2   (0 IsCharacterContentPresentationAttribute(att) iff
3     ∃ n, c
4     (1 att = [n : c] and
5       n ∈ ['alignment';       'character fonts';      'character orientation';
6            'character path';  'character spacing';    'code extension announcers';
7            'first line offset'; 'formatting indicator';'graphic character sets';
8            'graphic rendition';'indentation';         'graphic character subrepertoire';
9            'initial offset';  'itemization';          'kerning offset';
10           'line layout table'; 'line progression';   'line spacing';
11           'orphan size';     'pairwise kerning';     'proportional line spacing';
12           'widow size'] and
13      (2 n = 'alignment' impl
14           (3 IsPlaceholder^1.19(c) or IsAlignmentValue^6.15(c)3)2) and
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

15 $({}_{4} n = \text{'character fonts'}\ \underline{impl}$
16 $({}_{5}\text{IsPlaceholder}^{1.19}(c)\ \underline{or}\ \text{IsCharacterFontsValue}^{6.16}(c)_{5})_{4})\ \underline{and}$
17 $({}_{6} n = \text{'character orientation'}\ \underline{impl}$
18 $({}_{7}\text{IsPlaceholder}^{1.19}(c)\ \underline{or}\ \text{IsCharacterOrientationValue}^{6.17}(c)_{7})_{6})\ \underline{and}$
19 $({}_{8} n = \text{'character path'}\ \underline{impl}$
20 $({}_{9}\text{IsPlaceholder}^{1.19}(c)\ \underline{or}\ \text{IsCharacterPathValue}^{6.19}(c)_{9})_{8})\ \underline{and}$
21 $({}_{10} n = \text{'character spacing'}\ \underline{impl}$
22 $({}_{11}\text{IsPlaceholder}^{1.19}(c)\ \underline{or}\ \text{IsCharacterSpacingValue}^{6.21}(c)_{11})_{10})\ \underline{and}$
23 $({}_{12} n = \text{'code extension announcers'}\ \underline{impl}$
24 $({}_{13}\text{IsPlaceholder}^{1.19}(c)\ \underline{or}\ \text{IsCodeExtensionAnnouncersValue}^{6.38}(c)_{13})_{12})\ \underline{and}$
25 $({}_{14} n = \text{'first line offset'}\ \underline{impl}$
26 $({}_{15}\text{IsPlaceholder}^{1.19}(c)\ \underline{or}\ \text{IsFirstLineOffsetValue}^{6.22}(c)_{15})_{14})\ \underline{and}$
27 $({}_{16} n = \text{'formatting indicator'}\ \underline{impl}$
28 $({}_{17}\text{IsPlaceholder}^{1.19}(c)\ \underline{or}\ \text{IsFormattingIndicatorValue}^{6.18}(c)_{17})_{16})\ \underline{and}$
29 $({}_{18} n = \text{'graphic character sets'}\ \underline{impl}$
30 $({}_{19}\text{IsPlaceholder}^{1.19}(c)\ \underline{or}\ \text{IsCharacterSetsValue}^{6.23}(c)_{19})_{18})\ \underline{and}$
31 $({}_{20} n = \text{'graphic character subrepertoire'}\ \underline{impl}$
32 $({}_{21}\text{IsPlaceholder}^{1.19}(c)\ \underline{or}\ \text{IsGraphicCharacterSubrepertoireValue}^{6.24}(c)_{21})_{20})\ \underline{and}$
33 $({}_{22} n = \text{'graphic rendition'}\ \underline{impl}$
34 $({}_{23}\text{IsPlaceholder}^{1.19}(c)\ \underline{or}\ \text{IsGraphicRenditionValue}^{6.27}(c)_{23})_{22})\ \underline{and}$
35 $({}_{24} n = \text{'indentation'}\ \underline{impl}$
36 $({}_{25}\text{IsPlaceholder}^{1.19}(c)\ \underline{or}\ \text{IsIndentationValue}^{6.34}(c)_{25})_{24})\ \underline{and}$
37 $({}_{26} n = \text{'initial offset'}\ \underline{impl}$
38 $({}_{27}\text{IsPlaceholder}^{1.19}(c)\ \underline{or}\ \text{IsCharacterInitialOffsetValue}^{6.35}(c)_{27})_{26})\ \underline{and}$
39 $({}_{28} n = \text{'itemization'}\ \underline{impl}$
40 $({}_{29}\text{IsPlaceholder}^{1.19}(c)\ \underline{or}\ \text{IsItemizationValue}^{6.46}(c)_{29})_{28})\ \underline{and}$
41 $({}_{30} n = \text{'kerning offset'}\ \underline{impl}$
42 $({}_{31}\text{IsPlaceholder}^{1.19}(c)\ \underline{or}\ \text{IsKerningOffsetValue}^{6.36}(c)_{31})_{30})\ \underline{and}$
43 $({}_{32} n = \text{'line layout table'}\ \underline{impl}$
44 $({}_{33}\text{IsPlaceholder}^{1.19}(c)\ \underline{or}\ \text{IsLineLayoutTableValue}^{6.42}(c)_{33})_{32})\ \underline{and}$
45 $({}_{34} n = \text{'line spacing'}\ \underline{impl}$
46 $({}_{35}\text{IsPlaceholder}^{1.19}(c)\ \underline{or}\ \text{IsLineSpacingValue}^{6.44}(c)_{35})_{34})\ \underline{and}$
47 $({}_{36} n = \text{'line progression'}\ \underline{impl}$
48 $({}_{37}\text{IsPlaceholder}^{1.19}(c)\ \underline{or}\ \text{IsCharacterLineProgressionValue}^{6.45}(c)_{37})_{36})\ \underline{and}$
49 $({}_{38} n = \text{'orphan size'}\ \underline{impl}$
50 $({}_{39}\text{IsPlaceholder}^{1.19}(c)\ \underline{or}\ \text{IsOrphanSizeValue}^{6.37}(c)_{39})_{38})\ \underline{and}$
51 $({}_{40} n = \text{'pairwise kerning'}\ \underline{impl}$
52 $({}_{41}\text{IsPlaceholder}^{1.19}(c)\ \underline{or}\ \text{IsPairwiseKerningValue}^{6.39}(c)_{41})_{40})\ \underline{and}$
53 $({}_{42} n = \text{'proportional line spacing'}\ \underline{impl}$
54 $({}_{43}\text{IsPlaceholder}^{1.19}(c)\ \underline{or}\ \text{IsProportionalLineSpacingValue}^{6.41}(c)_{43})_{42})\ \underline{and}$
55 $({}_{44} n = \text{'widow size'}\ \underline{impl}$
56 $({}_{45}\text{IsPlaceholder}^{1.19}(c)\ \underline{or}\ \text{IsWidowSizeValue}^{6.40}(c)_{45})_{44})_{1})_{0})$

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Semiformal Description 6.13

Predicate "is a character content coding attribute" (clause 10.1)

A character content coding attribute consists of the attribute 'type of coding' with its corresponding value.

Definition 6.13

1 $\forall att$
2 $({}_0$IsCharacterContentCodingAttribute(att) *iff*
3 $\exists n, c$
4 $({}_1\ att = [n : c]$ *and*
5 $n =$ 'type of coding' *impl* IsCharacterContentTypeOfCodingValue$^{6.20}(c)_1)_0)$

9.3 Attributes of the Character Content Architecture

Semiformal Description 6.14

Predicate "is character content portion attribute set"

A character content architecture portion attribute set contains the listed attributes with corresponding attribute values.

Definition 6.14

1 $\forall as$
2 $({}_0$IsCharacterContentPortionAttributeSet(as) *iff*
3 $({}_1$IsNeNom$^{1.2}(as)$ *and*
4 $\forall a \in \hat{}as \bullet$
5 $({}_2$N $a =$ 'type of coding' *impl*
6 $({}_3$IsPlaceholder$^{1.19}$(C $a)_3)_2)$ *and*
7 $({}_4$N $a =$ 'content information' *impl* IsCharacterContentInformationValue$^{6.111}$(C $a)_4)_1)_0)$

Semiformal Description 6.15

Predicate "is an alignment value" (clause 7.2.1)

The value of the attribute 'alignment' is 'start-aligned', 'end-aligned', 'centred' or 'justified'.

Definition 6.15

1 $\forall v$
2 $({}_0$IsAlignmentValue(v) *iff*
3 $v \in$ ['start-aligned'; 'end-aligned'; 'centred'; 'justified'$]_0)$

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 6.16

Predicate "is a character fonts value" (clause 9.1.2)

The value of the attribute 'character fonts' can contain up to ten pairs of parameters (3) named 'primary font', 'first alternative font', 'second alternative font', 'third alternative font', 'fourth alternative font', 'fifth alternative font', 'sixth alternative font', 'seventh alternative font', 'eighth alternative font' and 'ninth alternative font' (4-8), each with sub-parameters 'font size' and 'font identifier' (10), both having a positive integer value (11).

Definition 6.16

```
1   ∀ v
2   (0 IsCharacterFontsValue(v) iff
3     (1 IsNom(v) and   CARD (v) ≤ 10 and
4       NAMS1.18(v) ⊆ ['primary font';          'first alternative font';
5                      'second alternative font';'third alternative font';
6                      'fourth alternative font'; 'fifth alternative font';
7                      'sixth alternative font';  'seventh alternative font';
8                      'eighth alternative font'; 'ninth alternative font'] and
9       ∀ a ∈̂ ^v •
10      (2 IsNom(C a) and NAMS1.18(C a) = ['font size'; 'font identifier'] and
11      ∀ b ∈ ^(C a) • IsNat(C b)2)1)0)
```

Semiformal Description 6.17

Predicate "is a character orientation value" (clause 9.1.3)

The value of the attribute 'character orientation' is '0°', '90°', '180°' or '270°'.

Definition 6.17

```
1   ∀ v
2   (0 IsCharacterOrientationValue(v) iff
3     v ∈ ['0°'; '90°'; '180°'; '270°']0)
```

Semiformal Description 6.18

Predicate "is a formatting indicator value" (clause 9.2.1)

The attribute 'formatting indicator' has the value 'yes' or 'no'.

Definition 6.18

```
1   ∀ v
2   (0 IsFormattingIndicatorValue(v) iff
3     v ∈ ['yes'; 'no']0)
```

Semiformal Description 6.19

Predicate "is a character path value" (clause 9.1.4)

The value of the attribute 'character path' is '0°', '90°', '180°' or '270°'.

Definition 6.19

```
1   ∀ v
2   (0 IsCharacterPathValue(v) iff
3     v ∈ ['0°'; '90°'; '180°'; '270°']0)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 6.20

Predicate "is a character content type of coding value" (clause 10.1)

The value of the attribute 'type of coding' is an ASN.1 object indentifier '2 8 3 6 0'

Definition 6.20

1 $\forall v$
2 $(_{o}$IsCharacterContentTypeOfCodingValue(v) *iff*
3 v = '2 8 3 6 0'$_{o})$

Semiformal Description 6.21

Predicate "is a character spacing value" (clause 9.1.5)

The value of the attribute 'character spacing' is any positive integer.

Definition 6.21

1 $\forall v$
2 $(_{o}$IsCharacterSpacingValue(v) *iff*
3 IsNat$(v)_{o})$

Semiformal Description 6.22

Predicate "is a first line offset value" (clause 9.1.7)

The value of the attribute 'first line offset' is any integer.

Definition 6.22

1 $\forall v$
2 $(_{o}$IsFirstLineOffsetValue(v) *iff*
3 IsInt$(v)_{o})$

Semiformal Description 6.23

Predicate "is a character sets value" (clause 9.1.8)

The value of the attribute 'graphic character sets' is a string of escape sequences (in accordance with ISO 2022 and ISO 2375) found in the *ISO International register of coded character sets to be used with escape sequences* (held by the *European Computer Manufacturers Association* (ECMA) – the registration authority). These escape sequences are used to designate one or more graphic character sets, and any locking shift functions needed for their invocation.

Definition 6.23

1 $\forall v$
2 $(_{o}$IsCharacterSetsValue(v) *iff*
3 IsISO2022String$^{6.92}(v)_{o})$

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 6.24

Predicate "is a graphic character subrepertoire value" (clause 9.1.9)

The value of the attribute **'graphic character subrepertoire'** is either 0 or an identifier of a subrepertoire.

Definition 6.24

```
1   ∀ v
2   (0 IsGraphicCharacterSubrepertoireValue(v) iff
3     v = 0 or IsISO6937SubrepertoireId^6.25(v)0)
```

Semiformal Description 6.25

Predicate "is a ISO 6937 subrepertoire identifier" (clause 9.1.9)

An ISO 6937 subrepertoire identifier is considered atomic for this definition.

Definition 6.25

```
1   ∀ v
2   (0 IsISO6937SubrepertoireId(v) iff
3     IsAtom(v)0)
```

Semiformal Description 6.26

Predicate "is a character content architecture class value" (clause 9.4.1)

A content architecture class attribute value is an ASN.1 object identifier '2 8 2 6 1', '2 8 2 6 2' or '2 8 2 6 0'.

Definition 6.26

```
1   ∀ v
2   (0 IsCharacterContentArchitectureClassValue(v) iff
3     v ∈ ['2 8 2 6 1'; '2 8 2 6 2'; '2 8 2 6 0']0)
```

Semiformal Description 6.27

Predicate "is a graphic rendition value" (clauses 9.1.10 and 8.1)

The value of the attribute **'graphic rendition'** can contain any sequence of integer values from those specified below (4-12).

Definition 6.27

```
1   ∀ v
2   (0 IsGraphicRenditionValue(v) iff
3     IsNeCat^1.3(v) and
4     (1 v = [ → 0 → ] or
5       ∀ x ∈̂ ^v •
6       (2 IsWeightValue^6.28(x) or
7         IsPostureValue^6.29(x) or
8         IsUnderliningValue^6.30(x) or
9         IsBlinkingValue^6.31(x) or
10        IsImageInversionValue^6.32(x) or
11        IsCrossingOutValue^6.33(x) or
12        x ∈ [10; 11;12; 13; 14; 15; 16; 17; 18; 19; 26; 38; 48; 50]2)1)0)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Semiformal Description 6.28

Predicate "is a weight value" (clause 8.1.1)

A weight value is either the integer 2, which represents a rendition *feint* (decreased intensity), the integer 22, which represents a rendition *normal intensity* (neither *feint* nor *bold*), or the integer 1, which represents a rendition *bold* (increased intensity).

Definition 6.28

1 $\forall v$
2 $(_0\text{IsWeightValue}(v)$ *iff*
3 $v \in [2;22;1]_0)$

Semiformal Description 6.29

Predicate "is a posture value" (clause 8.1.2)

A posture value is either the integer 23, which represents a rendition *not italicized*, or the integer 3, which represents a rendition *italicized*.

Definition 6.29

1 $\forall v$
2 $(_0\text{IsPostureValue}(v)$ *iff*
3 $v \in [23;3]_0)$

Semiformal Description 6.30

Predicate "is an underlining value" (clause 8.1.3)

An underlining value is the integer 24, which represents a rendition *not underlined*, the integer 4, which represents a rendition *underlined*, or the integer 21, which represents a rendition *doubly underlined*.

Definition 6.30

1 $\forall v$
2 $(_0\text{IsUnderliningValue}(v)$ *iff*
3 $v \in [24;4;21]_0)$

Semiformal Description 6.31

Predicate "is a blinking value" (clause 8.1.4)

A blinking value is the integer 25, which represents a rendition *steady* (not blinking), the integer 5, which represents a rendition *slowly blinking*, or the integer 6, which represents a rendition *rapidly blinking*.

Definition 6.31

1 $\forall v$
2 $(_0\text{IsBlinkingValue}(v)$ *iff*
3 $v \in [25;5;6]_0)$

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 6.32

Predicate "is an image inversion value" (clause 8.1.5)

An image inversion value is either the integer 27, which represents a rendition *positive image*, or the integer 7, which represents a rendition *negative image*.

Definition 6.32

1 $\forall v$
2 $(_0$IsImageInversionValue(v) *iff*
3 $v \in [27;7]_0)$

Semiformal Description 6.33

Predicate "is a crossing-out value" (clause 8.1.6)

A crossing-out value is either the integer 29, which represents a rendition *not crossed-out*, or the integer 9, which represents a rendition *crossed-out*.

Definition 6.33

1 $\forall v$
2 $(_0$IsCrossingOutValue(v) *iff*
3 $v \in [29;9]_0)$

Semiformal Description 6.34

Predicate "is an indentation value" (clause 9.3.1)

The value of the attribute 'indentation' is any non-negative integer.

Definition 6.34

1 $\forall v$
2 $(_0$IsIndentationValue(v) *iff*
3 IsNnInt$^{1.7}(v)_0)$

Semiformal Description 6.35

Predicate "is an initial offset value" (clause 9.2.2)

The value of the attribute 'initial offset' consists of two parameters 'horizontal coordinate' and 'vertical coordinate' each having a non-negative integer value.

NOTE: Since the attribute 'initial offset' is also used within the raster graphics content architectures the qualifier "Character" has been added to the predicate name.

Definition 6.35

1 $\forall v$
2 $(_0$IsCharacterInitialOffsetValue(v) *iff*
3 $(_1$IsNeNom$^{1.2}(v)$ *and*
4 NAMS$^{1.18}(v)$ = ['horizontal coordinate';'vertical coordinate'] *and*
5 $\forall b \in \hat{}v \bullet$
6 $(_2$N b = 'horizontal coordinate' *impl* IsNnInt$^{1.7}$(C b) *and*
7 N b = 'vertical coordinate' *impl* IsNnInt$^{1.7}$(C $b)_2)_1)_0)$

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 6.36

Predicate "is a kerning offset value" (clause 9.1.12)

The value of the attribute 'kerning offset' consists of two parameters 'start edge offset' and 'end edge offset' each having a non-negative integer value.

Definition 6.36

1 $\forall v$
2 $(_0$IsKerningOffsetValue(v) *iff*
3 $(_1$IsNom(v) *and*
4 NAMS$^{1.18}(v)$ = ['start edge offset'; 'end edge offset'] *and*
5 $\forall b \in \hat{}v \bullet$
6 $(_2$N b = 'start edge offset' *impl* IsNnInt$^{1.7}$(C b) *and*
7 N b = 'end edge offset' *impl* IsNnInt$^{1.7}$(C $b)_2)_1)_0)$

Semiformal Description 6.37

Predicate "is an orphan size value" (clause 9.3.2)

The value of the attribute 'orphan size' is any positive integer.

Definition 6.37

1 $\forall v$
2 $(_0$IsOrphanSizeValue(v) *iff*
3 IsNat$(v)_0)$

Semiformal Description 6.38

Predicate "is a code extension announcers value" (clause 9.1.6)

The value of the attribute 'code extension announcers' is a string of escape sequences in accordance with ISO 2022, used to announce code extension features.

Definition 6.38

1 $\forall v$
2 $(_0$IsCodeExtensionAnnouncersValue(v) *iff*
3 IsISO2022String$^{6.92}(v)_0)$

Semiformal Description 6.39

Predicate "is a pairwise kerning value" (clause 9.1.16)

The value of the attribute 'pairwise kerning' is either 'yes' or 'no'.

Definition 6.39

1 $\forall v$
2 $(_0$IsPairwiseKerningValue(v) *iff*
3 $v \in$ ['yes'; 'no']$_0)$

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 6.40

Predicate "is a widow size value" (clause 9.3.4)

The value of the attribute 'widow size' is any positive integer.

Definition 6.40

1 $\forall v$
2 $({}_0$IsWidowSizeValue(v) *iff*
3 IsNat$(v)_0$)

Semiformal Description 6.41

Predicate "is a proportional line spacing value" (clause 9.3.3)

The value of the attribute 'proportional line spacing' is either 'yes' or 'no'.

Definition 6.41

1 $\forall v$
2 $({}_0$IsProportionalLineSpacingValue(v) *iff*
3 $v \in$ ['yes';'no']$_0$)

Semiformal Description 6.42

Predicate "is a line layout table value" (clause 9.1.13)

The value of the attribute 'line layout table' can consists of four parameters: 'tab reference', 'tab position', 'alignment' and conditionally 'alignment string' each with corresponding values. In the case of 'alignment string' this parameter is only specified if the value 'aligned-around' is specified for the parameter 'alignment' (9-13). The value of 'alignment string' is represented as an atomic structure (13).

Definition 6.42

1 $\forall v$
2 $({}_0$IsLineLayoutTableValue(v) *iff*
3 $({}_1$IsNom(v) *and*
4 NAMS$^{1.18}(v) \supseteq$ ['tab reference';'tab position';'alignment'] *and*
5 NAMS$^{1.18}(v) \subseteq$ ['tab reference';'tab position';'alignment';'alignment string'] *and*
6 $\forall b \in \hat{}v$ •
7 $({}_2$N b = 'tab reference' *impl* IsTabReferenceString$^{6.43}$(C b) *and*
8 N b = 'tab position' *impl* IsNnInt$^{1.7}$(C b) *and*
9 N b = 'alignment' *impl*
10 $({}_3$C $b \in$ ['start-aligned';'end-aligned';'centred';'aligned-around'] *and*
11 C b = 'aligned-around' *impl*
12 $({}_4$'alignment string' $\in$ NAMS$^{1.18}(v)_4)_3$) *and*
13 N b = 'alignment string' *impl* IsAtom(C $b)_2)_1)_0$)

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 6.43

Predicate "is a tab reference string" (clause 9.1.13)

A tab reference string in this context consists of a concatenated sequence of integers whose length is greater than or equal to 1 and less than or equal to 4.

Definition 6.43

1 $\forall v$
2 $(_0$IsTabReferenceString(v) *iff*
3 IsNeCat$^{1.3}(v)$ *and*
4 (LENGTH$^{1.16}$ $(v) \geq 1$ *and* LENGTH$^{1.16}$ $(v) \leq 4)$ *and*
5 $\forall x \hat{\in} \hat{}v \bullet$ (IsInt(C x))$_0$)

Semiformal Description 6.44

Predicate "is a line spacing value" (clause 9.1.15)

The value of the attribute 'line spacing' is any positive integer.

Definition 6.44

1 $\forall v$
2 $(_0$IsLineSpacingValue(v) *iff*
3 IsNat(v)$_0$)

Semiformal Description 6.45

Predicate "is a line progression value" (clause 9.1.14)

The value of the attribute 'line progression' is '90°' or '270°'.

NOTE: Since the attribute 'line progression' is also used within the raster graphics content architectures the qualifier "Character" has been added to the predicate name.

Definition 6.45

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

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 6.46

Predicate "is an itemization value" (clause 9.1.11)

The value of the attribute 'itemization' consists of three parameters 'identifier alignment', 'identifier start offset' and 'identifier end offset'. The values of 'identifier start offset' and 'identifier end offset' are integers (9, 10). The value of 'identifier alignment' is an element of the set 'no itemization', 'start-aligned' and 'end-aligned' (8).

Definition 6.46

$\forall v$
$(_0$ IsItemizationValue(v) *iff*
  $(_1$ IsNom(v) *and*
    NAMS$^{1.18}(v)$ = ['identifier alignment'; 'identifier start offset';
      'identifier end offset']$_1$) *and*
    $\forall a \in \hat{} v \bullet$
    $(_2$ N a = 'identifier alignment' *impl*
      $(_3$ C $a \in$ ['no itemization';'start-aligned'; 'end-aligned']$_3$) *and*
      N a = 'identifier start offset' *impl* IsInt(C a) *and*
      N a = 'identifier end offset' *impl* IsInt(C $a)_2)_0$)

9.4 Elements of the Character Content Information

Semiformal Description 6.47

Predicate "contains formatted elements" (clause 6)

A character string represents a valid formatted content if the string has a valid syntax (3) and valid semantics (4) for formatted content.

Definition 6.47

$\forall v$
$(_0$ ContainsFormattedElements(v) *iff*
  IsValidFormattedContentSyntax$^{6.50}(v)$ *and*
  IsValidFormattedContentSemantics$^{6.57}(v)_0$)

Semiformal Description 6.48

Predicate "contains processable elements" (clause 6)

A character string represents a valid processable content if the string has a valid syntax (3) and valid semantics (4) for processable content.

Definition 6.48

$\forall v$
$(_0$ ContainsProcessableElements(v) *iff*
  IsValidProcessableContentSyntax$^{6.51}(v)$ *and*
  IsValidProcessableContentSemantics$^{6.58}(v)_0$)

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 6.49

Predicate "contains formatted processable elements" (clause 6)

A character string represents a valid formatted processable content if the string has a valid syntax (3) and valid semantics (4) for formatted processable content.

Definition 6.49

1 $\forall v$
2 $({}_{0}$ ContainsFormattedProcessableElements(v) *iff*
3 IsValidFormattedProcessableContentSyntax$^{6.52}(v)$ *and*
4 IsValidFormattedProcessableContentSemantics$^{6.59}(v)_{0})$

Semiformal Description 6.50

Predicate "is a valid formatted content syntax" (clause 6)

A valid formatted character content syntax is an empty string (3) or a string in which the first element is a formatted content element (4) and the remainder of the string has a valid formatted content syntax (5).

Definition 6.50

1 $\forall v$
2 $({}_{0}$ IsValidFormattedContentSyntax(v) *iff*
3 IsEmptyCat$^{1.6}(v)$ *or*
4 $({}_{1}$ IsFormattedContentElement$^{6.53}$(HEAD$^{1.13}(v))$ *and*
5 IsValidFormattedContentSyntax$^{6.50}$(TAIL$^{1.15}(v))_{1})_{0})$

Semiformal Description 6.51

Predicate "is a valid processable content syntax" (clause 6)

A valid processable character content syntax is an empty string (3) or a string in which the first element is processable content element (4) and the remainder of the string has a valid processable content syntax (5).

Definition 6.51

1 $\forall v$
2 $({}_{0}$ IsValidProcessableContentSyntax(v) *iff*
3 IsEmptyCat$^{1.6}(v)$ *or*
4 $({}_{1}$ IsProcessableContentElement$^{6.54}$(HEAD$^{1.13}(v))$ *and*
5 IsValidProcessableContentSyntax$^{6.51}$(TAIL$^{1.15}(v))_{1})_{0})$

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 6.52

Predicate "is a valid formatted processable content syntax" (clause 6)

A valid formatted processable character content syntax is an empty string (3) or a string in which the first element is a formatted processable content element (4) and the remainder of the string has a valid formatted processable content syntax (5).

Definition 6.52

1 $\forall v$
2 $(_0$ IsValidFormattedProcessableContentSyntax(v) *iff*
3 IsEmptyCat$^{1.6}(v)$ *or*
4 $(_1$ IsFormattedProcessableContentElement$^{6.55}$(HEAD$^{1.13}(v)$) *and*
5 IsValidFormattedProcessableContentSyntax$^{6.52}$(TAIL$^{1.15}(v))_1)_0$)

Semiformal Description 6.53

Predicate "is a formatted content element" (clause 6)

A formatted character content element is either a content element common to all character content architectures (3) or a layout control function (4).

Definition 6.53

1 $\forall v$
2 $(_0$ IsFormattedContentElement(v) *iff*
3 IsCommonContentElement$^{6.56}(v)$ *or*
4 IsLayoutControlFn$^{6.100}(v)_0$)

Semiformal Description 6.54

Predicate "is a processable content element" (clause 6)

A processable character content element is either a content element common to all character content architectures (3) or a logical control function (4).

Definition 6.54

1 $\forall v$
2 $(_0$ IsProcessableContentElement(v) *iff*
3 IsCommonContentElement$^{6.56}(v)$ *or*
4 IsLogicalControlFn$^{6.104}(v)_0$)

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 6.55

Predicate "is a formatted processable content element" (clause 6)

A formatted processable character content element is a content element common to all character content architectures (3), a layout control function (4), a logical control function (5), or a delimiter control function (6).

Definition 6.55

1 $\forall v$
2 $(_0$IsFormattedProcessableContentElement(v) *iff*
3 IsCommonContentElement$^{6.56}(v)$ *or*
4 IsLayoutControlFn$^{6.100}(v)$ *or*
5 IsLogicalControlFn$^{6.104}(v)$ *or*
6 IsDelimiterControlFn$^{6.108}(v)_0$)

Semiformal Description 6.56

Predicate "is common content element" (clause 6)

A content element common to all character content architectures is a shared control function (3), a space control function (4), a code extension control function (5), an escape control function (6), or a graphic character string (7).

Definition 6.56

1 $\forall v$
2 $(_0$IsCommonContentElement(v) *iff*
3 IsSharedControlFn$^{6.96}(v)$ *or*
4 IsSpaceControlFn$^{6.109}(v)$ *or*
5 IsCodeExtensionControlFn$^{6.91}(v)$ *or*
6 IsEscapeControlFn$^{6.110}(v)$ *or*
7 IsGraphicCharacterString$^{6.112}(v)_0$)

Semiformal Description 6.57

Predicate "is a valid formatted content semantics" (clause 6)

A character content with valid formatted content semantics must have valid semantics for its common content elements (3).

Definition 6.57

1 $\forall v$
2 $(_0$IsValidFormattedContentSemantics(v) *iff*
3 IsValidCommonContentSemantics$^{6.60}(v)_0$)

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 6.58

Predicate "is a valid processable content semantics" (clause 6)

A character content with valid processable content semantics must have valid semantics for its common content elements (3) and its logical control elements (4).

Definition 6.58

1 $\forall v$
2 $(_0$IsValidProcessableContentSemantics(v) *iff*
3 IsValidCommonContentSemantics$^{6.60}(v)$ *and*
4 IsValidLogicalControlSemantics$^{6.61}(v)_0)$

Semiformal Description 6.59

Predicate "is a valid formatted processable content semantics" (clause 6)

A character content with valid formatted processable content semantics must have valid semantics for its common content elements (3), its logical control elements (4), and its delimiter control elements (5).

Definition 6.59

1 $\forall v$
2 $(_0$IsValidFormattedProcessableContentSemantics(v) *iff*
3 IsValidCommonContentSemantics$^{6.60}(v)$ *and*
4 IsValidLogicalControlSemantics$^{6.61}(v)$ *and*
5 IsValidDelimiterControlSemantics$^{6.62}(v)_0)$

Semiformal Description 6.60

Predicate "is a valid common content semantics" (clause 13.1)

A character content with valid common content semantics must have valid shared control semantics (3).

Definition 6.60

1 $\forall v$
2 $(_0$IsValidCommonContentSemantics(v) *iff*
3 IsValidSharedControlSemantics$^{6.63}(v)_0)$

Semiformal Description 6.61

Predicate "is a valid logical control semantics" (clause 13.3)

A character content with valid logical control semantics must have valid PTX function sequence semantics (3).

Definition 6.61

1 $\forall v$
2 $(_0$IsValidLogicalControlSemantics(v) *iff*
3 IsValidPTX$^{6.82}(v)_0)$

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 6.62

Predicate "is a valid delimiter control semantics" (clause 13.4)

A character content with valid logical control semantics must have valid SOS function sequence semantics (3).

Definition 6.62

1 $\forall v$
2 $(_0$IsValidDelimiterControlSemantics(v) *iff*
3 IsValidSOS$^{6.85}(v)_0$)

Semiformal Description 6.63

Predicate "is a valid shared control semantics" (clause 13.1)

A character content with valid shared control semantics must have valid semantics for the function sequences LF (3), GCC (4,5), PLU-PLD (6,8), SRS (7,8,10), VPB-VPR (9-10).

Definition 6.63

1 $\forall v$
2 $(_0$IsValidSharedControlSemantics(v) *iff*
3 IsValidLF$^{6.67}(v)$ *and*
4 IsValidGCC0$^{6.64}(v)$ *and*
5 IsValidGCC12$^{6.65}(v)$ *and*
6 IsValidPLUD$^{6.70}(v)$ *and*
7 IsValidSRS$^{6.73}(v)$ *and*
8 IsValidPLUDInSRS$^{6.76}(v)$ *and*
9 IsValidVP$^{6.80}(v)$ *and*
10 IsValidVPInSRS$^{6.88}(v)_0$)

Semiformal Description 6.64

Predicate "is a valid GCC0" (clause 13.1.2)

A character content string with valid GCC0 control function and parameter semantics is: (a) an empty string (3), or (b) a string that begins with a GCC control function with the parameter 0 followed by two graphic characters a and b (5, 6) and for which the remainder of the content string has valid GCC0 semantics (7), or (c) a non-empty string that does not begin with a GCC control function with the parameter 0 (9, 10) and for which the remainder of the content string has valid GCC0 semantics (11).

Definition 6.64

1 $\forall v$
2 $(_0$IsValidGCC0(v) *iff*
3 IsEmptyCat$^{1.6}(v)$ *or*
4 $\exists t, a, b$
5 $(_1 v = [\rightarrow [\rightarrow$ 'GCC' $\rightarrow 0 \rightarrow] \rightarrow a \rightarrow b \rightarrow]//t$ *and*
6 IsGraphicCharacter$^{6.113}(a)$ *and* IsGraphicCharacter$^{6.113}(b)$ *and*
7 IsValidGCC0$^{6.64}(t)_1$) *or*
8 $\exists t$
9 $(_2 v \neq [\rightarrow [\rightarrow$ 'GCC' $\rightarrow 0 \rightarrow] \rightarrow]//t$ *and*
10 IsNeCat$^{1.3}(v)$ *and*
11 IsValidGCC0$^{6.64}$(TAIL$^{1.15}(v))_2)_0$)

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 6.65

Predicate "is a valid GCC12" (clause 13.1.2)

A character content string with valid GCC12 control function semantics and parameter is: (a) an empty string (3), or (b) a string that begins with a GCC control function with the parameter 1 (5) and for which the remainder of the content string has a length of at least 2 (7) and valid GCC12 semantics (6), or (c) a non-empty string that does not begin with a GCC control function with the parameter 1 or 2 (9, 10) and for which the remainder of the content string has valid GCC12 semantics (11).

Definition 6.65

1 $\forall v$
2 $(_0$IsValidGCC12(v) *iff*
3 IsEmptyCat$^{1.6}(v)$ *or*
4 $\exists t$
5 $(_1 v = [\rightarrow [\rightarrow$ 'GCC' $\rightarrow 1 \rightarrow] \rightarrow]//t$ *and*
6 IsValidGCC12End$^{6.66}(t)$ *and*
7 LENGTH$^{1.16}(t) \geq 2_1)$ *and*
8 $\exists t, n$
9 $(_2$ *not* $(v = [\rightarrow [\rightarrow$ 'GCC' $\rightarrow n \rightarrow] \rightarrow]//t$ *and* $n \in [1;2])$ *and*
10 IsNeCat$^{1.3}(v)$ *and*
11 IsValidGCC12$^{6.65}$(TAIL$^{1.15}(v))_2)_0)$

Semiformal Description 6.66

Predicate "is a valid GCC12 end" (clause 13.1.2)

A character string with valid GCC12 end semantics is (a) a string which begins with the control function GCC with the parameter 2 and followed by a string with valid GCC12 semantics (3–5) or (b) a non-empty character string which does not start with the control function GCC with the parameter 1, 2, or 3 and where the first character is a graphic character (7, 8) and where the remainder of the string has valid GCC12 end semantics (9).

Definition 6.66

1 $\forall v$
2 $(_0$IsValidGCC12End(v) *iff*
3 $\exists t$
4 $(_1 v = [\rightarrow [\rightarrow$ 'GCC' $\rightarrow 2 \rightarrow] \rightarrow]//t$ *and*
5 IsValidGCC12$^{6.65}(t)_1)$ *or*
6 $\exists t, n$
7 $(_2$ *not* $(v = [\rightarrow [\rightarrow$ 'GCC' $\rightarrow n \rightarrow] \rightarrow]//t$ *and* $n \in [0;1;2])$ *and*
8 IsNeCat$^{1.3}(v)$ *and* IsGraphicCharacter$^{6.113}$(HEAD$^{1.13}(v))$ *and*
9 IsValidGCC12End$^{6.66}$(TAIL$^{1.15}(v))_2)_0)$

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 6.67

Predicate "is a valid LF" (clause 13.1.4)

A character content string with valid LF function semantics is: (a) an empty string (3), or (b) a string that begins with an LF function (5) and for which the remainder of the content string has valid LF semantics assuming a leading LF function (6), or (c) a non-empty string that does not begin with an LF function (8,9) and for which the remainder of the content string has valid LF semantics assuming that there is no leading LF function (10).

Definition 6.67

1 $\forall v$
2 $(_0$IsValidLF(v) *iff*
3 IsEmptyCat$^{1.6}(v)$ *or*
4 $\exists t$
5 $(_1 v = [\rightarrow [\rightarrow$ 'LF' $\rightarrow] \rightarrow]//t$ *and*
6 IsValidNextLF1$^{6.69}(t)_1)$ *or*
7 $\exists t$
8 $(_2 v \neq [\rightarrow [\rightarrow$ 'LF' $\rightarrow] \rightarrow]//t$ *and*
9 IsNeCat$^{1.3}(v)$ *and*
10 IsValidNextLF0$^{6.68}$(TAIL$^{1.15}(v))_2)_0)$

Semiformal Description 6.68

Predicate "is a valid next LF zero" (clause 13.1.4)

A character content string with valid LF function semantics assuming no leading LF is: (a) an empty string (3), or (b) a string that begins with CR-LF functions (5) and for which the remainder of the content string has valid LF semantics assuming a leading LF function (6), or (c) a non-empty string that does not begin with CR-LF functions (8,9) and for which the remainder of the content string has valid LF semantics assuming that there are no leading LF functions (10).

Definition 6.68

1 $\forall v$
2 $(_0$IsValidNextLF0(v) *iff*
3 IsEmptyCat$^{1.6}(v)$ *or*
4 $\exists t$
5 $(_1 v = [\rightarrow [\rightarrow$ 'CR' $\rightarrow] \rightarrow [\rightarrow$ 'LF' $\rightarrow] \rightarrow]//t$ *and*
6 IsValidNextLF1$^{6.69}(t)_1)$ *or*
7 $\exists t$
8 $(_2 v \neq [\rightarrow [\rightarrow$ 'CR' $\rightarrow] \rightarrow [\rightarrow$ 'LF' $\rightarrow] \rightarrow]//t$ *and*
9 IsNeCat$^{1.3}(v)$ *and*
10 IsValidNextLF0$^{6.68}$(TAIL$^{1.15}(v))_2)_0)$

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 6.69

Predicate "is a valid next LF one" (clause 13.1.4)

A character content string with valid LF function semantics assuming a leading LF is: (a) an empty string (3), or (b) a string that begins with an LF function (5) and for which the remainder of the content string has valid LF semantics assuming a leading LF function (6), or (c) a string that begins with CR-LF functions (8) and for which the remainder of the content string has valid LF semantics assuming a leading LF function (9), or (d) a string that begins with a CR function with no subsequent LF function (11) and for which the remainder of the content string has valid LF semantics assuming that there are no leading LF functions (12), or (e) a non-empty string that does not begin with CR or LF functions (14,15) and for which the remainder of the content string has valid LF semantics assuming that there are no leading LF functions (16).

Definition 6.69

1 $\forall v$
2 $(_0$IsValidNextLF1(v) *iff*
3 IsEmptyCat$^{1.6}(v)$ *or*
4 $\exists t$
5 $(_1 v = [\rightarrow [\rightarrow$ 'LF' $\rightarrow] \rightarrow]//t$ *and*
6 IsValidNextLF1$^{6.69}(t)_1)$ *or*
7 $\exists t$
8 $(_2 v = [\rightarrow [\rightarrow$ 'CR' $\rightarrow] \rightarrow [\rightarrow$ 'LF' $\rightarrow] \rightarrow]//t$ *and*
9 IsValidNextLF1$^{6.69}(t)_2)$ *or*
10 $\exists t, x$
11 $(_3 v = [\rightarrow [\rightarrow$ 'CR' $\rightarrow] \rightarrow x \rightarrow]//t$ *and* $x \neq [\rightarrow$ 'LF' $\rightarrow]$ *and*
12 IsValidNextLF0$^{6.68}([\rightarrow x \rightarrow]//t)_3)$ *or*
13 $\exists t, x$
14 $(_4$ *not* $(v = [\rightarrow [\rightarrow x \rightarrow] \rightarrow]//t$ *and* $x \in$ ['CR';'LF']) *and*
15 IsNeCat$^{1.3}(v)$ *and*
16 IsValidNextLF0$^{6.68}$(TAIL$^{1.15}(v))_4)_0)$

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 6.70

Predicate "is a valid PLUD" (clauses 13.1.5 and 13.1.6)

A character content string with valid PLU-PLD function sequence semantics is: (a) an empty string (3), or (b) a string that begins with a PLU function (5) and for which the remainder of the content string has valid incomplete PLU-PLD semantics (6), or (c) a string that begins with a PLD function (8) and for which the remainder of the content string has valid incomplete PLD-PLU semantics (9), or (d) a non-empty string that does not begin with PLU or PLD functions (11,12) and for which the remainder of the content string has valid PLU-PLD function sequence semantics (13).

Definition 6.70

1 $\forall v$
2 $(_0$IsValidPLUD(v) *iff*
3 IsEmptyCat$^{1.6}(v)$ *or*
4 $\exists t$
5 $(_1 v = [\rightarrow [\rightarrow$ 'PLU' $\rightarrow] \rightarrow]//t$ *and*
6 IsIncompletePLU$^{6.71}(t)_1)$ *or*
7 $\exists t$
8 $(_2 v = [\rightarrow [\rightarrow$ 'PLD' $\rightarrow] \rightarrow]//t$ *and*
9 IsIncompletePLD$^{6.72}(t)_2)$ *or*
10 $\exists t, x$
11 $(_3$ *not* $(v = [\rightarrow [\rightarrow x \rightarrow] \rightarrow]//t$ *and* $x \in$ ['PLU'; 'PLD']) *and*
12 IsNeCat$^{1.3}(v)$ *and*
13 IsValidPLUD$^{6.70}$(TAIL$^{1.15}(v))_3)_0)$

Semiformal Description 6.71

Predicate "is incomplete PLU" (clause 13.1.6)

A character content string with valid incomplete PLU-PLD semantics is either: (a) a string that begins with a PLD function (4) and for which the remainder of the content string has valid PLU-PLD function sequence semantics (5), or (b) a non-empty string that does not begin with a PLU, PLD or LF function (7-8) and for which the remainder of the content string has valid incomplete PLU-PLD semantics (9).

Definition 6.71

1 $\forall v$
2 $(_0$IsIncompletePLU(v) *iff*
3 $\exists t$
4 $(_1 v = [\rightarrow [\rightarrow$ 'PLD' $\rightarrow] \rightarrow]//t$ *and*
5 IsValidPLUD$^{6.70}(t)_1)$ *or*
6 $\exists t, x$
7 $(_2$ *not* $(v = [\rightarrow [\rightarrow x \rightarrow] \rightarrow]//t$ *and* $x \in$ ['PLU'; 'PLD'; 'LF']) *and*
8 IsNeCat$^{1.3}(v)$ *and*
9 IsIncompletePLU$^{6.71}$(TAIL$^{1.15}(v))_2)_0)$

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 6.72

Predicate "is incomplete PLD" (clause 13.1.5)

A character content string with valid incomplete PLD-PLU semantics is either: (a) a string that begins with a PLU function (4) and for which the remainder of the content string has valid PLU-PLD function sequence semantics (5), or (b) a non-empty string that does not begin with a PLU, PLD or LF function (7-8) and for which the remainder of the content string has valid incomplete PLD-PLU semantics (9).

Definition 6.72

```
1   ∀v
2   (0 IsIncompletePLD(v) iff
3     ∃t
4     (1 v = [ → [ → 'PLU' → ] → ]//t and
5       IsValidPLUD^6.70(t)1) or
6     ∃t, x
7     (2 not (v = [ → [ → x → ] → ]//t and x ∈ ['PLU';'PLD';'LF']) and
8           IsNeCat^1.3(v) and
9           IsIncompletePLD^6.72(TAIL^1.15(v))2)0)
```

Semiformal Description 6.73

Predicate "is a valid SRS" (clause 13.1.11)

A character content string with valid SRS function sequence semantics is: (a) an empty string (3), or (b) a string that begins with an SRS-1 function (5) and for which the remainder of the content string has valid ending SRS semantics (6), or (c) a non-empty string that does not begin with an SRS-1 or SRS-0 function (8) and for which the remainder of the content string has valid SRS function sequence semantics (10).

Definition 6.73

```
1   ∀v
2   (0 IsValidSRS(v) iff
3     IsEmptyCat^1.6(v) or
4     ∃t
5     (1 v = [ → [ → 'SRS' → 1 → ] → ]//t and
6       IsValidSRSEnd^6.74(t)1) or
7     ∃t, n
8     (2 not (v = [ → [ → 'SRS' → n → ] → ]//t and n ∈ [0;1]) and
9           IsNeCat^1.3(v) and
10          IsValidSRS^6.73(TAIL^1.15(v))2)0)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 6.74

Predicate "is a valid SRS end" (clause 13.1.11)

A character content string with valid ending SRS semantics is: (a) a string that begins with an SRS-0 function which terminates the sequence (4) and for which the remainder of the content string has valid SRS function sequence semantics (5), or (b) a string that begins with an SRS-1 function, which indicates the start of a nested SRS function sequence (7) and for which the remainder of the content string has two parts, the first of which has valid nested SRS function sequence semantics (8) and the remainder of which has valid ending SRS semantics (9), or (c) a non-empty string that does not begin with an SRS-1, SRS-0, or CR LF function pair (10-13) and for which the remainder of the content string has valid ending SRS semantics (14).

Definition 6.74

1 $\forall v$
2 $({}_0\text{IsValidSRSEnd}(v)$ *iff*
3 $\exists t$
4 $({}_1 v = [\rightarrow [\rightarrow \text{'SRS'} \rightarrow 0 \rightarrow] \rightarrow]//t$ *and*
5 $\text{IsValidSRS}^{6.73}(t)_1)$ *or*
6 $\exists s, t$
7 $({}_2 v = [\rightarrow [\rightarrow \text{'SRS'} \rightarrow 1 \rightarrow] \rightarrow]//s$ *and*
8 $\text{IsValidSRSNest}^{6.75}(s, t)$ *and*
9 $\text{IsValidSRSEnd}^{6.74}(t)_2)$ *or*
10 $\exists t, n$
11 $({}_3 v \neq [\rightarrow [\rightarrow \text{'CR'} \rightarrow] \rightarrow [\rightarrow \text{'LF'} \rightarrow] \rightarrow]//t$ *and*
12 $v \neq [\rightarrow [\rightarrow \text{'SRS'} \rightarrow n \rightarrow] \rightarrow]//t$ *and* $n \in [0; 1]$ *and*
13 $\text{IsNeCat}^{1.3}(v)$ *and*
14 $\text{IsValidSRSEnd}^{6.74}(\text{TAIL}^{1.15}(v))_3)_0)$

Semiformal Description 6.75

Predicate "is a valid SRS nest" (clause 13.1.11)

A character content string and substring with valid nested SRS function sequence semantics are: (a) a string that begins with an SRS-0 function which terminates the nested sequence and for which the substring remains (4), or (b) a string that begins with an SRS-1 function, which indicates the start of a nested SRS function sequence (6) and for which the remainder of the content string has two parts, the first of which has valid nested SRS function sequence semantics (7) and the remainder is the substring (8), or (c) a non-empty string that does not begin with an SRS-1, SRS-0, or CR LF function pair (10-12) and for which the remainder of the content string has valid nested SRS function sequence semantics with the substring as remainder (13).

Definition 6.75

1 $\forall v, v_1$
2 $({}_0\text{IsValidSRSNest}(v, v_1)$ *iff*
3 $\exists t$
4 $({}_1 v = [\rightarrow [\rightarrow \text{'SRS'} \rightarrow 0 \rightarrow] \rightarrow]//t$ *and* $t = v_{1\,1})$ *or*
5 $\exists s, t$
6 $({}_2 v = [\rightarrow [\rightarrow \text{'SRS'} \rightarrow 1 \rightarrow] \rightarrow]//s$ *and*
7 $\text{IsValidSRSNest}^{6.75}(s, t)$ *and*
8 $\text{IsValidSRSNest}^{6.75}(t, v_1)_2)$ *or*
9 $\exists t, n$
10 $({}_3 v \neq [\rightarrow [\rightarrow \text{'CR'} \rightarrow] \rightarrow [\rightarrow \text{'LF'} \rightarrow] \rightarrow]//t$ *and*
11 $v \neq [\rightarrow [\rightarrow \text{'SRS'} \rightarrow n \rightarrow] \rightarrow]//t$ *and* $n \in [0; 1]$ *and*
12 $\text{IsNeCat}^{1.3}(v)$ *and*
13 $\text{IsValidSRSNest}^{6.75}(\text{TAIL}^{1.15}(v), v_1)_3)_0)$

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 6.76

Predicate "is a valid PLUD in SRS" (clause 13.1.11)

A character content string with valid PLU-PLD in SRS function sequence semantics is: (a) an empty string (3), or (b) a string that begins with an SRS-1 function (5) and for which the remainder of the content string has PLU-PLD in ending SRS semantics (6), or (c) a non-empty string that does not begin with an SRS-1 function (8) and for which the remainder of the content string has valid PLU-PLD in SRS function sequence semantics (9).

Definition 6.76

1 $\forall v$
2 $(_0$IsValidPLUDInSRS(v) *iff*
3 IsEmptyCat$^{1.6}(v)$ *or*
4 $\exists t$
5 $(_1 v = [\rightarrow [\rightarrow$ 'SRS' $\rightarrow 1 \rightarrow] \rightarrow]//t$ *and*
6 IsValidPLUDInSRSEnd$^{6.77}(t)_1)$ *or*
7 $\exists t$
8 $(_2 v \neq [\rightarrow [\rightarrow$ 'SRS' $\rightarrow 1 \rightarrow] \rightarrow]//t$ *and* IsNeCat$^{1.3}(v)$ *and*
9 IsValidPLUDInSRS$^{6.76}$(TAIL$^{1.15}(v))_2)_0)$

Semiformal Description 6.77

Predicate "is a valid PLUD in SRS end" (clause 13.1.11)

A character content string with valid PLU-PLD in ending SRS semantics is: (a) a string that begins with an SRS-0 function (4) and for which the remainder of the content string has valid PLU-PLD in SRS function sequeunce semantics (5), or (b) a string that begins with an SRS-1 function (7) and for which the remainder of the string has valid PLU-PLD in ending SRS semantics (8), or (c) a string that begins with a PLU function (10) and for which the remainder of the content string has valid incomplete PLU-PLD in SRS semantics (11), or (d) a string that begins with a PLD function (13) and for which the remainder of the content string has valid incomplete PLD-PLU in SRS semantics (14), or (e) a non-empty string that does not begin with PLU, PLD, SRS-0, or SRS-1 functions (16-18) and for which the remainder of the content string has valid PLU-PLD in ending SRS semantics (19).

Definition 6.77

1 $\forall v$
2 $(_0$IsValidPLUDInSRSEnd(v) *iff*
3 $\exists t$
4 $(_1 v = [\rightarrow [\rightarrow$ 'SRS' $\rightarrow 0 \rightarrow] \rightarrow]//t$ *and*
5 IsValidPLUDInSRS$^{6.76}(t)_1)$ *or*
6 $\exists t$
7 $(_2 v = [\rightarrow [\rightarrow$ 'SRS' $\rightarrow 1 \rightarrow] \rightarrow]//t$ *and*
8 IsValidPLUDInSRSEnd$^{6.77}(t)_2)$ *or*
9 $\exists t$
10 $(_3 v = [\rightarrow [\rightarrow$ 'PLU' $\rightarrow] \rightarrow]//t$ *and*
11 IsIncompletePLUInSRS$^{6.78}(t)_3)$ *or*
12 $\exists t$
13 $(_4 v = [\rightarrow [\rightarrow$ 'PLD' $\rightarrow] \rightarrow]//t$ *and*
14 IsIncompletePLDInSRS$^{6.79}(t)_4)$ *or*
15 $\exists t, x, n$
16 $(_5 v \neq [\rightarrow [\rightarrow x \rightarrow] \rightarrow]//t$ *and* $x \in$ ['PLU';'PLD'] *and*
17 $v \neq [\rightarrow [\rightarrow$ 'SRS' $\rightarrow n \rightarrow] \rightarrow]//t$ *and* $n \in [0;1]$ *and*
18 IsNeCat$^{1.3}(v)$ *and*
19 IsValidPLUDInSRSEnd$^{6.77}$(TAIL$^{1.15}(v))_5)_0)$

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 6.78

Predicate "is incomplete PLU in SRS" (clause 13.1.11)

A character content string with valid incomplete PLU-PLD in SRS semantics is either: (a) a string that begins with a PLD function (4) and for which the remainder of the content string has valid PLU-PLD in ending SRS semantics (5), or (b) a non-empty string that does not begin with PLD or SRS-0 (7,8) and for which the remainder of the content string has valid incomplete PLU-PLD in SRS semantics (9).

Definition 6.78

```
1  ∀v
2  (0 IsIncompletePLUInSRS(v) iff
3    ∃t
4    (1 v = [ → [ → 'PLD' → ] → ]//t and
5      IsValidPLUDInSRSEnd^6.77(t)1) or
6    ∃t
7    (2 v ≠ [ → [ → 'SRS' → 0 → ] → ]//t and
8      v ≠ [ → [ → 'PLD' → ] → ]//t and IsNeCat^1.3(v) and
9      IsIncompletePLUInSRS^6.78(TAIL^1.15(v))2)0)
```

Semiformal Description 6.79

Predicate "is incomplete PLD in SRS" (clause 13.1.11)

A character content string with valid incomplete PLD-PLU in SRS semantics is either: (a) a string that begins with a PLU function (4) and for which the remainder of the content string has valid PLU-PLD in ending SRS semantics (5), or (b) a non-empty string that does not begin with PLU or SRS-0 (7,8) and for which the remainder of the content string has valid incomplete PLD-PLU in SRS semantics (9).

Definition 6.79

```
1  ∀v
2  (0 IsIncompletePLDInSRS(v) iff
3    ∃t
4    (1 v = [ → [ → 'PLU' → ] → ]//t and
5      IsValidPLUDInSRSEnd^6.77(t)1) or
6    ∃t
7    (2 v ≠ [ → [ → 'SRS' → 0 → ] → ]//t and
8      v ≠ [ → [ → 'PLU' → ] → ]//t and IsNeCat^1.3(v) and
9      IsIncompletePLDInSRS^6.79(TAIL^1.15(v))2)0)
```

Semiformal Description 6.80

Predicate "is a valid VP" (clauses 13.1.15 and 13.1.16)

A character content string with valid VPB-VPR function sequence semantics is a string in which the sum of the VP changes is zero.

Definition 6.80

```
1  ∀v
2  (0 IsValidVP(v) iff
3    IsValidVPSum^6.81(v, 0)0)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 6.81

Predicate "is a valid VP sum" (clause 13.1.15)

n is a valid sum of VP changes for a character content string v if v and n satisfy: (a) v is an empty string and $n = 0$ (3, 4); or (b) v begins with VPB-i, where i is a natural number and $n + i$ is a valid sum of VP changes for the remainder of v (5–7); or (c) v begins with VPR-i, where i is a natural number and $n - i$ is a valid sum of VP changes for the remainder of v (8–10); or (d) v begins with the control functions CR-LF and $n = 0$ and zero is a valid sum of VP changes for the remainder of v (11-13); or (e) v is non-empty and does not begin with VPB, VPR or CR-LF, and n is a valid sum of VP changes for the remainder of v (14–18).

Definition 6.81

```
1   ∀v, n
2   (0 IsValidVPSum(v, n) iff
3     (1 IsInt(n) and
4       (2 IsEmptyCat^1.6(v) and n = 0 2) or
5       ∃t, i
6       (3 v = [ → [ → 'VPB' → i → ] → ]//t and IsNat(i) and
7         IsValidVPSum^6.81(t, n + i) 3) or
8       ∃t, i
9       (4 v = [ → [ → 'VPR' → i → ] → ]//t and IsNat(i) and
10        IsValidVPSum^6.81(t, n − i) 4) or
11      ∃t
12      (5 v = [ → [ → 'CR' → ] → [ → 'LF' → ] → ]//t and n = 0 and
13        IsValidVPSum^6.81(t, 0) 5) or
14      ∃t, i, x
15      (6 v ≠ [ → [ → x → i → ] → ]//t and x ∈ ['VPB'; 'VPR'] and
16        v ≠ [ → [ → 'CR' → ] → [ → 'LF' → ] → ]//t and
17        IsNat(i) and IsNeCat^1.3(v) and
18        IsValidVPSum^6.81(TAIL^1.15(v), n) 6) 1) 0)
```

Semiformal Description 6.82

Predicate "is a valid PTX" (clause 13.3.3)

A character content string with valid PTX function sequence semantics is: (a) an empty string (3), or (b) a string that begins with a PTX-1 function (5) and for which the remainder of the content string has valid next stage PTX semantics (6), or (c) a non-empty string that does not begin with PTX-0, PTX-1 or PTX-3 (8,9) and for which the remainder of the content string has valid PTX function sequence semantics (10).

Definition 6.82

```
1   ∀v
2   (0 IsValidPTX(v) iff
3     IsEmptyCat^1.6(v) or
4     ∃t
5     (1 v = [ → [ → 'PTX' → 1 → ] → ]//t and
6       IsValidPTXNext^6.83(t) 1) or
7     ∃t, n
8     (2 not (v = [ → [ → 'PTX' → n → ] → ]//t and n ∈ [0; 1; 3]) and
9           IsNeCat^1.3(v) and
10          IsValidPTX^6.82(TAIL^1.15(v)) 2) 0)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 6.83

Predicate "is valid PTX next" (clause 13.3.3)

A character content string with valid next stage PTX semantics is either: (a) a string that begins with a PTX-3 function (4) and for which the remainder of the content string has valid ending PTX semantics (5), or (b) a non-empty string that does not begin with PTX-0, PTX-1 or PTX-3 (7,8) and for which the remainder of the content string has valid next stage PTX semantics (9).

Definition 6.83

$\forall v$
$(_0$IsValidPTXNext(v) *iff*
  $\exists t$
  $(_1 v = [\rightarrow [\rightarrow$ 'PTX' $\rightarrow 3 \rightarrow] \rightarrow]//t$ *and*
    IsValidPTXEnd$^{6.84}(t)_1)$ *or*
  $\exists t, n$
  $(_2$ *not* $(v = [\rightarrow [\rightarrow$ 'PTX' $\rightarrow n \rightarrow] \rightarrow]//t$ *and* $n \in [0;1;3])$ *and*
      IsNeCat$^{1.3}(v)$ *and*
      IsValidPTXNext$^{6.83}$(TAIL$^{1.15}(v))_2)_0)$

Semiformal Description 6.84

Predicate "is a valid PTX end" (clause 13.3.3)

A character content string with valid ending PTX semantics is either: (a) a string that begins with a PTX-0 function (4) and for which the remainder of the content string has valid ending PTX semantics (5), or (b) a non-empty string that does not begin with PTX-0, PTX-1 or PTX-3 (7,8) and for which the remainder of the content string has valid ending PTX semantics (9).

Definition 6.84

$\forall v$
$(_0$IsValidPTXEnd(v) *iff*
  $\exists t$
  $(_1 v = [\rightarrow [\rightarrow$ 'PTX' $\rightarrow 0 \rightarrow] \rightarrow]//t$ *and*
    IsValidPTX$^{6.82}(t)_1)$ *or*
  $\exists t, n$
  $(_2$ *not* $(v = [\rightarrow [\rightarrow$ 'PTX' $\rightarrow n \rightarrow] \rightarrow]//t$ *and* $n \in [0;1;3])$ *and*
      IsNeCat$^{1.3}(v)$ *and*
      IsValidPTXEnd$^{6.84}$(TAIL$^{1.15}(v))_2)_0)$

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 6.85

Predicate "is a valid SOS" (clause 13.4.1)

A character content string with valid SOS function sequence semantics is: (a) an empty string (3), or (b) a string that begins with an SOS function (5) and for which the remainder of the content string has valid ending SOS semantics (6), or (c) a non-empty string that does not begin with an SOS function (8) and for which the remainder of the content string has valid SOS function sequence semantics (9).

Definition 6.85

```
1   ∀ v
2   (0 IsValidSOS(v) iff
3     IsEmptyCat^1.6(v) or
4     ∃ t
5     (1 v = [ → [ → 'SOS' → ] → ]//t and
6       IsValidSOSEnd^6.86(t)1) or
7     ∃ t
8     (2 v ≠ [ → [ → 'SOS' → ] → ]//t and IsNeCat^1.3(v) and
9       IsValidSOS^6.85(TAIL^1.15(v))2)0)
```

Semiformal Description 6.86

Predicate "is a valid SOS end" (clause 13.4.1)

A character content string with valid ending SOS semantics is: (a) a string that begins with an ST function which terminates the sequence (4) and for which the remainder of the content string has valid SOS function sequence semantics (5), or (b) a string that begins with an SOS function, which indicates the start of a nested SOS function sequence and for which the remainder of the content string has two parts, the first of which has valid nested SOS function sequence semantics (8) and the remainder of which ending SOS semantics (9), or (c) a non-empty string that does not begin with an SOS or ST function (11,12) and for which the remainder of the content string has valid ending SOS semantics (13).

Definition 6.86

```
1   ∀ v
2   (0 IsValidSOSEnd(v) iff
3     ∃ t
4     (1 v = [ → [ → 'ST' → ] → ]//t and
5       IsValidSOS^6.85(t)1) or
6     ∃ s, t
7     (2 v = [ → [ → 'SOS' → ] → ]//s and
8       IsValidSOSNest^6.87(s, t) and
9       IsValidSOSEnd^6.86(t)2) or
10    ∃ t, a
11    (3 not (v = [ → [ → a → ] → ]//t and a ∈ ['SOS'; 'ST']) and
12          IsNeCat^1.3(v) and
13          IsValidSOSEnd^6.86(TAIL^1.15(v))3)0)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 6.87

Predicate "is a valid SOS nest" (clause 13.4.1)

A character content string and substring with valid nested SOS function sequence semantics are: (a) a string that begins with an ST function which terminates the nested sequence and for which the substring remains (4), or (b) a string that begins with an SOS function, which indicates the start of a nested SOS function sequence (6) and for which the remainder of the content string has has two parts, the first of which has valid nested SOS function sequence semantics and the remainder is the substring (7,8), or (c) a non-empty string that does not begin with an SOS or ST function (10,11) and for which the remainder of the content string has valid nested SOS function sequence semantics with the substring as remainder (12).

Definition 6.87

1 $\forall v, v_1$
2 $(_0$ IsValidSOSNest(v, v_1) *iff*
3 $\exists t$
4 $(_1 v = [\rightarrow [\rightarrow$ 'ST' $\rightarrow] \rightarrow]//t$ *and* $t = v_{1\,1})$ *or*
5 $\exists s, t$
6 $(_2 v = [\rightarrow [\rightarrow$ 'SOS' $\rightarrow] \rightarrow]//s$ *and*
7 IsValidSOSNest$^{6.87}(s, t)$ *and*
8 IsValidSOSNest$^{6.87}(t, v_1)_2)$ *or*
9 $\exists t, a$
10 $(_3$ *not* $(v = [\rightarrow [\rightarrow a \rightarrow] \rightarrow]//t$ *and* $a \in$ ['SOS'; 'ST']) *and*
11 IsNeCat$^{1.3}(v)$ *and*
12 IsValidSOSNest$^{6.87}$(TAIL$^{1.15}(v), v_1)_3)_0)$

Semiformal Description 6.88

Predicate "is a valid VP in SRS" (clause 13.1.11)

A character content string with valid VPB-VPR in SRS function sequence semantics is a string in which the sum of the VP changes in an SRS sequence is zero.

Definition 6.88

1 $\forall v$
2 $(_0$ IsValidVPInSRS(v) *iff*
3 IsValidVPSumInSRS$^{6.89}(v, 0)_0)$

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 6.89

Predicate "is a valid VP sum in SRS" (clause 13.1.11)

n is a valid sum of VP changes in SRS for a character content string v if v and n satisfy: (a) v is an empty string and $n = 0$ (3, 4); or (b) v begins with SRS-1 and n is a valid sum of VP changes in SRS end for the remainder of v (5–7); or (b) v is a non-empty string which does not begin with SRS-1 and n is a valid sum of VP changes in SRS for the remainder of v (8–10);

Definition 6.89

```
1   ∀v, n
2   (0 IsValidVPSumInSRS(v, n) iff
3     (1 IsInt(n) and
4       (2 IsEmptyCat^1.6(v) and n = 0 2) or
5       ∃t
6       (3 v = [ → [ → 'SRS' → 1 → ] → ]//t and
7         IsValidVPSumInSRSEnd^6.90(t, n) 3) or
8       ∃t
9       (4 v ≠ [ → [ → 'SRS' → 1 → ] → ]//t and IsNeCat^1.3(v) and
10        IsValidVPSumInSRS^6.89(TAIL^1.15(v), n) 4) 1) 0)
```

Semiformal Description 6.90

Predicate "is a valid VP sum in SRS end" (clause 13.1.11)

n is a valid sum of VP changes in SRS end for a character content string v if v and n satisfy: (a) v begins with VPB-i, where i is a natural number and $n + i$ is a valid sum of VP changes in SRS end for the remainder of v (4–6); or (b) v begins with VPR-i, where i is a natural number and $n - i$ is a valid sum of VP changes in SRS end for the remainder of v (7–9); or (c) v begins with SRS-0 and $n = 0$ and zero is a valid sum of VP changes in SRS end for the remainder of v (10-12); or (d) v is non-empty and does not begin with VPB, VPR or SRS with a positive parameter, and n is a valid sum of VP changes in SRS end for the remainder of v (13–16).

Definition 6.90

```
1   ∀v, n
2   (0 IsValidVPSumInSRSEnd(v, n) iff
3     (1 IsInt(n) and
4       ∃t, i
5       (2 v = [ → [ → 'VPB' → i → ] → ]//t and IsNat(i) and
6         IsValidVPSumInSRSEnd^6.90(t, n + i) 2) or
7       ∃t, i
8       (3 v = [ → [ → 'VPR' → i → ] → ]//t and IsNat(i) and
9         IsValidVPSumInSRSEnd^6.90(t, n − i) 3) or
10      ∃t
11      (4 v = [ → [ → 'SRS' → 0 → ] → ]//t and n = 0 and
12        IsValidVPSumInSRSEnd^6.90(t, 0) 4) or
13      ∃t, i, x
14      (5 not (v = [ → [ → x → i → ] → ]//t and x ∈ ['VPB'; 'VPR'; 'SRS']) and
15            IsNat(i) and IsNeCat^1.3(v) and
16            IsValidVPSumInSRSEnd^6.90(TAIL^1.15(v), n) 5) 1) 0)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 6.91

Predicate "is code extension control function element" (clause 13)

An element is a code extension control function if it is a code extension control function for the designation and invocation of graphic character sets and control functions.

Definition 6.91

1 $\forall v$
2 $(_0$ IsCodeExtensionControlFn(v) *iff*
3 IsISO2022String$^{6.92}(v)$ *or* IsISO6429String$^{6.94}(v)_0$)

Semiformal Description 6.92

Predicate "is ISO 2022 string" (clause 13)

An ISO 2022 string is a catenation whose elements are ISO 2022 code extension control function characters which are used for the designation and invocation of graphic character sets.

Definition 6.92

1 $\forall v$
2 $(_0$ IsISO2022String(v) *iff*
3 IsNeCat$^{1.3}(v)$ *and* $\forall a \hat{\in} \hat{} v \bullet$ (IsISO2022Character$^{6.93}$(C a))$_0$)

Semiformal Description 6.93

Predicate "is ISO 2022 character" (clause 13)

An ISO 2022 character is an element of ISO 2022 and is thus considered an atomic construct in the formal specification.

Definition 6.93

1 $\forall v$
2 $(_0$ IsISO2022Character(v) *iff*
3 IsAtom$(v)_0$)

Semiformal Description 6.94

Predicate "is ISO 6429 string" (clause 13)

An ISO 6429 string is a catenation whose elements are ISO 6429 code extension control function characters which are used for the designation and invocation of graphic character sets.

Definition 6.94

1 $\forall v$
2 $(_0$ IsISO6429String(v) *iff*
3 IsNeCat$^{1.3}(v)$ *and* $\forall a \hat{\in} \hat{} v \bullet$ (IsISO6429Character$^{6.95}$(C a))$_0$)

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 6.95

Predicate "is ISO 6429 character" (clause 13)

An ISO 6429 character is an element of ISO 6429 and is thus considered an atomic construct in the formal specification.

Definition 6.95

1 $\forall v$
2 $({}_0$IsISO6429Character(v) *iff*
3 IsAtom$(v)_0)$

Semiformal Description 6.96

Predicate "is shared control function" (clause 13.1)

A shared control function is a single shared control function with a parameter or without a parameter.

Definition 6.96

1 $\forall v$
2 $({}_0$IsSharedControlFn(v) *iff*
3 $\exists n$
4 $({}_1 v = [\rightarrow n \rightarrow]$ *and*
5 IsParameterlessSharedControlFn$^{6.97}(n)_1)$ *or*
6 $\exists n, p$
7 $({}_2 v = [\rightarrow n \rightarrow p \rightarrow]$ *and*
8 IsParameterizedSharedControlFn$^{6.99}(n,p)_2)_0)$

Semiformal Description 6.97

Predicate "is parameterless shared control function" (clause 13)

An element is a parameterless shared control function if it is one of the control functions: 'CR', 'LF', 'PLD', 'PLU' and 'SUB'.

Definition 6.97

1 $\forall v$
2 $({}_0$IsParameterlessSharedControlFn(v) *iff*
3 $v \in$ ['CR';'LF';'PLD';'PLU';'SUB']$_0)$

Semiformal Description 6.98

Predicate "is parameterized shared control function name" (clause 13)

A parameterized shared control function name is one of the control function names: 'GCC', 'IGS', 'SCS', 'SGR', 'SHS', 'SLS', 'SRS', 'STAB', 'SVS', 'VPB' and 'VPR'.

Definition 6.98

1 $\forall v$
2 $({}_0$IsParameterizedSharedControlFnName(v) *iff*
3 $v \in$ ['GCC';'IGS';'SCS';'SGR';'SHS';'SLS';'SRS';'STAB';'SVS';'VPB';'VPR']$_0)$

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 6.99

Predicate "is parameterized shared control function" (clause 13.1)

A parameterized shared control function comprises the name of a shared control function and a parameter value valid for that function.

Definition 6.99

1 $\forall n, p$
2 $(_0$IsParameterizedSharedControlFn(n, p) *iff*
3 IsParameterizedSharedControlFnName$^{6.98}(n)$ *and*
4 n = 'GCC' *impl* $p \in [0; 1; 2]$ *and*
5 n = 'SGR' *impl* $(_1$IsNnInt$^{1.7}(p)$ *and* $p \leq 50$ *and* $p \notin [8; 20; 28; 39; 49]_1)$ *and*
6 n = 'SHS' *impl* $p \in [0; 1; 2; 3; 4]$ *and*
7 n = 'SRS' *impl* $p \in [0; 1]$ *and*
8 $n \in$ ['IGS'; 'STAB'] *impl* IsNnInt$^{1.7}(p)$ *and*
9 n = 'SVS' *impl* $p \in [0; 1; 2; 3; 4; 9]$ *and*
10 $n \in$ ['SCS'; 'SLS'; 'VPB'; 'VPR'] *impl* IsNat$(p)_0$)

Semiformal Description 6.100

Predicate "is layout control function" (clause 13.1)

A layout control function is a single layout control function with a parameter or without a parameter.

Definition 6.100

1 $\forall v$
2 $(_0$IsLayoutControlFn(v) *iff*
3 $\exists n$
4 $(_1 v = [\rightarrow n \rightarrow]$ *and*
5 IsParameterlessLayoutControlFn$^{6.101}(n)_1)$ *or*
6 $\exists n, p$
7 $(_2 v = [\rightarrow n \rightarrow p \rightarrow]$ *and*
8 IsParameterizedLayoutControlFn$^{6.103}(n, p)_2)_0$)

Semiformal Description 6.101

Predicate "is parameterless layout control function" (clause 13)

An element is a parameterless layout control function if it is 'BS'.

Definition 6.101

1 $\forall v$
2 $(_0$IsParameterlessLayoutControlFn(v) *iff*
3 v = 'BS'$_0$)

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 6.102

Predicate "is parameterized layout control function name" (clause 13)

A parameterized layout control function name is one of the control function names 'HPB', 'HPR', 'JFY', 'SACS', 'SRCS' and 'SSW'.

Definition 6.102

```
1  ∀ v
2  (₀IsParameterizedLayoutControlFnName(v) iff
3    v ∈ ['HPB';'HPR';'JFY';'SACS';'SRCS';'SSW']₀)
```

Semiformal Description 6.103

Predicate "is a parameterized layout control function" (clause 13.2)

A parameterized layout control function comprises the name of a layout control function and a parameter value valid for that function.

Definition 6.103

```
1  ∀ n, p
2  (₀IsParameterizedLayoutControlFn(n, p) iff
3    IsParameterizedLayoutControlFnName^6.102(n) and
4    n ∈ ['HPB';'HPR';'SSW'] impl IsNat(p) and
5    n = 'JFY' impl p = 0 and
6    n ∈ ['SACS';'SRCS'] impl IsNnInt^1.7(p)₀)
```

Semiformal Description 6.104

Predicate "is logical control function" (clause 13.1)

A logical control function is a single control function with a parameter or without a parameter.

Definition 6.104

```
1  ∀ v
2  (₀IsLogicalControlFn(v) iff
3    ∃ n
4    (₁v = [ → n → ] and
5      IsParameterlessLogicalControlFn^6.105(n)₁) or
6    ∃ n, p
7    (₂v = [ → n → p → ] and
8      IsParameterizedLogicalControlFn^6.107(n, p)₂)₀)
```

Semiformal Description 6.105

Predicate "is parameterless logical control function" (clause 13.3)

A parameterless logical control function is one of the control functions; 'BPH' and 'NBH'.

Definition 6.105

```
1  ∀ v
2  (₀IsParameterlessLogicalControlFn(v) iff
3    v ∈ ['BPH';'NBH']₀)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 6.106

Predicate "is parameterized logical control function name" (clause 13.3.3)

A parameterized logical control function name is 'PTX'.

Definition 6.106

1 $\forall v$
2 $(_0$IsParameterizedLogicalControlFnName(v) *iff*
3 $v =$ 'PTX' $_0)$

Semiformal Description 6.107

Predicate "is parameterized logical control function" (clause 13.1)

A parameterized logical control function comprises the name of a parameterized logical control function where the parameter is 0, 1 or 3 for the control function 'PTX'.

Definition 6.107

1 $\forall n, p$
2 $(_0$IsParameterizedLogicalControlFn(n, p) *iff*
3 IsParameterizedLogicalControlFnName$^{6.106}(n)$ *and*
4 $n =$ 'PTX' *impl* $p \in [0; 1; 3]_0)$

Semiformal Description 6.108

Predicate "is delimiter control function" (clause 13.4)

An element is a control function if it is one of the control functions 'SOS' and 'ST'.

Definition 6.108

1 $\forall v$
2 $(_0$IsDelimiterControlFn(v) *iff*
3 $v \in [[\rightarrow$ 'SOS' $\rightarrow]; [\rightarrow$ 'ST' $\rightarrow]]_0)$

Semiformal Description 6.109

Predicate "is parameterless space control function element" (clause 13.5)

A space control function is the single catenation 'SP'.

Definition 6.109

1 $\forall v$
2 $(_0$IsSpaceControlFn(v) *iff*
3 $v = [\rightarrow$ 'SP' $\rightarrow]_0)$

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 6.110

Predicate "is parameterless escape control function element" (clause 13)

A parameterless escape control function is the single catenation 'ESC'.

Definition 6.110

1 $\forall v$
2 $(_0$IsEscapeControlFn(v) *iff*
3 $v = [\rightarrow$ 'ESC' $\rightarrow]_0)$

Semiformal Description 6.111

Predicate "is a character content information value" (clause 10.3)

The value of the attribute 'content information' consists of processable information, formatted information, or formatted processable information.

Definition 6.111

1 $\forall v$
2 $(_0$IsCharacterContentInformationValue(v) *iff*
3 ContainsProcessableElements$^{6.48}(v)$ *or*
4 ContainsFormattedElements$^{6.47}(v)$ *or*
5 ContainsFormattedProcessableElements$^{6.49}(v)_0)$

Semiformal Description 6.112

Predicate "is a graphic character string" (clauses 6.5 and 12)

A graphic character string is the concatenation of characters from character sets whose invocation and determination are specified according to ISO 2022.

Definition 6.112

1 $\forall v$
2 $(_0$IsGraphicCharacterString(v) *iff*
3 IsNeCat$^{1.3}(v)$ *and* $\forall m \hat{\in} \hat{} v \bullet$ (IsGraphicCharacter$^{6.113}$(C $m))_0)$

Semiformal Description 6.113

Predicate "is a graphic character" (clause 12)

A graphic character is an element of a character set identified by the use of ISO 2022 and is thus considered atomic in the formal specification.

Definition 6.113

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

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

10 Formal specification of the raster graphics content architectures

This clause gives a formal specification of the raster graphics content architectures as described in part 7 of ISO/IEC 8613.

Clause 10.1 provides the interface to the document profile and its formal specification.

Clauses 10.2 and 10.3 provide the interface to the document architecture by giving a formal specification of raster graphics presentation attributes and content portion attributes that apply to raster graphics content portions.

At any time a clause number is specified in the semi-formal descriptions this refers to a clause number in ISO/IEC 8613-7.

What follows is the outline of that part of the overall formula (see clause 5) which specifies the raster graphics content architecture.

... IsDefaultableRasterGraphicsContentArchitectureAttribute$^{7.1}(att)$ ...
and ... IsRasterGraphicsCodingSpecification$^{7.2}(v)$ ...
and ... IsRasterGraphicsPresentationFeature$^{7.3}(v)$ ...
and ... SatisfiesRasterGraphicsContentArchitectureConstraints$^{7.4}(prof, doby)$ ...
and ... IsFormattedProcessableRasterGraphicsContent$^{7.5}(cont)$ ...
and ... IsFormattedRasterGraphicsContent$^{7.6}(cont)$ ...
and ... IsRasterGraphicsContentPortionDescription$^{7.7}(cont)$ ...
and ... IsRasterGraphicsContentPresentationAttribute$^{7.8}(att)$ ...
and ... IsRasterGraphicsContentCodingAttribute$^{7.9}(att)$ ...
and ... IsRasterGraphicsContentPortionAttributeSet$^{7.10}(as)$ ...
and ... IsClippingValue$^{7.11}(v)$ ...
and ... IsRasterGraphicsLineProgressionValue$^{7.12}(v)$ ...
and ... IsPelPathValue$^{7.13}(v)$ ...
and ... IsRasterGraphicsInitialOffsetValue$^{7.14}(v)$ ...
and ... IsPelTransmissionDensityValue$^{7.15}(v)$ ...
and ... IsImageDimensionsValue$^{7.16}(v)$ ...
and ... IsPelSpacingValue$^{7.17}(v)$ ...
and ... IsSpacingRatioValue$^{7.18}(v)$ ...
and ... IsRasterGraphicsContentArchitectureClassValue$^{7.19}(v)$ ...
and ... IsRasterGraphicsContentTypeOfCodingValue$^{7.20}(v)$ ...
and ... IsCompressionValue$^{7.21}(v)$ ...
and ... IsNumberOfLinesValue$^{7.22}(v)$ ...
and ... IsNumberOfPelsPerLineValue$^{7.23}(v)$ ...
and ... IsNumberOfDiscardedPelsValue$^{7.24}(v)$ ...
and ... IsRasterGraphicsContentInformationValue$^{7.25}(v)$ ...
and ... IsNumberOfLinesPerTileValue$^{7.26}(v)$ ...
and ... IsNumberOfPelsPerTileLineValue$^{7.27}(v)$ ...
and ... IsTilingOffsetValue$^{7.28}(v)$ ...
and ... IsTileTypesValue$^{7.29}(v)$ ...
and ... IsTileEncodingValue$^{7.30}(v)$ ...
and ... IsBitsPerColourComponentValue$^{7.31}(v)$ ...
and ... IsInterleavingFormatValue$^{7.32}(v)$ ...

NOTE — Other predicates or operators which are used here, but are defined in other clauses, are not listed here.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

10.1 Interface to the Document Profile

Semiformal Description 7.1

Predicate "is a defaultable raster graphics content architecture attribute"

A defaultable raster graphics content architecture attribute is one of the attributes 'bits per colour component', 'clipping', 'compression', 'image dimensions', 'initial offset', 'interleaving format', 'line progression', 'number of discarded pels', 'number of lines per tile', 'number of pels per line', 'number of pels per tile line', 'pel path', 'pel spacing', 'pel transmission density', 'spacing ratio', 'tile types', 'tiling offset' or 'type of coding' with an appropriate value.

Definition 7.1

$\forall att$
$({}_0$IsDefaultableRasterGraphicsContentArchitectureAttribute(att) *iff*
$\exists n, c$
$({}_1 att = [n : c]$ *and*
$n \in$ ['bits per colour component'; 'clipping';
'compression'; 'image dimensions';
'initial offset'; 'interleaving format';
'line progression'; 'number of discarded pels';
'number of lines per tile'; 'number of pels per line';
'number of pels per tile line'; 'pel path';
'pel spacing'; 'pel transmission density';
'spacing ratio'; 'tile types';
'tiling offset'; 'type of coding'] *and*
$({}_2 n =$ 'bits per colour component' *impl* IsBitsPerColourComponentValue$^{7.31}(c)\ {}_2$) *and*
$({}_3 n =$ 'clipping' *impl* IsClippingValue$^{7.11}(c)\ {}_3$) *and*
$({}_4 n =$ 'compression' *impl* IsCompressionValue$^{7.21}(c)\ {}_4$) *and*
$({}_5 n =$ 'image dimensions' *impl* IsImageDimensionsValue$^{7.16}(c)\ {}_5$) *and*
$({}_6 n =$ 'initial offset' *impl* IsRasterGraphicsInitialOffsetValue$^{7.14}(c)\ {}_6$) *and*
$({}_7 n =$ 'interleaving format' *impl* IsInterleavingFormatValue$^{7.32}(c)\ {}_7$) *and*
$({}_8 n =$ 'line progression' *impl* IsRasterGraphicsLineProgressionValue$^{7.12}(c)\ {}_8$) *and*
$({}_9 n =$ 'number of discarded pels' *impl* IsNumberOfDiscardedPelsValue$^{7.24}(c)\ {}_9$) *and*
$({}_{10} n =$ 'number of lines per tile' *impl* IsNumberOfLinesPerTileValue$^{7.26}(c)\ {}_{10}$) *and*
$({}_{11} n =$ 'number of pels per line' *impl* IsNumberOfPelsPerLineValue$^{7.23}(c)\ {}_{11}$) *and*
$({}_{12} n =$ 'number of pels per tile line' *impl* IsNumberOfPelsPerTileLineValue$^{7.27}(c)\ {}_{12}$) *and*
$({}_{13} n =$ 'pel path' *impl* IsPelPathValue$^{7.13}(c)\ {}_{13}$) *and*
$({}_{14} n =$ 'pel spacing' *impl* IsPelSpacingValue$^{7.17}(c)\ {}_{14}$) *and*
$({}_{15} n =$ 'pel transmission density' *impl* IsPelTransmissionDensityValue$^{7.15}(c)\ {}_{15}$) *and*
$({}_{16} n =$ 'spacing ratio' *impl* IsSpacingRatioValue$^{7.18}(c)\ {}_{16}$) *and*
$({}_{17} n =$ 'tile types' *impl* IsTileTypesValue$^{7.29}(c)\ {}_{17}$) *and*
$({}_{18} n =$ 'tiling offset' *impl* IsTilingOffsetValue$^{7.28}(c)\ {}_{18}$) *and*
$({}_{19} n =$ 'type of coding' *impl* IsRasterGraphicsContentTypeOfCodingValue$^{7.20}(c)\ {}_{19})_1)_0)$

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Semiformal Description 7.2

Predicate "is a raster graphics coding specification"

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

Definition 7.2

1 $\forall v$
2 $(_0$IsRasterGraphicsCodingSpecification(v) *iff*
3 $(_1$IsNom(v) *and*
4 $\forall b \in \hat{} v \bullet$ (IsRasterGraphicsContentCodingAttribute$^{7.9}$(C b)) $_1)_0)$

Semiformal Description 7.3

Predicate "is a raster graphics presentation feature"

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

Definition 7.3

1 $\forall v$
2 $(_0$IsRasterGraphicsPresentationFeature(v) *iff*
3 $(_1$IsNom(v) *and*
4 $\forall b \in \hat{} v \bullet$ (IsRasterGraphicsContentPresentationAttribute$^{7.8}$(C b)) $_1)_0)$

10.2 Interface to the Document Architecture

Semiformal Description 7.4

Predicate "satisfies raster graphics content architecture constraints"

A document profile *prof* and a document body *doby* satisfy the constraints imposed by the raster graphics content architecture if the following holds:

For all constituents for which the attribute 'content architecture class' is specified and has the value '2 8 2 7 2' (4), the content portion is a formatted processable raster graphics content portion (5).

For all constituents for which the attribute 'content architecture class' is specified and has the value '2 8 2 7 0' (6), the content portion is a formatted raster graphics content portion (7).

The value '2 8 2 7 2' appears in the document profile attribute 'content architecture classes' if and only if there exists a content portion for which the attribute 'content architecture class' has also this value (8, 9).

The value '2 8 2 7 0' appears in the document profile attribute 'content architecture classes' if and only if there exists a content portion for which the attribute 'content architecture class' has also this value (10. 11).

NOTE — This predicate is used in clause 7 of this part of ISO/IEC 8613.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Definition 7.4

1 $\forall prof, doby$
2 $({}_0$SatisfiesRasterGraphicsContentArchitectureConstraints$(prof, doby)$ *iff*
3 $\forall cont \in doby$
4 $({}_1({}_2$C ^$cont$ • 'content architecture class' = '2 8 2 7 2' *impl*
5 IsFormattedProcessableRasterGraphicsContent$^{7.5}(cont)_2)$ *and*
6 $({}_3$C ^$cont$ • 'content architecture class' = '2 8 2 7 0' *impl*
7 IsFormattedRasterGraphicsContent$^{7.6}(cont)_3)_1)$ *and*
8 '2 8 2 7 2' ∈ C ^$prof$ • 'content architecture classes' *iff*
9 $\exists cont \in doby$ (C ^$cont$ • 'content architecture class' = '2 8 2 7 2') *and*
10 '2 8 2 7 0' ∈ C ^$prof$ • 'content architecture classes' *iff*
11 $\exists cont \in doby$ (C ^$cont$ • 'content architecture class' = '2 8 2 7 0') $_0)$

Semiformal Description 7.5

Predicate "is formatted processable raster graphics content" (clause 6.1.2)

A formatted processable raster graphics content portion is a raster graphics content portion description. The attribute 'number of pels per line' is explicitly specified and the attribute 'number of discarded pels' is not specified.

Definition 7.5

1 $\forall cont$
2 $({}_0$IsFormattedProcessableRasterGraphicsContent$(cont)$ *iff*
3 $({}_1$ IsRasterGraphicsContentPortionDescription$^{7.7}(cont)$ *and*
4 *not* IsPlaceholder$^{1.19}$(C ^$cont$ • 'number of pels per line') *and*
5 *not* 'number of discarded pels' ∈ NAMS$^{1.18}(cont)_1)_0)$

Semiformal Description 7.6

Predicate "is formatted raster graphics content" (clause 6.1.1)

A formatted raster graphics content portion is a raster graphics content portion description. The attributes 'number of lines', 'number of lines per tile', 'number of pels per tile line', 'tile types' and 'tiling offset' are not specified.

Definition 7.6

1 $\forall cont$
2 $({}_0$IsFormattedRasterGraphicsContent$(cont)$ *iff*
3 $({}_1$ IsRasterGraphicsContentPortionDescription$^{7.7}(cont)$ *and*
4 ['number of lines'; 'number of lines per tile'; 'number of pels per tile line';
5 'tile types'; 'tiling offset'] ∩ NAMS$^{1.18}(cont) = [:]_1)_0)$

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 7.7

Predicate "is a raster graphics content portion description" (clause 9)

A raster graphics content portion description is a set of raster graphics content portion attributes, which are 'bits per colour component', 'compression', 'content information', 'interleaving format', 'number of discarded pels', 'number of lines', 'number of lines per tile', 'number of pels per line', 'number of pels per tile line', 'tile types', 'tiling offset' and 'type of coding' (6–11). The attributes 'compression', 'content information', 'number of pels per line' and 'type of coding' must be present (4, 5). If the value of the attribute 'type of coding' is '2 8 3 7 5' (tiled encoding), the attributes 'number of lines per tile', 'number of pels per tile line', 'tile types' and 'tiling offset' must also be specified (12–14). If the attribute 'tiling offset' is specified, the first component of the value of this attribute must be less than the value of the attribute 'number of pels per tile line' and the last (second) component must be less than the value of the attribute 'number of lines per tile' (15–17).

NOTE — This predicate is used in clause 7 of this part of ISO/IEC 8613.

Definition 7.7

```
1   ∀ cont
2   (0 IsRasterGraphicsContentPortionDescription(cont) iff
3     (1 IsRasterGraphicsContentPortionAttributeSet^7.10(cont) and
4       NAMS^1.18(cont) ⊇ ['compression';              'content information';
5                          'number of pels per line';  'type of coding'] and
6       NAMS^1.18(cont) ⊆ ['bits per colour component'; 'compression';
7                          'content information';      'interleaving format';
8                          'number of discarded pels'; 'number of lines';
9                          'number of lines per tile'; 'number of pels per line';
10                         'number of pels per tile line'; 'tile types';
11                         'tiling offset';            'type of coding'] and
12      (2 C ^cont • 'type of coding' = '2 8 3 7 5' impl
13        NAMS^1.18(cont) ⊇
14        ['number of lines per tile'; 'number of pels per tile line'; 'tile types'; 'tiling offset']2) and
15      (3 'tiling offset' ∈ NAMS^1.18(cont) impl
16        (4 HEAD^1.13(C ^cont • 'tiling offset') < C ^cont • 'number of pels per tile line' and
17          LASTC^1.14 (C ^cont • 'tiling offset') < C ^cont • 'number of lines per tile'4)3)1)0)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 7.8

Predicate "is a raster graphics content presentation attribute" (clause 8)

A raster graphics content presentation attribute is one of the attributes 'clipping', 'image dimensions', 'initial offset', 'line progression', 'pel path', 'pel spacing', 'pel transmission density' or 'spacing ratio' with an appropriate value.

NOTE — This predicate is used in clause 7 of this part of ISO/IEC 8613.

Definition 7.8

$\forall\, att$
$({}_0\,\text{IsRasterGraphicsContentPresentationAttribute}(att)\ \underline{iff}$
$\quad \exists\, n, c$
$\quad ({}_1\, att = [n : c]\ \underline{and}$
$\quad\quad n \in$ ['clipping'; 'image dimensions';
$\quad\quad\quad$ 'initial offset'; 'line progression';
$\quad\quad\quad$ 'pel path'; 'pel spacing';
$\quad\quad\quad$ 'pel transmission density'; 'spacing ratio'] $\underline{and}$
$\quad\quad ({}_2\, n = \text{'clipping'}\ \underline{impl}$
$\quad\quad\quad ({}_3\, \text{IsPlaceholder}^{1.19}(c)\ \underline{or}\ \text{IsClippingValue}^{7.11}(c)\ {}_3)_2)\ \underline{and}$
$\quad\quad ({}_4\, n = \text{'image dimensions'}\ \underline{impl}$
$\quad\quad\quad ({}_5\, \text{IsPlaceholder}^{1.19}(c)\ \underline{or}\ \text{IsImageDimensionsValue}^{7.16}(c)\ {}_5)_4)\ \underline{and}$
$\quad\quad ({}_6\, n = \text{'initial offset'}\ \underline{impl}$
$\quad\quad\quad ({}_7\, \text{IsPlaceholder}^{1.19}(c)\ \underline{or}\ \text{IsRasterGraphicsInitialOffsetValue}^{7.14}(c)\ {}_7)_6)\ \underline{and}$
$\quad\quad ({}_8\, n = \text{'line progression'}\ \underline{impl}$
$\quad\quad\quad ({}_9\, \text{IsPlaceholder}^{1.19}(c)\ \underline{or}\ \text{IsRasterGraphicsLineProgressionValue}^{7.12}(c)\ {}_9)_8)\ \underline{and}$
$\quad\quad ({}_{10}\, n = \text{'pel path'}\ \underline{impl}$
$\quad\quad\quad ({}_{11}\, \text{IsPlaceholder}^{1.19}(c)\ \underline{or}\ \text{IsPelPathValue}^{7.13}(c)\ {}_{11})_{10})\ \underline{and}$
$\quad\quad ({}_{12}\, n = \text{'pel spacing'}\ \underline{impl}$
$\quad\quad\quad ({}_{13}\, \text{IsPlaceholder}^{1.19}(c)\ \underline{or}\ \text{IsPelSpacingValue}^{7.17}(c)\ {}_{13})_{12})\ \underline{and}$
$\quad\quad ({}_{14}\, n = \text{'pel transmission density'}\ \underline{impl}$
$\quad\quad\quad ({}_{15}\, \text{IsPlaceholder}^{1.19}(c)\ \underline{or}\ \text{IsPelTransmissionDensityValue}^{7.15}(c)\ {}_{15})_{14})\ \underline{and}$
$\quad\quad ({}_{16}\, n = \text{'spacing ratio'}\ \underline{impl}$
$\quad\quad\quad ({}_{17}\, \text{IsPlaceholder}^{1.19}(c)\ \underline{or}\ \text{IsSpacingRatioValue}^{7.18}(c)\ {}_{17})_{16})_1)_0)$

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Semiformal Description 7.9

Predicate "is a raster graphics content coding attribute" (clauses 9.1 and 9.2)

A raster graphics content coding attribute is one of the attributes 'bits per colour component', 'compression', 'interleaving format', 'number of discarded pels', 'number of lines', 'number of lines per tile', 'number of pels per line', 'number of pels per tile line', 'tile types', 'tiling offset' and 'type of coding' with an appropriate value.

NOTE — This predicate is used in clause 7 of this part of ISO/IEC 8613.

Definition 7.9

$\forall\, att$
$(_0$ IsRasterGraphicsContentCodingAttribute(att) *iff*
$\exists\, n, c$
$(_1\, att = [n : c]$ *and*
$n \in$ ['bits per colour component'; 'compression'; 'interleaving format'; 'number of discarded pels';
'number of lines'; 'number of lines per tile'; 'number of pels per line'; 'number of pels per tile line';
'tile types'; 'tiling offset'; 'type of coding'] *and*
$(_2\, n =$ 'bits per colour component' *impl*
$(_3$ IsPlaceholder$^{1.19}(c)$ *or* IsBitsPerColourComponentValue$^{7.31}(c)\ _3)_2)$ *and*
$(_4\, n =$ 'compression' *impl*
$(_5$ IsPlaceholder$^{1.19}(c)$ *or* IsCompressionValue$^{7.21}(c)\ _5)_4)$ *and*
$(_6\, n =$ 'interleaving format' *impl*
$(_7$ IsPlaceholder$^{1.19}(c)$ *or* IsInterleavingFormatValue$^{7.32}(c)\ _7)_6)$ *and*
$(_8\, n =$ 'number of discarded pels' *impl*
$(_9$ IsPlaceholder$^{1.19}(c)$ *or* IsNumberOfDiscardedPelsValue$^{7.24}(c)\ _9)_8)$ *and*
$(_{10}\, n =$ 'number of lines' *impl* IsNumberOfLinesValue$^{7.22}(c)\ _{10})$ *and*
$(_{11}\, n =$ 'number of lines per tile' *impl*
$(_{12}$ IsPlaceholder$^{1.19}(c)$ *or* IsNumberOfLinesPerTileValue$^{7.26}(c)\ _{12})_{11})$ *and*
$(_{13}\, n =$ 'number of pels per line' *impl*
$(_{14}$ IsPlaceholder$^{1.19}(c)$ *or* IsNumberOfPelsPerLineValue$^{7.23}(c)\ _{14})_{13})$ *and*
$(_{15}\, n =$ 'number of pels per tile line' *impl*
$(_{16}$ IsPlaceholder$^{1.19}(c)$ *or* IsNumberOfPelsPerTileLineValue$^{7.27}(c)\ _{16})_{15})$ *and*
$(_{17}\, n =$ 'tile types' *impl*
$(_{18}$ IsPlaceholder$^{1.19}(c)$ *or* IsTileTypesValue$^{7.29}(c)\ _{18})_{17})$ *and*
$(_{19}\, n =$ 'tiling offset' *impl*
$(_{20}$ IsPlaceholder$^{1.19}(c)$ *or* IsTilingOffsetValue$^{7.28}(c)\ _{20})_{19})$ *and*
$(_{21}\, n =$ 'type of coding' *impl*
$(_{22}$ IsPlaceholder$^{1.19}(c)$ *or* IsRasterGraphicsContentTypeOfCodingValue$^{7.20}(c)\ _{22})_{21})_1)_0)$

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

10.3 Attributes of the Raster Graphics Content Architecture

Semiformal Description 7.10

Predicate "is a raster graphics content portion attribute set"

A raster graphics content portion attribute set contains one or more of the attributes 'bits per colour component', 'compression', 'content information', 'interleaving format', 'number of discarded pels', 'number of lines', 'number of lines per tile', 'number of pels per line', 'number of pels per tile line', 'tile types', 'tiling offset' and 'type of coding' with an appropriate value.

Definition 7.10

```
1   ∀as
2   (0 IsRasterGraphicsContentPortionAttributeSet(as) iff
3     (1 IsNeNom1.2(as) and
4       ∀a ∈̂ ^as •
5       (2(3 N a = 'bits per colour component' impl
6           (4 IsPlaceholder1.19(C a) or IsBitsPerColourComponentValue7.31(C a) 4)3) and
7         (5 N a = 'compression' impl
8           (6 IsPlaceholder1.19(C a) or IsCompressionValue7.21(C a) 6)5) and
9         (7 N a = 'content information' impl IsRasterGraphicsContentInformationValue7.25(C a) 7) and
10        (8 N a = 'interleaving format' impl
11          (9 IsPlaceholder1.19(C a) or IsInterleavingFormatValue7.32(C a) 9)8) and
12        (10 N a = 'number of discarded pels' impl
13          (11 IsPlaceholder1.19(C a) or IsNumberOfDiscardedPelsValue7.24(C a) 11)10) and
14        (12 N a = 'number of lines' impl IsNumberOfLinesValue7.22(C a) 12) and
15        (13 N a = 'number of lines per tile' impl
16          (14 IsPlaceholder1.19(c) or IsNumberOfLinesPerTileValue7.26(c) 14)13) and
17        (15 N a = 'number of pels per line' impl
18          (16 IsPlaceholder1.19(C a) or IsNumberOfPelsPerLineValue7.23(C a) 16)15) and
19        (17 N a = 'number of pels per tile line' impl
20          (18 IsPlaceholder1.19(c) or IsNumberOfPelsPerTileLineValue7.27(c) 18)17) and
21        (19 N a = 'tile types' impl
22        (20 IsPlaceholder1.19(c) or IsTileTypesValue7.29(c) 20)19) and
23        (21 N a = 'tiling offset' impl
24          (22 IsPlaceholder1.19(c) or IsTilingOffsetValue7.28(c) 22)21) and
25        (23 N a = 'type of coding' impl
26          (24 IsPlaceholder1.19(C a) or IsRasterGraphicsContentTypeOfCodingValue7.20(C a) 24)23)2)1)0)
```

Semiformal Description 7.11

Predicate "is a clipping value" (clause 8.1.1)

The value of the attribute 'clipping' is a catenation of two elements where each element is a pair of non-negative integers.

Definition 7.11

```
1   ∀v
2   (0 IsClippingValue(v) iff
3     ∃l, r
4     (1 v = [ → l → r → ] and
5       IsPairOfNnInt1.8(l) and IsPairOfNnInt1.8(r) 1)0)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 7.12

Predicate "is a raster graphics line progression value" (clause 8.1.2)

The value of the attribute 'line progression' is either '90°' or '270°'.

NOTE — Since the attribute 'line progression' is also used within the character content architectures the qualifier "RasterGraphics" has been added to the predicate name.

Definition 7.12

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

Semiformal Description 7.13

Predicate "is a pel path value" (clause 8.1.3)

The value of the attribute 'pel path' is one of '0°', '90°', '180°' or '270°'.

Definition 7.13

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

Semiformal Description 7.14

Predicate "is a raster graphics initial offset value" (clause 8.2.1)

The value of the attribute 'initial offset' is a nomination with two elements. The names are 'horizontal coordinate' and 'vertical coordinate' and the corresponding components are integers.

NOTE — Since the attribute 'initial offset' is also used within the character content architectures the qualifier "RasterGraphics" has been added to the predicate name.

Definition 7.14

1 $\forall v$
2 $(_0$IsRasterGraphicsInitialOffsetValue(v) *iff*
3 $(_1$IsNeNom$^{1.2}(v)$ *and*
4 NAMS$^{1.18}(v)$ = ['horizontal coordinate'; 'vertical coordinate'] *and*
5 $\forall b \hat{\in} \hat{}v \bullet$ (IsInt(C b)) $_1)_0$)

Semiformal Description 7.15

Predicate "is a pel transmission density value" (clause 8.2.2)

The value of the attribute pel transmission density is one of '1 BMU', '2 BMU', '3 BMU', '4 BMU', '5 BMU' or '6 BMU'.

Definition 7.15

1 $\forall v$
2 $(_0$IsPelTransmissionDensityValue(v) *iff*
3 $v \in$ ['1 BMU'; '2 BMU'; '3 BMU'; '4 BMU'; '5 BMU'; '6 BMU']$_0$)

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 7.16

Predicate "is an image dimensions value" (clause 8.3.1)

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

Definition 7.16

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

Semiformal Description 7.17

Predicate "is a pel spacing value" (clause 8.3.2)

The value of the attribute 'pel spacing' is either 'null' or a nomination with two elements. The names are 'length' and 'pel spaces' and the corresponding components are positive integers.

Definition 7.17

```
1   ∀ v
2   (0 IsPelSpacingValue(v) iff
3     (1 (v = 'null') or
4       (2 IsNeNom1.2(v) and
5         NAMS1.18(v) = ['length'; 'pel spaces'] and
6         ∀ b ∈̂ ^v • (IsNat(C b)) 2)1)0)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 7.18

Predicate "is a spacing ratio value" (clause 8.3.3)

The value of the attribute 'spacing ratio' a nomination with two elements. The names are 'line spacing value' and 'pel spacing value' and the corresponding components are positive integers.

Definition 7.18

```
1  ∀ v
2  (0 IsSpacingRatioValue(v) iff
3    (1 IsNeNom^1.2(v) and
4      NAMS^1.18(v) = ['line spacing value'; 'pel spacing value'] and
5      ∀ b ∈̂ ^v • (IsNat(C b)) 1)0)
```

Semiformal Description 7.19

Predicate "is a raster graphics content architecture class value" (clause 8.4.1)

The value of the attribute 'content architecture class' is an ASN.1 object identifier with either the value '2 8 2 7 0' or '2 8 2 7 2' for raster graphics content.

NOTE — This predicate is used in clause 7 of this part of ISO/IEC 8613.

Definition 7.19

```
1  ∀ v
2  (0 IsRasterGraphicsContentArchitectureClassValue(v) iff
3    v ∈ ['2 8 2 7 0'; '2 8 2 7 2']0)
```

Semiformal Description 7.20

Predicate "is a raster graphics content type of coding value" (clause 9.1.1)

For a raster graphics content portion the value of the attribute 'type of coding' is either 1 or an ASN.1 object identifier with one of the values '2 8 3 7 0', '2 8 3 7 1', '2 8 3 7 2', '2 8 3 7 3', '2 8 3 7 5', '2 8 3 7 9', '2 8 3 7 10' or '2 8 3 7 11'.

NOTE — This predicate is used in clause 7 of this part of ISO/IEC 8613.

Definition 7.20

```
1  ∀ v
2  (0 IsRasterGraphicsContentTypeOfCodingValue(v) iff
3    (1 v = 1 or v ∈ ['2 8 3 7 0'; '2 8 3 7 1'; '2 8 3 7 2'; '2 8 3 7 3';
4                     '2 8 3 7 5'; '2 8 3 7 9'; '2 8 3 7 10'; '2 8 3 7 11']1)0)
```

Semiformal Description 7.21

Predicate "is a compression value" (clause 9.2.1)

The value of the attribute 'compression' is either 'compressed' or 'uncompressed'.

Definition 7.21

```
1  ∀ v
2  (0 IsCompressionValue(v) iff
3    v ∈ ['compressed'; 'uncompressed']0)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 7.22

Predicate "is a number of lines value" (clause 9.2.2)

The value of the attribute **'number of lines'** is a positive integer.

Definition 7.22

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

Semiformal Description 7.23

Predicate "is a number of pels per line value" (clause 9.2.3)

The value of the attribute **'number of pels per line'** is a non-negative integer.

Definition 7.23

1 $\forall v$
2 $({}_{0}\text{IsNumberOfPelsPerLineValue}(v)$ *iff*
3 $\text{IsNnInt}^{1.7}(v)_{0})$

Semiformal Description 7.24

Predicate "is a number of discarded pels value" (clause 9.2.4)

The value of the attribute **'number of discarded pels'** is a non-negative integer.

Definition 7.24

1 $\forall v$
2 $({}_{0}\text{IsNumberOfDiscardedPelsValue}(v)$ *iff*
3 $\text{IsNnInt}^{1.7}(v)_{0})$

Semiformal Description 7.25

Predicate "is a raster graphics content information value" (clause 9.3)

For a raster graphics content portion the value of the attribute **'content information'** is an octet string, representing the pel array.

NOTE — This predicate is used in clause 7 of this part of ISO/IEC 8613.

Definition 7.25

1 $\forall v$
2 $({}_{0}\text{IsRasterGraphicsContentInformationValue}(v)$ *iff*
3 $\text{IsOctetString}^{1.10}(v)_{0})$

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 7.26

Predicate "is a number of lines per tile value" (clause 9.2.5)

For a raster graphics content portion the value of the attribute 'number of lines per tile' is a positive integer.

Definition 7.26

1 $\forall v$
2 $(_0$IsNumberOfLinesPerTileValue(v) *iff*
3 IsNat$(v)_0)$

Semiformal Description 7.27

Predicate "is a number of pels per tile line" (clause 9.2.6)

For a raster graphics content portion the value of the attribute 'number of pels per tile line' is a positive integer.

Definition 7.27

1 $\forall v$
2 $(_0$IsNumberOfPelsPerTileLineValue(v) *iff*
3 IsNat$(v)_0)$

Semiformal Description 7.28

Predicate "is a tiling offset value" (clause 9.2.7)

For a raster graphics content portion the value of the attribute 'tiling offset' is a sequence of two non-negative integers.

Definition 7.28

1 $\forall v$
2 $(_0$IsTilingOffsetValue(v) *iff*
3 $\exists a, b$ *and*
4 $(_1 v = [\rightarrow a \rightarrow b \rightarrow]$ *and* IsNnInt$^{1.7}(a)$ *and* IsNnInt$^{1.7}(b)$ $_1)_0)$

Semiformal Description 7.29

Predicate "is a tile types value" (clause 9.2.8)

For a raster graphics content portion the value of the attribute 'tile types' is a non-empty catenation of tile encoding values.

Definition 7.29

1 $\forall v$
2 $(_0$IsTileTypesValue(v) *iff*
3 $(_1$ IsNeCat$^{1.3}(v)$ *and*
4 $\forall a \hat{\in} \hat{} v \bullet$ IsTileEncodingValue$^{7.30}(a)$ $_1)_0)$

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 7.30

Predicate "is a tile encoding value" (clause 9.2.8)

A tile encoding value is one of 'null background', 'null foreground', 'bitmap encoded', 'T6 encoded', 'T4 one dimensional encoded' or 'T4 two dimensional encoded'.

Definition 7.30

```
1  ∀ v
2  (0 IsTileEncodingValue(v) iff
3    v ∈ ['null background'; 'null foreground'; 'bitmap encoded'; 'T6 encoded';
4         'T4 one dimensional encoded'; 'T4 two dimensional encoded']0)
```

Semiformal Description 7.31

Predicate "is a bits per colour component value" (clause 9.2.9)

A bits per colour component value is either a positive integer or a sequence of non-negative integers of which at least one is positive.

Definition 7.31

```
1  ∀ v
2  (0 IsBitsPerColourComponentValue(v) iff
3    IsNat(v) or
4    (1 IsNeCat^1.3(v) and
5      ∀ a ∈ v • (2 IsNnInt^1.7(a)2) and
6      ∃ a ∈ v • (3 IsNat(a)3)1)0)
```

Semiformal Description 7.32

Predicate "is an interleaving format value" (clause 9.2.10)

An interleaving format value is one of 'line', 'pel' or 'plane'.

Definition 7.32

```
1  ∀ v
2  (0 IsInterleavingFormatValue(v) iff
3    v ∈ ['line'; 'pel'; 'plane']0)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

11 Formal specification of the geometric graphics content architectures

This clause gives a formal specification of the geometric graphics content architectures as described in ISO/IEC 8613-8.

Clause 11.1 provides the interface to the document profile and its formal specification.

Clauses 11.2 and 11.3 provide the interface to the document architecture by giving a formal specification of geometric graphics presentation attributes and content portion attributes that apply to geometric graphics content portions.

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

What follows is the outline of that part of the overall formula (see clause 5) which specifies the geometric graphics content architectures.

... IsDefaultableGeometricGraphicsContentArchitectureAttribute$^{8.1}(att)$ ...
and ... IsGeometricGraphicsCodingSpecification$^{8.2}(v)$ ...
and ... IsGeometricGraphicsPresentationFeature$^{8.3}(v)$ ...
and ... SatisfiesGeometricGraphicsContentArchitectureConstraints$^{8.4}(prof, doby)$ ...
and ... IsGeometricGraphicsContentPortionDescription$^{8.5}(cont)$ ...
and ... IsGeometricGraphicsContentPresentationAttribute$^{8.6}(att)$ ...
and ... IsGeometricGraphicsContentCodingAttribute$^{8.7}(att)$ ...
and ... IsGeometricGraphicsContentPortionAttributeSet$^{8.8}(as)$ ...
and ... IsGeometricGraphicsEncodingAnnouncerValue$^{8.9}(v)$ ...
and ... IsDirectColourValuePair$^{8.10}(v)$ ...
and ... IsLineRenditionValue$^{8.11}(v)$ ...
and ... IsSetOfLineBundleSpecifications$^{8.12}(v)$ ...
and ... IsLineBundleSpecification$^{8.13}(v)$ ...
and ... IsGeometricGraphicsColourValue$^{8.14}(v)$ ...
and ... IsWidthValue$^{8.15}(v)$ ...
and ... IsSpecificationModeValue$^{8.16}(v)$ ...
and ... IsLineTypeValue$^{8.17}(v)$ ...
and ... IsMarkerRenditionValue$^{8.18}(v)$ ...
and ... IsSetOfMarkerBundleSpecifications$^{8.19}(v)$ ...
and ... IsMarkerBundleSpecification$^{8.20}(v)$ ...
and ... IsMarkerTypeValue$^{8.21}(v)$ ...
and ... IsTextRenditionValue$^{8.22}(v)$ ...
and ... IsCharacterSetList$^{8.23}(v)$ ...
and ... IsCharacterSetSpecification$^{8.24}(v)$ ...
and ... IsSetOfTextBundleSpecifications$^{8.25}(v)$ ...
and ... IsRegisteredDesignationSequenceTail$^{8.26}(v)$ ...
and ... IsTextBundleSpecification$^{8.27}(v)$ ...
and ... IsOrientationVectorPair$^{8.28}(v)$ ...
and ... IsFontListSet$^{8.29}(v)$ ...
and ... IsTextAlignmentSpecification$^{8.30}(v)$ ...
and ... IsFilledAreaRenditionValue$^{8.31}(v)$ ...
and ... IsSetOfFillBundleSpecifications$^{8.32}(v)$ ...
and ... IsFillBundleSpecification$^{8.33}(v)$ ...
and ... IsSetOfPatternTableSpecifications$^{8.34}(v)$ ...
and ... IsPatternTableSpecification$^{8.35}(v)$ ...
and ... IsSeqOfDirectColourValue$^{8.36}(v)$ ...
and ... IsHatchIndexValue$^{8.37}(v)$ ...
and ... IsPatternSizeValue$^{8.38}(v)$ ...
and ... IsEdgeRenditionValue$^{8.39}(v)$ ...
and ... IsSetOfEdgeBundleSpecifications$^{8.40}(v)$ ...
and ... IsEdgeBundleSpecification$^{8.41}(v)$ ...

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

and ... IsEdgeTypeValue$^{8.42}(v)$ ...
and ... IsColourRepresentationsValue$^{8.43}(v)$ ...
and ... IsSetOfColourTableSpecifications$^{8.44}(v)$ ...
and ... IsColourTableSpecification$^{8.45}(v)$ ...
and ... IsTransparencySpecificationValue$^{8.46}(v)$ ...
and ... IsTransformationSpecificationValue$^{8.47}(v)$ ...
and ... IsRegionOfInterestSpecificationValue$^{8.48}(v)$ ...
and ... IsPictureOrientationValue$^{8.49}(v)$ ...
and ... IsPictureDimensionsValue$^{8.50}(v)$ ...
and ... IsGeometricGraphicsContentArchitectureClassValue$^{8.51}(v)$ ...
and ... IsGeometricGraphicsContentTypeOfCodingValue$^{8.52}(v)$ ...
and ... IsGeometricGraphicsContentInformationValue$^{8.53}(v)$ ...
and ... IsDirectColourValue$^{8.54}(v)$ ...
and ... IsVDCPair$^{8.55}(v)$ ...
and ... IsVDCValue$^{8.56}(v)$ ...
and ... IsNnVDCValue$^{8.57}(v)$ ...
and ... IsRegisteredLineType$^{8.58}(v)$ ...
and ... IsRegisteredMarkerType$^{8.59}(v)$ ...
and ... IsRegisteredHatchIndex$^{8.60}(v)$ ...
and ... IsRegisteredEdgeType$^{8.61}(v)$ ...

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

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

11.1 Interface to the Document Profile

Semiformal Description 8.1

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

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

Definition 8.1

1 $\forall att$
2 $(_0$ IsDefaultableGeometricGraphicsContentArchitectureAttribute(att) *iff*
3 IsGeometricGraphicsContentPresentationAttribute$^{8.6}$(att) $_0)$

Semiformal Description 8.2

Predicate "is a profile geometric graphics coding specification"

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

Definition 8.2

1 $\forall v$
2 $(_0$ IsGeometricGraphicsCodingSpecification(v) *iff*
3 IsNom(v) *and*
4 $\forall b \hat{\in} \hat{} v \bullet$ (IsGeometricGraphicsContentCodingAttribute$^{8.7}$(C b)) $_0)$

Semiformal Description 8.3

Predicate "is a profile geometric graphics presentation feature"

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

Definition 8.3

1 $\forall v$
2 $(_0$ IsGeometricGraphicsPresentationFeature(v) *iff*
3 IsNom(v) *and*
4 $\forall b \hat{\in} \hat{} v \bullet$ (IsGeometricGraphicsContentPresentationAttribute$^{8.6}$(C b)) $_0)$

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

11.2 Interface to the Document Architecture

Semiformal Description 8.4

Predicate "satisfies geometric graphics content architecture constraints"

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

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

Definition 8.4

1 $\forall prof, doby$
2 $(_0$SatisfiesGeometricGraphicsContentArchitectureConstraints$(prof, doby)$ *iff*
3 $\forall cst, cont \in doby$
4 $(_1 (_2 (cont)$DescribesContPortOf$^{2.153}(cst)$ *and*
5 C ^cst • 'content architecture class' = '2 8 2 8 0'$_2$) *impl*
6 $(_3$IsGeometricGraphicsContentPortionDescription$^{8.5}(cont)$ *and*
7 '2 8 2 8 0' ∈ C ^$prof$ • 'content architecture classes'$_3)_1)_0$)

Semiformal Description 8.5

Predicate "is a geometric graphics content portion description"

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

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

Definition 8.5

1 $\forall cont$
2 $(_0$IsGeometricGraphicsContentPortionDescription$(cont)$ *iff*
3 IsGeometricGraphicsContentPortionAttributeSet$^{8.8}(cont)_0$)

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 8.6

Predicate "is a geometric graphics content presentation attribute"

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

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

Definition 8.6

$\forall\, att$
$(_0$ IsGeometricGraphicsContentPresentationAttribute(att) *iff*
$\exists\, n, c$
$(_1\, att = [n : c]$ *and*
$n \in$ ['colour representations'; 'edge rendition';
'filled area rendition'; 'geometric graphics encoding announcer';
'line rendition'; 'marker rendition';
'picture dimensions'; 'picture orientation';
'region of interest specification'; 'text rendition';
'transformation specification'; 'transparency specification'] *and*
$(_2\, n =$ 'colour representations' *impl*
$(_3$ IsColourRepresentationsValue$^{8.43}(c)$ *or* IsPlaceholder$^{1.19}(c)_{3})\ _2)$ *and*
$(_4\, n =$ 'edge rendition' *impl*
$(_5$ IsEdgeRenditionValue$^{8.39}(c)$ *or* IsPlaceholder$^{1.19}(c)_{5})\ _4)$ *and*
$(_6\, n =$ 'filled area rendition' *impl*
$(_7$ IsFilledAreaRenditionValue$^{8.31}(c)$ *or* IsPlaceholder$^{1.19}(c)_{7})\ _6)$ *and*
$(_8\, n =$ 'geometric graphics encoding announcer' *impl*
$(_9$ IsGeometricGraphicsEncodingAnnouncerValue$^{8.9}(c)$ *or* IsPlaceholder$^{1.19}(c)_{9})\ _8)$ *and*
$(_{10}\, n =$ 'line rendition' *impl*
$(_{11}$ IsLineRenditionValue$^{8.11}(c)$ *or* IsPlaceholder$^{1.19}(c)_{11})\ _{10})$ *and*
$(_{12}\, n =$ 'marker rendition' *impl*
$(_{13}$ IsMarkerRenditionValue$^{8.18}(c)$ *or* IsPlaceholder$^{1.19}(c)_{13})\ _{12})$ *and*
$(_{14}\, n =$ 'picture dimensions' *impl*
$(_{15}$ IsPictureDimensionsValue$^{8.50}(c)$ *or* IsPlaceholder$^{1.19}(c)_{15})\ _{14})$ *and*
$(_{16}\, n =$ 'picture orientation' *impl*
$(_{17}$ IsPictureOrientationValue$^{8.49}(c)$ *or* IsPlaceholder$^{1.19}(c)_{17})\ _{16})$ *and*
$(_{18}\, n =$ 'region of interest specification' *impl*
$(_{19}$ IsRegionOfInterestSpecificationValue$^{8.48}(c)$ *or* IsPlaceholder$^{1\ 19}(c)_{19})\ _{18})$ *and*
$(_{20}\, n =$ 'text rendition' *impl*
$(_{21}$ IsTextRenditionValue$^{8.22}(c)$ *or* IsPlaceholder$^{1.19}(c)_{21})\ _{20})$ *and*
$(_{22}\, n =$ 'transformation specification' *impl*
$(_{23}$ IsTransformationSpecificationValue$^{8.47}(c)$ *or* IsPlaceholder$^{1.19}(c)_{23})\ _{22})$ *and*
$(_{24}\, n =$ 'transparency specification' *impl*
$(_{25}$ IsTransparencySpecificationValue$^{8.46}(c)$ *or* IsPlaceholder$^{1.19}(c)_{25})\ _{24})_1)_0)$

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 8.7

Predicate "is a geometric graphics content coding attribute"

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

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

Definition 8.7

```
1  ∀ att
2  (0 IsGeometricGraphicsContentCodingAttribute(att) iff
3    ∃ n, c
4    (1 att = [n : c] and
5      n = 'type of coding' and IsGeometricGraphicsContentTypeOfCodingValue^8.52(c) 1)0)
```

11.3 Attributes of the Geometric Graphics Content Architecture

Semiformal Description 8.8

Predicate "is a geometric graphics content portion attribute set"

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

Definition 8.8

```
1  ∀ as
2  (0 IsGeometricGraphicsContentPortionAttributeSet(as) iff
3    IsNeNom^1.2(as) and
4    ∀ a ∈ ^as •
5    (1 N a = 'content information' impl IsGeometricGraphicsContentInformationValue^8.53(C a) 1)0)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 8.9

Predicate "is a geometric graphics encoding announcer value" (clause 8.1.1.1)

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

Definition 8.9

```
1   ∀ v
2   (₀IsGeometricGraphicsEncodingAnnouncerValue(v) iff
3     IsNom(v) and
4     NAMS^1.18(v) = ['Colour Index Precision'; 'Colour Precision';
5                     'Colour Selection Mode'; 'Colour Value Extent';
6                     'Index Precision';       'Integer Precision';
7                     'Maximum Colour Index'; 'Real Precision';
8                     'VDC Integer Precision'; 'VDC Real Precision';
9                     'VDC Type'] and
10    ∀ a ∈ ^v •
11    (₁(₂N a ∈ ['Colour Index Precision'; 'Colour Precision'; 'Index Precision'; 'Integer Precision'] impl
12        (₃IsPlaceholder^1.19(C a) or C a ∈ [8; 16; 24; 32]₃)₂) and
13      (₄ N a = 'Colour Selection Mode' impl
14          (₅IsPlaceholder^1.19(C a) or C a ∈ ['indexed'; 'direct']₅)₄) and
15      (₆ N a = 'Colour Value Extent' impl
16          (₇IsPlaceholder^1.19(C a) or IsDirectColourValuePair^8.10(C a)₇)₆) and
17      (₈ N a = 'Maximum Colour Index' impl
18          (₉IsPlaceholder^1.19(C a) or IsNnInt^1.7(C a)₉)₈) and
19      (₁₀N a ∈ ['Real Precision'; 'VDC Real Precision'] impl
20          (₁₁IsPlaceholder^1.19(C a) or
21            C a ∈ ['floating point format, 9, 23'; 'floating point format, 12, 52';
22                   'fixed point format, 16, 16'; 'fixed point format, 32, 32']₁₁)₁₀) and
23    (₁₂N a = 'VDC Integer Precision' impl
24        (₁₃IsPlaceholder^1.19(C a) or C a ∈ [16; 24; 32]₁₃)₁₂) and
25    (₁₄N a = 'VDC Type' impl
26        (₁₅IsPlaceholder^1.19(C a) or C a ∈ ['integer'; 'real']₁₅)₁₄)₁)₀)
```

Semiformal Description 8.10

Predicate "is a direct colour value pair" (clause 8.1.1.1)

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

Definition 8.10

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


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 8.11

Predicate "is a line rendition value" (clause 8.1.1.2)

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

Definition 8.11

```
1   ∀ v
2   (0 IsLineRenditionValue(v) iff
3     IsNom(v) and
4     NAMS1.18(v) = ['line aspect source flags'; 'line bundle specifications';
5                    'Line Bundle Index';      'Line Colour';
6                    'Line Type';              'Line Width';
7                    'Line Width Specification Mode'] and
8     ∀ a ∈̇ ^v •
9     (1 (2 N a = 'line aspect source flags' impl
10        (3 IsPlaceholder1.19(C a) or
11          (4 IsNom(C a) and
12            NAMS1.18(C a) = ['line colour asf'; 'line type asf'; 'line width asf'] and
13            ∀ b ∈̇ ^(C a) •
14            (5 IsPlaceholder1.19(C b) or C b ∈ ['bundled'; 'individual'])5)4)3)2) and
15      (6 N a = 'line bundle specifications' impl
16        (7 IsPlaceholder1.19(C a) or IsSetOfLineBundleSpecifications8.12(C a)7)6) and
17      (8 N a = 'Line Bundle Index' impl
18        (9 IsPlaceholder1.19(C a) or IsNat(C a)9)8) and
19      (10 N a = 'Line Colour' impl
20        (11 IsPlaceholder1.19(C a) or IsGeometricGraphicsColourValue8.14(C a)11)10) and
21      (12 N a = 'Line Type' impl
22        (13 IsPlaceholder1.19(C a) or IsLineTypeValue8.17(C a)13)12) and
23      (14 N a = 'Line Width' impl
24        (15 IsPlaceholder1.19(C a) or IsWidthValue8.15(C a)15)14) and
25      (16 N a = 'Line Width Specification Mode' impl
26        (17 IsPlaceholder1.19(C a) or IsSpecificationModeValue8.16(C a)17)16)1)0)
```

Semiformal Description 8.12

Predicate "is a set of line bundle specifications" (clause 8.1.1.2)

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

Definition 8.12

```
1   ∀ v
2   (0 IsSetOfLineBundleSpecifications(v) iff
3     IsCol(v) and
4     ∀ a ∈ v (IsLineBundleSpecification8.13(a))0)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Semiformal Description 8.13

Predicate "is a line bundle specification" (clause 8.1.1.2)

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

Definition 8.13

1 $\forall v$
2 $(_0$IsLineBundleSpecification(v) *iff*
3 IsNom(v) *and*
4 NAMS$^{1.18}(v)$ = ['line bundle representation'; 'Line Bundle Index'] *and*
5 $\forall a \hat{\in} \hat{}v \bullet$
6 $(_1 (_2$N a = 'Line Bundle Index' *impl* IsNat(C $a)_2$) *and*
7 $(_3$N a = 'line bundle representation' *impl*
8 $(_4$IsNom(C a) *and*
9 NAMS$^{1.18}$(C a) = ['Line Colour'; 'Line Type'; 'Line Width'] *and*
10 $\forall b \hat{\in} \hat{}$(C a) $\bullet$
11 $(_5 (_6$N b = 'Line Colour' *impl* IsGeometricGraphicsColourValue$^{8.14}$(C $b)_6$) *and*
12 $(_7$N b = 'Line Type' *impl* IsLineTypeValue$^{8.17}$(C $b)_7$) *and*
13 $(_8$N b = 'Line Width' *impl* IsWidthValue$^{8.15}$(C $b)_8)_5)_4)_3)_1)_0$)

Semiformal Description 8.14

Predicate "is a geometric graphics colour value" (clauses 8.1.1.2, 8.1.1.3, 8.1.1.4, 8.1.1.5, 8.1.1.6)

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

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

Definition 8.14

1 $\forall v$
2 $(_0$IsGeometricGraphicsColourValue(v) *iff*
3 IsNnInt$^{1.7}(v)$ *or* IsDirectColourValue$^{8.54}(v)$ $_0$)

Semiformal Description 8.15

Predicate "is a width value" (clauses 8.1.1.2, 8.1.1.3, 8.1.1.6)

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

Definition 8.15

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

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 8.16

Predicate "is a specification mode value" (clauses 8.1.1.2, 8.1.1.3, 8.1.1.6)

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

Definition 8.16

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

Semiformal Description 8.17

Predicate "is a line type value" (clause 8.1.1.2)

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

Definition 8.17

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

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Semiformal Description 8.18

Predicate "is a marker rendition value" (clause 8.1.1.3)

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

Definition 8.18

```
1   ∀v
2   (0 IsMarkerRenditionValue(v) iff
3     IsNom(v) and
4     NAMS1.18(v) = ['marker aspect source flags'; 'marker bundle specifications';
5                    'Marker Bundle Index';      'Marker Colour';
6                    'Marker Size';              'Marker Size Specification Mode';
7                    'Marker Type'] and
8     ∀a ∈ ^v •
9     (1 (2 N a = 'marker aspect source flags' impl
10        (3 IsPlaceholder1.19(C a) or
11          (4 IsNom(C a) and
12            NAMS1.18(C a) = ['marker colour asf'; 'marker size asf'; 'marker type asf'] and
13            ∀b ∈ ^(C a) •
14            (5 IsPlaceholder1.19(C b) or C b ∈ ['bundled'; 'individual']5)4)3)2) and
15      (6 N a = 'marker bundle specifications' impl
16        (7 IsPlaceholder1.19(C a) or IsSetOfMarkerBundleSpecifications8.19(C a)7)6) and
17      (8 N a = 'Marker Bundle Index' impl
18        (9 IsPlaceholder1.19(C a) or IsNat(C a)9)8) and
19      (10 N a = 'Marker Colour' impl
20        (11 IsPlaceholder1.19(C a) or IsGeometricGraphicsColourValue8.14(C a)11)10) and
21      (12 N a = 'Marker Size' impl
22        (13 IsPlaceholder1.19(C a) or IsWidthValue8.15(C a)13)12) and
23      (14 N a = 'Marker Size Specification Mode' impl
24        (15 IsPlaceholder1.19(C a) or IsSpecificationModeValue8.16(C a)15)14) and
25      (16 N a = 'Marker Type' impl
26        (17 IsPlaceholder1.19(C a) or IsMarkerTypeValue8.21(C a)17)16)1)0)
```

Semiformal Description 8.19

Predicate "is a set of marker bundle specifications" (clause 8.1.1.3)

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

Definition 8.19

```
1   ∀v
2   (0 IsSetOfMarkerBundleSpecifications(v) iff
3     IsCol(v) and
4     ∀a ∈ v (IsMarkerBundleSpecification8.20(a))0)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 8.20

Predicate "is a marker bundle specification" (clause 8.1.1.3)

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

Definition 8.20

$$
\begin{array}{ll}
1 & \forall\, v \\
2 & (_0\,\mathrm{IsMarkerBundleSpecification}(v)\ \underline{\textit{iff}} \\
3 & \quad \mathrm{IsNom}(v)\ \underline{\textit{and}} \\
4 & \quad \mathrm{NAMS}^{1.18}(v) = [\text{'marker bundle representation'}; \text{'Marker Bundle Index'}]\ \underline{\textit{and}} \\
5 & \quad \forall\, a \hat{\in} \hat{}v \bullet \\
6 & \quad (_1\,(_2\,\mathrm{N}\ a = \text{'Marker Bundle Index'}\ \underline{\textit{impl}}\ \mathrm{IsNat}(\mathrm{C}\ a)_2)\ \underline{\textit{and}} \\
7 & \qquad (_3\,\mathrm{N}\ a = \text{'marker bundle representation'}\ \underline{\textit{impl}} \\
8 & \qquad\quad (_4\,\mathrm{IsNom}(\mathrm{C}\ a)\ \underline{\textit{and}} \\
9 & \qquad\qquad \mathrm{NAMS}^{1.18}(\mathrm{C}\ a) = [\text{'Marker Colour'}; \text{'Marker Size'}; \text{'Marker Type'}]\ \underline{\textit{and}} \\
10 & \qquad\qquad \forall\, b \hat{\in} \hat{}(\mathrm{C}\ a) \bullet \\
11 & \qquad\qquad (_5\,(_6\,\mathrm{N}\ b = \text{'Marker Colour'}\ \underline{\textit{impl}}\ \mathrm{IsGeometricGraphicsColourValue}^{8.14}(\mathrm{C}\ b)_6)\ \underline{\textit{and}} \\
12 & \qquad\qquad\quad (_7\,\mathrm{N}\ b = \text{'Marker Size'}\ \underline{\textit{impl}}\ \mathrm{IsWidthValue}^{8.15}(\mathrm{C}\ b)_7)\ \underline{\textit{and}} \\
13 & \qquad\qquad\quad (_8\,\mathrm{N}\ b = \text{'Marker Type'}\ \underline{\textit{impl}}\ \mathrm{IsMarkerTypeValue}^{8.21}(\mathrm{C}\ b)_8)_5)_4)_3)_1)_0)
\end{array}
$$

Semiformal Description 8.21

Predicate "is a marker type value" (clause 8.1.1.3)

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

Definition 8.21

$$
\begin{array}{ll}
1 & \forall\, v \\
2 & (_0\,\mathrm{IsMarkerTypeValue}(v)\ \underline{\textit{iff}} \\
3 & \quad v \in [1; 2; 3; 4; 5]\ \underline{\textit{or}}\ \mathrm{IsRegisteredMarkerType}^{8.59}(v)_0)
\end{array}
$$

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 8.22

Predicate "is a text rendition value" (clause 8.1.1.4)

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

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Definition 8.22

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


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Semiformal Description 8.23

Predicate "is a character set list" (clause 8.1.1.4)

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

Definition 8.23

```
1  ∀ v
2  (0 IsCharacterSetList(v) iff
3    IsCol(v) and
4    ∀ a ∈ v (IsCharacterSetSpecification^8.24(a))0)
```

Semiformal Description 8.24

Predicate "is a character set specification" (clause 8.1.1.4)

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

Definition 8.24

```
1  ∀ v
2  (0 IsCharacterSetSpecification(v) iff
3    IsNom(v) and
4    NAMS^1 18(v) = ['character set type';'designation sequence tail'] and
5    ∀ a ∈̂ ^v •
6    (1 (2 N a = 'character set type' impl
7        C a ∈ ['94-character sets';'96-character sets';'94-character multibyte sets';
8              '96-character multibyte sets';'complete code']2) and
9      (3 N a = 'designation sequence tail' impl IsRegisteredDesignationSequenceTail^8.26(C a))3)1)0)
```

Semiformal Description 8.25

Predicate "is a set of text bundle specifications" (clause 8.1.1.4)

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

Definition 8.25

```
1  ∀ v
2  (0 IsSetOfTextBundleSpecifications(v) iff
3    IsCol(v) and
4    ∀ a ∈ v (IsTextBundleSpecification^8.27(a))0)
```

Semiformal Description 8.26

Predicate "is a registered designation sequence tail" (clause 8.1.1.4)

A registered designation sequence tail is considered an atomic construct.

Definition 8.26

```
1  ∀ v
2  (0 IsRegisteredDesignationSequenceTail(v) iff
3    IsAtom(v)0)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 8.27

Predicate "is a text bundle specification" (clause 8.1.1.4)

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

Definition 8.27

```
1   ∀v
2   (0 IsTextBundleSpecification(v) iff
3     IsNom(v) and
4     NAMS1.18(v) = ['text bundle representation'; 'Text Bundle Index'] and
5     ∀a ∈̂ ^v •
6     (1 (2 N a = 'Text Bundle Index' impl IsNat(C a)2) and
7       (3 N a = 'text bundle representation' impl
8         (4 IsNom(C a) and
9           NAMS1.18(C a) = ['Character Expansion Factor'; 'Character Spacing';
10                           'Text Colour'; 'Text Font Index'; 'Text Precision'] 4) and
11      ∀b ∈̂ ~(C a) •
12      (5 (6 N b = 'Character Expansion Factor' impl IsPosReal1.25(C b)6) and
13        (7 N b = 'Character Spacing' impl IsReal(C b)7) and
14        (8 N b = 'Text Colour' impl IsGeometricGraphicsColourValue8.14(C b)8) and
15        (9 N b = 'Text Font Index' impl IsNat(C b)9) and
16        (10 N b = 'Text Precision' impl C b ∈ ['character'; 'string'; 'stroke']10)5)3)1)0)
```

Semiformal Description 8.28

Predicate "is an orientation vector pair" (clause 8.1.1.4)

An orientation vector pair is a VDC pair.

Definition 8.28

```
1   ∀v
2   (0 IsOrientationVectorPair(v) iff
3     IsVDCPair8.55(v)0)
```

Semiformal Description 8.29

Predicate "is a font list set" (clause 8.1.1.4)

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

Definition 8.29

```
1   ∀v
2   (0 IsFontListSet(v) iff
3     IsCol(v) and
4     ∀a ∈ v (IsAtom(a)) 0)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 8.30

Predicate "is a text alignment specification" (clause 8.1.1.4)

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

Definition 8.30

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


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 8.31

Predicate "is a filled area rendition value" (clause 8.1.1.5)

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

Definition 8.31

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


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 8.32

Predicate "is a set of fill bundle specifications" (clause 8.1.1.5)

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

Definition 8.32

1 $\forall v$
2 $(_0$IsSetOfFillBundleSpecifications(v) *iff*
3 IsCol(v) *and*
4 $\forall a \in v$ (IsFillBundleSpecification$^{8.33}$(a)) $_0$)

Semiformal Description 8.33

Predicate "is a fill bundle specification" (clause 8.1.1.5)

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

Definition 8.33

1 $\forall v$
2 $(_0$IsFillBundleSpecification(v) *iff*
3 IsNom(v) *and*
4 NAMS$^{1.18}$(v) = ['fill bundle representation'; 'Fill Bundle Index'] *and*
5 $\forall a \hat{\in} \hat{}v \bullet$
6 $(_1 (_2$N a = 'Fill Bundle Index' *impl* IsNat(C a)$_2$) *and*
7 $(_3$N a = 'fill bundle representation' *impl*
8 $(_4$IsNom(C a) *and*
9 NAMS$^{1.18}$(C a) = ['Fill Colour'; 'Hatch Index'; 'Interior Style'; 'Pattern Index'] *and*
10 $\forall b \hat{\in} \hat{}$(C a)$\bullet$
11 $(_5 (_6$N b = 'Fill Colour' *impl* IsGeometricGraphicsColourValue$^{8.14}$(C b)$_6$) *and*
12 $(_7$N b = 'Hatch Index' *impl* IsHatchIndexValue$^{8.37}$(C b)$_7$) *and*
13 $(_8$N b = 'Interior Style' *impl* C $b \in$ ['empty'; 'hatch'; 'hollow'; 'pattern'; 'solid']$_8$) *and*
14 $(_9$N b = 'Pattern Index' *impl* IsNat(C b)$_9$)$_5$)$_4$)$_3$)

Semiformal Description 8.34

Predicate "is a set of pattern table specifications" (clause 8.1.1.5)

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

Definition 8.34

1 $\forall v$
2 $(_0$IsSetOfPatternTableSpecifications(v) *iff*
3 IsCol(v) *and*
4 $\forall a \in v$ (IsPatternTableSpecification$^{8.35}$(a)) $_0$)

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 8.35

Predicate "is a pattern table specification" (clause 8.1.1.5)

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

Definition 8.35

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

Semiformal Description 8.36

Predicate "is a sequence of direct colour values" (clause 8.1.1.5)

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

Definition 8.36

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

Semiformal Description 8.37

Predicate "is a hatch index value" (clause 8.1.1.5)

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

Definition 8.37

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


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 8.38

Predicate "is a pattern size value" (clause 8.1.1.5)

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

Definition 8.38

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


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 8.39

Predicate "is an edge rendition value" (clause 8.1.1.6)

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

Definition 8.39

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


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 8.40

Predicate "is a set of edge bundle specifications" (clause 8.1.1.6)

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

Definition 8.40

1 $\forall v$
2 $(_0$ IsSetOfEdgeBundleSpecifications(v) *iff*
3 IsCol(v) *and*
4 $\forall a \in v$ (IsEdgeBundleSpecification$^{8.41}(a)$) $_0)$

Semiformal Description 8.41

Predicate "is an edge bundle specification" (clause 8.1.1.6)

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

Definition 8.41

1 $\forall v$
2 $(_0$ IsEdgeBundleSpecification(v) *iff*
3 IsNom(v) *and*
4 NAMS$^{1.18}(v)$ = ['edge bundle representation'; 'Edge Bundle Index'] *and*
5 $\forall a \hat{\in} \hat{}v \bullet$
6 $(_1 (_2$N a = 'Edge Bundle Index' *impl* IsNat(C $a)_2$) *and*
7 $(_3$N a = 'edge bundle representation' *impl*
8 $(_4$ IsNom(C a) *and*
9 NAMS$^{1.18}$(C a) = ['Edge Colour'; 'Edge Type'; 'Edge Width'] *and*
10 $\forall b \hat{\in} \hat{}$(C a) $\bullet$
11 $(_5 (_6$N b = 'Edge Colour' *impl* IsGeometricGraphicsColourValue$^{8.14}$(C $b)_6$) *and*
12 $(_7$N b = 'Edge Type' *impl* IsEdgeTypeValue$^{8.42}$(C $b)_7$) *and*
13 $(_8$N b = 'Edge Width' *impl* IsWidthValue$^{8.15}$(C $b)_8)_5)_4)_3)_1)_0)$

Semiformal Description 8.42

Predicate "is a edge type value" (clause 8.1.1.6)

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

Definition 8.42

1 $\forall v$
2 $(_0$ IsEdgeTypeValue(v) *iff*
3 $v \in$ [1; 2; 3; 4; 5] *or* IsRegisteredEdgeType$^{8.61}(v)_0)$

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 8.43

Predicate "is a colour representations value" (clause 8.1.1.7)

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

Definition 8.43

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

Semiformal Description 8.44

Predicate "is a set of colour table specifications" (clause 8.1.1.7)

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

Definition 8.44

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

Semiformal Description 8.45

Predicate "is a colour table specification" (clause 8.1.1.7)

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

Definition 8.45

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


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 8.46

Predicate "is a transparency specification value" (clause 8.1.1.8)

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

Definition 8.46

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

Semiformal Description 8.47

Predicate "is a transformation specification value" (clause 8.1.1.9)

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

Definition 8.47

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


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 8.48

Predicate "is a region of interest specification value" (clause 8.1.2)

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

Definition 8.48

```
1   ∀ v
2   (0 IsRegionOfInterestSpecificationValue(v) iff
3     IsNom(v) and
4     (1 NAMS^1.18(v) = ['rectangle'] or NAMS^1.18(v) = ['automatic']1) and
5     ∀ a ∈ ~v •
6     (2 (3 N a = 'rectangle' impl IsVDCPair^8.55(C a)3) and
7       (4 N a = 'automatic' impl C a = 'null'4)2)0)
```

Semiformal Description 8.49

Predicate "is a picture orientation value" (clause 8.1.3)

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

Definition 8.49

```
1   ∀ v
2   (0 IsPictureOrientationValue(v) iff
3     v ∈ ['0°'; '90°'; '180°'; '270°']0)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Semiformal Description 8.50

Predicate "is a picture dimensions value" (clause 8.2.1)

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

Definition 8.50

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

Semiformal Description 8.51

Predicate "is a geometric graphics content architecture class value" (clause 8.3)

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

NOTE: This predicate is used in clause 7.

Definition 8.51

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


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 8.52

Predicate "is a geometric graphics type of coding value" (clause 9.1)

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

NOTE: This predicate is used in clause 7.

Definition 8.52

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

Semiformal Description 8.53

Predicate "is a geometric graphics content information value" (clause 9.2)

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

NOTE: This predicate is used in clause 7.

Definition 8.53

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

Semiformal Description 8.54

Predicate "is a direct colour value"

A direct colour value is considered an atomic construct.

Definition 8.54

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

Semiformal Description 8.55

Predicate "is a VDC Pair"

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

Definition 8.55

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

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 8.56

Predicate "is a VDC value"

A VDC value is considered an atomic construct.

Definition 8.56

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

Semiformal Description 8.57

Predicate "is a non-negative VDC value"

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

Definition 8.57

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

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

Semiformal Description 8.58

Predicate "is a registered line type"

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

Definition 8.58

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

Semiformal Description 8.59

Predicate "is a registered marker type"

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

Definition 8.59

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

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 8.60

Predicate "is a registered hatch index"

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

Definition 8.60

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

Semiformal Description 8.61

Predicate "is a registered edge type"

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

Definition 8.61

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

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

12 Formal specification of the defaulting mechanism for defaultable attributes

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

Clause 12.1 defines a set of general functions which are used in clause 12.2.

Clause 12.2 defines a set of functions which describe the derivation of all defaultable attributes in ISO/IEC 8613. It is substructured into clauses 12.2.1 (defaultable attributes of ISO/IEC 8613-2), clauses 12.2.2 (defaultable attributes of ISO/IEC 8613-6), clauses 12.2.3 (defaultable attributes of ISO/IEC 8613-7) and clauses 12.2.4 (defaultable attributes of ISO/IEC 8613-8).

12.1 General functions

This clause defines the general functions for determining the values of defaultable attributes according to clause 9.1.2.4 of ISO/IEC 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 $({}_{0}x$ 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, if a layout style is referenced by a constituent *cst*, test if the layout style is a derived layout style and the value can be found in a superior layout style. 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* $att \in \text{NAMS}^{1.18}(\text{REF_LAY_STYLE}^{9.6}(cst, doby))$
4 THEN C ^$(\text{REF_LAY_STYLE}^{9.6}(cst, doby)) \bullet att$
5 ELSE IF 'layout style' $\in \text{NAMS}^{1.18}(cst)$
6 THEN $\text{SUP_LAY_STYLE_VALUE}^{9.3}(\text{REF_LAY_STYLE}^{9.6}(cst, doby), doby, att)$
7 $\text{ORELSE}^{9.1}\text{UNDEF}_{0})$

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 9.3

Function "Attribute value from a superior layout style"

If the layout style *lsty* is derived from another layout style and if the attribute *att* is specified for that layout style (3), then its value is returned (4). Otherwise, if the layout style *lsty* is derived from another layout style (5), test if that layout style is a derived layout style and the value can be found in a superior layout style (6). Otherwise the value is undefined.

Definition 9.3

1 $\forall\, lsty, doby, att$
2 $({}_{0}\mathrm{SUP_LAY_STYLE_VALUE}(lsty, doby, att) =$
3 IF 'derived from' $\in \mathrm{NAMS}^{1.18}(lsty)$ *and* $att \in \mathrm{NAMS}^{1.18}(\mathrm{SUP_LAY_STYLE}^{9.8}(lsty, doby))$
4 THEN C $\hat{}$$(\mathrm{SUP_LAY_STYLE}^{9.8}(lsty, doby)) \bullet att$
5 ELSE IF 'derived from' $\in \mathrm{NAMS}^{1.18}(lsty)$
6 THEN $\mathrm{SUP_LAY_STYLE_VALUE}^{9.3}(\mathrm{SUP_LAY_STYLE}^{9.8}(lsty, doby), doby, att)$
7 ORELSE$^{9.1}\mathrm{UNDEF}_{0}$)

Semiformal Description 9.4

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, if a presentation style is referenced by a constituent *cst*, test if the presentation style is a derived presentation style and the value can be found in a superior presentation style. Otherwise the value is undefined by step BP.

Definition 9.4

1 $\forall\, cst, doby, att$
2 $({}_{0}\mathrm{STEP_BP_VALUE}(cst, doby, att) =$
3 IF 'presentation style' $\in \mathrm{NAMS}^{1.18}(cst)$ *and* $att \in \mathrm{NAMS}^{1.18}(\mathrm{REF_PRES_STYLE}^{9.7}(cst, doby))$
4 THEN C $\hat{}$$(\mathrm{REF_PRES_STYLE}^{9.7}(cst, doby)) \bullet att$
5 ELSE IF 'presentation style' $\in \mathrm{NAMS}^{1.18}(cst)$
6 THEN $\mathrm{SUP_PRES_STYLE_VALUE}^{9.5}(\mathrm{REF_PRES_STYLE}^{9.7}(cst, doby), doby, att)$
7 ORELSE$^{9.1}\mathrm{UNDEF}_{0}$)

Semiformal Description 9.5

Function "Attribute value from a superior presentation style"

If the presentation style *psty* is derived from another presentation style and if the attribute *att* is specified for that presentation style (3), then its value is returned (4). Otherwise, if the presentation style *psty* is derived from another presentation style (5), test if that presentation style is a derived presentation style and the value can be found in a superior presentation style (6). Otherwise the value is undefined.

Definition 9.5

1 $\forall\, psty, doby, att$
2 $({}_{0}\mathrm{SUP_PRES_STYLE_VALUE}(psty, doby, att) =$
3 IF 'derived from' $\in \mathrm{NAMS}^{1.18}(psty)$ *and* $att \in \mathrm{NAMS}^{1.18}(\mathrm{SUP_PRES_STYLE}^{9.9}(psty, doby))$
4 THEN C $\hat{}$$(\mathrm{SUP_PRES_STYLE}^{9.9}(psty, doby)) \bullet att$
5 ELSE IF 'derived from' $\in \mathrm{NAMS}^{1.18}(psty)$
6 THEN $\mathrm{SUP_PRES_STYLE_VALUE}^{9.5}(\mathrm{SUP_PRES_STYLE}^{9.9}(psty, doby), doby, att)$
7 ORELSE$^{9.1}\mathrm{UNDEF}_{0}$)

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 9.6

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 9.6

```
1   ∀ cst, doby
2   (0 REF_LAY_STYLE(cst, doby) =
3     C ^doby • <IsLayoutStyle2.55(C xs) and C xs • 'layout style identifier' = C ^cst • 'layout style'>
4     ORELSE9.1 [ : ]0)
```

Semiformal Description 9.7

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 9.7

```
1   ∀ cst, doby
2   (0 REF_PRES_STYLE(cst, doby) =
3     C ^doby • <IsPresentationStyle2.56(C xs) and
4               C xs • 'presentation style identifier' = C ^cst • 'presentation style'>
5     ORELSE9.1 [ : ]0)
```

Semiformal Description 9.8

Function "Superior 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 'derived from' of a particular layout style *lsty* this is called the superior layout style (3) which is returned by this function.

Definition 9.8

```
1   ∀ lsty, doby
2   (0 SUP_LAY_STYLE(lsty, doby) =
3     THAT x (1 IsLayoutStyle2.55(x) and C x • 'layout style identifier' = C ^lsty • 'derived from'1)0)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 9.9

Function "Superior 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 'derived from' of a particular presentation style *psty* this is called the superior presentation style (3, 4) which is returned by this function.

Definition 9.9

1 $\forall\, psty, doby$
2 $({}_0\mathrm{SUP_PRES_STYLE}(psty, doby) =$
3 THAT x $({}_1\mathrm{IsPresentationStyle}^{2.56}(x)$ *and* C x • 'presentation style identifier' = C ^$psty$ • 'derived from' ${}_1)_0)$

Semiformal Description 9.10

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 9.10

1 $\forall\, cst, doby, att$
2 $({}_0\mathrm{STEP_C_VALUE}(cst, doby, att) =$
3 IF 'object class' $\in \mathrm{NAMS}^{1.18}(cst)$ *and* $att \in \mathrm{NAMS}^{1.18}(\mathrm{REF_OBJECT_CLASS}^{9.11}(cst, doby))$
4 THEN C ^$(\mathrm{REF_OBJECT_CLASS}^{9.11}(cst, doby))$ • att
5 ELSE $\mathrm{UNDEF}_0)$

Semiformal Description 9.11

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 9.11

1 $\forall\, cst, doby$
2 $({}_0\mathrm{REF_OBJECT_CLASS}(cst, doby) =$
3 C ^$doby$ • <$\mathrm{IsObjectClassDescription}^{2.35}$(C xs) *and* C xs • 'object class identifier' = C ^cst • 'object class'>
4 $\mathrm{ORELSE}^{9.1}$ [:] $_0)$

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Semiformal Description 9.12

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, if an object class is referenced by a constituent *cst* and a presentation style is referenced by this object class, test if the presentation style is a derived presentation style and the value can be found in a superior presentation style. Otherwise the value is undefined by step DP.

Definition 9.12

```
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^9.11(cst, doby)) and
5           att ∈ NAMS^1.18(REF_PRES_STYLE^9.7(REF_OBJECT_CLASS^9.11(cst, doby), doby))
6     THEN C ^(REF_PRES_STYLE^9.7(REF_OBJECT_CLASS^9.11(cst, doby), doby)) • att
7     ELSE IF 'presentation style' ∈ NAMS^1.18(REF_OBJECT_CLASS^9.11(cst, doby)) and
8           THEN SUP_PRES_STYLE_VALUE^9.5(
9                 REF_PRES_STYLE^9.7(REF_OBJECT_CLASS^9.11(cst, doby), doby), doby, att)
10          ORELSE^9.1 UNDEF0)
```

Semiformal Description 9.13

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, if an object class is referenced by a constituent *cst* and a layout style is referenced by this object class, test if the layout style is a derived layout style and the value can be found in a superior layout style. Otherwise the value is undefined by step DL.

Definition 9.13

```
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^9.11(cst, doby)) and
5           att ∈ NAMS^1.18(REF_LAY_STYLE^9.6(REF_OBJECT_CLASS^9.11(cst, doby), doby))
6     THEN C ^(REF_LAY_STYLE^9.6(REF_OBJECT_CLASS^9.11(cst, doby), doby)) • att
7     ELSE IF 'layout style' ∈ NAMS^1.18(REF_OBJECT_CLASS^9.11(cst, doby)) and
8           THEN SUP_LAY_STYLE_VALUE^9.3(
9                 REF_LAY_STYLE^9.6(REF_OBJECT_CLASS^9.11(cst, doby), doby), doby, att)
10          ORELSE^9.1 UNDEF0)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 9.14

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 9.14

$$
\begin{array}{l}
1 \quad \forall\, cst, doby, rdoby, prof, att \\
2 \quad ({}_{0}\mathrm{STEP_E_VALUE}(cst, doby, rdoby, prof, att) = \\
3 \quad\ \ \mathrm{IF}\ \texttt{'object class'} \in \mathrm{NAMS}^{1.18}(cst)\ \underline{and} \\
4 \quad\ \ \ \ \texttt{'resource'} \in \mathrm{NAMS}^{1.18}(\mathrm{REF_OBJECT_CLASS}^{9.11}(cst, doby))\ \underline{and} \\
5 \quad\ \ \ \ att \in \mathrm{NAMS}^{1.18}(\mathrm{RESOURCE_OBJECT_CLASS}^{9.15}(rdoby, \\
6 \quad\ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \mathrm{OBJECT_CLASSID_INRES}^{9.16}(prof, \mathrm{REF_OBJECT_CLASS}^{9.11}(cst, doby)))) \\
7 \quad\ \ \mathrm{THEN\ C\ \hat{}(RESOURCE_OBJECT_CLASS}^{9.15}(rdoby, \\
8 \quad\ \ \ \ \ \ \ \ \ \ \mathrm{OBJECT_CLASSID_INRES}^{9.16}(prof, \mathrm{REF_OBJECT_CLASS}^{9.11}(cst, doby))))\bullet att \\
9 \quad\ \ \mathrm{ELSE\ UNDEF}_{0})
\end{array}
$$

Semiformal Description 9.15

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 9.15

$$
\begin{array}{l}
1 \quad \forall\, rdoby, objcid \\
2 \quad ({}_{0}\mathrm{RESOURCE_OBJECT_CLASS}(rdoby, objcid) = \\
3 \quad\ \ \mathrm{C\ \hat{}}rdoby \bullet \texttt{<IsObjectClassDescription}^{2.35}(\mathrm{C}\ xs)\ \underline{and}\ \mathrm{C}\ xs \bullet \texttt{'object class identifier'} = objcid\texttt{>} \\
4 \quad\ \ \mathrm{ORELSE}^{9.1}\ [:]_{0})
\end{array}
$$

Semiformal Description 9.16

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 9.16

$$
\begin{array}{l}
1 \quad \forall\, prof, cst \\
2 \quad ({}_{0}\mathrm{OBJECT_CLASSID_INRES}(prof, cst) = \\
3 \quad\ \ \mathrm{LASTC}^{1.14}\ (\mathrm{C\ \hat{}}prof \bullet \texttt{'resources'} \bullet \texttt{<HEAD}^{1.13}(\mathrm{C}\ xs) = \mathrm{C\ \hat{}}cst \bullet \texttt{'resource'>}) \bullet {}_{0})
\end{array}
$$

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 9.17

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, 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, test if the presentation style is a derived presentation style and the value can be found in a superior presentation style in the resource document. Otherwise the value is undefined by step FP.

Definition 9.17

```
1   ∀ cst, doby, rdoby, prof, att
2   (₀STEP_FP_VALUE(cst, doby, rdoby, prof, att) =
3     IF 'object class' ∈ NAMS^1.18(cst) and
4        'resource' ∈ NAMS^1.18(REF_OBJECT_CLASS^9.11(cst, doby)) and
5        'presentation style' ∈ NAMS^1.18(RESOURCE_OBJECT_CLASS^9.15(rdoby,
6                              OBJECT_CLASSID_INRES^9.16(prof, REF_OBJECT_CLASS^9.11(cst, doby)))) and
7        att ∈ NAMS^1.18(RESOURCE_PRES_STYLE^9.18(rdoby,
8                        OBJECT_CLASSID_INRES^9.16(prof, REF_OBJECT_CLASS^9.11(cst, doby))))
9     THEN C ^(RESOURCE_PRES_STYLE^9.18(rdoby,
10              OBJECT_CLASSID_INRES^9.16(prof, REF_OBJECT_CLASS^9.11(cst, doby)))).att
11    ELSE IF 'presentation style' ∈ NAMS^1.18(RESOURCE_OBJECT_CLASS^9.15(rdoby,
12                OBJECT_CLASSID_INRES^9.16(prof, REF_OBJECT_CLASS^9.11(cst, doby))))
13          THEN SUP_PRES_STYLE_VALUE^9.5(RESOURCE_PRES_STYLE^9.18(rdoby,
14                OBJECT_CLASSID_INRES^9.16(prof, REF_OBJECT_CLASS^9.11(cst, doby))), rdoby, att)
15          ORELSE^9.1 UNDEF₀)
```

Semiformal Description 9.18

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 9.18

```
1   ∀ rdoby, objcid
2   (₀RESOURCE_PRES_STYLE(rdoby, objcid) =
3     THAT rpres
4     (₁ rpres ∈ rdoby and IsPresentationStyle^2.56(rpres) and
5       ∃ rcst ∈ rdoby
6       (₂C ^rpres.'presentation style identifier' = C ^rcst.'presentation style' and
7         C ^rcst.'object class identifier' = objcid₂)₁)₀)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 9.19

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, 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, test if the layout style is a derived layout style and the value can be found in a superior layout style in the resource document. Otherwise the value is undefined by step FL.

Definition 9.19

```
1   ∀ cst, doby, rdoby, prof, att
2   (0 STEP_FL_VALUE(cst, doby, rdoby, prof, att) =
3     IF 'object class' ∈ NAMS^1.18(cst) and
4        'resource' ∈ NAMS^1.18(REF_OBJECT_CLASS^9.11(cst, doby)) and
5        'layout style' ∈ NAMS^1.18(RESOURCE_OBJECT_CLASS^9.15(rdoby,
6                          OBJECT_CLASSID_INRES^9.16(prof, REF_OBJECT_CLASS^9.11(cst, doby)))) and
7       att ∈ NAMS^1.18(RESOURCE_LAY_STYLE^9.20(rdoby,
8                          OBJECT_CLASSID_INRES^9.16(prof, REF_OBJECT_CLASS^9.11(cst, doby))))
9     THEN C ^(RESOURCE_LAY_STYLE^9.20(rdoby,
10              OBJECT_CLASSID_INRES^9.16(prof, REF_OBJECT_CLASS^9.11(cst, doby)))) • att
11    ELSE IF 'layout style' ∈ NAMS^1.18(RESOURCE_OBJECT_CLASS^9.15(rdoby,
12                 OBJECT_CLASSID_INRES^9.16(prof, REF_OBJECT_CLASS^9.11(cst, doby))))
13          THEN SUP_LAY_STYLE_VALUE^9.3(RESOURCE_LAY_STYLE^9.20(rdoby,
14                 OBJECT_CLASSID_INRES^9.16(prof, REF_OBJECT_CLASS^9.11(cst, doby))), rdoby, att)
15          ORELSE^9.1 UNDEF_0)
```

Semiformal Description 9.20

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 9.20

```
1   ∀ rdoby, objcid
2   (0 RESOURCE_LAY_STYLE(rdoby, objcid) =
3     THAT rlay
4     (1 rlay ∈ rdoby and IsLayoutStyle^2.55(rlay) and
5       ∃ rcst ∈ rdoby
6       (2 C ^rlay • 'layout style identifier' = C ^rcst • 'layout style' and
7         C ^rcst • 'object class identifier' = objcid_2)_1)_0)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 9.21

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 9.21

$$
\begin{array}{l}
\forall\, cst, doby, rdoby, prof, att, ctype \\
({}_0\mathrm{STEP_G_VALUE}(cst, doby, rdoby, prof, att, ctype) = \\
\quad \mathrm{SUPERIOR_VALUE}^{9.22}({}_1\mathrm{THAT}\ sup\ ({}_2 sup \in doby\ \underline{and}\ (cst)\mathrm{DescribesImSubOf}^{2.152}(sup)_2), \\
\qquad\qquad doby, rdoby, prof, att, ctype_1) \\
\quad \mathrm{ORELSE}^{9.1}\ \mathrm{STEP_G_VALUE}^{9.21}({}_3\mathrm{THAT}\ sup\ ({}_4 sup \in doby\ \underline{and}\ (cst)\mathrm{DescribesImSubOf}^{2.152}(sup)_4), \\
\qquad\qquad doby, rdoby, prof, att, ctype_3)_0)
\end{array}
$$

Semiformal Description 9.22

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 9.22

$$
\begin{array}{l}
\forall\, cst, doby, rdoby, prof, att, ctype \\
({}_0\mathrm{SUPERIOR_VALUE}(cst, doby, rdoby, prof, att, ctype) = \\
\quad \mathrm{DEFAULT_VALUE}^{9.23}(cst, att, ctype, doby)\ \mathrm{ORELSE}^{9.1} \\
\quad ({}_1\mathrm{IF}\ \texttt{'object class'} \in \mathrm{NAMS}^{1.18}(cst) \\
\qquad \mathrm{THEN}\ ({}_2\mathrm{DEFAULT_VALUE}^{9.23}(\mathrm{REF_OBJECT_CLASS}^{9.11}(cst, doby), att, ctype, doby)\ \mathrm{ORELSE}^{9.1} \\
\qquad ({}_3\mathrm{IF}\ \texttt{'resource'} \in \mathrm{NAMS}^{1.18}(\mathrm{REF_OBJECT_CLASS}^{9.11}(cst, doby)) \\
\qquad\quad \mathrm{THEN}\ ({}_4\mathrm{RES_DEFAULT_VALUE}^{9.25}(\mathrm{REF_OBJECT_CLASS}^{9.11}(cst, doby), rdoby, prof, att, ctype) \\
\qquad\qquad\quad \mathrm{ORELSE}^{9.1}\ \mathrm{UNDEF}_4) \\
\qquad\quad \mathrm{ELSE}\ \mathrm{UNDEF}_3)_2) \\
\qquad \mathrm{ELSE}\ \mathrm{UNDEF}_1)_0)
\end{array}
$$

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Semiformal Description 9.23

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 9.23

```
1  ∀ cst, att, ctype, doby
2  (0 DEFAULT_VALUE(cst, att, ctype, 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 • att5) or
8              (6 IsCol(C v) and a = v • • att6) 4) 3) 2)
9    ORELSE^9.1 DEFAULT_VALUE_BY_STYLE^9.24(cst, doby, att, ctype) 1) 0)
```

Semiformal Description 9.24

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 9.24

```
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^9.6(C v, doby))
9          THEN C ^(REF_LAY_STYLE^9.6(C v, doby)) • att3) or
10       (4 IsNom(C v) and a = v • 'presentation style' and
11         att ∈ NAMS^1.18(REF_PRES_STYLE^9.7(C v, doby))
12         THEN C ^(REF_PRES_STYLE^9.7(C v, doby)) • att4) or
13       (5 IsCol(C v) and a = v • • 'presentation style' and
14         att ∈ NAMS^1.18(REF_PRES_STYLE^9.7(C v •, doby))
15         THEN C ^(REF_PRES_STYLE^9.7(C v •, doby)) • att5) 2) 1) 0)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Semiformal Description 9.25

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 9.25

$$
\begin{aligned}
&\forall\, cst, rdoby, prof, att, ctype \\
&({}_0\mathrm{RES_DEFAULT_VALUE}(cst, rdoby, prof, att, ctype) = \\
&\quad \mathrm{DEFAULT_VALUE}^{9.23}(\mathrm{RESOURCE_OBJECT_CLASS}^{9.15}(rdoby, \\
&\qquad \mathrm{OBJECT_CLASSID_INRES}^{9.16}(prof, cst)), att, ctype, rdoby)_0)
\end{aligned}
$$

Semiformal Description 9.26

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 9.26

$$
\begin{aligned}
&\forall\, prof, att \\
&({}_0\mathrm{STEP_H_VALUE}(prof, att) = \\
&\quad \mathrm{C}\ ({}_1\mathrm{THAT}\ a\ ({}_2 a \hat{\in} \hat{}prof \bullet \text{'document application profile defaults'}\ \underline{and}\ \mathrm{N}\ a = att_2)_1)_0)
\end{aligned}
$$

Semiformal Description 9.27

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, if a layout style is referenced by a constituent *cst*, test if the layout style is a derived layout style and the value can be found in a superior layout style. Otherwise the value is undefined by step BL.

Definition 9.27

$$
\begin{aligned}
&\forall\, cst, doby, att, par \\
&({}_0\mathrm{STEP_BL_PAR_VALUE}(cst, doby, att, par) = \\
&\quad \mathrm{IF}\ \text{'layout style'} \in \mathrm{NAMS}^{1.18}(cst)\ \underline{and}\ att \in \mathrm{NAMS}^{1.18}(\mathrm{REF_LAY_STYLE}^{9.6}(cst, doby))\ \underline{and} \\
&\qquad par \in \mathrm{NAMS}^{1.18}(\mathrm{C}\ \hat{}\mathrm{REF_LAY_STYLE}^{9.6}(cst, doby) \bullet att) \\
&\quad \mathrm{THEN}\ \mathrm{C}\ \hat{}(\mathrm{REF_LAY_STYLE}^{9.6}(cst, doby)) \bullet att \bullet par \\
&\quad \mathrm{ELSE\ IF}\ \text{'layout style'} \in \mathrm{NAMS}^{1.18}(cst) \\
&\qquad \mathrm{THEN\ SUP_LAY_STYLE_PAR_VALUE}^{9.28}(\mathrm{REF_LAY_STYLE}^{9.6}(cst, doby), doby, att, par) \\
&\qquad \mathrm{ORELSE}^{9.1}\mathrm{UNDEF}_0)
\end{aligned}
$$

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 9.28

Function "Parameter value from a superior layout style"

If the layout style *lsty* is derived from another layout style and the attribute *att* is specified for that layout style (3) and the parameter is specified for this attribute, then its value is returned (5). Otherwise, if the layout style *lsty* is derived from another layout style (6), test if that layout style is a derived layout style and the value can be found in a superior layout style (7). Otherwise the value is undefined.

Definition 9.28

1 $\forall\, lsty, doby, att, par$
2 $({}_0\text{SUP_LAY_STYLE_PAR_VALUE}(lsty, doby, att, par) =$
3 IF 'derived from' $\in \text{NAMS}^{1.18}(lsty)$ *and* $att \in \text{NAMS}^{1.18}(\text{SUP_LAY_STYLE}^{9.8}(lsty, doby))$ *and*
4 $par \in \text{NAMS}^{1.18}(\text{C } \hat{}\text{SUP_LAY_STYLE}^{9.8}(lsty, doby) \bullet att)$
5 THEN C $\hat{}(\text{SUP_LAY_STYLE}^{9.8}(lsty, doby)) \bullet att \bullet par$
6 ELSE IF 'derived from' $\in \text{NAMS}^{1.18}(lsty)$
7 THEN $\text{SUP_LAY_STYLE_PAR_VALUE}^{9.28}(\text{SUP_LAY_STYLE}^{9.8}(lsty, doby), doby, att, par)$
8 $\text{ORELSE}^{9.1}\text{UNDEF}_0)$

Semiformal Description 9.29

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 9.29

1 $\forall\, cst, doby, att, par$
2 $({}_0\text{STEP_C_PAR_VALUE}(cst, doby, att, par) =$
3 IF 'object class' $\in \text{NAMS}^{1.18}(cst)$ *and* $att \in \text{NAMS}^{1.18}(\text{REF_OBJECT_CLASS}^{9.11}(cst, doby))$ *and*
4 $par \in \text{NAMS}^{1.18}(\text{C } \hat{}\text{REF_OBJECT_CLASS}^{9.11}(cst, doby) \bullet att)$
5 THEN C $\hat{}(\text{REF_OBJECT_CLASS}^{9.11}(cst, doby)) \bullet att \bullet par$
6 ELSE $\text{UNDEF}_0)$

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 9.30

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, if an object class is referenced by a constituent *cst* and a layout style is referenced by this object class, test if the layout style is a derived layout style and the value can be found in a superior layout style. Otherwise the value is undefined by step DL.

Definition 9.30

$$
\begin{array}{ll}
1 & \forall\, cst, doby, att, par \\
2 & ({}_{0}\mathrm{STEP_DL_PAR_VALUE}(cst, doby, att, par) = \\
3 & \quad \mathrm{IF} \quad \text{'object class'} \in \mathrm{NAMS}^{1.18}(cst) \ \underline{and} \\
4 & \qquad \text{'layout style'} \in \mathrm{NAMS}^{1.18}(\mathrm{REF_OBJECT_CLASS}^{9.11}(cst, doby)) \ \underline{and} \\
5 & \qquad att \in \mathrm{NAMS}^{1.18}(\mathrm{REF_LAY_STYLE}^{9.6}(\mathrm{REF_OBJECT_CLASS}^{9.11}(cst, doby), doby)) \ \underline{and} \\
6 & \qquad par \in \mathrm{NAMS}^{1.18}(\mathrm{C\ \hat{}REF_LAY_STYLE}^{9.6}(\mathrm{REF_OBJECT_CLASS}^{9.11}(cst, doby), doby) \bullet att) \\
7 & \quad \mathrm{THEN\ C\ \hat{}(REF_LAY_STYLE}^{9.6}(\mathrm{REF_OBJECT_CLASS}^{9.11}(cst, doby), doby) \bullet att \bullet par \\
8 & \quad \mathrm{ELSE\ IF} \ \text{'layout style'} \in \mathrm{NAMS}^{1.18}(\mathrm{REF_OBJECT_CLASS}^{9.11}(cst, doby)) \ \underline{and} \\
9 & \qquad \mathrm{THEN\ SUP_LAY_STYLE_PAR_VALUE}^{9.28}(\\
10 & \qquad\quad \mathrm{REF_LAY_STYLE}^{9.6}(\mathrm{REF_OBJECT_CLASS}^{9.11}(cst, doby), doby), doby, att, par) \\
11 & \qquad \mathrm{ORELSE}^{9.1}\mathrm{UNDEF}_{0})
\end{array}
$$

Semiformal Description 9.31

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 9.31

$$
\begin{array}{ll}
1 & \forall\, cst, doby, rdoby, prof, att, par \\
2 & ({}_{0}\mathrm{STEP_E_PAR_VALUE}(cst, doby, rdoby, prof, att, par) = \\
3 & \quad \mathrm{IF} \ \text{'object class'} \in \mathrm{NAMS}^{1.18}(cst) \ \underline{and} \\
4 & \qquad \text{'resource'} \in \mathrm{NAMS}^{1.18}(\mathrm{REF_OBJECT_CLASS}^{9.11}(cst, doby)) \ \underline{and} \\
5 & \qquad att \in \mathrm{NAMS}^{1.18}(\ \mathrm{RESOURCE_OBJECT_CLASS}^{9.15}(rdoby, \\
6 & \qquad\qquad \mathrm{OBJECT_CLASSID_INRES}^{9.16}(prof, \mathrm{REF_OBJECT_CLASS}^{9.11}(cst, doby)))) \ \underline{and} \\
7 & \qquad par \in \mathrm{NAMS}^{1.18}(\mathrm{C\ \hat{}RESOURCE_OBJECT_CLASS}^{9.15}(rdoby, \\
8 & \qquad\qquad \mathrm{OBJECT_CLASSID_INRES}^{9.16}(prof, \mathrm{REF_OBJECT_CLASS}^{9.11}(cst, doby))) \bullet att) \\
9 & \quad \mathrm{THEN\ C\ \hat{}(RESOURCE_OBJECT_CLASS}^{9.15}(rdoby, \\
10 & \qquad \mathrm{OBJECT_CLASSID_INRES}^{9.16}(prof, \mathrm{REF_OBJECT_CLASS}^{9.11}(cst, doby)))) \bullet att \bullet par \\
11 & \quad \mathrm{ELSE\ UNDEF}_{0})
\end{array}
$$

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 9.32

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, 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, test if the layout style is a derived layout style and the value can be found in a superior layout style in the resource document. Otherwise the value is undefined by step FL.

Definition 9.32

```
1   ∀ cst, doby, rdoby, prof, att, par
2   (0STEP_FL_PAR_VALUE(cst, doby, rdoby, prof, att, par) =
3     IF 'object class' ∈ NAMS1.18(cst) and
4        'resource' ∈ NAMS1.18(REF_OBJECT_CLASS9.11(cst, doby)) and
5        'layout style' ∈ NAMS1.18(RESOURCE_OBJECT_CLASS9.15(rdoby,
6                          OBJECT_CLASSID_INRES9.16(prof, REF_OBJECT_CLASS9.11(cst, doby)))) and
7        att ∈ NAMS1.18( RESOURCE_LAY_STYLE9.20(rdoby,
8                          OBJECT_CLASSID_INRES9.16(prof, REF_OBJECT_CLASS9.11(cst, doby)))) and
9        par ∈ NAMS1.18(C ^RESOURCE_LAY_STYLE9.20(rdoby,
10                         OBJECT_CLASSID_INRES9.16(prof, REF_OBJECT_CLASS9.11(cst, doby))) • att)
11    THEN C ^(RESOURCE_LAY_STYLE9.20(rdoby,
12               OBJECT_CLASSID_INRES9.16(prof, REF_OBJECT_CLASS9.11(cst, doby)))) • att • par
13    ELSE IF 'layout style' ∈ NAMS1.18(RESOURCE_OBJECT_CLASS9.15(rdoby,
14                 OBJECT_CLASSID_INRES9.16(prof, REF_OBJECT_CLASS9.11(cst, doby))))
15          THEN SUP_LAY_STYLE_PAR_VALUE9.28(RESOURCE_LAY_STYLE9.20(rdoby,
16                 OBJECT_CLASSID_INRES9.16(prof, REF_OBJECT_CLASS9.11(cst, doby))), rdoby, att, par)
17          ORELSE9.1 UNDEF0)
```

Semiformal Description 9.33

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 9.33

```
1   ∀ cst, doby, rdoby, prof, att, par, ctype
2   (0STEP_G_PAR_VALUE(cst, doby, rdoby, prof, att, par, ctype) =
3     SUPERIOR_PAR_VALUE9.35(1 THAT sup (2 sup ∈ doby and (cst)DescribesImSubOf2.152(sup)2),
4                                  doby, rdoby, prof, att, par, ctype1)
5     ORELSE9.1 STEP_G_PAR_VALUE9.33(3 THAT sup (4 sup ∈ doby and (cst)DescribesImSubOf2.152(sup)4),
6                                  doby, rdoby, prof, att, par, ctype3)0)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Semiformal Description 9.34

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 9.34

```
1  ∀prof, att, par
2  (0STEP_H_PAR_VALUE(prof, att, par) =
3    C (1 THAT a
4        (2 a ∈̂ ^prof • 'document application profile defaults' and N a = att and
5          ∃p (3 p ∈̂ ^prof • 'document application profile defaults' • att and N p = par3)2)1)0)
```

Semiformal Description 9.35

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 9.35

```
1  ∀cst, doby, rdoby, prof, att, ctype, par
2  (0SUPERIOR_PAR_VALUE(cst, doby, rdoby, prof, att, ctype, par) =
3    DEFAULT_PAR_VALUE^9.36(cst, att, ctype, par, doby) ORELSE^9.1
4    (1 IF 'object class' ∈ NAMS^1.18(cst)
5      THEN (2 DEFAULT_PAR_VALUE^9.36(REF_OBJECT_CLASS^9.11(cst, doby), att, ctype, par, doby)
6      ORELSE^9.1 (3 IF 'resource' ∈ NAMS^1.18(REF_OBJECT_CLASS^9.11(cst, doby))
7                  THEN (4 RES_DEFAULT_PAR_VALUE^9.38(REF_OBJECT_CLASS^9.11(cst, doby),
8                                                     rdoby, prof, att, ctype, par)
9                        ORELSE^9.1 UNDEF4)
10                 ELSE UNDEF3)2)
11     ELSE UNDEF1)0)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 9.36

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 9.36

```
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 • par5) or
8              (6 IsCol(C v) and a = v • • att • par6)4)3)2)
9    ORELSE^9.1 DEFAULT_PAR_VALUE_BY_STYLE^9.37(cst, doby, att, ctype, par)1)0)
```

Semiformal Description 9.37

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 9.37

```
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^9.6(C v, doby)) and
9          par ∈ NAMS^1.18(REF_LAY_STYLE^9.6(C v, doby) • par)
10         THEN C ^(REF_LAY_STYLE^9.6(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^9.7(C v, doby)) and
13         par ∈ NAMS^1.18(REF_PRES_STYLE^9.7(C v, doby) • att)
14         THEN C ^(REF_PRES_STYLE^9.7(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^9.7(C v •, doby)) and
17         par ∈ NAMS^1.18(REF_PRES_STYLE^9.7(C v •, doby) • att)
18         THEN C ^(REF_PRES_STYLE^9.7(C v •, doby)) • att • par5)2)1)0)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 9.38

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 9.38

$\forall\, cst, rdoby, prof, att, ctype, par$
$({}_{0}\text{RES_DEFAULT_PAR_VALUE}(cst, rdoby, prof, att, ctype, par) =$
  $\text{DEFAULT_PAR_VALUE}^{9.36}(\text{RESOURCE_OBJECT_CLASS}^{9.15}(rdoby,$
    $\text{OBJECT_CLASSID_INRES}^{9.16}(prof, cst)), att, ctype, par, rdoby)_{0})$

Semiformal Description 9.39

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 9.39

$\forall\, cst, doby, att, par, subpar$
$({}_{0}\text{STEP_C_SUBPAR_VALUE}(cst, doby, att, par, subpar) =$
  IF 'object class' $\in \text{NAMS}^{1.18}(cst)$ *and* $att \in \text{NAMS}^{1.18}(\text{REF_OBJECT_CLASS}^{9.11}(cst, doby))$ *and*
    $par \in \text{NAMS}^{1.18}(\text{C}\ \hat{}\text{REF_OBJECT_CLASS}^{9.11}(cst, doby) \bullet att)$ *and*
    $subpar \in \text{NAMS}^{1.18}(\text{C}\ \hat{}\text{REF_OBJECT_CLASS}^{9.11}(cst, doby) \bullet att \bullet par)$
  THEN C $\hat{}(\text{REF_OBJECT_CLASS}^{9.11}(cst, doby)) \bullet att \bullet par \bullet subpar$
  ELSE UNDEF_{0})

Semiformal Description 9.40

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 9.40

$\forall\, cst, doby, rdoby, prof, att, par, subpar$
$({}_{0}\text{STEP_E_SUBPAR_VALUE}(cst, doby, rdoby, prof, att, par, subpar) =$
  IF 'object class' $\in \text{NAMS}^{1.18}(cst)$ *and*
    'resource' $\in \text{NAMS}^{1.18}(\text{REF_OBJECT_CLASS}^{9.11}(cst, doby))$ *and*
    $att \in \text{NAMS}^{1.18}(\text{RESOURCE_OBJECT_CLASS}^{9.15}(rdoby,$
      $\text{OBJECT_CLASSID_INRES}^{9.16}(prof, \text{REF_OBJECT_CLASS}^{9.11}(cst, doby)))))$ *and*
    $par \in \text{NAMS}^{1.18}(\text{C}\ \hat{}\text{RESOURCE_OBJECT_CLASS}^{9.15}(rdoby,$
      $\text{OBJECT_CLASSID_INRES}^{9.16}(prof, \text{REF_OBJECT_CLASS}^{9.11}(cst, doby))) \bullet att)$ *and*
    $subpar \in \text{NAMS}^{1.18}(\text{C}\ \hat{}\text{RESOURCE_OBJECT_CLASS}^{9.15}(rdoby,$
      $\text{OBJECT_CLASSID_INRES}^{9.16}(prof, \text{REF_OBJECT_CLASS}^{9.11}(cst, doby))) \bullet att \bullet par)$
  THEN C $\hat{}(\text{RESOURCE_OBJECT_CLASS}^{9.15}(rdoby,$
    $\text{OBJECT_CLASSID_INRES}^{9.16}(prof, \text{REF_OBJECT_CLASS}^{9.11}(cst, doby)))) \bullet att \bullet par \bullet subpar$
  ELSE UNDEF_{0})

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 9.41

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 9.41

$\forall\, cst, doby, rdoby, prof, att, par, subpar, ctype$
$(_0\text{STEP_G_SUBPAR_VALUE}(cst, doby, rdoby, prof, att, par, subpar, ctype) =$
$\quad \text{SUPERIOR_PAR_VALUE}^{9.35}(_1 \text{THAT } sup\ (_2 sup \in doby\ \underline{and}$
$\qquad\qquad (cst)\text{DescribesImSubOf}^{2.152}(sup)_2),$
$\qquad\qquad doby, rdoby, prof, att, par \bullet subpar, ctype_1)$
$\quad \text{ORELSE}^{9.1}\ \text{STEP_G_SUBPAR_VALUE}^{9.41}(_3 \text{THAT } sup\ (_4 sup \in doby\ \underline{and}$
$\qquad\qquad (cst)\text{DescribesImSubOf}^{2.152}(sup)_4),$
$\qquad\qquad doby, rdoby, prof, att, par, subpar, ctype_3)_0)$

12.2 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.

12.2.1 Defaultable attributes of ISO/IEC 8613-2

Semiformal Description 9.42

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 9.42

$\forall\, cst, doby, prof, rdoby$
$(_0\text{APPLICATION_COMMENTS_VALUE}(cst, doby, prof, rdoby) =$
$\quad \text{IF IsAttributeSet}^{2.61}(cst)\ \underline{and}\ \text{'application comments'} \in \text{NAMS}^{1.18}(cst)\ \underline{and}$
$\qquad \underline{not}\ \text{IsPlaceholder}^{1.19}(\text{C } \hat{}cst \bullet \text{'application comments'})$
$\quad \text{THEN C } \hat{}cst \bullet \text{'application comments'}$
$\quad \text{ELSE STEP_C_VALUE}^{9.10}(cst, doby, \text{'application comments'})\ \text{ORELSE}^{9.1}$
$\quad (_1 \text{STEP_E_VALUE}^{9.14}(cst, doby, rdoby, prof, \text{'application comments'})$
$\qquad \text{ORELSE}^{9.1}\ [\rightarrow]_1)_0)$

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 9.43

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 9.43

$$
\begin{array}{l}
\forall\, cst, doby, prof, rdoby \\
({}_0\mathrm{BALANCE_VALUE}(cst, doby, prof, rdoby) = \\
\quad \mathrm{IF\ IsAttributeSet}^{2.61}(cst)\ \underline{and}\ \text{'balance'} \in \mathrm{NAMS}^{1.18}(cst)\ \underline{and} \\
\quad\quad \underline{not}\ \mathrm{IsPlaceholder}^{1.19}(\mathrm{C}\ \hat{}cst \bullet \text{'balance'}) \\
\quad \mathrm{THEN\ C}\ \hat{}cst \bullet \text{'balance'} \\
\quad \mathrm{ELSE\ STEP_C_VALUE}^{9.10}(cst, doby, \text{'balance'})\ \mathrm{ORELSE}^{9.1} \\
\quad ({}_1\mathrm{STEP_E_VALUE}^{9.14}(cst, doby, rdoby, prof, \text{'balance'}) \\
\quad\quad \mathrm{ORELSE}^{9.1}\ \text{'null'}\,{}_1)_0)
\end{array}
$$

Semiformal Description 9.44

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 9.44

$$
\begin{array}{l}
\forall\, cst, doby, prof, rdoby \\
({}_0\mathrm{BINDINGS_VALUE}(cst, doby, prof, rdoby) = \\
\quad \mathrm{IF\ IsAttributeSet}^{2.61}(cst)\ \underline{and}\ \text{'bindings'} \in \mathrm{NAMS}^{1.18}(cst)\ \underline{and} \\
\quad\quad \underline{not}\ \mathrm{IsPlaceholder}^{1.19}(\mathrm{C}\ \hat{}cst \bullet \text{'bindings'}) \\
\quad \mathrm{THEN\ C}\ \hat{}cst \bullet \text{'bindings'} \\
\quad \mathrm{ELSE\ STEP_C_VALUE}^{9.10}(cst, doby, \text{'bindings'})\ \mathrm{ORELSE}^{9.1} \\
\quad ({}_1\mathrm{STEP_E_VALUE}^{9.14}(cst, doby, rdoby, prof, \text{'bindings'}) \\
\quad\quad \mathrm{ORELSE}^{9.1}\ []\,{}_1)_0)
\end{array}
$$

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 9.45

Function "Value of the attribute 'block alignment'"

The value of the attribute 'block alignment' may be found by step BL (3). If this is not the case, test if the attribute value can be found by step DL (4). If this is not the case, test if the attribute value can be found by step FL (5). If this is not the case, test if the attribute value can be found by step G (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 'block alignment' is 'right-hand aligned'.

Definition 9.45

```
1  ∀ cst, doby, prof, rdoby
2  (0 BLOCK_ALIGNMENT_VALUE(cst, doby, prof, rdoby) =
3    STEP_BL_VALUE^9.2(cst, doby, 'block alignment') ORELSE^9.1
4    (1 STEP_DL_VALUE^9.13(cst, doby, 'block alignment') ORELSE^9.1
5      (2 STEP_FL_VALUE^9.19(cst, doby, rdoby, prof, 'block alignment') ORELSE^9.1
6        (3 STEP_G_VALUE^9.21(cst, doby, rdoby, prof, 'block alignment', OBJECT_TYPE_VALUE^9.74(cst, doby))
7          ORELSE^9.1 (4 STEP_H_VALUE^9.26(prof, 'block alignment')
8            ORELSE^9.1 'right-hand aligned'4)3)2)1)0)
```

Semiformal Description 9.46

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 four parameters where the value of the parameters 'border line width' and 'border freespace width' is 0, the value of the parameter 'border line type' is 'solid' and the value of the parameter 'border line colour' is 'implementation-defined'.

Definition 9.46

```
1  ∀ cst, doby, prof, rdoby
2  (0 BORDER_VALUE(cst, doby, prof, rdoby) =
3    IF IsAttributeSet^2.61(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^9.4(cst, doby, 'border') ORELSE^9.1
7    (1 STEP_C_VALUE^9.10(cst, doby, 'border') ORELSE^9.1
8      (2 STEP_DP_VALUE^9.12(cst, doby, 'border') ORELSE^9.1
9        (3 STEP_E_VALUE^9.14(cst, doby, rdoby, prof, 'border') ORELSE^9.1
10         (4 STEP_FP_VALUE^9.17(cst, doby, rdoby, prof, 'border') ORELSE^9.1
11           (5 STEP_G_VALUE^9.21(cst, doby, rdoby, prof, 'border', OBJECT_TYPE_VALUE^9.74(cst, doby))
12             ORELSE^9.1 (6 STEP_H_VALUE^9.26(prof, 'border')
13               ORELSE^9.1 ['border line width' : 0; 'border line type' : 'solid';
14                           'border freespace width' : 0;
15                           'border line colour' : 'implementation-defined']6)5)4)3)2)1)0)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Semiformal Description 9.47

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 'colour of media'.

Definition 9.47

```
1   ∀ cst, doby, prof, rdoby
2   (0 COLOUR_VALUE(cst, doby, prof, rdoby) =
3     IF IsAttributeSet^2.61(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^9.4(cst, doby, 'colour') ORELSE^9.1
7     (1 STEP_C_VALUE^9.10(cst, doby, 'colour') ORELSE^9.1
8       (2 STEP_DP_VALUE^9.12(cst, doby, 'colour') ORELSE^9.1
9         (3 STEP_E_VALUE^9.14(cst, doby, rdoby, prof, 'colour') ORELSE^9.1
10          (4 STEP_FP_VALUE^9.17(cst, doby, rdoby, prof, 'colour') ORELSE^9.1
11            (5 STEP_G_VALUE^9.21(cst, doby, rdoby, prof, 'colour', OBJECT_TYPE_VALUE^9.74(cst, doby))
12              ORELSE^9.1 (6 STEP_H_VALUE^9.26(prof, 'colour')
13                ORELSE^9.1 'colour of media'6)5)4)3)2)1)0)
```

Semiformal Description 9.48

Function "Value of the attribute 'concatenation'"

The value of the attribute 'concatenation' may be found by step BL (3). If this is not the case, test if the attribute value can be found by step DL (4). If this is not the case, test if the attribute value can be found by step FL (5). If this is not the case, test if the attribute value can be found by step G (6). If this is not the case, the value of the attribute 'concatenation' is 'non-concatenated'.

Definition 9.48

```
1   ∀ cst, doby, prof, rdoby
2   (0 CONCATENATION_VALUE(cst, doby, prof, rdoby) =
3     STEP_BL_VALUE^9.2(cst, doby, 'concatenation') ORELSE^9.1
4     (1 STEP_DL_VALUE^9.13(cst, doby, 'concatenation') ORELSE^9.1
5       (2 STEP_FL_VALUE^9.19(cst, doby, rdoby, prof, 'concatenation') ORELSE^9.1
6         (3 STEP_G_VALUE^9.21(cst, doby, rdoby, prof, 'concatenation', OBJECT_TYPE_VALUE^9.74(cst, doby))
7           ORELSE^9.1 'non-concatenated'3)2)1)0)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 9.49

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 9.49

```
1   ∀ cst, doby, prof, rdoby
2   (0 CONTENT_ARCHITECTURE_CLASS_VALUE(cst, doby, prof, rdoby) =
3     IF IsAttributeSet^2.61(cst) and 'content architecture class' ∈ 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^9.10(cst, doby, 'content architecture class') ORELSE^9.1
7     (1 STEP_E_VALUE^9.14(cst, doby, rdoby, prof, 'content architecture class') ORELSE^9.1
8       (2 STEP_G_VALUE^9.21(cst, doby, rdoby, prof, 'content architecture class',
9                            OBJECT_TYPE_VALUE^9.74(cst, doby)) ORELSE^9.1
10        (3 STEP_H_VALUE^9.26(prof, 'content architecture class')
11          ORELSE^9.1 '2 8 2 6 0'3)2)1)0)
```

Semiformal Description 9.50

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 9.50

```
1   ∀ cst, doby, prof, rdoby
2   (0 DIMENSIONS_VALUE(cst, doby, prof, rdoby) =
3     ['horizontal dimension' : HORIZONTAL_DIMENSION_VALUE^9.51(cst, doby, prof, rdoby);
4      'vertical dimension' : VERTICAL_DIMENSION_VALUE^9.52(cst, doby, prof, rdoby)]0)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 9.51

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 9.51

```
1   ∀ cst, doby, prof, rdoby
2   (0 HORIZONTAL_DIMENSION_VALUE(cst, doby, prof, rdoby) =
3     IF IsAttributeSet^2.61(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^9.29(cst, doby, 'dimensions', 'horizontal dimension') ORELSE^9.1
8     (1 STEP_E_PAR_VALUE^9.31(cst, doby, rdoby, prof, 'dimensions', 'horizontal dimension') ORELSE^9.1
9       (2 STEP_G_PAR_VALUE^9.33(cst, doby, rdoby, prof, 'dimensions', 'horizontal dimension',
10                                OBJECT_TYPE_VALUE^9.74(cst, doby)) ORELSE^9.1
11        (3 (4 IF OBJECT_TYPE_VALUE^9.74(cst, doby) = 'composite or basic page'
12            THEN STEP_H_PAR_VALUE^9.34(prof, 'dimensions', 'horizontal dimension')
13                 ORELSE^9.1 UNDEF4)
14          ORELSE^9.1 ['fixed dimension' : HORIZONTAL_FIXED_DIMENSION_VALUE^9.53(cst)]3)2)1)0)
```

Semiformal Description 9.52

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 9.52

```
1   ∀ cst, doby, prof, rdoby
2   (0 VERTICAL_DIMENSION_VALUE(cst, doby, prof, rdoby) =
3     IF IsAttributeSet^2.61(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^9.29(cst, doby, 'dimensions', 'vertical dimension') ORELSE^9.1
8     (1 STEP_E_PAR_VALUE^9.31(cst, doby, rdoby, prof, 'dimensions', 'vertical dimension') ORELSE^9.1
9       (2 STEP_G_PAR_VALUE^9.33(cst, doby, rdoby, prof, 'dimensions', 'vertical dimension',
10                                OBJECT_TYPE_VALUE^9.74(cst, doby)) ORELSE^9.1
11        (3 (4 IF OBJECT_TYPE_VALUE^9.74(cst, doby) = 'composite or basic page'
12            THEN STEP_H_PAR_VALUE^9.34(prof, 'dimensions', 'vertical dimension')
13                 ORELSE^9.1 UNDEF4)
14          ORELSE^9.1 ['fixed dimension' : VERTICAL_FIXED_DIMENSION_VALUE^9.55(cst)]3)2)1)0)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 9.53

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 9.53

1 $\forall\, cst$
2 $({}_{0}\,\text{HORIZONTAL_FIXED_DIMENSION_VALUE}(cst) =$
3 IF $\text{OBJECT_TYPE_VALUE}^{9.74}(cst, doby) =$ 'composite or basic page'
4 THEN 9240
5 ELSE $\text{AVAILABLE_HORIZONTAL_DIMENSION_VALUE}^{9.54}(cst)_{0})$

Semiformal Description 9.54

Function "available horizontal dimension value"

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

Definition 9.54

1 $\forall\, cst$
2 $({}_{0}\,\text{IsAtom}(\text{AVAILABLE_HORIZONTAL_DIMENSION_VALUE}(cst))_{0})$

Semiformal Description 9.55

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 9.55

1 $\forall\, cst$
2 $({}_{0}\,\text{VERTICAL_FIXED_DIMENSION_VALUE}(cst) =$
3 IF $\text{OBJECT_TYPE_VALUE}^{9.74}(cst, doby) =$ 'composite or basic page'
4 THEN 13200
5 ELSE $\text{AVAILABLE_VERTICAL_DIMENSION_VALUE}^{9.56}(cst)_{0})$

Semiformal Description 9.56

Function "available vertical dimension value"

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

Definition 9.56

1 $\forall\, cst$
2 $({}_{0}\,\text{IsAtom}(\text{AVAILABLE_VERTICAL_DIMENSION_VALUE}(cst))_{0})$

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 9.57

Function "Value of the attribute 'fill order'"

The value of the attribute 'fill order' may be found by step BL (3). If this is not the case, test if the attribute value can be found by step DL (4). If this is not the case, test if the attribute value can be found by step FL (5). If this is not the case, test if the attribute value can be found by step G (6). If this is not the case, the value of the attribute 'fill order' is 'normal order'.

Definition 9.57

$$
\begin{array}{l}
\forall\, cst, doby, prof, rdoby \\
({}_0\mathrm{FILL_ORDER_VALUE}(cst, doby, prof, rdoby) = \\
\quad \mathrm{STEP_BL_VALUE}^{9.2}(cst, doby, \text{'fill order'})\ \mathrm{ORELSE}^{9.1} \\
\quad ({}_1\mathrm{STEP_DL_VALUE}^{9.13}(cst, doby, \text{'fill order'})\ \mathrm{ORELSE}^{9.1} \\
\quad\quad ({}_2\mathrm{STEP_FL_VALUE}^{9.19}(cst, doby, rdoby, prof, \text{'fill order'})\ \mathrm{ORELSE}^{9.1} \\
\quad\quad\quad ({}_3\mathrm{STEP_G_VALUE}^{9.21}(cst, doby, rdoby, prof, \text{'fill order'}, \mathrm{OBJECT_TYPE_VALUE}^{9.74}(cst, doby)) \\
\quad\quad\quad\quad \mathrm{ORELSE}^{9.1}\ \text{'normal order'}_3)_2)_1)_0)
\end{array}
$$

Semiformal Description 9.58

Function "Value of the attribute 'floatability range'"

The value of the attribute 'floatability range' comprises the parameters 'backward limit' whose value is a backward limit value and 'forward limit' whose value is a forward limit value.

Definition 9.58

$$
\begin{array}{l}
\forall\, cst, doby, prof, rdoby \\
({}_0\mathrm{FLOATABILITY_RANGE_VALUE}(cst, doby, prof, rdoby) = \\
\quad [\text{'backward limit'} : \mathrm{BACKWARD_LIMIT_VALUE}^{9.59}(cst, doby, prof, rdoby); \\
\quad\ \text{'forward limit'} : \mathrm{FORWARD_LIMIT_VALUE}^{9.60}(cst, doby, prof, rdoby)]_0)
\end{array}
$$

Semiformal Description 9.59

Function "Value of the parameter 'backward limit'"

The value of the parameter 'backward limit' may be found by step BL (3). If this is not the case, test if the parameter value can be found by step DL (4). If this is not the case, test if the parameter value can be found by step FL (5). If this is not the case, test if the parameter value can be found by step G (6). If this is not the case, the value of the parameter 'backward limit' comprises the sub-parameters 'layout object' and 'logical object' whose values are 'null'.

Definition 9.59

$$
\begin{array}{l}
\forall\, cst, doby, prof, rdoby \\
({}_0\mathrm{BACKWARD_LIMIT_VALUE}(cst, doby, prof, rdoby) = \\
\quad \mathrm{STEP_BL_VALUE}^{9.2}(cst, doby, \text{'backward limit'})\ \mathrm{ORELSE}^{9.1} \\
\quad ({}_1\mathrm{STEP_DL_VALUE}^{9.13}(cst, doby, \text{'backward limit'})\ \mathrm{ORELSE}^{9.1} \\
\quad\quad ({}_2\mathrm{STEP_FL_VALUE}^{9.19}(cst, doby, rdoby, prof, \text{'backward limit'})\ \mathrm{ORELSE}^{9.1} \\
\quad\quad\quad ({}_3\mathrm{STEP_G_VALUE}^{9.21}(cst, doby, rdoby, prof, \text{'backward limit'}, \mathrm{OBJECT_TYPE_VALUE}^{9.74}(cst, doby)) \\
\quad\quad\quad\quad \mathrm{ORELSE}^{9.1}\ [\text{'layout object'} : \text{'null'};\ \text{'logical object'} : \text{'null'}]_3)_2)_1)_0)
\end{array}
$$

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Semiformal Description 9.60

Function "Value of the parameter 'forward limit'"

The value of the parameter 'forward limit' may be found by step BL (3). If this is not the case, test if the parameter value can be found by step DL (4). If this is not the case, test if the parameter value can be found by step FL (5). If this is not the case, test if the parameter value can be found by step G (6). If this is not the case, the value of the parameter 'forward limit' comprises the sub-parameters 'layout object' and 'logical object' whose values are 'null'.

Definition 9.60

1 $\forall\, cst, doby, prof, rdoby$
2 $({}_0\text{FORWARD_LIMIT_VALUE}(cst, doby, prof, rdoby) =$
3 STEP_BL_VALUE$^{9.2}(cst, doby,$ 'forward limit') ORELSE$^{9.1}$
4 $({}_1\text{STEP_DL_VALUE}^{9.13}(cst, doby,$ 'forward limit') ORELSE$^{9.1}$
5 $({}_2\text{STEP_FL_VALUE}^{9.19}(cst, doby, rdoby, prof,$ 'forward limit') ORELSE$^{9.1}$
6 $({}_3\text{STEP_G_VALUE}^{9.21}(cst, doby, rdoby, prof,$ 'forward limit', OBJECT_TYPE_VALUE$^{9.74}(cst, doby))$
7 ORELSE$^{9.1}$ ['layout object' : 'null'; 'logical object' : 'null']$_3)_2)_1)_0)$

Semiformal Description 9.61

Function "Value of the attribute 'indivisibility'"

The value of the attribute 'indivisibility' may be found by step BL (3). If this is not the case, test if the attribute value can be found by step DL (4). If this is not the case, test if the attribute value can be found by step FL (5). If this is not the case, test if the attribute value can be found by step G (6). If this is not the case, the value of the attribute 'indivisibility' is 'null'.

Definition 9.61

1 $\forall\, cst, doby, prof, rdoby$
2 $({}_0\text{INDIVISIBILITY_VALUE}(cst, doby, prof, rdoby) =$
3 STEP_BL_VALUE$^{9.2}(cst, doby,$ 'indivisibility') ORELSE$^{9.1}$
4 $({}_1\text{STEP_DL_VALUE}^{9.13}(cst, doby,$ 'indivisibility') ORELSE$^{9.1}$
5 $({}_2\text{STEP_FL_VALUE}^{9.19}(cst, doby, rdoby, prof,$ 'indivisibility') ORELSE$^{9.1}$
6 $({}_3\text{STEP_G_VALUE}^{9.21}(cst, doby, rdoby, prof,$ 'indivisibility', OBJECT_TYPE_VALUE$^{9.74}(cst, doby))$
7 ORELSE$^{9.1}$ 'null'$_3)_2)_1)_0)$

Semiformal Description 9.62

Function "Value of the attribute 'layout category'"

The value of the attribute 'layout category' may be found by step BL (3). If this is not the case, test if the attribute value can be found by step DL (4). If this is not the case, test if the attribute value can be found by step FL (5). If this is not the case, test if the attribute value can be found by step G (6). If this is not the case, the value of the attribute 'layout category' is 'null'.

Definition 9.62

1 $\forall\, cst, doby, prof, rdoby$
2 $({}_0\text{LAYOUT_CATEGORY_VALUE}(cst, doby, prof, rdoby) =$
3 STEP_BL_VALUE$^{9.2}(cst, doby,$ 'layout category') ORELSE$^{9.1}$
4 $({}_1\text{STEP_DL_VALUE}^{9.13}(cst, doby,$ 'layout category') ORELSE$^{9.1}$
5 $({}_2\text{STEP_FL_VALUE}^{9.19}(cst, doby, rdoby, prof,$ 'layout category') ORELSE$^{9.1}$
6 $({}_3\text{STEP_G_VALUE}^{9.21}(cst, doby, rdoby, prof,$ 'layout category', OBJECT_TYPE_VALUE$^{9.74}(cst, doby))$
7 ORELSE$^{9.1}$ 'null'$_3)_2)_1)_0)$

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 9.63

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

The value of the attribute 'layout object class' may be found by step BL (3). If this is not the case, test if the attribute value can be found by step DL (4). If this is not the case, test if the attribute value can be found by step FL (5). If this is not the case, test if the attribute value can be found by step G (6, 7). If this is not the case, the value of the attribute 'layout object class' is 'null'.

Definition 9.63

```
1  ∀ cst, doby, prof, rdoby
2  (0 LAYOUT_OBJECT_CLASS_VALUE(cst, doby, prof, rdoby) =
3    STEP_BL_VALUE^9.2(cst, doby, 'layout object class') ORELSE^9.1
4    (1 STEP_DL_VALUE^9.13(cst, doby, 'layout object class') ORELSE^9.1
5      (2 STEP_FL_VALUE^9.19(cst, doby, rdoby, prof, 'layout object class') ORELSE^9.1
6        (3 STEP_G_VALUE^9.21(cst, doby, rdoby, prof, 'layout object class',
7                             OBJECT_TYPE_VALUE^9.74(cst, doby))
8          ORELSE^9.1 'null'3)2)1)0)
```

Semiformal Description 9.64

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 9.64

```
1  ∀ cst, doby, prof, rdoby
2  (0 LAYOUT_PATH_VALUE(cst, doby, prof, rdoby) =
3    IF IsAttributeSet^2.61(cst) and 'layout path' ∈ NAMS^1.18(cst) and
4       not IsPlaceholder^1.19(C ^cst • 'layout path')
5    THEN C ^cst • 'layout path'
6    ELSE STEP_C_VALUE^9.10(cst, doby, 'layout path') ORELSE^9.1
7    (1 STEP_E_VALUE^9.14(cst, doby, rdoby, prof, 'layout path') ORELSE^9.1
8      (2 STEP_G_VALUE^9.21(cst, doby, rdoby, prof, 'layout path', OBJECT_TYPE_VALUE^9.74(cst, doby))
9        ORELSE^9.1 (3 STEP_H_VALUE^9.26(prof, 'layout path')
10         ORELSE^9.1 '270°'3)2)1)0)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 9.65

Function "Value of the attribute 'layout stream categories'"

The value of the attribute 'layout stream 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, the value of the attribute 'layout stream categories' is an empty string.

Definition 9.65

```
1  ∀ cst, doby, prof, rdoby
2  (0 LAYOUT_STREAM_CATEGORIES_VALUE(cst, doby, prof, rdoby) =
3    IF IsAttributeSet^2.61(cst) and 'layout stream categories' ∈ NAMS^1.18(cst) and
4       not IsPlaceholder^1.19(C ^cst • 'layout stream categories')
5    THEN C ^cst • 'layout stream categories'
6    ELSE STEP_C_VALUE^9.10(cst, doby, 'layout stream categories') ORELSE^9.1
7    (1 STEP_E_VALUE^9.14(cst, doby, rdoby, prof, 'layout stream categories') ORELSE^9.1
8       ORELSE^9.1 ''1)0)
```

Semiformal Description 9.66

Function "Value of the attribute 'layout stream sub-categories'"

The value of the attribute 'layout stream sub-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, the value of the attribute 'layout stream sub-categories' is an empty string.

Definition 9.66

```
1  ∀ cst, doby, prof, rdoby
2  (0 LAYOUT_STREAM_SUBCATEGORIES_VALUE(cst, doby, prof, rdoby) =
3    IF IsAttributeSet^2.61(cst) and 'layout stream sub-categories' ∈ NAMS^1.18(cst) and
4       not IsPlaceholder^1.19(C ^cst • 'layout stream sub-categories')
5    THEN C ^cst • 'layout stream sub-categories'
6    ELSE STEP_C_VALUE^9.10(cst, doby, 'layout stream sub-categories') ORELSE^9.1
7    (1 STEP_E_VALUE^9.14(cst, doby, rdoby, prof, 'layout stream sub-categories') ORELSE^9.1
8       ORELSE^9.1 ''1)0)
```

Semiformal Description 9.67

Function "Value of the attribute 'logical stream category'"

The value of the attribute 'logical stream category' may be found by step BL (3). If this is not the case, test if the attribute value can be found by step DL (4). If this is not the case, test if the attribute value can be found by step FL (5, 6). If this is not the case, the value of the attribute 'logical stream category' is 'null'.

Definition 9.67

```
1  ∀ cst, doby, prof, rdoby
2  (0 LOGICAL_STREAM_CATEGORY_VALUE(cst, doby, prof, rdoby) =
3    STEP_BL_VALUE^9.2(cst, doby, 'logical stream category') ORELSE^9.1
4    (1 STEP_DL_VALUE^9.13(cst, doby, rdoby, prof, 'logical stream category') ORELSE^9.1
5      (2 STEP_FL_VALUE^9.19(cst, doby, rdoby, prof, 'logical stream category',
6                            OBJECT_TYPE_VALUE^9.74(cst, doby))
7        ORELSE^9.1 'null'2)1)0)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 9.68

Function "Value of the attribute 'logical stream sub-category'"

The value of the attribute 'logical stream sub-category' may be found by step BL (3). If this is not the case, test if the attribute value can be found by step DL (4). If this is not the case, test if the attribute value can be found by step FL (5, 6). If this is not the case, the value of the attribute 'logical stream sub-category' is 'null'.

Definition 9.68

1 $\forall\, cst, doby, prof, rdoby$
2 $({}_0$LOGICAL_STREAM_SUBCATEGORY_VALUE$(cst, doby, prof, rdoby) =$
3 STEP_BL_VALUE$^{9.2}(cst, doby,$ 'logical stream sub-category') ORELSE$^{9.1}$
4 $({}_1$STEP_DL_VALUE$^{9.13}(cst, doby, rdoby, prof,$ 'logical stream sub-category') ORELSE$^{9.1}$
5 $({}_2$STEP_FL_VALUE$^{9.19}(cst, doby, rdoby, prof,$ 'logical stream sub-category',
6 OBJECT_TYPE_VALUE$^{9.74}(cst, doby)$)
7 ORELSE$^{9.1}$ 'null'$_2)_1)_0)$

Semiformal Description 9.69

Function "Value of the attribute 'medium type'"

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

Definition 9.69

1 $\forall\, cst, doby, prof, rdoby$
2 $({}_0$MEDIUM_TYPE_VALUE$(cst, doby, prof, rdoby) =$
3 ['colour of medium' : COLOUR_OF_MEDIUM_VALUE$^{9.72}(cst, doby, prof, rdoby)$;
4 'side of sheet' : SIDE_OF_SHEET_VALUE$^{9.71}(cst, doby, prof, rdoby)$;
5 'nominal page size' : NOMINAL_PAGE_SIZE_VALUE$^{9.70}(cst, doby, prof, rdoby)]_0)$

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 9.70

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 9.70

```
1   ∀ cst, doby, prof, rdoby
2   (0 NOMINAL_PAGE_SIZE_VALUE(cst, doby, prof, rdoby) =
3     IF IsAttributeSet^2.61(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^9.29(cst, doby, 'medium type', 'nominal page size') ORELSE^9.1
8     (1 STEP_E_PAR_VALUE^9.31(cst, doby, rdoby, prof, 'medium type', 'nominal page size') ORELSE^9.1
9       (2 STEP_G_PAR_VALUE^9.33(cst, doby, rdoby, prof, 'medium type', 'nominal page size',
10                                 OBJECT_TYPE_VALUE^9.74(cst, doby)) ORELSE^9.1
11        (3 STEP_H_PAR_VALUE^9.34(prof, 'medium type', 'nominal page size')
12          ORELSE^9.1 [ → 9920 → 14030 → ]3)2)1)0)
```

Semiformal Description 9.71

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 9.71

```
1   ∀ cst, doby, prof, rdoby
2   (0 SIDE_OF_SHEET_VALUE(cst, doby, prof, rdoby) =
3     IF IsAttributeSet^2.61(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^9.29(cst, doby, 'medium type', 'side of sheet') ORELSE^9.1
8     (1 STEP_E_PAR_VALUE^9.31(cst, doby, rdoby, prof, 'medium type', 'side of sheet') ORELSE^9.1
9       (2 STEP_G_PAR_VALUE^9.33(cst, doby, rdoby, prof, 'medium type', 'side of sheet',
10                                 OBJECT_TYPE_VALUE^9.74(cst, doby)) ORELSE^9.1
11        (3 STEP_H_PAR_VALUE^9.34(prof, 'medium type', 'side of sheet')
12          ORELSE^9.1 'unspecified'3)2)1)0)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 9.72

Function "Value of the parameter 'colour of medium'"

The value of the parameter 'colour of medium' 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, 10). 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 'colour of medium' is 'unspecified'.

Definition 9.72

```
1   ∀ cst, doby, prof, rdoby
2   (0 COLOUR_OF_MEDIUM_VALUE(cst, doby, prof, rdoby) =
3     IF IsAttributeSet2.61(cst) and 'medium type' ∈ NAMS1.18(cst) and
4        'colour of medium' ∈ NAMS1.18(C ^cst • 'medium type') and
5        not IsPlaceholder1.19(C ^cst • 'medium type' • 'colour of medium')
6     THEN C ^cst • 'medium type' • 'colour of medium'
7     ELSE STEP_C_PAR_VALUE9.29(cst, doby, 'medium type', 'colour of medium') ORELSE9.1
8     (1 STEP_E_PAR_VALUE9.31(cst, doby, rdoby, prof, 'medium type', 'colour of medium') ORELSE9.1
9       (2 STEP_G_PAR_VALUE9.33(cst, doby, rdoby, prof, 'medium type', 'colour of medium',
10                                OBJECT_TYPE_VALUE9.74(cst, doby)) ORELSE9.1
11        (3 STEP_H_PAR_VALUE9.34(prof, 'medium type', 'colour of medium')
12          ORELSE9.1 'unspecified'3)2)1)0)
```

Semiformal Description 9.73

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

The value of the attribute 'new layout object' may be found by step BL (3). If this is not the case, test if the attribute value can be found by step DL (4). If this is not the case, test if the attribute value can be found by step FL (5). If this is not the case, test if the attribute value can be found by step G (6, 7). If this is not the case, the value of the attribute 'new layout object' is 'null'.

Definition 9.73

```
1   ∀ cst, doby, prof, rdoby
2   (0 NEW_LAYOUT_OBJECT_VALUE(cst, doby, prof, rdoby) =
3     STEP_BL_VALUE9.2(cst, doby, 'new layout object') ORELSE9.1
4     (1 STEP_DL_VALUE9.13(cst, doby, 'new layout object') ORELSE9.1
5       (2 STEP_FL_VALUE9.19(cst, doby, rdoby, prof, 'new layout object') ORELSE9.1
6         (3 STEP_G_VALUE9.21(cst, doby, rdoby, prof, 'new layout object',
7                               OBJECT_TYPE_VALUE9.74(cst, doby))
8           ORELSE9.1 'null'3)2)1)0)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 9.74

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 9.74

1 $\forall\, cst, doby$
2 $({}_0\text{OBJECT_TYPE_VALUE}(cst, doby) =$
3 IF $\text{IsAttributeSet}^{2.61}(cst)$ *and* 'object type' $\in \text{NAMS}^{1.18}(cst)$ *and*
4 *not* $\text{IsPlaceholder}^{1.19}(C\ \hat{}\ cst \bullet \text{'object type'})$
5 THEN $C\ \hat{}\ cst \bullet$ 'object type'
6 ELSE $\text{STEP_C_VALUE}^{9.10}(cst, doby, \text{'object type'})_0)$

Semiformal Description 9.75

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 9.75

1 $\forall\, cst, doby, prof, rdoby$
2 $({}_0\text{OFFSET_VALUE}(cst, doby, prof, rdoby) =$
3 ['leading offset' : $\text{LEADING_OFFSET_VALUE}^{9.76}(cst, doby, prof, rdoby)$;
4 'trailing offset' : $\text{TRAILING_OFFSET_VALUE}^{9.77}(cst, doby, prof, rdoby)$;
5 'left-hand offset' : $\text{LEFTHAND_OFFSET_VALUE}^{9.78}(cst, doby, prof, rdoby)$;
6 'right-hand offset' : $\text{RIGHTHAND_OFFSET_VALUE}^{9.79}(cst, doby, prof, rdoby)]_0)$

Semiformal Description 9.76

Function "Value of the parameter 'leading offset'"

The value of the parameter 'leading offset' may be found by step BL (3). If this is not the case, test if the attribute value can be found by step DL (4). If this is not the case, test if the attribute value can be found by step FL (5). If this is not the case, test if the attribute value can be found by step G (6, 7). If this is not the case, the value of the parameter 'leading offset' is 0.

Definition 9.76

1 $\forall\, cst, doby, prof, rdoby$
2 $({}_0\text{LEADING_OFFSET_VALUE}(cst, doby, prof, rdoby) =$
3 $\text{STEP_BL_PAR_VALUE}^{9.27}(cst, doby, \text{'offset'}, \text{'leading offset'})\ \text{ORELSE}^{9.1}$
4 $({}_1\text{STEP_DL_PAR_VALUE}^{9.30}(cst, doby, \text{'offset'}, \text{'leading offset'})\ \text{ORELSE}^{9.1}$
5 $({}_2\text{STEP_FL_PAR_VALUE}^{9.32}(cst, doby, rdoby, prof, \text{'offset'}, \text{'leading offset'})\ \text{ORELSE}^{9.1}$
6 $({}_3\text{STEP_G_PAR_VALUE}^{9.33}(cst, doby, rdoby, prof, \text{'offset'}, \text{'leading offset'},$
7 $\text{OBJECT_TYPE_VALUE}^{9.74}(cst, doby))$
8 $\text{ORELSE}^{9.1}\ 0_3)_2)_1)_0)$

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 9.77

Function "Value of the parameter 'trailing offset'"

The value of the parameter 'trailing offset' may be found by step BL (3). If this is not the case, test if the attribute value can be found by step DL (4). If this is not the case, test if the attribute value can be found by step FL (5). If this is not the case, test if the attribute value can be found by step G (6, 7). If this is not the case, the value of the parameter 'trailing offset' is 0.

Definition 9.77

$\forall\, cst, doby, prof, rdoby$
$({}_0\text{TRAILING_OFFSET_VALUE}(cst, doby, prof, rdoby) =$
STEP_BL_PAR_VALUE$^{9.27}(cst, doby,$ 'offset', 'trailing offset') ORELSE$^{9.1}$
$({}_1$STEP_DL_PAR_VALUE$^{9.30}(cst, doby,$ 'offset', 'trailing offset') ORELSE$^{9.1}$
$({}_2$STEP_FL_PAR_VALUE$^{9.32}(cst, doby, rdoby, prof,$ 'offset', 'trailing offset') ORELSE$^{9.1}$
$({}_3$STEP_G_PAR_VALUE$^{9.33}(cst, doby, rdoby, prof,$ 'offset', 'trailing offset',
OBJECT_TYPE_VALUE$^{9.74}(cst, doby)$)
ORELSE$^{9.1}$ $0_3)_2)_1)_0)$

Semiformal Description 9.78

Function "Value of the parameter 'left-hand offset'"

The value of the parameter 'left-hand offset' may be found by step BL (3). If this is not the case, test if the attribute value can be found by step DL (4). If this is not the case, test if the attribute value can be found by step FL (5). If this is not the case, test if the attribute value can be found by step G (6, 7). If this is not the case, the value of the parameter 'left-hand offset' is 0.

Definition 9.78

$\forall\, cst, doby, prof, rdoby$
$({}_0\text{LEFTHAND_OFFSET_VALUE}(cst, doby, prof, rdoby) =$
STEP_BL_PAR_VALUE$^{9.27}(cst, doby,$ 'offset', 'left-hand offset') ORELSE$^{9.1}$
$({}_1$STEP_DL_PAR_VALUE$^{9.30}(cst, doby,$ 'offset', 'left-hand offset') ORELSE$^{9.1}$
$({}_2$STEP_FL_PAR_VALUE$^{9.32}(cst, doby, rdoby, prof,$ 'offset', 'left-hand offset') ORELSE$^{9.1}$
$({}_3$STEP_G_PAR_VALUE$^{9.33}(cst, doby, rdoby, prof,$ 'offset', 'left-hand offset',
OBJECT_TYPE_VALUE$^{9.74}(cst, doby)$)
ORELSE$^{9.1}$ $0_3)_2)_1)_0)$

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 9.79

Function "Value of the parameter 'right-hand offset'"

The value of the parameter 'right-hand offset' may be found by step BL (3). If this is not the case, test if the attribute value can be found by step DL (4). If this is not the case, test if the attribute value can be found by step FL (5). If this is not the case, test if the attribute value can be found by step G (6, 7). If this is not the case, the value of the parameter 'right-hand offset' is 0.

Definition 9.79

```
1  ∀ cst, doby, prof, rdoby
2  (0 RIGHTHAND_OFFSET_VALUE(cst, doby, prof, rdoby) =
3    STEP_BL_PAR_VALUE^9.27(cst, doby, 'offset', 'right-hand offset') ORELSE^9.1
4    (1 STEP_DL_PAR_VALUE^9.30(cst, doby, 'offset', 'right-hand offset') ORELSE^9.1
5      (2 STEP_FL_PAR_VALUE^9.32(cst, doby, rdoby, prof, 'offset', 'right-hand offset') ORELSE^9.1
6        (3 STEP_G_PAR_VALUE^9.33(cst, doby, rdoby, prof, 'offset', 'right-hand offset',
7                                 OBJECT_TYPE_VALUE^9.74(cst, doby))
8          ORELSE^9.1 0 3) 2) 1) 0)
```

Semiformal Description 9.80

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 9.4.3.4).

Definition 9.80

```
1  ∀ cst, doby, prof, rdoby
2  (0 PAGE_POSITION_VALUE(cst, doby, prof, rdoby) =
3    IF IsAttributeSet^2.61(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^9.10(cst, doby, 'page position') ORELSE^9.1
7    (1 STEP_E_VALUE^9.14(cst, doby, rdoby, prof, 'page position') ORELSE^9.1
8      (2 STEP_G_VALUE^9.21(cst, doby, rdoby, prof, 'page position',
9                           OBJECT_TYPE_VALUE^9.74(cst, doby)) ORELSE^9.1
10       (3 STEP_H_VALUE^9.26(prof, 'page position')
11         ORELSE^9.1 MINIMUM_EDGE_LOSS ^9.81(cst) 3) 2) 1) 0)
```

Semiformal Description 9.81

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 9.81

```
1  ∀ cst
2  (0 IsAtom(MINIMUM_EDGE_LOSS (cst)) 0)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 9.82

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 9.82

1 $\forall\, cst, doby, prof, rdoby$
2 $({}_0$PERMITTED_CATEGORIES_VALUE$(cst, doby, prof, rdoby) =$
3 IF IsAttributeSet$^{2.61}(cst)$ *and* 'permitted categories' $\in$ 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$^{9.10}(cst, doby,$ 'permitted categories') ORELSE$^{9.1}$
7 $({}_1$STEP_E_VALUE$^{9.14}(cst, doby, rdoby, prof,$ 'permitted categories') ORELSE$^{9.1}$
8 $({}_2$STEP_G_VALUE$^{9.21}(cst, doby, rdoby, prof,$ 'permitted categories',
9 OBJECT_TYPE_VALUE$^{9.74}(cst, doby)$)
10 ORELSE$^{9.1}$ 'null'$_2)_1)_0$)

Semiformal Description 9.83

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 9.83

1 $\forall\, cst, doby, prof, rdoby$
2 $({}_0$POSITION_VALUE$(cst, doby, prof, rdoby) =$
3 ['fixed position' : FIXED_POSITION_VALUE$^{9.84}(cst, doby, prof, rdoby)]_0$)

Semiformal Description 9.84

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 9.84

1 $\forall\, cst, doby, prof, rdoby$
2 $({}_0$FIXED_POSITION_VALUE$(cst, doby, prof, rdoby) =$
3 ['horizontal position' : HORIZONTAL_POSITION_VALUE$^{9.85}(cst, doby, prof, rdoby)$;
4 'vertical position' : VERTICAL_POSITION_VALUE$^{9.86}(cst, doby, prof, rdoby)]_0$)

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 9.85

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 9.85

$\forall\, cst, doby, prof, rdoby$
$({}_0\text{HORIZONTAL_POSITION_VALUE}(cst, doby, prof, rdoby) =$
IF $\text{IsAttributeSet}^{2.61}(cst)$ *and* 'position' $\in \text{NAMS}^{1.18}(cst)$ *and*
'horizontal position' $\in \text{NAMS}^{1.18}(\text{C}\ \hat{}cst \bullet \text{'position'} \bullet \text{'fixed position'})$ *and*
not $\text{IsPlaceholder}^{1.19}(\text{C}\ \hat{}cst \bullet \text{'position'} \bullet \text{'fixed position'} \bullet \text{'horizontal position'})$
THEN C $\hat{}cst \bullet$ 'position' $\bullet$ 'fixed position' $\bullet$ 'horizontal position'
ELSE $\text{STEP_C_SUBPAR_VALUE}^{9.39}(cst, doby, \text{'position'}, \text{'fixed position'}, \text{'horizontal position'})$ $\text{ORELSE}^{9.1}$
$({}_1\text{STEP_E_SUBPAR_VALUE}^{9.40}(cst, doby, rdoby, prof, \text{'position'}, \text{'fixed position'},$
'horizontal position') $\text{ORELSE}^{9.1}$
$({}_2\text{STEP_G_SUBPAR_VALUE}^{9.41}(cst, doby, rdoby, prof, \text{'position'}, \text{'fixed position'},$
'horizontal position', $\text{OBJECT_TYPE_VALUE}^{9.74}(cst, doby))$
$\text{ORELSE}^{9.1}\ 0_2)_1)_0)$

Semiformal Description 9.86

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 9.86

$\forall\, cst, doby, prof, rdoby$
$({}_0\text{VERTICAL_POSITION_VALUE}(cst, doby, prof, rdoby) =$
IF $\text{IsAttributeSet}^{2.61}(cst)$ *and* 'position' $\in \text{NAMS}^{1.18}(cst)$ *and*
'vertical position' $\in \text{NAMS}^{1.18}(\text{C}\ \hat{}cst \bullet \text{'position'} \bullet \text{'fixed position'})$ *and*
not $\text{IsPlaceholder}^{1.19}(\text{C}\ \hat{}cst \bullet \text{'position'} \bullet \text{'fixed position'} \bullet \text{'vertical position'})$
THEN C $\hat{}cst \bullet$ 'position' $\bullet$ 'fixed position' $\bullet$ 'vertical position'
ELSE $\text{STEP_C_SUBPAR_VALUE}^{9.39}(cst, doby, \text{'position'}, \text{'fixed position'}, \text{'vertical position'})$ $\text{ORELSE}^{9.1}$
$({}_1\text{STEP_E_SUBPAR_VALUE}^{9.40}(cst, doby, rdoby, prof, \text{'position'}, \text{'fixed position'},$
'vertical position') $\text{ORELSE}^{9.1}$
$({}_2\text{STEP_G_SUBPAR_VALUE}^{9.41}(cst, doby, rdoby, prof, \text{'position'}, \text{'fixed position'},$
'vertical position', $\text{OBJECT_TYPE_VALUE}^{9.74}(cst, doby))$
$\text{ORELSE}^{9.1}\ 0_2)_1)_0)$

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 9.87

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 9.87

```
1  ∀ cst, doby, prof, rdoby
2  (0 PROTECTION_VALUE(cst, doby, prof, rdoby) =
3    IF IsAttributeSet^2.61(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^9.10(cst, doby, 'protection') ORELSE^9.1
7    (1 STEP_E_VALUE^9.14(cst, doby, rdoby, prof, 'protection') ORELSE^9.1
8      (2 STEP_G_VALUE^9.21(cst, doby, rdoby, prof, 'protection', OBJECT_TYPE_VALUE^9.74(cst, doby))
9        ORELSE^9.1 'unprotected'2)1)0)
```

Semiformal Description 9.88

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

The value of the attribute 'same layout object' may be found by step BL (3). If this is not the case, test if the attribute value can be found by step DL (4). If this is not the case, test if the attribute value can be found by step FL (5). If this is not the case, test if the attribute value can be found by step G (6, 7). 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 9.88

```
1  ∀ cst, doby, prof, rdoby
2  (0 SAME_LAYOUTOBJECT_VALUE(cst, doby, prof, rdoby) =
3    STEP_BL_VALUE^9.2(cst, doby, 'same layout object') ORELSE^9.1
4    (1 STEP_DL_VALUE^9.13(cst, doby, 'same layout object') ORELSE^9.1
5      (2 STEP_FL_VALUE^9.19(cst, doby, rdoby, prof, 'same layout object') ORELSE^9.1
6        (3 STEP_G_VALUE^9.21(cst, doby, rdoby, prof, 'same layout object',
7                             OBJECT_TYPE_VALUE^9.74(cst, doby))
8          ORELSE^9.1 ['logical object' : 'null']3)2)1)0)
```

Semiformal Description 9.89

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 9.89

```
1  ∀ cst, doby, prof, rdoby
2  (0 SEPARATION_VALUE(cst, doby, prof, rdoby) =
3    ['leading-edge' : LEADING_EDGE_VALUE^9.90(cst, doby, prof, rdoby);
4     'trailing-edge' : TRAILING_EDGE_VALUE^9.91(cst, doby, prof, rdoby);
5     'centre separation' : CENTRE_SEPARATION_VALUE^9.92(cst, doby, prof, rdoby)]0)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 9.90

Function "Value of the parameter 'leading-edge'"

The value of the parameter 'leading-edge' may be found by step BL (3). If this is not the case, test if the attribute value can be found by step DL (4). If this is not the case, test if the attribute value can be found by step FL (5). If this is not the case, test if the attribute value can be found by step G (6, 7). If this is not the case, the value of the parameter 'leading-edge' is 0.

Definition 9.90

$$
\begin{aligned}
&\forall\, cst, doby, prof, rdoby \\
&({}_0\mathrm{LEADING_EDGE_VALUE}(cst, doby, prof, rdoby) = \\
&\quad \mathrm{STEP_BL_PAR_VALUE}^{9.27}(cst, doby, \text{'separation'}, \text{'leading-edge'})\ \mathrm{ORELSE}^{9.1} \\
&\quad ({}_1\mathrm{STEP_DL_PAR_VALUE}^{9.30}(cst, doby, \text{'separation'}, \text{'leading-edge'})\ \mathrm{ORELSE}^{9.1} \\
&\qquad ({}_2\mathrm{STEP_FL_PAR_VALUE}^{9.32}(cst, doby, rdoby, prof, \text{'separation'}, \text{'leading-edge'})\ \mathrm{ORELSE}^{9.1} \\
&\qquad\quad ({}_3\mathrm{STEP_G_PAR_VALUE}^{9.33}(cst, doby, rdoby, prof, \text{'separation'}, \text{'leading-edge'}, \\
&\qquad\qquad\qquad \mathrm{OBJECT_TYPE_VALUE}^{9.74}(cst, doby)) \\
&\qquad\quad\ \ \mathrm{ORELSE}^{9.1}\ 0_3)_2)_1)_0)
\end{aligned}
$$

Semiformal Description 9.91

Function "Value of the parameter 'trailing-edge'"

The value of the parameter 'trailing-edge' may be found by step BL (3). If this is not the case, test if the attribute value can be found by step DL (4). If this is not the case, test if the attribute value can be found by step FL (5). If this is not the case, test if the attribute value can be found by step G (6, 7). If this is not the case, the value of the parameter 'trailing-edge' is 0.

Definition 9.91

$$
\begin{aligned}
&\forall\, cst, doby, prof, rdoby \\
&({}_0\mathrm{TRAILING_EDGE_VALUE}(cst, doby, prof, rdoby) = \\
&\quad \mathrm{STEP_BL_PAR_VALUE}^{9.27}(cst, doby, \text{'separation'}, \text{'trailing-edge'})\ \mathrm{ORELSE}^{9.1} \\
&\quad ({}_1\mathrm{STEP_DL_PAR_VALUE}^{9.30}(cst, doby, \text{'separation'}, \text{'trailing-edge'})\ \mathrm{ORELSE}^{9.1} \\
&\qquad ({}_2\mathrm{STEP_FL_PAR_VALUE}^{9.32}(cst, doby, rdoby, prof, \text{'separation'}, \text{'trailing-edge'})\ \mathrm{ORELSE}^{9.1} \\
&\qquad\quad ({}_3\mathrm{STEP_G_PAR_VALUE}^{9.33}(cst, doby, rdoby, prof, \text{'separation'}, \text{'trailing-edge'}, \\
&\qquad\qquad\qquad \mathrm{OBJECT_TYPE_VALUE}^{9.74}(cst, doby)) \\
&\qquad\quad\ \ \mathrm{ORELSE}^{9.1}\ 0_3)_2)_1)_0)
\end{aligned}
$$

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Semiformal Description 9.92

Function "Value of the parameter 'centre separation'"

The value of the parameter 'centre separation' may be found by step BL (3). If this is not the case, test if the attribute value can be found by step DL (4). If this is not the case, test if the attribute value can be found by step FL (5). If this is not the case, test if the attribute value can be found by step G (6, 7). If this is not the case, the value of the parameter 'centre separation' is 0.

Definition 9.92

$$
\begin{array}{l}
\forall\, cst, doby, prof, rdoby \\
({}_0\mathrm{CENTRE_SEPARATION_VALUE}(cst, doby, prof, rdoby) = \\
\quad \mathrm{STEP_BL_PAR_VALUE}^{9.27}(cst, doby, \text{'separation'}, \text{'centre separation'})\ \mathrm{ORELSE}^{9.1} \\
\quad ({}_1\mathrm{STEP_DL_PAR_VALUE}^{9.30}(cst, doby, \text{'separation'}, \text{'centre separation'})\ \mathrm{ORELSE}^{9.1} \\
\quad\quad ({}_2\mathrm{STEP_FL_PAR_VALUE}^{9.32}(cst, doby, rdoby, prof, \text{'separation'}, \text{'centre separation'})\ \mathrm{ORELSE}^{9.1} \\
\quad\quad\quad ({}_3\mathrm{STEP_G_PAR_VALUE}^{9.33}(cst, doby, rdoby, prof, \text{'separation'}, \text{'centre separation'}, \\
\quad\quad\quad\quad\quad \mathrm{OBJECT_TYPE_VALUE}^{9.74}(cst, doby)) \\
\quad\quad\quad\quad \mathrm{ORELSE}^{9.1}\ 0_3)_2)_1)_0)
\end{array}
$$

Semiformal Description 9.93

Function "Value of the attribute 'synchronization'"

The value of the attribute 'synchronization' may be found by step BL (3). If this is not the case, test if the attribute value can be found by step DL (4). If this is not the case, test if the attribute value can be found by step FL (5). If this is not the case, test if the attribute value can be found by step G (6). If this is not the case, the value of the attribute 'synchronization' is 'null'.

Definition 9.93

$$
\begin{array}{l}
\forall\, cst, doby, prof, rdoby \\
({}_0\mathrm{SYNCHRONIZATION_VALUE}(cst, doby, prof, rdoby) = \\
\quad \mathrm{STEP_BL_VALUE}^{9.2}(cst, doby, \text{'synchronization'})\ \mathrm{ORELSE}^{9.1} \\
\quad ({}_1\mathrm{STEP_DL_VALUE}^{9.13}(cst, doby, \text{'synchronization'})\ \mathrm{ORELSE}^{9.1} \\
\quad\quad ({}_2\mathrm{STEP_FL_VALUE}^{9.19}(cst, doby, rdoby, prof, \text{'synchronization'})\ \mathrm{ORELSE}^{9.1} \\
\quad\quad\quad ({}_3\mathrm{STEP_G_VALUE}^{9.21}(cst, doby, rdoby, prof, \text{'synchronization'}, \mathrm{OBJECT_TYPE_VALUE}^{9.74}(cst.doby)) \\
\quad\quad\quad\quad \mathrm{ORELSE}^{9.1}\ \text{'null'}_3)_2)_1)_0)
\end{array}
$$

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Semiformal Description 9.94

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 9.94

```
1   ∀ cst, doby, prof, rdoby
2   (0 TRANSPARENCY_VALUE(cst, doby, prof, rdoby) =
3     IF IsAttributeSet2.61(cst) and 'transparency' ∈ NAMS1.18(cst) and
4        not IsPlaceholder1.19(C ^cst • 'transparency')
5     THEN C ^cst • 'transparency'
6     ELSE STEP_BP_VALUE9.4(cst, doby, 'transparency') ORELSE9.1
7     (1 STEP_C_VALUE9.10(cst, doby, 'transparency') ORELSE9.1
8       (2 STEP_DP_VALUE9.12(cst, doby, 'transparency') ORELSE9.1
9         (3 STEP_E_VALUE9.14(cst, doby, rdoby, prof, 'transparency') ORELSE9.1
10          (4 STEP_FP_VALUE9.17(cst, doby, rdoby, prof, 'transparency') ORELSE9.1
11            (5 STEP_G_VALUE9.21(cst, doby, rdoby, prof, 'transparency',
12                                OBJECT_TYPE_VALUE9.74(cst, doby))
13              ORELSE9.1(6 STEP_H_VALUE9.26(prof, 'transparency')
14               ORELSE9.1 'transparent'6)5)4)3)2)1)0)
```

Semiformal Description 9.95

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 9.95

```
1   ∀ cst, doby, prof
2   (0 TYPE_OF_CODING_VALUE(cst, doby, prof) =
3     IF IsAttributeSet2.61(cst) and 'type of coding' ∈ NAMS1.18(cst) and
4        not IsPlaceholder1.19(C ^cst • 'type of coding')
5     THEN C ^cst • 'type of coding'
6     ELSE STEP_C_VALUE9.10(cst, doby, 'type of coding') ORELSE9.1
7     (1 STEP_H_VALUE9.26(prof, 'type of coding') ORELSE9.1
8       (2 IF IsCharacterContentPortionDescription6.11(cst) THEN '2 8 3 6 0' ELSE
9         (3 IF IsRasterGraphicsContentPortionDescription7.7(cst) THEN '2 8 3 7 0' ELSE
10          (4 IF IsGeometricGraphicsContentPortionDescription8.5(cst) THEN '2 8 3 8 0'4)3)2)1)0)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Semiformal Description 9.96

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 9.96

```
1  ∀ cst, doby, prof, rdoby
2  (0 USER_READABLE_COMMENTS_VALUE(cst, doby, prof, rdoby) =
3    IF IsAttributeSet^2.61(cst) and 'user-readable comments' ∈ 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^9.10(cst, doby, 'user-readable comments') ORELSE^9.1
7    (1 STEP_E_VALUE^9.14(cst, doby, rdoby, prof, 'user-readable comments')
8      ORELSE^9.1 [ → ]1)0)
```

Semiformal Description 9.97

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 9.97

```
1  ∀ cst, doby, prof, rdoby
2  (0 USER_VISIBLE_NAME_VALUE(cst, doby, prof, rdoby) =
3    IF IsAttributeSet^2.61(cst) and 'user-visible name' ∈ 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^9.10(cst, doby, 'user-visible name') ORELSE^9.1
7    (1 STEP_E_VALUE^9.14(cst, doby, rdoby, prof, 'user-visible name')
8      ORELSE^9.1 [ → ]1)0)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 9.98

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

The value of the attribute 'colour of 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 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 'colour of layout object' is the parameter 'colour access mode' with the value 'direct', the parameter 'colour space id' with the value 0, the parameter 'colour specification' whose value is a sequence of three ones, and the parameter 'colour tolerance' with the sub-parameter 'unspecified tolerance' whose value is 'infinite'.

Definition 9.98

$$
\begin{aligned}
&\forall\, cst, doby, prof, rdoby \\
&({}_0\mathrm{COLOUR_OF_LAYOUT_OBJECT_VALUE}(cst, doby, prof, rdoby) = \\
&\quad \mathrm{IF\ IsAttributeSet}^{2.61}(cst)\ \underline{and}\ \text{'colour of layout object'} \in \mathrm{NAMS}^{1.18}(cst)\ \underline{and} \\
&\qquad \underline{not}\ \mathrm{IsPlaceholder}^{1.19}(\mathrm{C}\ \hat{}cst \bullet \text{'colour of layout object'}) \\
&\quad \mathrm{THEN\ C}\ \hat{}cst \bullet \text{'colour of layout object'} \\
&\quad \mathrm{ELSE\ STEP_BP_VALUE}^{9.4}(cst, doby, \text{'colour of layout object'})\ \mathrm{ORELSE}^{9.1} \\
&\quad ({}_1\mathrm{STEP_C_VALUE}^{9.10}(cst, doby, \text{'colour of layout object'})\ \mathrm{ORELSE}^{9.1} \\
&\qquad ({}_2\mathrm{STEP_DP_VALUE}^{9.12}(cst, doby, \text{'colour of layout object'})\ \mathrm{ORELSE}^{9.1} \\
&\qquad\quad ({}_3\mathrm{STEP_E_VALUE}^{9.14}(cst, doby, rdoby, prof, \text{'colour of layout object'})\ \mathrm{ORELSE}^{9.1} \\
&\qquad\qquad ({}_4\mathrm{STEP_FP_VALUE}^{9.17}(cst, doby, rdoby, prof, \text{'colour of layout object'})\ \mathrm{ORELSE}^{9.1} \\
&\qquad\qquad\quad ({}_5\mathrm{STEP_G_VALUE}^{9.21}(cst, doby, rdoby, prof, \text{'colour of layout object'}, \\
&\qquad\qquad\qquad\qquad \mathrm{OBJECT_TYPE_VALUE}^{9.74}(cst, doby)) \\
&\qquad\qquad\quad \mathrm{ORELSE}^{9.1}\ ({}_6\mathrm{STEP_H_VALUE}^{9.26}(prof, \text{'colour of layout object'}) \\
&\qquad\qquad\quad \mathrm{ORELSE}^{9.1}\ [\text{'colour access mode'} : \text{'direct'}; \text{'colour space id'} : 0 \\
&\qquad\qquad\qquad\qquad \text{'colour specification'} : [\to 1 \to 1 \to 1 \to]; \\
&\qquad\qquad\qquad\qquad \text{'colour tolerance'} : [\text{'unspecified tolerance'} : \text{'infinite'}]]_6)_5)_4)_3)_2)_1)_0)
\end{aligned}
$$

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Semiformal Description 9.99

Function "Value of the attribute 'object colour table'"

The value of the attribute 'object colour 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, 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 'object colour table' is the parameter 'colour space id' with the value 0 and the parameter 'colour table entries' with eight entries for colour tables as indicated.

Definition 9.99

```
1   ∀ cst, doby, prof, rdoby
2   (0 OBJECT_COLOUR_TABLE_VALUE(cst, doby, prof, rdoby) =
3     IF IsAttributeSet^2.61(cst) and 'object colour table' ∈ NAMS^1.18(cst) and
4       not IsPlaceholder^1.19(C ^cst • 'object colour table')
5     THEN C ^cst • 'object colour table'
6     ELSE STEP_BP_VALUE^9.4(cst, doby, 'object colour table') ORELSE^9.1
7     (1 STEP_C_VALUE^9.10(cst, doby, 'object colour table') ORELSE^9.1
8       (2 STEP_DP_VALUE^9.12(cst, doby, 'object colour table') ORELSE^9.1
9         (3 STEP_E_VALUE^9.14(cst, doby, rdoby, prof, 'object colour table') ORELSE^9.1
10          (4 STEP_FP_VALUE^9.17(cst, doby, rdoby, prof, 'object colour table') ORELSE^9.1
11            (5 STEP_G_VALUE^9.21(cst, doby, rdoby, prof, 'object colour table',
12                                 OBJECT_TYPE_VALUE^9.74(cst, doby))
13              ORELSE^9.1 (6 STEP_H_VALUE^9.26(prof, 'object colour table')
14               ORELSE^9.1
15                ['colour space id' : 0; 'colour table entries' :
16                 [['colour specification' : [ → 1.0 → 1.0 → 1.0 → ];
17                   'colour tolerance' : ['unspecified tolerance' : 'infinite']; 'index' : 0];
18                  ['colour specification' : [ → 0.0 → 0.0 → 0.0 → ];
19                   'colour tolerance' : ['unspecified tolerance' : 'infinite']; 'index' : 1];
20                  ['colour specification' : [ → 1.0 → 0.0 → 0.0 → ];
21                   'colour tolerance' : ['unspecified tolerance' : 'infinite']; 'index' : 2];
22                  ['colour specification' : [ → 0.0 → 1.0 → 0.0 → ];
23                   'colour tolerance' : ['unspecified tolerance' : 'infinite']; 'index' : 3];
24                  ['colour specification' : [ → 0.0 → 0.0 → 1.0 → ];
25                   'colour tolerance' : ['unspecified tolerance' : 'infinite']; 'index' : 4];
26                  ['colour specification' : [ → 1.0 → 1.0 → 0.0 → ];
27                   'colour tolerance' : ['unspecified tolerance' : 'infinite']; 'index' : 5];
28                  ['colour specification' : [ → 1.0 → 0.0 → 1.0 → ];
29                   'colour tolerance' : ['unspecified tolerance' : 'infinite']; 'index' : 6];
30                  ['colour specification' : [ → 0.0 → 1.0 → 1.0 → ];
31                   'colour tolerance' : ['unspecified tolerance' : 'infinite']; 'index' : 7]]] 6)5)4)3)2)1)0)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 9.100

Function "Value of the attribute 'content background colour'"

The value of the attribute 'content background 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, 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 'content background colour' is 'content background transparency'.

Definition 9.100

```
1   ∀ cst, doby, prof, rdoby
2   (0 CONTENT_BACKGROUND_COLOUR_VALUE(cst, doby, prof, rdoby) =
3     IF IsAttributeSet^2.61(cst) and 'content background colour' ∈ NAMS^1.18(cst) and
4        not IsPlaceholder^1.19(C ^cst • 'content background colour')
5     THEN C ^cst • 'content background colour'
6     ELSE STEP_BP_VALUE^9.4(cst, doby, 'content background colour') ORELSE^9.1
7     (1 STEP_C_VALUE^9.10(cst, doby, 'content background colour') ORELSE^9.1
8       (2 STEP_DP_VALUE^9.12(cst, doby, 'content background colour') ORELSE^9.1
9         (3 STEP_E_VALUE^9.14(cst, doby, rdoby, prof, 'content background colour') ORELSE^9.1
10          (4 STEP_FP_VALUE^9.17(cst, doby, rdoby, prof, 'content background colour') ORELSE^9.1
11            (5 STEP_G_VALUE^9.21(cst, doby, rdoby, prof, 'content background colour',
12                                 OBJECT_TYPE_VALUE^9.74(cst, doby))
13              ORELSE^9.1 (6 STEP_H_VALUE^9.26(prof, 'content background colour')
14               ORELSE^9.1 'content background transparency'6)5)4)3)2)1)0)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 9.101

Function "Value of the attribute 'content foreground colour'"

The value of the attribute 'content foreground 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, 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 'content foreground colour' is 'implementation-defined'.

Definition 9.101

```
1   ∀ cst, doby, prof, rdoby
2   (0 CONTENT_FOREGROUND_COLOUR_VALUE(cst, doby, prof, rdoby) =
3     IF IsAttributeSet^2.61(cst) and 'content foreground colour' ∈ NAMS^1.18(cst) and
4        not IsPlaceholder^1.19(C ^cst • 'content foreground colour')
5     THEN C ^cst • 'content foreground colour'
6     ELSE STEP_BP_VALUE^9.4(cst, doby, 'content foreground colour') ORELSE^9.1
7     (1 STEP_C_VALUE^9.10(cst, doby, 'content foreground colour') ORELSE^9.1
8       (2 STEP_DP_VALUE^9.12(cst, doby, 'content foreground colour') ORELSE^9.1
9         (3 STEP_E_VALUE^9.14(cst, doby, rdoby, prof, 'content foreground colour') ORELSE^9.1
10          (4 STEP_FP_VALUE^9.17(cst, doby, rdoby, prof, 'content foreground colour') ORELSE^9.1
11            (5 STEP_G_VALUE^9.21(cst, doby, rdoby, prof, 'content foreground colour',
12                                 OBJECT_TYPE_VALUE^9.74(cst, doby))
13              ORELSE^9.1 (6 STEP_H_VALUE^9.26(prof, 'content foreground colour')
14               ORELSE^9.1 'implementation-defined' 6)5)4)3)2)1)0)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 9.102

Function "Value of the attribute 'content colour table'"

The value of the attribute 'content colour 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, 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 'content colour table' is the parameter 'colour space id' with the value 0 and the parameter 'colour table entries' with eight entries for colour tables as indicated.

Definition 9.102

```
1   ∀ cst, doby, prof, rdoby
2   (0 CONTENT_COLOUR_TABLE_VALUE(cst, doby, prof, rdoby) =
3     IF IsAttributeSet^2.61(cst) and 'content colour table' ∈ NAMS^1.18(cst) and
4        not IsPlaceholder^1.19(C ^cst • 'content colour table')
5     THEN C ^cst • 'content colour table'
6     ELSE STEP_BP_VALUE^9.4(cst, doby, 'content colour table') ORELSE^9.1
7     (1 STEP_C_VALUE^9.10(cst, doby, 'content colour table') ORELSE^9.1
8       (2 STEP_DP_VALUE^9.12(cst, doby, 'content colour table') ORELSE^9.1
9         (3 STEP_E_VALUE^9.14(cst, doby, rdoby, prof, 'content colour table') ORELSE^9.1
10          (4 STEP_FP_VALUE^9.17(cst, doby, rdoby, prof, 'content colour table') ORELSE^9.1
11            (5 STEP_G_VALUE^9.21(cst, doby, rdoby, prof, 'content colour table',
12                                 OBJECT_TYPE_VALUE^9.74(cst, doby))
13             ORELSE^9.1 (6 STEP_H_VALUE^9.26(prof, 'content colour table')
14              ORELSE^9.1
15                ['colour space id' : 0; 'colour table entries' :
16                 [['colour specification' : [ → 1.0 → 1.0 → 1.0 → ];
17                   'colour tolerance' : ['unspecified tolerance' : 'infinite']; 'index' : 0];
18                  ['colour specification' : [ → 0.0 → 0.0 → 0.0 → ];
19                   'colour tolerance' : ['unspecified tolerance' : 'infinite']; 'index' : 1];
20                  ['colour specification' : [ → 1.0 → 0.0 → 0.0 → ];
21                   'colour tolerance' : ['unspecified tolerance' : 'infinite']; 'index' : 2];
22                  ['colour specification' : [ → 0.0 → 1.0 → 0.0 → ];
23                   'colour tolerance' : ['unspecified tolerance' : 'infinite']; 'index' : 3];
24                  ['colour specification' : [ → 0.0 → 0.0 → 1.0 → ];
25                   'colour tolerance' : ['unspecified tolerance' : 'infinite']; 'index' : 4];
26                  ['colour specification' : [ → 1.0 → 1.0 → 0.0 → ];
27                   'colour tolerance' : ['unspecified tolerance' : 'infinite']; 'index' : 5];
28                  ['colour specification' : [ → 1.0 → 0.0 → 1.0 → ];
29                   'colour tolerance' : ['unspecified tolerance' : 'infinite']; 'index' : 6];
30                  ['colour specification' : [ → 0.0 → 1.0 → 1.0 → ];
31                   'colour tolerance' : ['unspecified tolerance' : 'infinite']; 'index' : 7]]] 6)5)4)3)2)1)0)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

12.2.2 Defaultable attributes of ISO/IEC 8613-6

Semiformal Description 9.103

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 9.103

```
1   ∀ cst, doby, prof, rdoby
2   (0 ALIGNMENT_VALUE(cst, doby, prof, rdoby) =
3     IF IsAttributeSet^2.61(cst) and 'alignment' ∈ NAMS^1.18(cst) and
4        not IsPlaceholder^1.19(C ^cst • 'alignment')
5     THEN C ^cst • 'alignment'
6     ELSE STEP_BP_VALUE^9.4(cst, doby, 'alignment') ORELSE^9.1
7     (1 IF IsBasicLayoutObjectDescription^2.52(cst)
8       THEN STEP_C_VALUE^9.10(cst, doby, 'alignment') ORELSE^9.1
9       (2 STEP_DP_VALUE^9.12(cst, doby, 'alignment') ORELSE^9.1
10        (3 IF IsBasicLayoutObjectDescription^2.52(cst)
11          THEN STEP_E_VALUE^9.14(cst, doby, rdoby, prof, 'alignment') ORELSE^9.1
12          (4 STEP_FP_VALUE^9.17(cst, doby, rdoby, prof, 'alignment') ORELSE^9.1
13            (5 STEP_G_VALUE^9.21(cst, doby, rdoby, prof, 'alignment', OBJECT_TYPE_VALUE^9.74(cst, doby))
14              ORELSE^9.1(6 STEP_H_VALUE^9.26(prof, 'alignment')
15                ORELSE^9.1 'start-aligned'6)5)4)3)2)1)0)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 9.104

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 9.104

```
1   ∀ cst, doby, prof, rdoby
2   (0 CHARACTER_FONTS_VALUE(cst, doby, prof, rdoby) =
3     IF IsAttributeSet^2.61(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^9.4(cst, doby, 'character fonts') ORELSE^9.1
7     (1 IF IsBasicLayoutObjectDescription^2.52(cst)
8       THEN STEP_C_VALUE^9.10(cst, doby, 'character fonts') ORELSE^9.1
9       (2 STEP_DP_VALUE^9.12(cst, doby, 'character fonts') ORELSE^9.1
10        (3 IF IsBasicLayoutObjectDescription^2.52(cst)
11          THEN STEP_E_VALUE^9.14(cst, doby, rdoby, prof, 'character fonts') ORELSE^9.1
12          (4 STEP_FP_VALUE^9.17(cst, doby, rdoby, prof, 'character fonts') ORELSE^9.1
13            (5 STEP_G_VALUE^9.21(cst, doby, rdoby, prof, 'character fonts',
14                                 OBJECT_TYPE_VALUE^9.74(cst, doby))
15              ORELSE^9.1(6 STEP_H_VALUE^9.26(prof, 'character fonts')
16               ORELSE^9.1 [ : ]6)5)4)3)2)1)0)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 9.105

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 9.105

```
1   ∀ cst, doby, prof, rdoby
2   (0 CHARACTER_ORIENTATION_VALUE(cst, doby, prof, rdoby) =
3     IF IsAttributeSet2.61(cst) and 'character orientation' ∈ NAMS1.18(cst) and
4        not IsPlaceholder1.19(C ^cst • 'character orientation')
5     THEN C ^cst • 'character orientation'
6     ELSE STEP_BP_VALUE9.4(cst, doby, 'character orientation') ORELSE9.1
7     (1 IF IsBasicLayoutObjectDescription2.52(cst)
8       THEN STEP_C_VALUE9.10(cst, doby, 'character orientation') ORELSE9.1
9       (2 STEP_DP_VALUE9.12(cst, doby, 'character orientation') ORELSE9.1
10        (3 IF IsBasicLayoutObjectDescription2.52(cst)
11          THEN STEP_E_VALUE9.14(cst, doby, rdoby, prof, 'character orientation') ORELSE9.1
12          (4 STEP_FP_VALUE9.17(cst, doby, rdoby, prof, 'character orientation') ORELSE9.1
13            (5 STEP_G_VALUE9.21(cst, doby, rdoby, prof, 'character orientation',
14                                OBJECT_TYPE_VALUE9.74(cst, doby))
15              ORELSE9.1(6 STEP_H_VALUE9.26(prof, 'character orientation')
16               ORELSE9.1 '0°'6)5)4)3)2)1)0)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 9.106

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 9.106

```
1   ∀ cst, doby, prof, rdoby
2   (0 CHARACTER_PATH_VALUE(cst, doby, prof, rdoby) =
3     IF IsAttributeSet^2.61(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^9.4(cst, doby, 'character path') ORELSE^9.1
7     (1 IF IsBasicLayoutObjectDescription^2.52(cst)
8       THEN STEP_C_VALUE^9.10(cst, doby, 'character path') ORELSE^9.1
9       (2 STEP_DP_VALUE^9.12(cst, doby, 'character path') ORELSE^9.1
10        (3 IF IsBasicLayoutObjectDescription^2.52(cst)
11          THEN STEP_E_VALUE^9.14(cst, doby, rdoby, prof, 'character path') ORELSE^9.1
12          (4 STEP_FP_VALUE^9.17(cst, doby, rdoby, prof, 'character path') ORELSE^9.1
13            (5 STEP_G_VALUE^9.21(cst, doby, rdoby, prof, 'character path',
14                                 OBJECT_TYPE_VALUE^9.74(cst, doby))
15              ORELSE^9.1 (6 STEP_H_VALUE^9.26(prof, 'character path')
16               ORELSE^9.1 '0°' 6) 5) 4) 3) 2) 1) 0)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Semiformal Description 9.107

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 9.107

```
1  ∀ cst, doby, prof, rdoby
2  (0 CHARACTER_SPACING_VALUE(cst, doby, prof, rdoby) =
3    IF IsAttributeSet^2.61(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^9.4(cst, doby, 'character spacing') ORELSE^9.1
7    (1 IF IsBasicLayoutObjectDescription^2.52(cst)
8      THEN STEP_C_VALUE^9.10(cst, doby, 'character spacing') ORELSE^9.1
9      (2 STEP_DP_VALUE^9.12(cst, doby, 'character spacing') ORELSE^9.1
10       (3 IF IsBasicLayoutObjectDescription^2.52(cst)
11         THEN STEP_E_VALUE^9.14(cst, doby, rdoby, prof, 'character spacing') ORELSE^9.1
12         (4 STEP_FP_VALUE^9.17(cst, doby, rdoby, prof, 'character spacing') ORELSE^9.1
13           (5 STEP_G_VALUE^9.21(cst, doby, rdoby, prof, 'character spacing',
14                                OBJECT_TYPE_VALUE^9.74(cst, doby))
15             ORELSE^9.1 (6 STEP_H_VALUE^9.26(prof, 'character spacing')
16              ORELSE^9.1 120 × BMU_EQUIVALENT^9.108(prof)6)5)4)3)2)1)0)
```

Semiformal Description 9.108

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 9.108

```
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)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 9.109

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 9.109

```
1   ∀ cst, doby, prof, rdoby
2   (0 CODE_EXTENSION_ANNOUNCERS_VALUE(cst, doby, prof, rdoby) =
3     IF IsAttributeSet^2.61(cst) and 'code extension announcers' ∈ NAMS^1.18(cst) and
4       not IsPlaceholder^1.19(C ^cst • 'code extension announcers')
5     THEN C ^cst • 'code extension announcers'
6     ELSE STEP_BP_VALUE^9.4(cst, doby, 'code extension announcers') ORELSE^9.1
7     (1 IF IsBasicLayoutObjectDescription^2.52(cst)
8       THEN STEP_C_VALUE^9.10(cst, doby, 'code extension announcers') ORELSE^9.1
9       (2 STEP_DP_VALUE^9.12(cst, doby, 'code extension announcers') ORELSE^9.1
10        (3 IF IsBasicLayoutObjectDescription^2.52(cst)
11          THEN STEP_E_VALUE^9.14(cst, doby, rdoby, prof, 'code extension announcers') ORELSE^9.1
12          (4 STEP_FP_VALUE^9.17(cst, doby, rdoby, prof, 'code extension announcers') ORELSE^9.1
13            (5 STEP_G_VALUE^9.21(cst, doby, rdoby, prof, 'code extension announcers',
14                                 OBJECT_TYPE_VALUE^9.74(cst, doby))
15              ORELSE^9.1(6 STEP_H_VALUE^9.26(prof, 'code extension announcers')
16               ORELSE^9.1 G0_G2_ESCAPE_SEQUENCE_ANNOUNCER^9.110 6)5)4)3)2)1)0)
```

Semiformal Description 9.110

Function "G0-G2 Escape Sequence Announcer"

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

Definition 9.110

```
1   ∀ cst
2   (0 IsAtom(G0_G2_ESCAPE_SEQUENCE_ANNOUNCER)0)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 9.111

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 9.111

```
1   ∀ cst, doby, prof, rdoby
2   (0 FIRST_LINE_OFFSET_VALUE(cst, doby, prof, rdoby) =
3     IF IsAttributeSet^2.61(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^9.4(cst, doby, 'first line offset') ORELSE^9.1
7     (1 IF IsBasicLayoutObjectDescription^2.52(cst)
8       THEN STEP_C_VALUE^9.10(cst, doby, 'first line offset') ORELSE^9.1
9       (2 STEP_DP_VALUE^9.12(cst, doby, 'first line offset') ORELSE^9.1
10        (3 IF IsBasicLayoutObjectDescription^2.52(cst)
11          THEN STEP_E_VALUE^9.14(cst, doby, rdoby, prof, 'first line offset') ORELSE^9.1
12          (4 STEP_FP_VALUE^9.17(cst, doby, rdoby, prof, 'first line offset') ORELSE^9.1
13            (5 STEP_G_VALUE^9.21(cst, doby, rdoby, prof, 'first line offset',
14                                 OBJECT_TYPE_VALUE^9.74(cst, doby))
15              ORELSE^9.1(6 STEP_H_VALUE^9.26(prof, 'first line offset')
16               ORELSE^9.1 0_6)_5)_4)_3)_2)_1)_0)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 9.112

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 9.112

```
1   ∀ cst, doby, prof, rdoby
2   (0 GRAPHIC_CHARACTER_SETS_VALUE(cst, doby, prof, rdoby) =
3     IF IsAttributeSet^2.61(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^9.4(cst, doby, 'graphic character sets') ORELSE^9.1
7     (1 IF IsBasicLayoutObjectDescription^2.52(cst)
8       THEN STEP_C_VALUE^9.10(cst, doby, 'graphic character sets') ORELSE^9.1
9       (2 STEP_DP_VALUE^9.12(cst, doby, 'graphic character sets') ORELSE^9.1
10        (3 IF IsBasicLayoutObjectDescription^2.52(cst)
11          THEN STEP_E_VALUE^9.14(cst, doby, rdoby, prof, 'graphic character sets') ORELSE^9.1
12          (4 STEP_FP_VALUE^9.17(cst, doby, rdoby, prof, 'graphic character sets') ORELSE^9.1
13            (5 STEP_G_VALUE^9.21(cst, doby, rdoby, prof, 'graphic character sets',
14                                 OBJECT_TYPE_VALUE^9.74(cst, doby))
15              ORELSE^9.1(6 STEP_H_VALUE^9.26(prof, 'graphic character sets')
16               ORELSE^9.1 G0_G2_ESCAPE_SEQUENCE^9.113 6)5)4)3)2)1)0)
```

Semiformal Description 9.113

Function "G0-G2 Escape Sequence"

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

Definition 9.113

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


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 9.114

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 9.114

```
1   ∀ cst, doby, prof, rdoby
2   (0 GRAPHIC_CHARACTER_SUBREPERTOIRE_VALUE(cst, doby, prof, rdoby) =
3     IF IsAttributeSet^2.61(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^9.4(cst, doby, 'graphic character subrepertoire') ORELSE^9.1
7     (1 IF IsBasicLayoutObjectDescription^2.52(cst)
8       THEN STEP_C_VALUE^9.10(cst, doby, 'graphic character subrepertoire') ORELSE^9.1
9       (2 STEP_DP_VALUE^9.12(cst, doby, 'graphic character subrepertoire') ORELSE^9.1
10        (3 IF IsBasicLayoutObjectDescription^2.52(cst)
11          THEN STEP_E_VALUE^9.14(cst, doby, rdoby, prof, 'graphic character subrepertoire') ORELSE^9.1
12          (4 STEP_FP_VALUE^9.17(cst, doby, rdoby, prof, 'graphic character subrepertoire') ORELSE^9.1
13            (5 STEP_G_VALUE^9.21(cst, doby, rdoby, prof, 'graphic character subrepertoire',
14                                 OBJECT_TYPE_VALUE^9.74(cst, doby))
15              ORELSE^9.1(6 STEP_H_VALUE^9.26(prof, 'graphic character subrepertoire')
16                ORELSE^9.1 0 6) 5) 4) 3) 2) 1) 0)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 9.115

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 9.115

```
1   ∀ cst, doby, prof, rdoby
2   (0 GRAPHIC_RENDITION_VALUE(cst, doby, prof, rdoby) =
3     IF IsAttributeSet^2.61(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^9.4(cst, doby, 'graphic rendition') ORELSE^9.1
7     (1 IF IsBasicLayoutObjectDescription^2.52(cst)
8       THEN STEP_C_VALUE^9.10(cst, doby, 'graphic rendition') ORELSE^9.1
9       (2 STEP_DP_VALUE^9.12(cst, doby, 'graphic rendition') ORELSE^9.1
10        (3 IF IsBasicLayoutObjectDescription^2.52(cst)
11          THEN STEP_E_VALUE^9.14(cst, doby, rdoby, prof, 'graphic rendition') ORELSE^9.1
12          (4 STEP_FP_VALUE^9.17(cst, doby, rdoby, prof, 'graphic rendition') ORELSE^9.1
13            (5 STEP_G_VALUE^9.21(cst, doby, rdoby, prof, 'graphic rendition',
14                                 OBJECT_TYPE_VALUE^9.74(cst, doby))
15              ORELSE^9.1 (6 STEP_H_VALUE^9.26(prof, 'graphic rendition')
16               ORELSE^9.1 0 6) 5) 4) 3) 2) 1) 0)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 9.116

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 9.116

```
1   ∀ cst, doby, prof, rdoby
2   (0 ITEMIZATION_VALUE(cst, doby, prof, rdoby) =
3     IF IsAttributeSet2.61(cst) and 'itemization' ∈ NAMS1.18(cst) and
4        not IsPlaceholder1.19(C ^cst . 'itemization')
5     THEN C ^cst . 'itemization'
6     ELSE STEP_BP_VALUE9.4(cst, doby, 'itemization') ORELSE9.1
7     (1 IF IsBasicLayoutObjectDescription2.52(cst)
8       THEN STEP_C_VALUE9.10(cst, doby, 'itemization') ORELSE9.1
9       (2 STEP_DP_VALUE9.12(cst, doby, 'itemization') ORELSE9.1
10        (3 IF IsBasicLayoutObjectDescription2.52(cst)
11          THEN STEP_E_VALUE9.14(cst, doby, rdoby, prof, 'itemization') ORELSE9.1
12          (4 STEP_FP_VALUE9.17(cst, doby, rdoby, prof, 'itemization') ORELSE9.1
13            (5 STEP_G_VALUE9.21(cst, doby, rdoby, prof, 'itemization', OBJECT_TYPE_VALUE9.74(cst, doby))
14              ORELSE9.1(6 STEP_H_VALUE9.26(prof, 'itemization')
15               ORELSE9.1 ['identifier alignment' : 'no itemization';
16                          'identifier start offset' : 0;
17                          'identifier end offset' : 0]6)5)4)3)2)1)0)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 9.117

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 9.117

```
1   ∀ cst, doby, prof, rdoby
2   (0 KERNING_OFFSET_VALUE(cst, doby, prof, rdoby) =
3     IF IsAttributeSet2.61(cst) and 'kerning offset' ∈ NAMS1.18(cst) and
4       not IsPlaceholder1.19(C ^cst • 'kerning offset')
5     THEN C ^cst • 'kerning offset'
6     ELSE STEP_BP_VALUE9.4(cst, doby, 'kerning offset') ORELSE9.1
7     (1 IF IsBasicLayoutObjectDescription2.52(cst)
8       THEN STEP_C_VALUE9.10(cst, doby, 'kerning offset') ORELSE9.1
9       (2 STEP_DP_VALUE9.12(cst, doby, 'kerning offset') ORELSE9.1
10        (3 IF IsBasicLayoutObjectDescription2.52(cst)
11          THEN STEP_E_VALUE9.14(cst, doby, rdoby, prof, 'kerning offset') ORELSE9.1
12          (4 STEP_FP_VALUE9.17(cst, doby, rdoby, prof, 'kerning offset') ORELSE9.1
13            (5 STEP_G_VALUE9.21(cst, doby, rdoby, prof, 'kerning offset',
14                                OBJECT_TYPE_VALUE9.74(cst, doby))
15              ORELSE9.1(6 STEP_H_VALUE9.26(prof, 'kerning offset')
16               ORELSE9.1 ['start edge offset' : 0; 'end edge offset' : 0]6)5)4)3)2)1)0)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 9.118

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 9.118

```
1   ∀ cst, doby, prof, rdoby
2   (0 LINE_LAYOUT_TABLE_VALUE(cst, doby, prof, rdoby) =
3     IF IsAttributeSet^2.61(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^9.4(cst, doby, 'line layout table') ORELSE^9.1
7     (1 IF IsBasicLayoutObjectDescription^2.52(cst)
8       THEN STEP_C_VALUE^9.10(cst, doby, 'line layout table') ORELSE^9.1
9       (2 STEP_DP_VALUE^9.12(cst, doby, 'line layout table') ORELSE^9.1
10        (3 IF IsBasicLayoutObjectDescription^2.52(cst)
11          THEN STEP_E_VALUE^9.14(cst, doby, rdoby, prof, 'line layout table') ORELSE^9.1
12          (4 STEP_FP_VALUE^9.17(cst, doby, rdoby, prof, 'line layout table') ORELSE^9.1
13            (5 STEP_G_VALUE^9.21(cst, doby, rdoby, prof, 'line layout table',
14                                 OBJECT_TYPE_VALUE^9.74(cst, doby))
15              ORELSE^9.1(6 STEP_H_VALUE^9.26(prof, 'line layout table')
16               ORELSE^9.1 []6)5)4)3)2)1)0)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 9.119

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 9.119

```
1   ∀ cst, doby, prof, rdoby
2   (0 CCA_LINE_PROGRESSION_VALUE(cst, doby, prof, rdoby) =
3     IF IsAttributeSet2.61(cst) and 'line progression' ∈ NAMS1.18(cst) and
4       not IsPlaceholder1.19(C ^cst • 'line progression')
5     THEN C ^cst • 'line progression'
6     ELSE STEP_BP_VALUE9.4(cst, doby, 'line progression') ORELSE9.1
7     (1 IF IsBasicLayoutObjectDescription2.52(cst)
8       THEN STEP_C_VALUE9.10(cst, doby, 'line progression') ORELSE9.1
9       (2 STEP_DP_VALUE9.12(cst, doby, 'line progression') ORELSE9.1
10        (3 IF IsBasicLayoutObjectDescription2.52(cst)
11          THEN STEP_E_VALUE9.14(cst, doby, rdoby, prof, 'line progression') ORELSE9.1
12          (4 STEP_FP_VALUE9.17(cst, doby, rdoby, prof, 'line progression') ORELSE9.1
13            (5 STEP_G_VALUE9.21(cst, doby, rdoby, prof, 'line progression',
14                                OBJECT_TYPE_VALUE9.74(cst, doby))
15              ORELSE9.1(6 STEP_H_VALUE9.26(prof, 'line progression')
16               ORELSE9.1 '270°' 6) 5) 4) 3) 2) 1) 0)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 9.120

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 9.120

1 $\forall\, cst, doby, prof, rdoby$
2 $({}_0\text{LINE_SPACING_VALUE}(cst, doby, prof, rdoby) =$
3 IF $\text{IsAttributeSet}^{2.61}(cst)$ *and* 'line spacing' $\in \text{NAMS}^{1.18}(cst)$ *and*
4 *not* $\text{IsPlaceholder}^{1.19}(\text{C } \hat{}cst \bullet \text{'line spacing'})$
5 THEN C $\hat{}cst \bullet$ 'line spacing'
6 ELSE $\text{STEP_BP_VALUE}^{9.4}(cst, doby, \text{'line spacing'})$ $\text{ORELSE}^{9.1}$
7 $({}_1$IF $\text{IsBasicLayoutObjectDescription}^{2.52}(cst)$
8 THEN $\text{STEP_C_VALUE}^{9.10}(cst, doby, \text{'line spacing'})$ $\text{ORELSE}^{9.1}$
9 $({}_2\text{STEP_DP_VALUE}^{9.12}(cst, doby, \text{'line spacing'})$ $\text{ORELSE}^{9.1}$
10 $({}_3$IF $\text{IsBasicLayoutObjectDescription}^{2.52}(cst)$
11 THEN $\text{STEP_E_VALUE}^{9.14}(cst, doby, rdoby, prof, \text{'line spacing'})$ $\text{ORELSE}^{9.1}$
12 $({}_4\text{STEP_FP_VALUE}^{9.17}(cst, doby, rdoby, prof, \text{'line spacing'})$ $\text{ORELSE}^{9.1}$
13 $({}_5\text{STEP_G_VALUE}^{9.21}(cst, doby, rdoby, prof, \text{'line spacing'}, \text{OBJECT_TYPE_VALUE}^{9.74}(cst, doby))$
14 $\text{ORELSE}^{9.1}({}_6\text{STEP_H_VALUE}^{9.26}(prof, \text{'line spacing'})$
15 $\text{ORELSE}^{9.1}$ $200 \times \text{BMU_EQUIVALENT}^{9.108}(prof)_6)_5)_4)_3)_2)_1)_0)$

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 9.121

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 9.121

```
1   ∀ cst, doby, prof, rdoby
2   (0 PAIRWISE_KERNING_VALUE(cst, doby, prof, rdoby) =
3     IF IsAttributeSet^2.61(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^9.4(cst, doby, 'pairwise kerning') ORELSE^9.1
7     (1 IF IsBasicLayoutObjectDescription^2.52(cst)
8       THEN STEP_C_VALUE^9.10(cst, doby, 'pairwise kerning') ORELSE^9.1
9       (2 STEP_DP_VALUE^9.12(cst, doby, 'pairwise kerning') ORELSE^9.1
10        (3 IF IsBasicLayoutObjectDescription^2.52(cst)
11          THEN STEP_E_VALUE^9.14(cst, doby, rdoby, prof, 'pairwise kerning') ORELSE^9.1
12          (4 STEP_FP_VALUE^9.17(cst, doby, rdoby, prof, 'pairwise kerning') ORELSE^9.1
13            (5 STEP_G_VALUE^9.21(cst, doby, rdoby, prof, 'pairwise kerning',
14                                 OBJECT_TYPE_VALUE^9.74(cst, doby))
15              ORELSE^9.1 (6 STEP_H_VALUE^9.26(prof, 'pairwise kerning')
16               ORELSE^9.1 'no' 6) 5) 4) 3) 2) 1) 0)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Semiformal Description 9.122

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 9.122

```
1   ∀ cst, doby, prof, rdoby
2   (0 FORMATTING_INDICATOR_VALUE(cst, doby, prof, rdoby) =
3     IF IsAttributeSet^2.61(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^9.4(cst, doby, 'formatting indicator') ORELSE^9.1
7     (1 IF IsBasicLayoutObjectDescription^2.52(cst)
8       THEN STEP_C_VALUE^9.10(cst, doby, 'formatting indicator') ORELSE^9.1
9       (2 STEP_DP_VALUE^9.12(cst, doby, 'formatting indicator') ORELSE^9.1
10        (3 IF IsBasicLayoutObjectDescription^2.52(cst)
11          THEN STEP_E_VALUE^9.14(cst, doby, rdoby, prof, 'formatting indicator') ORELSE^9.1
12          (4 STEP_FP_VALUE^9.17(cst, doby, rdoby, prof, 'formatting indicator') ORELSE^9.1
13            (5 STEP_G_VALUE^9.21(cst, doby, rdoby, prof, 'formatting indicator',
14                                 OBJECT_TYPE_VALUE^9.74(cst, doby))
15              ORELSE^9.1(6 STEP_H_VALUE^9.26(prof, 'formatting indicator')
16               ORELSE^9.1 'no' 6) 5) 4) 3) 2) 1) 0)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 9.123

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 9.123

```
1   ∀ cst, doby, prof, rdoby
2   (0 CCA_INITIAL_OFFSET_VALUE(cst, doby, prof, rdoby) =
3     IF IsAttributeSet^2.61(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^9.4(cst, doby, 'initial offset') ORELSE^9.1
7     (1 IF IsBasicLayoutObjectDescription^2.52(cst)
8       THEN STEP_C_VALUE^9.10(cst, doby, 'initial offset') ORELSE^9.1
9       (2 STEP_DP_VALUE^9.12(cst, doby, 'initial offset') ORELSE^9.1
10        (3 IF IsBasicLayoutObjectDescription^2.52(cst)
11          THEN STEP_E_VALUE^9.14(cst, doby, rdoby, prof, 'initial offset') ORELSE^9.1
12          (4 STEP_FP_VALUE^9.17(cst, doby, rdoby, prof, 'initial offset') ORELSE^9.1
13            (5 STEP_G_VALUE^9.21(cst, doby, rdoby, prof, 'initial offset', OBJECT_TYPE_VALUE^9.74(cst, doby))
14              ORELSE^9.1 (6 STEP_H_VALUE^9.26(prof, 'initial offset')
15                ORELSE^9.1 CCA_INITIAL_OFFSET_DEFAULT_VALUE^9.124(cst, doby, prof, rdoby)6)5)4)3)2)1)0)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 9.124

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 9.124

```
1   ∀ cst, doby, prof, rdoby
2   (0 CCA_INITIAL_OFFSET_DEFAULT_VALUE(cst, doby, prof, rdoby) =
3     IF CHARACTER_PATH_VALUE^9.106(cst, doby, prof, rdoby) = '0°' and
4        CCA_LINE_PROGRESSION_VALUE^9.119(cst, doby, prof, rdoby) = '270°'
5        THEN ['horizontal coordinate' : 0;
6              'vertical coordinate' : FIRST_LINE_BOX_BACKWARD_EXTENT^9.127(cst)] ELSE
7     (1 IF CHARACTER_PATH_VALUE^9.106(cst, doby, prof, rdoby) = '0°' and
8        CCA_LINE_PROGRESSION_VALUE^9.119(cst, doby, prof, rdoby) = '90°'
9        THEN ['horizontal coordinate' : 0;
10             'vertical coordinate' : VERTICAL_DIMENSION^9.126(cst, doby, prof, rdoby)−
11                                     FIRST_LINE_BOX_BACKWARD_EXTENT^9.127(cst)] ELSE
12    (2 IF CHARACTER_PATH_VALUE^9.106(cst, doby, prof, rdoby) = '90°' and
13       CCA_LINE_PROGRESSION_VALUE^9.119(cst, doby, prof, rdoby) = '270°'
14       THEN ['horizontal coordinate' : FIRST_LINE_BOX_BACKWARD_EXTENT^9.127(cst);
15             'vertical coordinate' : VERTICAL_DIMENSION^9.126(cst, doby, prof, rdoby)] ELSE
16     (3 IF CHARACTER_PATH_VALUE^9.106(cst, doby, prof, rdoby) = '90°' and
17        CCA_LINE_PROGRESSION_VALUE^9.119(cst, doby, prof, rdoby) = '90°'
18        THEN ['horizontal coordinate' : HORIZONTAL_DIMENSION^9.125(cst, doby, prof, rdoby)−
19                                        FIRST_LINE_BOX_BACKWARD_EXTENT^9.127(cst);
20              'vertical coordinate' :   VERTICAL_DIMENSION^9.126(cst, doby, prof, rdoby)] ELSE
21      (4 IF CHARACTER_PATH_VALUE^9.106(cst, doby, prof, rdoby) = '180°' and
22         CCA_LINE_PROGRESSION_VALUE^9.119(cst, doby, prof, rdoby) = '270°'
23         THEN ['horizontal coordinate' : HORIZONTAL_DIMENSION^9.125(cst, doby, prof, rdoby);
24               'vertical coordinate' : VERTICAL_DIMENSION^9.126(cst, doby, prof, rdoby)−
25                                       FIRST_LINE_BOX_BACKWARD_EXTENT^9.127(cst)] ELSE
26       (5 IF CHARACTER_PATH_VALUE^9.106(cst, doby, prof, rdoby) = '180°' and
27          CCA_LINE_PROGRESSION_VALUE^9.119(cst, doby, prof, rdoby) = '90°'
28          THEN ['horizontal coordinate' : HORIZONTAL_DIMENSION^9.125(cst, doby, prof, rdoby);
29                'vertical coordinate' : FIRST_LINE_BOX_BACKWARD_EXTENT^9.127(cst)] ELSE
30        (6 IF CHARACTER_PATH_VALUE^9.106(cst, doby, prof, rdoby) = '270°' and
31           CCA_LINE_PROGRESSION_VALUE^9.119(cst, doby, prof, rdoby) = '270°'
32           THEN ['horizontal coordinate' : HORIZONTAL_DIMENSION^9.125(cst, doby, prof, rdoby)−
33                                           FIRST_LINE_BOX_BACKWARD_EXTENT^9.127(cst);
34                 'vertical coordinate' : 0] ELSE
35          (7 IF CHARACTER_PATH_VALUE^9.106(cst, doby, prof, rdoby) = '270°' and
36             CCA_LINE_PROGRESSION_VALUE^9.119(cst, doby, prof, rdoby) = '90°'
37             THEN ['horizontal coordinate' : FIRST_LINE_BOX_BACKWARD_EXTENT^9.127(cst);
38                   'vertical coordinate' : 0] 7)6)5)4)3)2)1)0)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 9.125

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 9.125

1 $\forall\, cst, doby, prof, rdoby$
2 $({}_{0}\,\mathrm{HORIZONTAL_DIMENSION}(cst, doby, prof, rdoby) =$
3 C ^DIMENSIONS_VALUE$^{9.50}(cst, doby, prof, rdoby) \bullet$ 'horizontal dimension' ${}_{0})$

Semiformal Description 9.126

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 9.126

1 $\forall\, cst, doby, prof, rdoby$
2 $({}_{0}\,\mathrm{VERTICAL_DIMENSION}(cst, doby, prof, rdoby) =$
3 C ^DIMENSIONS_VALUE$^{9.50}(cst, doby, prof, rdoby) \bullet$ 'vertical dimension' ${}_{0})$

Semiformal Description 9.127

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 9.127

1 $\forall\, cst$
2 $({}_{0}\,\mathrm{IsAtom}(\mathrm{FIRST_LINE_BOX_BACKWARD_EXTENT}(cst))_{0})$

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 9.128

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 9.128

```
1   ∀ cst, doby, prof, rdoby
2   (0 INDENTATION_VALUE(cst, doby, prof, rdoby) =
3     IF IsAttributeSet^2.61(cst) and 'indentation' ∈ NAMS^1.18(cst) and
4        not IsPlaceholder^1.19(C ^cst • 'indentation')
5     THEN C ^cst • 'indentation'
6     ELSE STEP_BP_VALUE^9.4(cst, doby, 'indentation') ORELSE^9.1
7     (1 IF IsBasicLayoutObjectDescription^2.52(cst)
8       THEN STEP_C_VALUE^9.10(cst, doby, 'indentation') ORELSE^9.1
9       (2 STEP_DP_VALUE^9.12(cst, doby, 'indentation') ORELSE^9.1
10        (3 IF IsBasicLayoutObjectDescription^2.52(cst)
11          THEN STEP_E_VALUE^9.14(cst, doby, rdoby, prof, 'indentation') ORELSE^9.1
12          (4 STEP_FP_VALUE^9.17(cst, doby, rdoby, prof, 'indentation') ORELSE^9.1
13            (5 STEP_G_VALUE^9.21(cst, doby, rdoby, prof, 'indentation', OBJECT_TYPE_VALUE^9.74(cst, doby))
14              ORELSE^9.1(6 STEP_H_VALUE^9.26(prof, 'indentation')
15               ORELSE^9.1 0 6) 5) 4) 3) 2) 1) 0)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 9.129

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 9.129

```
1   ∀ cst, doby, prof, rdoby
2   (0 ORPHAN_SIZE_VALUE(cst, doby, prof, rdoby) =
3     IF IsAttributeSet^2.61(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^9.4(cst, doby, 'orphan size') ORELSE^9.1
7     (1 IF IsBasicLayoutObjectDescription^2.52(cst)
8       THEN STEP_C_VALUE^9.10(cst, doby, 'orphan size') ORELSE^9.1
9       (2 STEP_DP_VALUE^9.12(cst, doby, 'orphan size') ORELSE^9.1
10        (3 IF IsBasicLayoutObjectDescription^2.52(cst)
11          THEN STEP_E_VALUE^9.14(cst, doby, rdoby, prof, 'orphan size') ORELSE^9.1
12          (4 STEP_FP_VALUE^9.17(cst, doby, rdoby, prof, 'orphan size') ORELSE^9.1
13            (5 STEP_G_VALUE^9.21(cst, doby, rdoby, prof, 'orphan size', OBJECT_TYPE_VALUE^9.74(cst, doby))
14              ORELSE^9.1 (6 STEP_H_VALUE^9.26(prof, 'orphan size')
15                ORELSE^9.1 1 6) 5) 4) 3) 2) 1) 0)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 9.130

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 9.130

```
1   ∀ cst, doby, prof, rdoby
2   (0 PROPORTIONAL_LINE_SPACING_VALUE(cst, doby, prof, rdoby) =
3     IF IsAttributeSet^2.61(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^9.4(cst, doby, 'proportional line spacing') ORELSE^9.1
7     (1 IF IsBasicLayoutObjectDescription^2.52(cst)
8       THEN STEP_C_VALUE^9.10(cst, doby, 'proportional line spacing') ORELSE^9.1
9       (2 STEP_DP_VALUE^9.12(cst, doby, 'proportional line spacing') ORELSE^9.1
10        (3 IF IsBasicLayoutObjectDescription^2.52(cst)
11          THEN STEP_E_VALUE^9.14(cst, doby, rdoby, prof, 'proportional line spacing') ORELSE^9.1
12          (4 STEP_FP_VALUE^9.17(cst, doby, rdoby, prof, 'proportional line spacing') ORELSE^9.1
13            (5 STEP_G_VALUE^9.21(cst, doby, rdoby, prof, 'proportional line spacing',
14                                 OBJECT_TYPE_VALUE^9.74(cst, doby))
15              ORELSE^9.1 (6 STEP_H_VALUE^9.26(prof, 'proportional line spacing')
16               ORELSE^9.1 'no' 6) 5) 4) 3) 2) 1) 0)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Semiformal Description 9.131

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 9.131

```
1   ∀ cst, doby, prof, rdoby
2   (0 WIDOW_SIZE_VALUE(cst, doby, prof, rdoby) =
3     IF IsAttributeSet^2.61(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^9.4(cst, doby, 'widow size') ORELSE^9.1
7     (1 IF IsBasicLayoutObjectDescription^2.52(cst)
8       THEN STEP_C_VALUE^9.10(cst, doby, 'widow size') ORELSE^9.1
9       (2 STEP_DP_VALUE^9.12(cst, doby, 'widow size') ORELSE^9.1
10        (3 IF IsBasicLayoutObjectDescription^2.52(cst)
11          THEN STEP_E_VALUE^9.14(cst, doby, rdoby, prof, 'widow size') ORELSE^9.1
12          (4 STEP_FP_VALUE^9.17(cst, doby, rdoby, prof, 'widow size') ORELSE^9.1
13            (5 STEP_G_VALUE^9.21(cst, doby, rdoby, prof, 'widow size', OBJECT_TYPE_VALUE^9.74(cst, doby))
14              ORELSE^9.1 (6 STEP_H_VALUE^9.26(prof, 'widow size')
15                ORELSE^9.1 1 6) 5) 4) 3) 2) 1) 0)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

12.2.3 Defaultable attributes of ISO/IEC 8613-7

Semiformal Description 9.132

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 9.132

```
1   ∀ cst, doby, prof, rdoby
2   (0 CLIPPING_VALUE(cst, doby, prof, rdoby) =
3     IF IsAttributeSet^2.61(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^9.4(cst, doby, 'clipping') ORELSE^9.1
7     (1 IF IsBasicLayoutObjectDescription^2.52(cst)
8       THEN STEP_C_VALUE^9.10(cst, doby, 'clipping') ORELSE^9.1
9       (2 STEP_DP_VALUE^9.12(cst, doby, 'clipping') ORELSE^9.1
10        (3 IF IsBasicLayoutObjectDescription^2.52(cst)
11          THEN STEP_E_VALUE^9.14(cst, doby, rdoby, prof, 'clipping') ORELSE^9.1
12          (4 STEP_FP_VALUE^9.17(cst, doby, rdoby, prof, 'clipping') ORELSE^9.1
13            (5 STEP_G_VALUE^9.21(cst, doby, rdoby, prof, 'clipping', OBJECT_TYPE_VALUE^9.74(cst, doby))
14              ORELSE^9.1(6 STEP_H_VALUE^9.26(prof, 'clipping')
15                ORELSE^9.1 [→ [ → 0 → 0 → ] →
16                             [→ (NUMBER_OF_PELS_PER_LINE_VALUE^9.142(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)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Semiformal Description 9.133

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 9.133

```
1   ∀ cst, doby, prof, rdoby
2   (0 RGCA_LINE_PROGRESSION_VALUE(cst, doby, prof, rdoby) =
3     IF IsAttributeSet^2.61(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^9.4(cst, doby, 'line progression') ORELSE^9.1
7     (1 IF IsBasicLayoutObjectDescription^2.52(cst)
8       THEN STEP_C_VALUE^9.10(cst, doby, 'line progression') ORELSE^9.1
9       (2 STEP_DP_VALUE^9.12(cst, doby, 'line progression') ORELSE^9.1
10        (3 IF IsBasicLayoutObjectDescription^2.52(cst)
11          THEN STEP_E_VALUE^9.14(cst, doby, rdoby, prof, 'line progression') ORELSE^9.1
12          (4 STEP_FP_VALUE^9.17(cst, doby, rdoby, prof, 'line progression') ORELSE^9.1
13            (5 STEP_G_VALUE^9.21(cst, doby, rdoby, prof, 'line progression',
14                                 OBJECT_TYPE_VALUE^9.74(cst, doby))
15              ORELSE^9.1(6 STEP_H_VALUE^9.26(prof, 'line progression')
16               ORELSE^9.1 '270°'8)5)4)3)2)1)0)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 9.134

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 9.134

```
1   ∀ cst, doby, prof, rdoby
2   (0 PEL_PATH_VALUE(cst, doby, prof, rdoby) =
3     IF IsAttributeSet^2.61(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^9.4(cst, doby, 'pel path') ORELSE^9.1
7     (1 IF IsBasicLayoutObjectDescription^2.52(cst)
8       THEN STEP_C_VALUE^9.10(cst, doby, 'pel path') ORELSE^9.1
9       (2 STEP_DP_VALUE^9.12(cst, doby, 'pel path') ORELSE^9.1
10        (3 IF IsBasicLayoutObjectDescription^2.52(cst)
11          THEN STEP_E_VALUE^9.14(cst, doby, rdoby, prof, 'pel path') ORELSE^9.1
12          (4 STEP_FP_VALUE^9.17(cst, doby, rdoby, prof, 'pel path') ORELSE^9.1
13            (5 STEP_G_VALUE^9.21(cst, doby, rdoby, prof, 'pel path', OBJECT_TYPE_VALUE^9.74(cst, doby))
14              ORELSE^9.1(6 STEP_H_VALUE^9.26(prof, 'pel path')
15               ORELSE^9.1 '0°' 6)5)4)3)2)1)0)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 9.135

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 9.135

```
1   ∀ cst, doby, prof, rdoby
2   (0 RGCA_INITIAL_OFFSET_VALUE(cst, doby, prof, rdoby) =
3     IF IsAttributeSet^2.61(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^9.4(cst, doby, 'initial offset') ORELSE^9.1
7     (1 IF IsBasicLayoutObjectDescription^2.52(cst)
8       THEN STEP_C_VALUE^9.10(cst, doby, 'initial offset') ORELSE^9.1
9       (2 STEP_DP_VALUE^9.12(cst, doby, 'initial offset') ORELSE^9.1
10        (3 IF IsBasicLayoutObjectDescription^2.52(cst)
11          THEN STEP_E_VALUE^9.14(cst, doby, rdoby, prof, 'initial offset') ORELSE^9.1
12          (4 STEP_FP_VALUE^9.17(cst, doby, rdoby, prof, 'initial offset') ORELSE^9.1
13            (5 STEP_G_VALUE^9.21(cst, doby, rdoby, prof, 'initial offset',
14                                 OBJECT_TYPE_VALUE^9.74(cst, doby))
15              ORELSE^9.1(6 STEP_H_VALUE^9.26(prof, 'initial offset')
16               ORELSE^9.1 RGCA_INITIAL_OFFSET_DEFAULT_VALUE^9.136(cst, doby, prof, rdoby)
17  6)5)4)3)2)1)0)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 9.136

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 9.136

```
1  ∀ cst, doby, prof, rdoby
2  (0 RGCA_INITIAL_OFFSET_DEFAULT_VALUE(cst, doby, prof, rdoby) =
3    IF PEL_PATH_VALUE^9.134(cst, doby, prof, rdoby) = '0°' and
4       RGCA_LINE_PROGRESSION_VALUE^9.133(cst, doby, prof, rdoby) = '270°'
5       THEN ['horizontal coordinate' : 0; 'vertical coordinate' : 0] ELSE
6    (1 IF PEL_PATH_VALUE^9.134(cst, doby, prof, rdoby) = '0°' and
7          RGCA_LINE_PROGRESSION_VALUE^9.133(cst, doby, prof, rdoby) = '90°'
8          THEN ['horizontal coordinate' : 0;
9                'vertical coordinate' : VERTICAL_DIMENSION^9.126(cst, doby, prof, rdoby)] ELSE
10     (2 IF PEL_PATH_VALUE^9.134(cst, doby, prof, rdoby) = '270°' and
11         RGCA_LINE_PROGRESSION_VALUE^9.133(cst, doby, prof, rdoby) = '270°'
12         THEN ['horizontal coordinate' : HORIZONTAL_DIMENSION^9.125(cst, doby, prof, rdoby);
13               'vertical coordinate' : 0] ELSE
14       (3 IF PEL_PATH_VALUE^9.134(cst, doby, prof, rdoby) = '270°' and
15           RGCA_LINE_PROGRESSION_VALUE^9.133(cst, doby, prof, rdoby) = '90°'
16           THEN ['horizontal coordinate' : 0; 'vertical coordinate' : 0] ELSE
17         (4 IF PEL_PATH_VALUE^9.134(cst, doby, prof, rdoby) = '180°' and
18             RGCA_LINE_PROGRESSION_VALUE^9.133(cst, doby, prof, rdoby) = '270°'
19             THEN ['horizontal coordinate' : HORIZONTAL_DIMENSION^9.125(cst, doby, prof, rdoby);
20                   'vertical coordinate' : VERTICAL_DIMENSION^9.126(cst, doby, prof, rdoby)] ELSE
21           (5 IF PEL_PATH_VALUE^9.134(cst, doby, prof, rdoby) = '180°' and
22               RGCA_LINE_PROGRESSION_VALUE^9.133(cst, doby, prof, rdoby) = '90°'
23               THEN ['horizontal coordinate' : HORIZONTAL_DIMENSION^9.125(cst, doby, prof, rdoby);
24                     'vertical coordinate' : 0] ELSE
25             (6 IF PEL_PATH_VALUE^9.134(cst, doby, prof, rdoby) = '90°' and
26                 RGCA_LINE_PROGRESSION_VALUE^9.133(cst, doby, prof, rdoby) = '270°'
27                 THEN ['horizontal coordinate' : 0;
28                       'vertical coordinate' : VERTICAL_DIMENSION^9.126(cst, doby, prof, rdoby)] ELSE
29               (7 IF PEL_PATH_VALUE^9.134(cst, doby, prof, rdoby) = '90°' and
30                   RGCA_LINE_PROGRESSION_VALUE^9.133(cst, doby, prof, rdoby) = '90°'
31                   THEN ['horizontal coordinate' : HORIZONTAL_DIMENSION^9.125(cst, doby, prof, rdoby);
32                         'vertical coordinate' : VERTICAL_DIMENSION^9.126(cst, doby, prof, rdoby)]
33 7)6)5)4)3)2)1)0)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 9.137

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 9.137

```
1   ∀ cst, doby, prof, rdoby
2   (0 PEL_TRANSMISSION_DENSITY_VALUE(cst, doby, prof, rdoby) =
3     IF IsAttributeSet^2.61(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^9.4(cst, doby, 'pel transmission density') ORELSE^9.1
7     (1 IF IsBasicLayoutObjectDescription^2.52(cst)
8       THEN STEP_C_VALUE^9.10(cst, doby, 'pel transmission density') ORELSE^9.1
9       (2 STEP_DP_VALUE^9.12(cst, doby, 'pel transmission density') ORELSE^9.1
10        (3 IF IsBasicLayoutObjectDescription^2.52(cst)
11          THEN STEP_E_VALUE^9.14(cst, doby, rdoby, prof, 'pel transmission density') ORELSE^9.1
12          (4 STEP_FP_VALUE^9.17(cst, doby, rdoby, prof, 'pel transmission density') ORELSE^9.1
13            (5 STEP_G_VALUE^9.21(cst, doby, rdoby, prof, 'pel transmission density',
14                                 OBJECT_TYPE_VALUE^9.74(cst, doby))
15              ORELSE^9.1 (6 STEP_H_VALUE^9.26(prof, 'pel transmission density')
16               ORELSE^9.1 '6 BMU'6)5)4)3)2)1)0)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 9.138

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 9.138

```
1   ∀ cst, doby, prof, rdoby
2   (0 IMAGE_DIMENSIONS_VALUE(cst, doby, prof, rdoby) =
3     IF IsAttributeSet^2.61(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^9.4(cst, doby, 'image dimensions') ORELSE^9.1
7     (1 IF IsBasicLayoutObjectDescription^2.52(cst)
8       THEN STEP_C_VALUE^9.10(cst, doby, 'image dimensions') ORELSE^9.1
9       (2 STEP_DP_VALUE^9.12(cst, doby, 'image dimensions') ORELSE^9.1
10        (3 IF IsBasicLayoutObjectDescription^2.52(cst)
11          THEN STEP_E_VALUE^9.14(cst, doby, rdoby, prof, 'image dimensions') ORELSE^9.1
12          (4 STEP_FP_VALUE^9.17(cst, doby, rdoby, prof, 'image dimensions') ORELSE^9.1
13            (5 STEP_G_VALUE^9.21(cst, doby, rdoby, prof, 'image dimensions',
14                                 OBJECT_TYPE_VALUE^9.74(cst, doby))
15              ORELSE^9.1 (6 STEP_H_VALUE^9.26(prof, 'image dimensions')
16               ORELSE^9.1 'automatic' 6) 5) 4) 3) 2) 1) 0)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 9.139

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 9.139

```
1   ∀ cst, doby, prof, rdoby
2   (0 PEL_SPACING_VALUE(cst, doby, prof, rdoby) =
3     IF IsAttributeSet^2.61(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^9.4(cst, doby, 'pel spacing') ORELSE^9.1
7     (1 IF IsBasicLayoutObjectDescription^2.52(cst)
8       THEN STEP_C_VALUE^9.10(cst, doby, 'pel spacing') ORELSE^9.1
9       (2 STEP_DP_VALUE^9.12(cst, doby, 'pel spacing') ORELSE^9.1
10        (3 IF IsBasicLayoutObjectDescription^2.52(cst)
11          THEN STEP_E_VALUE^9.14(cst, doby, rdoby, prof, 'pel spacing') ORELSE^9.1
12          (4 STEP_FP_VALUE^9.17(cst, doby, rdoby, prof, 'pel spacing') ORELSE^9.1
13            (5 STEP_G_VALUE^9.21(cst, doby, rdoby, prof, 'pel spacing', OBJECT_TYPE_VALUE^9.74(cst, doby))
14              ORELSE^9.1(6 STEP_H_VALUE^9.26(prof, 'pel spacing')
15                ORELSE^9.1 ['length' : 4; 'pel spaces' : 1]6)5)4)3)2)1)0)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Semiformal Description 9.140

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 9.140

```
1   ∀ cst, doby, prof, rdoby
2   (0 SPACING_RATIO_VALUE(cst, doby, prof, rdoby) =
3     IF IsAttributeSet2.61(cst) and 'spacing ratio' ∈ NAMS1.18(cst) and
4        not IsPlaceholder1.19(C ^cst • 'spacing ratio')
5     THEN C ^cst • 'spacing ratio'
6     ELSE STEP_BP_VALUE9.4(cst, doby, 'spacing ratio') ORELSE9.1
7     (1 IF IsBasicLayoutObjectDescription2.52(cst)
8       THEN STEP_C_VALUE9.10(cst, doby, 'spacing ratio') ORELSE9.1
9       (2 STEP_DP_VALUE9.12(cst, doby, 'spacing ratio') ORELSE9.1
10        (3 IF IsBasicLayoutObjectDescription2.52(cst)
11          THEN STEP_E_VALUE9.14(cst, doby, rdoby, prof, 'spacing ratio') ORELSE9.1
12          (4 STEP_FP_VALUE9.17(cst, doby, rdoby, prof, 'spacing ratio') ORELSE9.1
13            (5 STEP_G_VALUE9.21(cst, doby, rdoby, prof, 'spacing ratio', OBJECT_TYPE_VALUE9.74(cst, doby))
14              ORELSE9.1(6 STEP_H_VALUE9.26(prof, 'spacing ratio')
15                ORELSE9.1 ['line spacing value' : 1; 'pel spacing value' : 1]6)5)4)3)2)1)0)
```

Semiformal Description 9.141

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 9.141

```
1   ∀ cst, doby, prof
2   (0 COMPRESSION_VALUE(cst, doby, prof) =
3     IF IsAttributeSet2.61(cst) and 'compression' ∈ NAMS1.18(cst) and
4        not IsPlaceholder1.19(C ^cst • 'compression')
5     THEN C ^cst • 'compression'
6     ELSE STEP_C_VALUE9.10(cst, doby, 'compression') ORELSE9.1
7     (1 STEP_H_VALUE9.26(prof, 'compression') ORELSE9.1 'compressed'1)0)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 9.142

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 9.142

```
1   ∀ cst, doby, prof, rdoby
2   (0 NUMBER_OF_PELS_PER_LINE_VALUE(cst, doby, prof, rdoby) =
3     IF IsAttributeSet^2.61(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^9.10(cst, doby, 'number of pels per line') ORELSE^9.1
7     (1 STEP_H_VALUE^9.26(prof, 'number of pels per line') ORELSE^9.1
8       (2 IF PEL_TRANSMISSION_DENSITY_VALUE^9.137(cst, doby, prof, rdoby) = '1 BMU'
9         THEN 10368 ELSE
10        (3 IF PEL_TRANSMISSION_DENSITY_VALUE^9.137(cst, doby, prof, rdoby) = '2 BMU'
11          THEN 5184 ELSE
12          (4 IF PEL_TRANSMISSION_DENSITY_VALUE^9.137(cst, doby, prof, rdoby) = '3 BMU'
13            THEN 3456 ELSE
14            (5 IF PEL_TRANSMISSION_DENSITY_VALUE^9.137(cst, doby, prof, rdoby) = '4 BMU'
15              THEN 2592 ELSE
16              (6 IF PEL_TRANSMISSION_DENSITY_VALUE^9.137(cst, doby, prof, rdoby) = '5 BMU'
17                THEN 2074 ELSE
18                (7 IF PEL_TRANSMISSION_DENSITY_VALUE^9.137(cst, doby, prof, rdoby) = '6 BMU'
19                  THEN 1728 7) 6) 5) 4) 3) 2) 1) 0)
```

Semiformal Description 9.143

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 9.143

```
1   ∀ cst, doby, prof, rdoby
2   (0 NUMBER_OF_DISCARDED_PELS_VALUE(cst, doby, prof, rdoby) =
3     IF IsAttributeSet^2.61(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^9.10(cst, doby, 'number of discarded pels') ORELSE^9.1
7     (1 IF MAX_PELS_PER_LINE^9.144(cst, doby, prof, rdoby) <
8          NUMBER_OF_PELS_PER_LINE_VALUE^9.142(cst, doby, prof, rdoby)
9       THEN (NUMBER_OF_PELS_PER_LINE_VALUE^9.142(cst, doby, prof, rdoby)−
10            MAX_PELS_PER_LINE^9.144(cst, doby, prof, rdoby))/2
11      ELSE 0 1)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 9.144

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 9.144

1 $\forall\, cst, doby, prof, rdoby$
2 $(_0\text{IsAtom}(\text{MAX_PELS_PER_LINE}(cst, doby, prof, rdoby)_0)$

Semiformal Description 9.145

Function "Value of the attribute 'number of lines per tile'"

The value of the attribute 'number of lines per tile' 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 lines per tile' is 512.

Definition 9.145

1 $\forall\, cst, doby, prof, rdoby$
2 $(_0\text{NUMBER_OF_LINES_PER_TILE_VALUE}(cst, doby, prof, rdoby) =$
3 IF $\text{IsAttributeSet}^{2.61}(cst)$ *and* 'number of lines per tile' $\in \text{NAMS}^{1.18}(cst)$ *and*
4 *not* $\text{IsPlaceholder}^{1.19}$(C ^$cst$ • 'number of lines per tile')
5 THEN C ^cst • 'number of lines per tile'
6 ELSE $\text{STEP_C_VALUE}^{9.10}(cst, doby,$ 'number of lines per tile') $\text{ORELSE}^{9.1}$
7 $(_1\text{STEP_H_VALUE}^{9.26}(prof,$ 'number of lines per tile') $\text{ORELSE}^{9.1}\ 512_1)_0)$

Semiformal Description 9.146

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

The value of the attribute 'number of pels per tile 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 tile line' is 512.

Definition 9.146

1 $\forall\, cst, doby, prof, rdoby$
2 $(_0\text{NUMBER_OF_PELS_PER_TILE_LINE_VALUE}(cst, doby, prof, rdoby) =$
3 IF $\text{IsAttributeSet}^{2.61}(cst)$ *and* 'number of pels per tile line' $\in \text{NAMS}^{1.18}(cst)$ *and*
4 *not* $\text{IsPlaceholder}^{1.19}$(C ^$cst$ • 'number of pels per tile line')
5 THEN C ^cst • 'number of pels per tile line'
6 ELSE $\text{STEP_C_VALUE}^{9.10}(cst, doby,$ 'number of pels per tile line') $\text{ORELSE}^{9.1}$
7 $(_1\text{STEP_H_VALUE}^{9.26}(prof,$ 'number of pels per tile line') $\text{ORELSE}^{9.1}\ 512_1)_0)$

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Semiformal Description 9.147

Function "Value of the attribute 'tiling offset'"

The value of the attribute 'tiling 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 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 'tiling offset' is a sequence of two zeros.

Definition 9.147

1 $\forall\, cst, doby, prof, rdoby$
2 $(_0$TILING_OFFSET_VALUE$(cst, doby, prof, rdoby) =$
3 IF IsAttributeSet$^{2.61}(cst)$ *and* 'tiling offset' $\in$ NAMS$^{1.18}(cst)$ *and*
4 *not* IsPlaceholder$^{1.19}$(C ^cst • 'tiling offset')
5 THEN C ^cst • 'tiling offset'
6 ELSE STEP_C_VALUE$^{9.10}(cst, doby,$ 'tiling offset') ORELSE$^{9.1}$
7 $(_1$STEP_H_VALUE$^{9.26}(prof,$ 'tiling offset') ORELSE$^{9.1}$ [$\rightarrow 0 \rightarrow 0 \rightarrow$] $_1)_0)$

Semiformal Description 9.148

Function "Value of the attribute 'tile types'"

The value of the attribute 'tile types' 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 'tile types' is a sequence of entries where each entry is 'T6 encoded'.

Definition 9.148

1 $\forall\, cst, doby, prof, rdoby$
2 $(_0$TILE_TYPES_VALUE$(cst, doby, prof, rdoby) =$
3 IF IsAttributeSet$^{2.61}(cst)$ *and* 'tile types' $\in$ NAMS$^{1.18}(cst)$ *and*
4 *not* IsPlaceholder$^{1.19}$(C ^cst • 'tile types')
5 THEN C ^cst • 'tile types'
6 ELSE STEP_C_VALUE$^{9.10}(cst, doby,$ 'tile types') ORELSE$^{9.1}$
7 $(_1$STEP_H_VALUE$^{9.26}(prof,$ 'tile types')
8 ORELSE$^{9.1}$ THAT v $(_2$IsNeCat$^{1.3}(v)$ *and* $\forall\, a \hat{\in}$ ^v • $(a =$ 'T6 encoded'$)_2)_1)_0)$

Semiformal Description 9.149

Function "Value of the attribute 'bits per colour component'"

The value of the attribute 'bits per colour component' 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 'bits per colour component' is 1.

Definition 9.149

1 $\forall\, cst, doby, prof, rdoby$
2 $(_0$BITS_PER_COLOUR_COMPONENT_VALUE$(cst, doby, prof, rdoby) =$
3 IF IsAttributeSet$^{2.61}(cst)$ *and* 'bits per colour component' $\in$ NAMS$^{1.18}(cst)$ *and*
4 *not* IsPlaceholder$^{1.19}$(C ^cst • 'bits per colour component')
5 THEN C ^cst • 'bits per colour component'
6 ELSE STEP_C_VALUE$^{9.10}(cst, doby,$ 'bits per colour component') ORELSE$^{9.1}$
7 $(_1$STEP_H_VALUE$^{9.26}(prof,$ 'bits per colour component') ORELSE$^{9.1}$ $1_1)_0)$

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 9.150

Function "Value of the attribute 'interleaving format'"

The value of the attribute 'interleaving format' 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 'interleaving format' is 'plane'.

Definition 9.150

```
1  ∀ cst, doby, prof, rdoby
2  (0 INTERLEAVING_FORMAT_VALUE(cst, doby, prof, rdoby) =
3    IF IsAttributeSet^2.61(cst) and 'interleaving format' ∈ NAMS^1.18(cst) and
4       not IsPlaceholder^1.19(C ^cst • 'interleaving format')
5    THEN C ^cst • 'interleaving format'
6    ELSE STEP_C_VALUE^9.10(cst, doby, 'interleaving format') ORELSE^9.1
7    (1 STEP_H_VALUE^9.26(prof, 'interleaving format') ORELSE^9.1 'plane'1)0)
```

12.2.4 Defaultable attributes of ISO/IEC 8613-8

Semiformal Description 9.151

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 9.151

```
1  ∀ cst, doby, prof, rdoby
2  (0 REGION_OF_INTEREST_SPECIFICATION_VALUE(cst, doby, prof, rdoby) =
3    IF IsAttributeSet^2.61(cst) and 'region of interest specification' ∈ NAMS^1.18(cst) and
4       not IsPlaceholder^1.19(C ^cst • 'region of interest specification')
5    THEN C ^cst • 'region of interest specification'
6    ELSE STEP_BP_VALUE^9.4(cst, doby, 'region of interest specification') ORELSE^9.1
7    (1 IF IsBasicLayoutObjectDescription^2.52(cst)
8      THEN STEP_C_VALUE^9.10(cst, doby, 'region of interest specification') ORELSE^9.1
9      (2 STEP_DP_VALUE^9.12(cst, doby, 'region of interest specification') ORELSE^9.1
10       (3 IF IsBasicLayoutObjectDescription^2.52(cst)
11         THEN STEP_E_VALUE^9.14(cst, doby, rdoby, prof, 'region of interest specification') ORELSE^9.1
12         (4 STEP_FP_VALUE^9.17(cst, doby, rdoby, prof, 'region of interest specification') ORELSE^9.1
13           (5 STEP_G_VALUE^9.21(cst, doby, rdoby, prof, 'region of interest specification',
14                                OBJECT_TYPE_VALUE^9.74(cst, doby))
15             ORELSE^9.1(6 STEP_H_VALUE^9.26(prof, 'region of interest specification')
16              ORELSE^9.1 ['automatic' : 'null']6)5)4)3)2)1)0)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 9.152

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 9.152

```
1   ∀ cst, doby, prof, rdoby
2   (0 PICTURE_ORIENTATION_VALUE(cst, doby, prof, rdoby) =
3     IF IsAttributeSet^2.61(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^9.4(cst, doby, 'picture orientation') ORELSE^9.1
7     (1 IF IsBasicLayoutObjectDescription^2.52(cst)
8       THEN STEP_C_VALUE^9.10(cst, doby, 'picture orientation') ORELSE^9.1
9       (2 STEP_DP_VALUE^9.12(cst, doby, 'picture orientation') ORELSE^9.1
10        (3 IF IsBasicLayoutObjectDescription^2.52(cst)
11          THEN STEP_E_VALUE^9.14(cst, doby, rdoby, prof, 'picture orientation') ORELSE^9.1
12          (4 STEP_FP_VALUE^9.17(cst, doby, rdoby, prof, 'picture orientation') ORELSE^9.1
13            (5 STEP_G_VALUE^9.21(cst, doby, rdoby, prof, 'picture orientation',
14                                 OBJECT_TYPE_VALUE^9.74(cst, doby))
15              ORELSE^9.1(6 STEP_H_VALUE^9.26(prof, 'picture orientation')
16               ORELSE^9.1 '0°'6)5)4)3)2)1)0)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 9.153

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 9.153

```
1   ∀ cst, doby, prof, rdoby
2   (0 PICTURE_DIMENSIONS_VALUE(cst, doby, prof, rdoby) =
3     IF IsAttributeSet^2.61(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^9.4(cst, doby, 'picture dimensions') ORELSE^9.1
7     (1 IF IsBasicLayoutObjectDescription^2.52(cst)
8       THEN STEP_C_VALUE^9.10(cst, doby, 'picture dimensions') ORELSE^9.1
9       (2 STEP_DP_VALUE^9.12(cst, doby, 'picture dimensions') ORELSE^9.1
10        (3 IF IsBasicLayoutObjectDescription^2.52(cst)
11          THEN STEP_E_VALUE^9.14(cst, doby, rdoby, prof, 'picture dimensions') ORELSE^9.1
12          (4 STEP_FP_VALUE^9.17(cst, doby, rdoby, prof, 'picture dimensions') ORELSE^9.1
13            (5 STEP_G_VALUE^9.21(cst, doby, rdoby, prof, 'picture dimensions',
14                                 OBJECT_TYPE_VALUE^9.74(cst, doby))
15              ORELSE^9.1(6 STEP_H_VALUE^9.26(prof, 'picture dimensions')
16               ORELSE^9.1 ['automatic' : 'null']6)5)4)3)2)1)0)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 9.154

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 9.154

```
1   ∀ cst, doby, prof, rdoby
2   (0 GEOMETRIC_GRAPHICS_ENCODING_ANNOUNCER_VALUE(cst, doby, prof, rdoby) =
3     IF IsAttributeSet^2.61(cst) and 'geometric graphics encoding announcer' ∈ NAMS^1.18(cst) and
4        not IsPlaceholder^1.19(C ^cst • 'geometric graphics encoding announcer')
5     THEN C ^cst • 'geometric graphics encoding announcer'
6     ELSE STEP_BP_VALUE^9.4(cst, doby, 'geometric graphics encoding announcer') ORELSE^9.1
7     (1 IF IsBasicLayoutObjectDescription^2.52(cst)
8       THEN STEP_C_VALUE^9.10(cst, doby, 'geometric graphics encoding announcer') ORELSE^9.1
9       (2 STEP_DP_VALUE^9.12(cst, doby, 'geometric graphics encoding announcer') ORELSE^9.1
10        (3 IF IsBasicLayoutObjectDescription^2.52(cst)
11          THEN STEP_E_VALUE^9.14(cst, doby, rdoby, prof, 'geometric graphics encoding announcer') ORELSE^9.1
12          (4 STEP_FP_VALUE^9.17(cst, doby, rdoby, prof, 'geometric graphics encoding announcer') ORELSE^9.1
13            (5 STEP_G_VALUE^9.21(cst, doby, rdoby, prof, 'geometric graphics encoding announcer',
14                                 OBJECT_TYPE_VALUE^9.74(cst, doby))
15              ORELSE^9.1(6 STEP_H_VALUE^9.26(prof, 'geometric graphics encoding announcer') ORELSE^9.1
16              ['Colour Index Precision' : 8;   'Colour Precision' : 8;
17               'Colour Selection Mode' : 'indexed';
18               'Colour Value Extent' : [ → [ → 0 → 0 → 0 → ] → [ → 255 → 255 → 255 → ] → ];
19               'Index Precision' : 16;          'Integer Precision' : 16;
20               'Maximum Colour Index' : 63; 'Real Precision' : 'fixed point format, 16, 16';
21               'VDC Integer Precision' : 16; 'VDC Real Precision' : 'fixed point format, 16, 16';
22               'VDC Type' : 'integer']6)5)4)3)2)1)0)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Semiformal Description 9.155

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 9.155

```
1   ∀ cst, doby, prof, rdoby
2   (0 LINE_RENDITION_VALUE(cst, doby, prof, rdoby) =
3     IF IsAttributeSet^2.61(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^9.4(cst, doby, 'line rendition') ORELSE^9.1
7     (1 IF IsBasicLayoutObjectDescription^2.52(cst)
8       THEN STEP_C_VALUE^9.10(cst, doby, 'line rendition') ORELSE^9.1
9       (2 STEP_DP_VALUE^9.12(cst, doby, 'line rendition') ORELSE^9.1
10        (3 IF IsBasicLayoutObjectDescription^2.52(cst)
11          THEN STEP_E_VALUE^9.14(cst, doby, rdoby, prof, 'line rendition') ORELSE^9.1
12          (4 STEP_FP_VALUE^9.17(cst, doby, rdoby, prof, 'line rendition') ORELSE^9.1
13            (5 STEP_G_VALUE^9.21(cst, doby, rdoby, prof, 'line rendition', OBJECT_TYPE_VALUE^9.74(cst, doby))
14              ORELSE^9.1(6 STEP_H_VALUE^9.26(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^9.156(cst, doby, prof, rdoby);
19               'Line Type' : 1;
20               'Line Width' : LINE_WIDTH_DEFAULT^9.157(cst, doby, prof, rdoby);
21               'Line Width Specification Mode' : 'scaled']6)5)4)3)2)1)0)
```

Semiformal Description 9.156

Function "default colour value"

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

Definition 9.156

```
1   ∀ cst, doby, prof, rdoby
2   (0 COLOUR_DEFAULT(cst, doby, prof, rdoby) =
3     IF C ^GEOMETRIC_GRAPHICS_ENCODING_ANNOUNCER_VALUE^9.154(cst, doby, prof.rdoby) •
4          'Colour Selection Mode' = 'direct'
5     THEN 'foreground' ELSE 1 0)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 9.157

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 9.157

$$
\begin{array}{l}
1 \quad \forall\, cst, doby, prof, rdoby \\
2 \quad ({}_{0}\mathrm{LINE_WIDTH_DEFAULT}(cst, doby, prof, rdoby) = \\
3 \quad\ \ \mathrm{IF\ LINE_WIDTH_SPECIFICATION_MODE}^{9.158}(cst, doby, prof, rdoby) = \text{'scaled'} \\
4 \quad\ \ \mathrm{THEN\ 1.0\ ELSE\ \ 0.001\ *\ LONGEST_VDC_EXTENT}^{9.159}(cst, doby, prof, rdoby)_{0})
\end{array}
$$

Semiformal Description 9.158

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 9.158

$$
\begin{array}{l}
1 \quad \forall\, cst, doby, prof, rdoby \\
2 \quad ({}_{0}\mathrm{LINE_WIDTH_SPECIFICATION_MODE}(cst, doby, prof, rdoby) = \\
3 \quad\ \ \mathrm{C\ \hat{}LINE_RENDITION_VALUE}^{9.155}(cst, doby, prof, rdoby) \bullet \text{'Line Width Specification Mode'}\,{}_{0})
\end{array}
$$

Semiformal Description 9.159

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 9.159

$$
\begin{array}{l}
1 \quad \forall\, cst, doby, prof, rdoby \\
2 \quad ({}_{0}\mathrm{LONGEST_VDC_EXTENT}(cst, doby, prof, rdoby) = \\
3 \quad\ \ \mathrm{IF}\ ({}_{,}\mathrm{HEIGHT}^{9.161}(\mathrm{C\ \hat{}TRANSFORMATION_SPECIFICATION_VALUE}^{9.179}(cst, doby, prof.rdoby) \bullet \\
4 \qquad\qquad \text{'VDC Extent'}) > \\
5 \qquad \mathrm{WIDTH}^{9.160}(\mathrm{C\ \hat{}TRANSFORMATION_SPECIFICATION_VALUE}^{9.179}(cst, doby, prof.rdoby) \bullet \\
6 \qquad\qquad \text{'VDC Extent'})_{1}) \\
7 \quad\ \ \mathrm{THEN\ HEIGHT}^{9.161}(\mathrm{C\ \hat{}TRANSFORMATION_SPECIFICATION_VALUE}^{9.179}(cst, doby, prof.rdoby) \bullet \\
8 \qquad\qquad \text{'VDC Extent'}) \\
9 \quad\ \ \mathrm{ELSE\ WIDTH}^{9.160}(\mathrm{C\ \hat{}TRANSFORMATION_SPECIFICATION_VALUE}^{9.179}(cst, doby, prof.rdoby) \bullet \\
10 \qquad\qquad \text{'VDC Extent'})_{0})
\end{array}
$$

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 9.160

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 9.160

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 9.161

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 9.161

1 $\forall v$
2 $({}_{0}\mathrm{HEIGHT}(v) =$
3 $\mathrm{LASTC}^{1.14}(\mathrm{LASTC}^{1.14}(v)) - \mathrm{LASTC}^{1.14}(\mathrm{HEAD}^{1.13}(v))_{0})$

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Semiformal Description 9.162

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 9.162

```
1   ∀ cst, doby, prof, rdoby
2   (0 MARKER_RENDITION_VALUE(cst, doby, prof, rdoby) =
3     IF IsAttributeSet^2.61(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^9.4(cst, doby, 'marker rendition') ORELSE^9.1
7     (1 IF IsBasicLayoutObjectDescription^2.52(cst)
8       THEN STEP_C_VALUE^9.10(cst, doby, 'marker rendition') ORELSE^9.1
9       (2 STEP_DP_VALUE^9.12(cst, doby, 'marker rendition') ORELSE^9.1
10        (3 IF IsBasicLayoutObjectDescription^2.52(cst)
11          THEN STEP_E_VALUE^9.14(cst, doby, rdoby, prof, 'marker rendition') ORELSE^9.1
12          (4 STEP_FP_VALUE^9.17(cst, doby, rdoby, prof, 'marker rendition') ORELSE^9.1
13            (5 STEP_G_VALUE^9.21(cst, doby, rdoby, prof, 'marker rendition',
14                                 OBJECT_TYPE_VALUE^9.74(cst, doby))
15              ORELSE^9.1(6 STEP_H_VALUE^9.26(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^9.156(cst, doby, prof, rdoby);
20               'Marker Size' : MARKER_SIZE_DEFAULT^9.163(cst, doby, prof, rdoby);
21               'Marker Size Specification Mode' : 'scaled';
22               'Marker Type' : 3]6)5)4)3)2)1)0)
```

Semiformal Description 9.163

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 9.163

```
1   ∀ cst, doby, prof, rdoby
2   (0 MARKER_SIZE_DEFAULT(cst, doby, prof, rdoby) =
3     IF MARKER_SIZE_SPECIFICATION_MODE^9.164(cst, doby, prof, rdoby) = 'scaled'
4     THEN 1.0 ELSE 0.01 * LONGEST_VDC_EXTENT^9.159(cst, doby, prof, rdoby)0)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 9.164

Function "marker size specification mode"

The marker size specification mode is the value of the parameter 'Marker Size Specification Mode' of the attribute 'marker rendition'.

Definition 9.164

1 $\forall cst, doby, prof, rdoby$
2 $({}_{o}\text{MARKER_SIZE_SPECIFICATION_MODE}(cst, doby, prof, rdoby) =$
3 C ˜MARKER_RENDITION_VALUE$^{9.162}(cst, doby, prof, rdoby) \bullet$ 'Marker Size Specification Mode'$_{o})$

Semiformal Description 9.165

Function "Value of the attribute 'text rendition'"

The value of the attribute 'text 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 'text rendition' comprises the parameters 'text aspect source flags' with the sub-parameters 'character expansion factor asf', 'character spacing asf', 'text colour asf', 'text font asf' and 'text precision asf' whose value is 'individual', 'text bundle specifications' whose value is an empty set, 'Alternate Character Set Index', 'Character Set Index', 'Text Bundle Index' and 'Text Font Index' with the value 1, 'Character Coding Announcer' with the value 'basic 7-bit', 'Character Expansion Factor' with the value 1.0, 'Character Height' with the default character height value, 'Character Orientation' with the value ((0,1),(1,0)), 'Character Set List' with the subparameters 'character set type' whose value is '94-character sets', 'designation sequence tail' whose value is the default designation sequence tail, 'Character Spacing' with the value 0.0, 'Font List' with the default font list, 'Text Alignment' with the sub-parameters 'horizontal alignment' whose value is 'normal horizontal' and 'vertical alignment' whose value is 'normal vertical', 'Text Colour' with the default colour value, 'Text Path' with the value 'right', and 'Text Precision' with the value 'string'.

NOTE: The defaulting mechanism may not provide a default value for the sub-parameters 'continuous horizontal alignment' and 'continuous vertical alignment'.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Definition 9.165

```
1   ∀ cst, doby, prof, rdoby
2   (0 TEXT_RENDITION_VALUE(cst, doby, prof, rdoby) =
3     IF IsAttributeSet^2.61(cst) and 'text rendition' ∈ NAMS^1.18(cst) and
4       not IsPlaceholder^1.19(C ^cst • 'text rendition')
5     THEN C ^cst • 'text rendition'
6     ELSE STEP_BP_VALUE^9.4(cst, doby, 'text rendition') ORELSE^9.1
7     (1 IF IsBasicLayoutObjectDescription^2.52(cst)
8       THEN STEP_C_VALUE^9.10(cst, doby, 'text rendition') ORELSE^9.1
9       (2 STEP_DP_VALUE^9.12(cst, doby, 'text rendition') ORELSE^9.1
10        (3 IF IsBasicLayoutObjectDescription^2.52(cst)
11          THEN STEP_E_VALUE^9.14(cst, doby, rdoby, prof, 'text rendition') ORELSE^9.1
12          (4 STEP_FP_VALUE^9.17(cst, doby, rdoby, prof, 'text rendition') ORELSE^9.1
13            (5 STEP_G_VALUE^9.21(cst, doby, rdoby, prof, 'text rendition',
14                                 OBJECT_TYPE_VALUE^9.74(cst, doby))
15              ORELSE^9.1(6 STEP_H_VALUE^9.26(prof, 'text rendition') ORELSE^9.1
16              ['text aspect source flags' : ['character expansion factor asf'; 'individual';
17                                             'character spacing asf' : 'individual'; 'text colour asf' : 'individual';
18                                             'text font asf' : 'individual'; 'text precision asf' : 'individual'];
19               'text bundle specifications' : []; 'Alternate Character Set Index' : 1;
20               'Character Coding Announcer' : 'basic 7-bit'; 'Character Expansion Factor' : 1.0;
21               'Character Height' : CHARACTER_HEIGHT_DEFAULT^9.166(cst, doby, prof, rdoby);
22               'Character Orientation' : [→ [→ 0 → 1 →] → [→ 1 → 0 →] →];
23               'Character Set Index' : 1;
24               'Character Set List' : ['character set type' : '94-character sets';
25                   'designation sequence tail' : DESIGNATION_SEQUENCE_TAIL_DEFAULT^9.167(cst)];
26               'Character Spacing' : 0.0;
27               'Font List' : FONT_LIST_DEFAULT^9.168(cst);
28               'Text Alignment' : ['horizontal alignment' : 'normal horizontal';
29                                   'vertical alignment' : 'normal vertical'];
30               'Text Bundle Index' : 1;
31               'Text Colour' : COLOUR_DEFAULT^9.156(cst, doby, prof, rdoby);
32               'Text Font Index' : 1; 'Text Path' : 'right'; 'Text Precision' : 'string'])6)5)4)3)2)1)0)
```

Semiformal Description 9.166

Function "default character height value"

The default character height is 0.01 times the longest side of the VDC extent.

Definition 9.166

```
1   ∀ cst, doby, prof, rdoby
2   (0 CHARACTER_HEIGHT_DEFAULT(cst, doby, prof, rdoby) =
3     0.01 * LONGEST_VDC_EXTENT^9.159(cst, doby, prof, rdoby)0)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 9.167

Function "default designation sequence tail"

The default designation sequence tail is considered an atom-valued function.

Definition 9.167

1 $\forall cst$

2 $(_{o}\text{IsAtom}(\text{DESIGNATION_SEQUENCE_TAIL_DEFAULT}(cst))_{o})$

Semiformal Description 9.168

Function "default font list"

The default font list is considered an atom-valued function.

Definition 9.168

1 $\forall cst$

2 $(_{o}\text{IsAtom}(\text{FONT_LIST_DEFAULT}(cst))_{o})$

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Semiformal Description 9.169

Function "Value of the attribute 'filled area rendition'"

The value of the attribute 'filled area 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 'filled area rendition' comprises the parameters 'fill aspect source flags' with the sub-parameters 'fill colour asf', 'interior style asf', 'hatch index asf' and 'pattern index asf' whose value is 'individual', 'fill bundle specifications' and 'pattern table specifications' whose value is the empty list, 'Fill Bundle Index', 'Hatch Index' and 'Pattern Index' with the value 1, 'Fill Colour' with the default colour value, 'Fill Reference Point' whose value is the first corner of the VDC extent, 'Interior Style' with the value 'hollow', and 'Pattern Size' with the sub-parameters 'height vector x component' whose value is 0, 'height vector y component' whose value is the height of the VDC extent, 'width vector x component' whose value is the width of the VDC extent and 'width vector y component' with the value 0.

Definition 9.169

```
1   ∀ cst, doby, prof, rdoby
2   (0 FILLED_AREA_RENDITION_VALUE(cst, doby, prof, rdoby) =
3     IF IsAttributeSet2.61(cst) and 'filled area rendition' ∈ NAMS1.18(cst) and
4       not IsPlaceholder1.19(C ^cst • 'filled area rendition')
5     THEN C ^cst • 'filled area rendition'
6     ELSE STEP_BP_VALUE9.4(cst, doby, 'filled area rendition') ORELSE9.1
7     (1 IF IsBasicLayoutObjectDescription2.52(cst)
8       THEN STEP_C_VALUE9.10(cst, doby, 'filled area rendition') ORELSE9.1
9       (2 STEP_DP_VALUE9.12(cst, doby, 'filled area rendition') ORELSE9.1
10        (3 IF IsBasicLayoutObjectDescription2.52(cst)
11          THEN STEP_E_VALUE9.14(cst, doby, rdoby, prof, 'filled area rendition') ORELSE9.1
12          (4 STEP_FP_VALUE9.17(cst, doby, rdoby, prof, 'filled area rendition') ORELSE9.1
13            (5 STEP_G_VALUE9.21(cst, doby, rdoby, prof, 'filled area rendition',
14                                  OBJECT_TYPE_VALUE9.74(cst, doby))
15              ORELSE9.1(6 STEP_H_VALUE9.26(prof, 'filled area rendition') ORELSE9.1
16              ['fill aspect source flags' : ['fill colour asf' : 'individual'; 'interior style asf' : 'individual';
17                                             'hatch index asf' : 'individual'; 'pattern index asf' : 'individual'];
18               'fill bundle specifications' : []; 'pattern table specifications' : [];
19               'Fill Bundle Index' : 1;
20               'Fill Colour' : COLOUR_DEFAULT9.156(cst, doby, prof, rdoby);
21               'Fill Reference Point' : FIRST_CORNER_OF_VDC_EXTENT9.170(cst, doby, prof, rdoby);
22               'Hatch Index' : 1; 'Interior Style' : 'hollow'; 'Pattern Index' : 1;
23               'Pattern Size' : ['height vector x component' : 0;
24                   'height vector y component' : HEIGHT_OF_VDC_EXTENT9.171(cst, doby, prof, rdoby);
25                   'width vector x component' : WIDTH_OF_VDC_EXTENT9.172(cst, doby, prof, rdoby);
26                   'width vector y component' : 0]]6)5)4)3)2)1)0)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 9.170

Function "first corner of the VDC extent"

The first corner of the VDC extent is the first component of the value of the parameter 'VDC Extent' of the attribute 'transformation specification'.

Definition 9.170

$\forall\, cst, doby, prof, rdoby$
$({}_{o}\mathrm{FIRST_CORNER_OF_VDC_EXTENT}(cst, doby, prof, rdoby) =$
$\quad \mathrm{HEAD}^{1.13}(\mathrm{C}\ \hat{}\mathrm{TRANSFORMATION_SPECIFICATION_VALUE}^{9.179}(cst, doby, prof.rdoby) \bullet$
$\quad\quad$ 'VDC Extent')$_{o}$)

Semiformal Description 9.171

Function "height of the VDC extent"

The height of the VDC extent is derived from the value of the parameter 'VDC Extent' of the attribute 'transformation specification'.

Definition 9.171

$\forall\, cst, doby, prof, rdoby$
$({}_{o}\mathrm{HEIGHT_OF_VDC_EXTENT}(cst, doby, prof, rdoby) =$
$\quad \mathrm{HEIGHT}^{9.161}(\mathrm{C}\ \hat{}\mathrm{TRANSFORMATION_SPECIFICATION_VALUE}^{9.179}(cst, doby, prof.rdoby) \bullet$
$\quad\quad$ 'VDC Extent')$_{o}$)

Semiformal Description 9.172

Function "width of the VDC extent"

The width of the VDC extent is derived from the value of the parameter 'VDC Extent' of the attribute 'transformation specification'.

Definition 9.172

$\forall\, cst, doby, prof, rdoby$
$({}_{o}\mathrm{WIDTH_OF_VDC_EXTENT}(cst, doby, prof, rdoby) =$
$\quad \mathrm{WIDTH}^{9.160}(\mathrm{C}\ \hat{}\mathrm{TRANSFORMATION_SPECIFICATION_VALUE}^{9.179}(cst, doby, prof.rdoby) \bullet$
$\quad\quad$ 'VDC Extent')$_{o}$)

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 9.173

Function "Value of the attribute 'edge rendition'"

The value of the attribute 'edge 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 'edge rendition' comprises the parameters 'edge aspect source flags' with the sub-parameters 'edge colour asf', 'edge type asf' and 'edge width asf' whose value is 'individual', 'edge bundle specifications' whose value is the empty collection, 'Edge Bundle Index' and 'Edge Type' with the value 1, 'Edge Colour' whose value is the default colour value, 'Edge Visibility' with the value 'off', 'Edge Width' with the default edge width value, and 'Edge Width Specification Mode' with the value 'scaled'.

Definition 9.173

$\forall\, cst, doby, prof, rdoby$
$(_0$EDGE_RENDITION_VALUE$(cst, doby, prof, rdoby) =$
IF IsAttributeSet$^{2.61}(cst)$ *and* 'edge rendition' $\in$ NAMS$^{1.18}(cst)$ *and*
not IsPlaceholder$^{1.19}$(C $\hat{}\,cst$ • 'edge rendition')
THEN C $\hat{}\,cst$ • 'edge rendition'
ELSE STEP_BP_VALUE$^{9.4}(cst, doby,$ 'edge rendition') ORELSE$^{9.1}$
$(_1$IF IsBasicLayoutObjectDescription$^{2.52}(cst)$
THEN STEP_C_VALUE$^{9.10}(cst, doby,$ 'edge rendition') ORELSE$^{9.1}$
$(_2$STEP_DP_VALUE$^{9.12}(cst, doby,$ 'edge rendition') ORELSE$^{9.1}$
$(_3$IF IsBasicLayoutObjectDescription$^{2.52}(cst)$
THEN STEP_E_VALUE$^{9.14}(cst, doby, rdoby, prof,$ 'edge rendition') ORELSE$^{9.1}$
$(_4$STEP_FP_VALUE$^{9.17}(cst, doby, rdoby, prof,$ 'edge rendition') ORELSE$^{9.1}$
$(_5$STEP_G_VALUE$^{9.21}(cst, doby, rdoby, prof,$ 'edge rendition',
OBJECT_TYPE_VALUE$^{9.74}(cst, doby)$)
ORELSE$^{9.1}(_6$STEP_H_VALUE$^{9.26}(prof,$ 'edge rendition') ORELSE$^{9.1}$
['edge aspect source flags' : ['edge colour asf' : 'individual'; 'edge type asf' : 'individual';
'edge width asf' : 'individual'];
'edge bundle specifications' : []; 'Edge Bundle Index' : 1;
'Edge Colour' : COLOUR_DEFAULT$^{9.156}(cst, doby, prof, rdoby)$;
'Edge Type' : 1; 'Edge Visibility' : 'off';
'Edge Width' : EDGE_WIDTH_DEFAULT$^{9.174}(cst, doby, prof, rdoby)$;
'Edge Width Specification Mode' : 'scaled']$_6)_5)_4)_3)_2)_1)_0$)

Semiformal Description 9.174

Function "default edge width value"

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

Definition 9.174

$\forall\, cst, doby, prof, rdoby$
$(_0$EDGE_WIDTH_DEFAULT$(cst, doby, prof, rdoby) =$
IF EDGE_WIDTH_SPECIFICATION_MODE$^{9.175}(cst, doby, prof, rdoby) =$ 'scaled'
THEN 1.0 ELSE 0.001 * LONGEST_VDC_EXTENT$^{9.159}(cst, doby, prof, rdoby)_0$)

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 9.175

Function "edge width specification mode"

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

Definition 9.175

1 $\forall cst, doby, prof, rdoby$
2 $({}_0$EDGE_WIDTH_SPECIFICATION_MODE$(cst, doby, prof, rdoby) =$
3 C ^EDGE_RENDITION_VALUE$^{9.173}(cst, doby, prof, rdoby)$ • 'Edge Width Specification Mode'$ {}_0)$

Semiformal Description 9.176

Function "Value of the attribute 'colour representations'"

The value of the attribute 'colour representations' 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 'colour representations' comprises the parameters 'colour table specifications' whose value is the empty collection, and 'Background Colour' with the value 'background'.

Definition 9.176

1 $\forall cst, doby, prof, rdoby$
2 $({}_0$COLOUR_REPRESENTATIONS_VALUE$(cst, doby, prof, rdoby) =$
3 IF IsAttributeSet$^{2.61}(cst)$ *and* 'colour representations' $\in$ NAMS$^{1.18}(cst)$ *and*
4 *not* IsPlaceholder$^{1.19}$(C ^cst • 'colour representations')
5 THEN C ^cst • 'colour representations'
6 ELSE STEP_BP_VALUE$^{9.4}(cst, doby,$ 'colour representations') ORELSE$^{9.1}$
7 $({}_1$IF IsBasicLayoutObjectDescription$^{2.52}(cst)$
8 THEN STEP_C_VALUE$^{9.10}(cst, doby,$ 'colour representations') ORELSE$^{9.1}$
9 $({}_2$STEP_DP_VALUE$^{9.12}(cst, doby,$ 'colour representations') ORELSE$^{9.1}$
10 $({}_3$IF IsBasicLayoutObjectDescription$^{2.52}(cst)$
11 THEN STEP_E_VALUE$^{9.14}(cst, doby, rdoby, prof,$ 'colour representations') ORELSE$^{9.1}$
12 $({}_4$STEP_FP_VALUE$^{9.17}(cst, doby, rdoby, prof,$ 'colour representations') ORELSE$^{9.1}$
13 $({}_5$STEP_G_VALUE$^{9.21}(cst, doby, rdoby, prof,$ 'colour representations',
14 OBJECT_TYPE_VALUE$^{9.74}(cst, doby)$)
15 ORELSE$^{9.1}({}_6$STEP_H_VALUE$^{9.26}(prof,$ 'colour representations') ORELSE$^{9.1}$
16 ['colour table specifications' : []; 'Background Colour' : 'background']$ {}_6)_5)_4)_3)_2)_1)_0)$

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 9.177

Function "Value of the attribute 'transparency specification'"

The value of the attribute 'transparency 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 'transparency specification' comprises the parameters 'Auxiliary Colour' whose value is the default auxiliary colour and 'Transparency' with the value 'on'.

Definition 9.177

```
1   ∀ cst, doby, prof, rdoby
2   (0 TRANSPARENCY_SPECIFICATION_VALUE(cst, doby, prof, rdoby) =
3     IF IsAttributeSet^2.61(cst) and 'transparency specification' ∈ NAMS^1.18(cst) and
4        not IsPlaceholder^1.19(C ^cst • 'transparency specification')
5     THEN C ^cst • 'transparency specification'
6     ELSE STEP_BP_VALUE^9.4(cst, doby, 'transparency specification') ORELSE^9.1
7     (1 IF IsBasicLayoutObjectDescription^2.52(cst)
8       THEN STEP_C_VALUE^9.10(cst, doby, 'transparency specification') ORELSE^9.1
9       (2 STEP_DP_VALUE^9.12(cst, doby, 'transparency specification') ORELSE^9.1
10        (3 IF IsBasicLayoutObjectDescription^2.52(cst)
11          THEN STEP_E_VALUE^9.14(cst, doby, rdoby, prof, 'transparency specification') ORELSE^9.1
12          (4 STEP_FP_VALUE^9.17(cst, doby, rdoby, prof, 'transparency specification') ORELSE^9.1
13            (5 STEP_G_VALUE^9.21(cst, doby, rdoby, prof, 'transparency specification',
14                                 OBJECT_TYPE_VALUE^9.74(cst, doby))
15              ORELSE^9.1(6 STEP_H_VALUE^9.26(prof, 'transparency specification') ORELSE^9.1
16              ['Auxiliary Colour' : AUXILIARY_COLOUR_DEFAULT^9.178(cst, doby, prof, rdoby);
17               'Transparency' : 'on']6)5)4)3)2)1)0)
```

Semiformal Description 9.178

Function "default auxiliary colour value"

If the colour selection mode is 'direct', the default auxiliary colour value is 'background', otherwise 0.

Definition 9.178

```
1   ∀ cst, doby, prof, rdoby
2   (0 AUXILIARY_COLOUR_DEFAULT(cst, doby, prof, rdoby) =
3     IF C ^GEOMETRIC_GRAPHICS_ENCODING_ANNOUNCER_VALUE^9.154(cst, doby, prof.rdoby) •
4          'Colour Selection Mode' = 'direct'
5     THEN 'background' ELSE 0)0
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Semiformal Description 9.179

Function "Value of the attribute 'transformation specification'"

The value of the attribute 'transformation 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 'transformation specification' comprises the parameters 'Clip Indicator' with the value 'on' and 'Clip Rectangle' and 'VDC Extent' both with a value of ((0,0),(1,1)).

Definition 9.179

```
1   ∀ cst, doby, prof, rdoby
2   (0 TRANSFORMATION_SPECIFICATION_VALUE(cst, doby, prof, rdoby) =
3     IF IsAttributeSet^2.61(cst) and 'transformation specification' ∈ NAMS^1.18(cst) and
4        not IsPlaceholder^1.19(C ^cst • 'transformation specification')
5     THEN C ^cst • 'transformation specification'
6     ELSE STEP_BP_VALUE^9.4(cst, doby, 'transformation specification') ORELSE^9.1
7     (1 IF IsBasicLayoutObjectDescription^2.52(cst)
8       THEN STEP_C_VALUE^9.10(cst, doby, 'transformation specification') ORELSE^9.1
9       (2 STEP_DP_VALUE^9.12(cst, doby, 'transformation specification') ORELSE^9.1
10        (3 IF IsBasicLayoutObjectDescription^2.52(cst)
11          THEN STEP_E_VALUE^9.14(cst, doby, rdoby, prof, 'transformation specification') ORELSE^9.1
12          (4 STEP_FP_VALUE^9.17(cst, doby, rdoby, prof, 'transformation specification') ORELSE^9.1
13            (5 STEP_G_VALUE^9.21(cst, doby, rdoby, prof, 'transformation specification',
14                                 OBJECT_TYPE_VALUE^9.74(cst, doby))
15              ORELSE^9.1(6 STEP_H_VALUE^9.26(prof, 'transformation specification') ORELSE^9.1
16              ['Clip Indicator' : 'on';
17               'Clip Rectangle' : [ → [ → 0 → 0 → ] → [ → 1 → 1 → ] → ];
18               'VDC Extent' : [ → [ → 0 → 0 → ] → [ → 1 → 1 → ] → ]]6)5)4)3)2)1)0)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

13 Index of predicate symbols, operator symbols and attribute names

The numbers refer to the definitions where the symbols are used or defined.

0° 2.95, 6.17, 6.19, 7.13, 8.49, 9.105, 9.106, 9.124, 9.134, 9.136, 9.152
1 BMU 7.15, 9.142
180° 2.95, 6.17, 6.19, 7.13, 8.49, 9.124, 9.136
2 8 2 6 0 6.4, 6.26, 9.49
2 8 2 6 1 6.4, 6.26
2 8 2 6 2 6.4, 6.26
2 8 2 7 0 7.4, 7.19
2 8 2 7 2 7.4, 7.19
2 8 2 8 0 8.4, 8.51
2 8 3 6 0 6.20, 9.95
2 8 3 7 0 7.20, 9.95
2 8 3 7 10 7.20
2 8 3 7 11 7.20
2 8 3 7 1 7.20
2 8 3 7 2 7.20
2 8 3 7 3 7.20
2 8 3 7 5 7.7, 7.20
2 8 3 7 9 7.20
2 8 3 8 0 8.52, 9.95
2 BMU 7.15, 9.142
270° 2.95, 6.17, 6.19, 6.45, 7.12, 7.13, 8.49, 9.64, 9.119, 9.124, 9.133, 9.136
3 BMU 7.15, 9.142
4 BMU 7.15, 9.142
5 BMU 7.15, 9.142
6 BMU 7.15, 9.137, 9.142
90° 2.95, 6.17, 6.19, 6.45, 7.12, 7.13, 8.49, 9.124, 9.136
94-character multibyte sets 8.24
94-character sets 8.24, 9.165
96-character multibyte sets 8.24
96-character sets 8.24

A 4.2
absolute 8.16
abstract 2.121, 4.1, 4.2
access rights 2.121, 4.1, 4.2
additional information 2.121, 4.1, 4.2, 4.52
aggregate 2.85
aligned-around 6.42
alignment string 6.42
alignment 2.91, 6.12, 6.42, 9.103
ALIGNMENT_VALUE 9.103
all 2.119
ALLCONT 2.141, 2.155
Alternate Character Set Index 8.22, 9.165
alternative feature sets 2.121, 4.1, 4.2
alternative representation character sets 2.121, 4.1, 4.2
alternative representation 2.54, 2.61
alternative 2.39, 2.44, 2.47, 2.61
application comments 2.20, 2.21, 2.22, 2.24, 2.25, 2.26, 2.27, 2.28, 2.29, 2.37, 2.38, 2.39, 2.41, 2.42, 2.43, 2.44, 2.46, 2.47, 2.54, 2.55, 2.61, 9.42
APPLICATION_COMMENTS_VALUE 9.42
applies 2.92
area controlled 7.16, 8.50

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

aspect ratio flag 7.16, 8.50
authorization 2.121, 4.1, 4.2
authors organization 4.36
authors 2.121, 4.1, 4.2
automatic 7.16, 8.48, 8.50, 9.138, 9.151, 9.153
Auxiliary Colour 8.46, 9.177
AUXILIARY_COLOUR_DEFAULT 9.177, 9.178
AVAILABLE_HORIZONTAL_DIMENSION_VALUE 9.53, 9.54
AVAILABLE_VERTICAL_DIMENSION_VALUE 9.55, 9.56

B value 4.89
B 4.2
Background Colour 8.43, 9.176
background 9.176, 9.178
backward limit 2.134, 9.58, 9.59
BACKWARD_LIMIT_VALUE 9.58, 9.59
balance 2.8, 2.15, 2.24, 2.25, 2.26, 2.28, 2.41, 2.42, 2.43, 2.46, 2.61, 9.43
BALANCE_VALUE 9.43
base 8.30
basic 7-bit 8.22, 9.165
basic 8-bit 8.22
basic logical object 2.22, 2.39, 2.76, 2.90
bindings 2.20, 2.21, 2.22, 2.24, 2.25, 2.26, 2.27, 2.28, 2.29, 2.35, 2.37, 2.38, 2.39, 2.40, 2.41, 2.42, 2.43, 2.44, 2.46, 2.47, 2.48, 2.61, 9.44
BINDINGS_VALUE 9.44
bitmap encoded 7.30
bits per colour component 7.1, 7.7, 7.9, 7.10, 9.149
BITS_PER_COLOUR_COMPONENT_VALUE 9.149
block alignment 2.55, 2.61, 4.10, 9.45
block alignments 2.121, 4.1, 4.2
block 2.29, 2.47, 2.76, 2.90
BLOCK_ALIGNMENT_VALUE 9.45
BMU_EQUIVALENT 9.107, 9.108, 9.120
border freespace width 2.93, 9.46
border line colour 2.93, 9.46
border line type 2.93, 9.46
border line width 2.93, 9.46
border 2.28, 2.29, 2.46, 2.47, 2.56, 2.61, 2.90, 4.10, 9.46
borders 2.121, 4.1, 4.2
BORDER_VALUE 9.46
both 4.76
bottom 8.30
BPH 6.105
BS 6.101
bundled 8.11, 8.18, 8.22, 8.31, 8.39

C value 4.95
calibration data 4.81
cap 8.30
CCA_INITIAL_OFFSET_DEFAULT_VALUE 9.123, 9.124
CCA_INITIAL_OFFSET_VALUE 9.123
CCA_LINE_PROGRESSION_VALUE 9.119, 9.124
centre separation 2.91, 2.109, 9.89, 9.92
centre 8.30
centred 2.91, 2.101, 6.15, 6.42
CENTRE_SEPARATION_VALUE 9.89, 9.92
Character Coding Announcer 8.22, 9.165

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

character coding attributes 4.24
character expansion factor asf 8.22, 9.165
Character Expansion Factor 8.22, 8.27, 9.165
character fonts 6.12, 9.104
Character Height 8.22, 9.165
character orientation 6.12, 9.105
Character Orientation 8.22, 9.165
character path 6.12, 9.106
character presentation features 4.25
Character Set Index 8.22, 9.165
Character Set List 8.22, 9.165
character set type 8.24, 9.165
character spacing asf 8.22, 9.165
character spacing 6.12, 9.107
Character Spacing 8.22, 8.27, 9.165
character 8.22, 8.27
CHARACTER_FONTS_VALUE 9.104
CHARACTER_HEIGHT_DEFAULT 9.165, 9.166
CHARACTER_ORIENTATION_VALUE 9.105
CHARACTER_PATH_VALUE 9.106, 9.124
CHARACTER_SPACING_VALUE 9.107
choice 2.85
CIELAB 2.130, 4.79, 4.82
CIELUV 2.130, 4.79, 4.82
Clip Indicator 8.47, 9.179
Clip Rectangle 8.47, 9.179
clipping 7.1, 7.8, 9.132
CLIPPING_VALUE 9.132
CMY 4.79, 4.82
CMYK 4.79, 4.82
code extension announcers 6.12, 9.109
CODE_EXTENSION_ANNOUNCERS_VALUE 9.109
coding attributes 2.54, 2.61, 2.121, 4.1, 4.2
COLC 1.17, 1.22, 2.147, 2.154
colour access mode 2.127, 2.128, 9.98
colour calibration type 4.77
colour characteristics 2.121, 4.1, 4.2
colour data scaling 4.81
Colour Index Precision 8.9, 9.154
colour list 8.45
colour look-up table 4.84
colour modes present 4.75
colour of layout object 2.26, 2.27, 2.28, 2.29, 2.43, 2.44, 2.46, 2.47, 2.56, 2.61, 2.90, 4.10, 9.98
colour of layout objects 2.121, 4.1, 4.2
colour of media 2.97, 9.47
colour of medium 2.98, 9.69, 9.72
colour offset 4.94
Colour Precision 8.9, 9.154
colour representations 8.6, 9.176
colour scale 4.94
Colour Selection Mode 8.9, 9.154, 9.156, 9.178
colour space id 2.128, 2.131, 4.81, 9.98, 9.99, 9.102
colour space name 4.81
colour space type 4.77, 4.81
colour spaces list 2.121, 4.1, 4.2
colour spaces present 4.75

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

colour specification 2.128, 2.133, 9.98, 9.99, 9.102
colour table entries 2.131, 9.99, 9.102
colour table specifications 8.43, 9.176
colour table 4.87
colour tolerance 2.128, 2.133, 9.98, 9.99, 9.102
Colour Value Extent 8.9, 9.154
colour 2.26, 2.27, 2.28, 2.29, 2.43, 2.44, 2.46, 2.47, 2.56, 2.61, 2.90, 4.10, 8.35, 9.47
COLOUR_DEFAULT 9.155, 9.156, 9.162, 9.165, 9.169, 9.173
coloured 2.97
COLOUR_OF_LAYOUT_OBJECT_VALUE 9.98
COLOUR_OF_MEDIUM_VALUE 9.69, 9.72
COLOUR_REPRESENTATIONS_VALUE 9.176
colours 2.121, 4.1, 4.2
COLOUR_VALUE 9.47
comments character sets 2.121, 4.1, 4.2
comments 4.85
complete code 8.24
complete generator set 2.5, 2.6, 2.62, 4.1, 4.2
composite logical object 2.21, 2.38, 2.76, 2.90
composite or basic page 2.26, 2.27, 2.43, 2.44, 2.76, 2.90, 9.51, 9.52, 9.53, 9.55
compressed 7.21, 9.141
compression 7.1, 7.7, 7.9, 7.10, 9.141
COMPRESSION_VALUE 9.141
concatenated 2.102
concatenation 2.55, 2.61, 9.48
CONCATENATION_VALUE 9.48
ContainsFormattedElements 6.8, 6.47, 6.111
ContainsFormattedProcessableElements 6.10, 6.49, 6.111
ContainsProcessableElements 6.6, 6.48, 6.111
content architecture class 2.22, 2.27, 2.29, 2.39, 2.44, 2.47, 2.61, 2.90, 4.10, 6.4, 7.4, 8.4, 9.49
content architecture classes 2.121, 4.1, 4.2, 6.4, 7.4, 8.4
content background colour 2.27, 2.29, 2.44, 2.47, 2.56, 2.61, 2.90, 4.10, 9.100
content background colours 2.121, 4.1, 4.2
content background transparency 2.61, 9.100
content colour table 2.27, 2.29, 2.44, 2.47, 2.56, 2.61, 2.90, 4.10, 9.102
content colour tables 2.121, 4.1, 4.2
content foreground colour 2.27, 2.29, 2.44, 2.47, 2.56, 2.61, 2.90, 4.10, 9.101
content foreground colours 2.121, 4.1, 4.2
content foreground transparency 2.61
content generator 2.22, 2.27, 2.29, 2.39, 2.61, 2.135, 2.136
content identifier layout 2.12, 2.15, 2.54, 2.61, 2.138, 2.153, 2.160
content identifier logical 2.11, 2.14, 2.54, 2.61, 2.138, 2.153, 2.159
content information 2.54, 2.61, 6.6, 6.8, 6.10, 6.14, 7.7, 7.10, 8.8
content portions 2.22, 2.27, 2.29, 2.39, 2.44, 2.47, 2.61, 2.135, 2.136, 2.138, 2.144
CONTENT_ARCHITECTURE_CLASS_VALUE 9.49
CONTENT_BACKGROUND_COLOUR_VALUE 9.100
CONTENT_COLOUR_TABLE_VALUE 9.102
CONTENT_FOREGROUND_COLOUR_VALUE 9.101
continuous horizontal alignment 8.30
continuous horizontal 8.30
continuous vertical alignment 8.30
continuous vertical 8.30
copyright dates 4.37
copyright information 4.37
copyright 2.121, 4.1, 4.2
CoveredBy 2.145, 2.146

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

CR 6.68, 6.69, 6.74, 6.75, 6.81, 6.97
creation date and time 2.121, 4.1, 4.2
CURRENT-INSTANCE 2.69, 2.70, 2.74
CURRENT-OBJECT 2.69, 2.70, 2.74

dash-dot-dot 2.93
dash-dot 2.93
dashed 2.93
DECREMENT 2.67, 2.68
default value lists 2.20, 2.21, 2.24, 2.25, 2.26, 2.28, 2.37, 2.38, 2.41, 2.42, 2.43, 2.46, 2.61, 9.23, 9.24, 9.36, 9.37
DEFAULT_PAR_VALUE 9.35, 9.36, 9.38
DEFAULT_PAR_VALUE_BY_STYLE 9.36, 9.37
DEFAULT_VALUE 9.22, 9.23, 9.25
DEFAULT_VALUE_BY_STYLE 9.23, 9.24
derived from 2.55, 2.56, 2.61, 9.3, 9.5, 9.8, 9.9, 9.28
DescribesClassImSubOf 2.20, 2.24, 2.148, 2.150, 2.151
DescribesClassSubOf 2.10, 2.151, 2.151
DescribesContPortOf 2.137, 2.153, 6.4, 8.4, 9.132
DescribesImSubOf 2.149, 2.152, 9.21, 9.33, 9.41
designation sequence tail 8.24, 9.165
DESIGNATION_SEQUENCE_TAIL_DEFAULT 9.165, 9.167
dimensions 2.26, 2.27, 2.28, 2.29, 2.43, 2.44, 2.46, 2.47, 2.61, 2.90, 4.10, 9.51, 9.52
DIMENSIONS_VALUE 9.50, 9.125, 9.126
direct 2.128, 4.76, 8.9, 9.98, 9.156, 9.178
distribution list 2.121, 4.1, 4.2
document application profile defaults 2.121, 4.1, 4.2, 9.26, 9.34
document application profile 2.121, 4.1, 4.2
document architecture class 2.4, 2.5, 2.6, 2.7, 2.62, 2.121, 4.1, 4.2
document body part seal 4.62
document date and time 2.121, 4.1, 4.2
document layout root 2.24, 2.41, 2.76
document logical root 2.20, 2.37, 2.76
document profile seal 4.69
document reference 2.121, 4.1, 4.2
document size 2.121, 4.1, 4.2
document type 2.121, 4.1, 4.2
dot 2.93
down 8.22

edge aspect source flags 8.39, 9.173
Edge Bundle Index 8.39, 8.41, 9.173
edge bundle representation 8.41
edge bundle specifications 8.39, 9.173
edge colour asf 8.39, 9.173
Edge Colour 8.39, 8.41, 9.173
edge rendition 8.6, 9.173
edge type asf 8.39, 9.173
Edge Type 8.39, 8.41, 9.173
Edge Visibility 8.39, 9.173
edge width asf 8.39, 9.173
Edge Width Specification Mode 8.39, 9.173, 9.175
Edge Width 8.39, 8.41, 9.173
EDGE_RENDITION_VALUE 9.173, 9.175
EDGE_WIDTH_DEFAULT 9.173, 9.174
EDGE_WIDTH_SPECIFICATION_MODE 9.174, 9.175
eighth alternative font 6.16
empty 8.31, 8.33

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

enciphered document profiles 2.121, 4.1, 4.2
enciphered information 2.58, 2.59, 2.60, 2.61
enciphered profiles 2.121, 4.1, 4.2
enciphered subordinates 2.119
enciphered 2.22, 2.27, 2.29, 2.37, 2.38, 2.39, 2.41, 2.42, 2.43, 2.44, 2.46, 2.47, 2.61
end edge offset 6.36, 9.117
end-aligned 6.15, 6.42, 6.46
ESC 6.110
expiry date and time 2.121, 4.1, 4.2
extended 7-bit 8.22
extended 8-bit 8.22
external-document class 2.121, 4.1, 4.2

factor set 2.7, 2.62, 4.1, 4.2
fifth alternative font 6.16
file name 4.44
fill aspect source flags 8.31, 9.169
Fill Bundle Index 8.31, 8.33, 9.169
fill bundle representation 8.33
fill bundle specifications 8.31, 9.169
fill colour asf 8.31, 9.169
Fill Colour 8.31, 8.33, 9.169
fill order 2.55, 2.61, 2.91, 9.57
fill orders 2.121, 4.1, 4.2
Fill Reference Point 8.31, 9.169
filled area rendition 8.6, 9.169
FILLED_AREA_RENDITION_VALUE 9.169
FILL_ORDER_VALUE 9.57
fingerprint information 4.66
fingerprint method 4.66
fingerprint 4.67
first alternative font 6.16
first component 4.93
first line offset 6.12, 9.111
FIRST_CORNER_OF_VDC_EXTENT 9.169, 9.170
FIRST_LINE_BOX_BACKWARD_EXTENT 9.124, 9.127
FIRST_LINE_OFFSET_VALUE 9.111
fixed dimension 2.92, 9.51, 9.52
fixed point format, 16, 16 8.9, 9.154
fixed point format, 32, 32 8.9
fixed position 2.91, 9.83, 9.85, 9.86
fixed 7.16, 8.50
FIXED_POSITION_VALUE 9.83, 9.84
floatability range 2.55, 2.61
FLOATABILITYRANGVALUE 9.58
floating point format, 12, 52 8.9
floating point format, 9, 23 8.9
font identifier 6.16
Font List 8.22, 9.165
font size 6.16
FONT_LIST_DEFAULT 9.165, 9.168
fonts list 2.121, 4.1, 4.2
foreground 9.156
formatted processable 2.4, 2.6, 2.62, 4.1, 4.2
formatted 2.4, 2.7, 2.62, 4.1, 4.2
formatting indicator 6.4, 6.12, 9.122

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

FORMATTING_INDICATOR_VALUE 9.122
forward limit 2.134, 9.58, 9.60
FORWARD_LIMIT_VALUE 9.58, 9.60
fourth alternative font 6.16
fourth component 4.93
frame 2.28, 2.46, 2.76, 2.90

G value 4.89
G0_G2_ESCAPE_SEQUENCE 9.112, 9.113
G0_G2_ESCAPE_SEQUENCE_ANNOUNCER 9.109, 9.110
GCC 6.64, 6.65, 6.66, 6.98, 6.99
generator for subordinates 2.10, 2.20, 2.21, 2.24, 2.25, 2.26, 2.28, 2.61, 2.136, 2.139, 2.142, 2.145, 2.165
generic layout structure 2.5, 2.6, 2.7, 2.62, 2.121, 4.1, 4.2
generic logical structure 2.5, 2.6, 2.7, 2.62, 2.121, 4.1, 4.2
geometric graphics encoding announcer 8.6, 9.154
geometric-graphics coding attributes 4.24
geometric-graphics presentation features 4.25
GEOMETRIC_GRAPHICS_ENCODING_ANNOUNCER_VALUE 9.154, 9.156, 9.178
givenname 4.47
graphic character sets 6.12, 9.112
graphic character subrepertoire 6.12, 9.114
graphic rendition 6.12, 9.115
GRAPHIC_CHARACTER_SETS_VALUE 9.112
GRAPHIC_CHARACTER_SUBREPERTOIRE_VALUE 9.114
GRAPHIC_RENDITION_VALUE 9.115
grid location 4.91
grid specification 4.85
grid value 4.91

half 8.30
hatch index asf 8.31, 9.169
Hatch Index 8.31, 8.33, 9.169
hatch 8.31, 8.33
HEAD 1.13, 2.57, 2.58, 2.59, 2.60, 2.77, 2.78, 2.80, 2.81, 2.89, 2.100, 2.110, 2.122, 2.123, 2.156, 2.161, 2.162, 4.56, 4.59, 4.61, 4.70, 6.50, 6.51, 6.52, 6.66, 7.7, 9.16, 9.108, 9.160, 9.161, 9.170
height controlled 7.16, 8.50
height vector x component 8.38, 9.169
height vector y component 8.38, 9.169
HEIGHT 9.159, 9.161, 9.171
HEIGHT_OF_VDC_EXTENT 9.169, 9.171
hollow 8.31, 8.33, 9.169
horizontal alignment 8.30, 9.165
horizontal coordinate 6.35, 7.14, 9.124, 9.136
horizontal dimension 2.92, 9.50, 9.51, 9.125
horizontal position 2.91, 9.84, 9.85
HORIZONTAL_DIMENSION 9.124, 9.125, 9.136
HORIZONTAL_DIMENSION_VALUE 9.50, 9.51
HORIZONTAL_FIXED_DIMENSION_VALUE 9.51, 9.53
HORIZONTAL_POSITION_VALUE 9.84, 9.85
HPB 6.102, 6.103
HPR 6.102, 6.103

identifier alignment 6.46, 9.116
identifier end offset 6.46, 9.116
identifier start offset 6.46, 9.116
IGS 6.98, 6.99
image dimensions 7.1, 7.8, 9.138

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

IMAGE_DIMENSIONS_VALUE 9.138
imaging order 2.8, 2.43, 2.46, 2.61
implementation-defined 2.61, 2.93, 9.46, 9.101
IMSUB 2.15, 2.154, 2.156
INCREMENT 2.67, 2.68
indentation 6.4, 6.12, 9.128
INDENTATION_VALUE 9.128
index array 8.35
Index Precision 8.9, 9.154
index 2.127, 2.133, 4.89, 9.99, 9.102
indexed 2.127, 4.76, 8.9, 9.154
individual 8.11, 8.18, 8.22, 8.31, 8.39, 9.155, 9.162, 9.165, 9.169, 9.173
indivisibility 2.55, 2.61, 9.61
INDIVISIBILITY_VALUE 9.61
infinite 2.130, 9.98, 9.99, 9.102
initial offset 6.4, 6.12, 7.1, 7.8, 9.123, 9.135
initials 4.47
Integer Precision 8.9, 9.154
integer 8.9, 9.154
interchange format class 2.121, 4.1, 4.2
interior style asf 8.31, 9.169
Interior Style 8.31, 8.33, 9.169
interleaving format 7.1, 7.7, 7.9, 7.10, 9.150
INTERLEAVING_FORMAT_VALUE 9.150
invisible 2.93
Is3x3Matrix 4.84, 4.92
IsAdditionalInformationValue 4.2, 4.42
IsAlignmentValue 6.12, 6.15
IsAlternativeCharacterSetsString 2.61, 2.115
IsAlternativeFeaturesSetsValue 4.2, 4.48
IsASN1ObjectIdentifier 4.6, 4.7, 4.9, 4.49, 4.53, 4.55, 4.68
IsASN1ObjectIdentifierSet 4.48, 4.49
IsAtom 1.10, 1.21, 2.115, 2.118, 4.5, 4.7, 4.30, 4.31, 4.40, 4.42, 6.25, 6.42, 6.93, 6.95, 6.113, 8.26, 8.29, 8.54, 8.56, 8.57, 8.58, 8.59, 8.60, 8.61, 9.54, 9.56, 9.81, 9.110, 9.113, 9.127, 9.144, 9.167, 9.168
IsAtomicStringExpression 2.64, 2.65
IsAttributeSet 2.20, 2.21, 2.22, 2.24, 2.25, 2.26, 2.27, 2.28, 2.29, 2.37, 2.38, 2.39, 2.41, 2.42, 2.43, 2.44, 2.46, 2.47, 2.54, 2.55, 2.56, 2.57, 2.58, 2.59, 2.60, 2.61, 2.90, 2.142, 2.143, 2.144, 2.145, 2.154, 2.156, 9.42, 9.43, 9.44, 9.46, 9.47, 9.49, 9.51, 9.52, 9.64, 9.65, 9.66, 9.70, 9.71, 9.72, 9.74, 9.80, 9.82, 9.85, 9.86, 9.87, 9.94, 9.95, 9.96, 9.97, 9.98, 9.99, 9.100, 9.101, 9.102, 9.103, 9.104, 9.105, 9.106, 9.107, 9.109, 9.111, 9.112, 9.114, 9.115, 9.116, 9.117, 9.118, 9.119, 9.120, 9.121, 9.122, 9.123, 9.128, 9.129, 9.130, 9.131, 9.132, 9.133, 9.134, 9.135, 9.137, 9.138, 9.139, 9.140, 9.141, 9.142, 9.143, 9.145, 9.146, 9.147, 9.148, 9.149, 9.150, 9.151, 9.152, 9.153, 9.154, 9.155, 9.162, 9.165, 9.169, 9.173, 9.176, 9.177, 9.179
IsAuthorizationValue 4.2, 4.45
IsAuthorsValue 4.2, 4.36
IsBalanceValue 2.61, 2.94
IsBasicLayoutObjectClassDescription 2.30, 2.33, 2.34
IsBasicLayoutObjectDescription 2.48, 2.50, 2.52, 2.135, 9.103, 9.104, 9.105, 9.106, 9.107, 9.109, 9.111, 9.112, 9.114, 9.115, 9.116, 9.117, 9.118, 9.119, 9.120, 9.121, 9.122, 9.123, 9.128, 9.129, 9.130, 9.131, 9.132, 9.133, 9.134, 9.135, 9.137, 9.138, 9.139, 9.140, 9.151, 9.152, 9.153, 9.154, 9.155, 9.162, 9.165, 9.169, 9.173, 9.176, 9.177, 9.179
IsBasicLogicalObjectClassDescription 2.22, 2.23, 2.34
IsBasicLogicalObjectDescription 2.39, 2.40, 2.50, 2.135
IsBasicObjectClassDescription 2.34, 2.138, 2.153
IsBasicObjectDescription 2.50, 2.138, 2.153
IsBasicPageClassDescription 2.27, 2.33, 2.158
IsBasicPageDescription 2.44, 2.45, 2.52
IsBindingName 2.72, 2.73

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

IsBindingReference 2.65, 2.67, 2.72
IsBindingReferenceExpression 2.71, 2.72
IsBindingSelectionFunction 2.71, 2.74
IsBindingsValue 2.61, 2.89
IsBindingsValueExpression 2.35, 2.40, 2.63, 2.89
IsBitsPerColourComponentValue 7.1, 7.9, 7.10, 7.31
IsBlinkingValue 6.27, 6.31
IsBlockAlignmentsValue 4.2, 4.17
IsBlockAlignmentValue 2.61, 2.101, 4.10, 4.17
IsBlockClassDescription 2.29, 2.33
IsBlockDescription 2.15, 2.47, 2.52, 2.136
IsBordersValue 4.2, 4.21
IsBorderValue 2.61, 2.93, 4.10, 4.21
IsCalibrationDataValue 4.81, 4.82
IsCat 1.3, 1.11, 1.12, 1.16, 2.83, 2.117, 2.129, 4.92
IsCharacter 2.117, 2.118
IsCharacterCodingSpecification 4.24, 6.2
IsCharacterContentArchitectureClassValue 2.87, 6.26
IsCharacterContentCodingAttribute 2.116, 6.2, 6.13
IsCharacterContentInformationValue 2.114, 6.14, 6.111
IsCharacterContentPortionAttributeSet 6.11, 6.14
IsCharacterContentPortionDescription 2.54, 6.5, 6.7, 6.9, 6.11, 9.95
IsCharacterContentPresentationAttribute 2.88, 6.1, 6.3, 6.12
IsCharacterContentTypeOfCodingValue 2.113, 6.13, 6.14, 6.20
IsCharacterFontsValue 6.12, 6.16
IsCharacterInitialOffsetValue 6.12, 6.35
IsCharacterLineProgressionValue 6.12, 6.45
IsCharacterOrientationValue 6.12, 6.17
IsCharacterPathValue 6.12, 6.19
IsCharacterPresentationFeature 4.25, 6.3
IsCharacterSetList 8.22, 8.23
IsCharacterSetSpecification 8.23, 8.24
IsCharacterSetsValue 4.2, 6.12, 6.23
IsCharacterSpacingValue 6.12, 6.21
IsCharSeq 2.61, 2.117, 4.27, 4.29, 4.53, 4.54, 4.55, 4.57, 4.68
IsCIECalibrationData 4.82, 4.83
IsCIEColour 4.91, 4.96
IsClassIdImSubIn 2.24, 2.25, 2.26, 2.28, 2.142, 2.150
IsClassImSubComplete 2.9, 2.139
IsClippingValue 7.1, 7.8, 7.11
IsCMYKCalibrationData 4.82, 4.85
IsCMYKColour 4.91, 4.95
IsCodeExtensionAnnouncersValue 6.12, 6.38
IsCodeExtensionControlFn 6.56, 6.91
IsCodingAttributesValue 2.61, 2.116
IsCol 1.1, 2.9, 2.13, 2.88, 2.90, 2.106, 2.116, 2.124, 2.137, 2.138, 2.139, 2.140, 2.148, 2.149, 2.155, 8.12, 8.19, 8.23, 8.25, 8.29, 8.32, 8.34, 8.40, 8.44, 9.23, 9.24, 9.36, 9.37
IsColourCalibrationTypeValue 4.77, 4.78
IsColourCharacteristicsValue 4.2, 4.75
IsColourDataScalingValue 4.81, 4.93
IsColourExpression 2.61, 2.93, 2.98, 2.126, 4.10, 4.71, 4.72, 4.73
IsColourLookupTable 4.84, 4.87
IsColourModesPresentValue 4.75, 4.76
IsColourOfLayoutObjectsValue 4.2, 4.71
IsColourRepresentationsValue 8.6, 8.43
IsColourSpaceDefinition 4.80, 4.81

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

IsColourSpacesListValue 4.2, 4.80
IsColourSpacesPresentValue 4.75, 4.77
IsColourSpaceTypeValue 4.77, 4.79, 4.81
IsColourSpecificationValue 2.128, 2.129, 2.133
IsColoursValue 4.2, 4.20
IsColourTable 2.61, 2.131, 4.10, 4.74
IsColourTableCorrection 4.88, 4.89
IsColourTableEntry 2.132, 2.133
IsColourTableList 4.87, 4.88
IsColourTableSpecification 8.44, 8.45
IsColourTablesValue 4.2, 4.74
IsColourToleranceValue 2.128, 2.130, 2.133, 4.75
IsColourValue 2.61, 2.97, 4.10, 4.20
IsCommonContentElement 6.53, 6.54, 6.55, 6.56
IsCompleteGenericDescription 2.5, 2.10, 2.136
IsCompletelyDescribedBy 2.5, 2.6, 2.136
IsCompositeLayoutObjectClassDescription 2.30, 2.31, 2.32
IsCompositeLayoutObjectDescription 2.48, 2.49, 2.51
IsCompositeLogicalObjectClassDescription 2.21, 2.23, 2.31
IsCompositeLogicalObjectDescription 2.38, 2.40, 2.49
IsCompositeObjectClassDescription 2.31, 2.136, 2.139
IsCompositeObjectDescription 2.49, 2.136, 2.140, 2.152
IsCompositePageClassDescription 2.26, 2.32, 2.158
IsCompositePageDescription 2.15, 2.43, 2.45, 2.51
IsCompressionValue 7.1, 7.9, 7.10, 7.21
IsConcatenationValue 2.61, 2.102
IsConstituent 2.157
IsConstituentId 4.64, 4.65
IsConstructionExpression 2.61, 2.84, 2.139, 2.142, 2.145, 2.146, 2.147
IsConstructionTerm 2.84, 2.85, 2.86
IsConstructionType 2.84, 2.85, 2.86
IsContainedIn 2.6, 2.141
IsContentArchitectureClassesValue 4.2, 4.11
IsContentArchitectureClassValue 2.61, 2.87, 4.10, 4.11
IsContentBackgroundColoursValue 4.2, 4.72
IsContentComplete 2.9, 2.13, 2.138
IsContentConnected 2.9, 2.13, 2.137
IsContentForegroundColoursValue 4.2, 4.73
IsContentInformationValue 2.61, 2.114
IsContentPortionDescription 2.9, 2.10, 2.11, 2.12, 2.13, 2.14, 2.15, 2.54, 2.137, 2.138, 2.153, 2.155, 2.157, 2.159, 2.160
IsCopyrightValue 4.2, 4.37
IsCrossingOutValue 6.27, 6.33
IsCurrentInstanceFunction 2.69, 2.74, 2.75
IsDateAndTimeSeq 4.2, 4.39
IsDateAndTimeValue 4.2, 4.31, 4.33, 4.39, 4.67
IsDateValue 4.12, 4.38, 4.40
IsDefaultableCharacterContentArchitectureAttribute 4.10, 6.1
IsDefaultableGeometricGraphicsContentArchitectureAttribute 4.10, 8.1
IsDefaultableRasterGraphicsContentArchitectureAttribute 4.10, 7.1
IsDefaultValueListsValue 2.61, 2.90
IsDelimiterControlFn 6.55, 6.108
IsDescribedBy 2.8, 2.135, 2.136
IsDimensionsValue 2.61, 2.92, 4.10
IsDirectColourExpression 2.126, 2.128
IsDirectColourValue 8.10, 8.14, 8.36, 8.43, 8.45, 8.54

--`,,```,,,,````-`-`,,`,,`,`,,`---

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

IsDirectColourValuePair 8.9, 8.10
IsDistributionListValue 4.2, 4.41
IsDocumentApplicationProfileDefaultsValue 4.2, 4.10
IsDocumentApplicationProfileValue 4.2, 4.9
IsDocumentBody 2.2, 2.5, 2.6, 2.7, 2.8
IsDocumentDescription 2.1, 2.4
IsDocumentProfile 2.2, 2.4, 2.5, 2.6, 2.7, 2.157, 4.1
IsDocumentSealValue 4.62, 4.63, 4.69
IsEdgeBundleSpecification 8.40, 8.41
IsEdgeRenditionValue 8.6, 8.39
IsEdgeTypeValue 8.39, 8.41, 8.42
IsEmptyCat 1.6, 2.105, 6.50, 6.51, 6.52, 6.64, 6.65, 6.67, 6.68, 6.69, 6.70, 6.73, 6.76, 6.81, 6.82, 6.85, 6.89
IsEmptyCol 1.4, 6.4
IsEmptyNom 1.5
IsEncipheredBodyPartId 2.119, 2.123
IsEncipheredDocumentProfileDescription 2.18, 2.58
IsEncipheredDocumentProfileId 4.50, 4.56
IsEncipheredDocumentProfilesValue 4.2, 4.50
IsEncipheredValue 2.61, 2.119
IsEscapeControlFn 6.56, 6.110
IsExternalReferenceValue 4.2, 4.6, 4.33, 4.43
IsFactorSet 2.165
IsFillBundleSpecification 8.32, 8.33
IsFilledAreaRenditionValue 8.6, 8.31
IsFillOrdersValue 4.2, 4.18
IsFillOrderValue 2.61, 2.103, 4.18
IsFirstLineOffsetValue 6.12, 6.22
IsFloatabilityRangeValue 2.61, 2.134
IsFontListSet 8.22, 8.29
IsFontReference 4.26, 4.27
IsFontsListValue 4.2, 4.26
IsFormatted 2.4, 2.7
IsFormattedCharacterContent 6.7, 6.8
IsFormattedCharacterDescription 6.4, 6.7
IsFormattedContentElement 6.50, 6.53
IsFormattedProcessable 2.4, 2.6
IsFormattedProcessableCharacterContent 6.9, 6.10
IsFormattedProcessableCharacterDescription 6.4, 6.9
IsFormattedProcessableContentElement 6.52, 6.55
IsFormattedProcessableRasterGraphicsContent 7.4, 7.5
IsFormattedRasterGraphicsContent 7.4, 7.6
IsFormattingIndicatorValue 6.12, 6.18
IsFrameClassDescription 2.28, 2.32
IsFrameDescription 2.15, 2.46, 2.51
IsGenericDescription 2.9, 2.10, 2.11, 2.12, 2.151, 2.165
IsGenericDocumentDescription 2.1, 2.2
IsGenericId 2.164
IsGenericLayoutDescription 2.5, 2.6, 2.7, 2.8, 2.12, 2.135
IsGenericLogicalDescription 2.5, 2.6, 2.8, 2.11, 2.135
IsGeometricGraphicsCodingSpecification 4.24, 8.2
IsGeometricGraphicsColourValue 8.11, 8.13, 8.14, 8.18, 8.20, 8.22, 8.27, 8.31, 8.33, 8.39, 8.41, 8.46
IsGeometricGraphicsContentArchitectureClassValue 2.87, 8.51
IsGeometricGraphicsContentCodingAttribute 2.116, 8.2, 8.7
IsGeometricGraphicsContentInformationValue 2.114, 8.8, 8.53
IsGeometricGraphicsContentPortionAttributeSet 8.5, 8.8
IsGeometricGraphicsContentPortionDescription 2.54, 8.4, 8.5, 9.95

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

IsGeometricGraphicsContentPresentationAttribute 2.88, 8.1, 8.3, 8.6
IsGeometricGraphicsContentTypeOfCodingValue 2.113, 8.7, 8.52
IsGeometricGraphicsEncodingAnnouncerValue 8.6, 8.9
IsGeometricGraphicsPresentationFeature 4.25, 8.3
IsGraphicCharacter 6.64, 6.66, 6.112, 6.113
IsGraphicCharacterString 6.56, 6.112
IsGraphicCharacterSubrepertoireValue 6.12, 6.24
IsGraphicRenditionValue 6.12, 6.27
IsGridSpecification 4.90, 4.91
IsGridSpecificationValue 4.85, 4.90
IsHatchIndexValue 8.31, 8.33, 8.37
IsIdContIn 2.144, 2.153
IsIdImSubIn 2.41, 2.42, 2.43, 2.46, 2.143
IsImageDimensionsValue 7.1, 7.8, 7.16
IsImageInversionValue 6.27, 6.32
IsImSubComplete 2.13, 2.140
IsIncompletePLD 6.70, 6.72, 6.72
IsIncompletePLDInSRS 6.77, 6.79, 6.79
IsIncompletePLU 6.70, 6.71, 6.71
IsIncompletePLUInSRS 6.77, 6.78, 6.78
IsIndentationValue 6.12, 6.34
IsIndexedColourExpression 2.126, 2.127
IsInitialGenericIn 2.10, 2.148
IsInitialSpecificIn 2.13, 2.149
IsInt 2.48, 2.67, 2.89, 2.92, 2.125, 2.129, 4.26, 4.62, 4.86, 4.87, 4.89, 4.92, 4.94, 4.95, 4.96, 6.22, 6.43, 6.46, 6.81, 6.89, 6.90, 7.14
IsInterchangeSet 2.1
IsInterleavingFormatValue 7.1, 7.9, 7.10, 7.32
IsISO2022Character 6.92, 6.93
IsISO2022String 6.23, 6.38, 6.91, 6.92
IsISO6429Character 6.94, 6.95
IsISO6429String 6.91, 6.94
IsISO6937MSCharacter 1.20, 1.21
IsISO6937MSString 1.20, 2.61, 2.73, 2.89, 2.105, 4.8
IsISO6937SubrepertoireId 6.24, 6.25
IsISO9541Part2AttributeSet 4.29, 4.30
IsItemizationValue 6.12, 6.46
IsKerningOffsetValue 6.12, 6.36
IsKeyAdditionalInformationValue 4.52, 4.54
IsKeyInformationValue 4.51, 4.52, 4.66
IsKeyMethodValue 4.52, 4.55
IsLayoutCatName 2.8, 2.61, 2.104, 2.105, 2.106, 2.108, 2.134
IsLayoutContentPortionDescription 2.160
IsLayoutContentPortionId 2.61, 2.111
IsLayoutControlFn 6.53, 6.55, 6.100
IsLayoutId 2.162
IsLayoutObjectClassDescription 2.12, 2.30, 2.35
IsLayoutObjectClassId 2.8, 2.12, 2.25, 2.26, 2.27, 2.28, 2.29, 2.41, 2.42, 2.43, 2.44, 2.46, 2.47, 2.55, 2.61, 2.81, 2.82, 2.94, 2.104, 2.108, 2.111, 2.134, 2.164
IsLayoutObjectDescription 2.15, 2.48, 2.53
IsLayoutObjectId 2.15, 2.42, 2.43, 2.44, 2.46, 2.47, 2.78, 2.79, 2.94, 2.111, 2.163
IsLayoutObjectReference 2.61, 2.104
IsLayoutPathsValue 4.2, 4.15
IsLayoutPathValue 2.61, 2.95, 4.10, 4.15
IsLayoutRootClassDescription 2.19, 2.24, 2.32
IsLayoutRootDescription 2.15, 2.36, 2.41, 2.51

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

IsLayoutStyle 2.16, 2.55, 2.157, 9.6, 9.8, 9.20
IsLayoutStyleId 2.8, 2.55, 2.61, 2.100, 4.65
IsLayoutStyleSet 2.5, 2.6, 2.8, 2.16
IsLineBundleSpecification 8.12, 8.13
IsLineLayoutTableValue 6.12, 6.42
IsLineRenditionValue 8.6, 8.11
IsLineSpacingValue 6.12, 6.44
IsLineTypeValue 8.11, 8.13, 8.17
IsLocalFileReferenceValue 4.2, 4.44
IsLocationValue 4.67, 4.68
IsLogicalContentPortionDescription 2.159
IsLogicalContentPortionId 2.61, 2.112
IsLogicalControlFn 6.54, 6.55, 6.104
IsLogicalId 2.161
IsLogicalObjectClassDescription 2.11, 2.23, 2.35
IsLogicalObjectClassId 2.11, 2.21, 2.22, 2.37, 2.38, 2.39, 2.61, 2.80, 2.82, 2.112, 2.164
IsLogicalObjectDescription 2.14, 2.40, 2.53
IsLogicalObjectId 2.14, 2.38, 2.39, 2.61, 2.77, 2.79, 2.108, 2.112, 2.134, 2.163
IsLogicalRootClassDescription 2.19, 2.20, 2.21
IsLogicalRootDescription 2.36, 2.37, 2.38
IsMarkerBundleSpecification 8.19, 8.20
IsMarkerRenditionValue 8.6, 8.18
IsMarkerTypeValue 8.18, 8.20, 8.21
IsMediumTypesValue 4.2, 4.14
IsMediumTypeValue 2.61, 2.98, 4.10, 4.14
IsMethodInformationValue 4.51, 4.53, 4.66
IsNat 1.7, 1.9, 2.92, 2.130, 4.2, 4.75, 6.16, 6.21, 6.37, 6.40, 6.44, 6.81, 6.90, 6.99, 6.103, 7.17, 7.18, 7.22, 7.26, 7.27, 7.31, 8.11, 8.13, 8.18, 8.20, 8.22, 8.27, 8.31, 8.33, 8.35, 8.39, 8.41
IsNeCat 1.3, 1.13, 1.14, 1.15, 1.20, 1.22, 1.23, 2.64, 2.85, 2.94, 4.4, 4.32, 4.39, 4.64, 6.27, 6.43, 6.64, 6.65, 6.66, 6.67, 6.68, 6.69, 6.70, 6.71, 6.72, 6.73, 6.74, 6.75, 6.76, 6.77, 6.78, 6.79, 6.81, 6.82, 6.83, 6.84, 6.85, 6.86, 6.87, 6.89, 6.90, 6.92, 6.94, 6.112, 7.29, 7.31, 8.36, 9.148
IsNeCol 1.1, 2.2, 2.3, 2.5, 2.6, 2.7, 2.8, 2.9, 2.13, 2.16, 2.17, 2.35, 2.40, 2.48, 2.89, 2.121, 2.132, 4.3, 4.8, 4.11, 4.13, 4.14, 4.15, 4.16, 4.17, 4.18, 4.19, 4.20, 4.21, 4.22, 4.23, 4.26, 4.28, 4.34, 4.35, 4.36, 4.37, 4.38, 4.41, 4.43, 4.44, 4.46, 4.48, 4.49, 4.71, 4.72, 4.73, 4.74, 4.76, 4.77, 4.80, 4.88, 4.90, 8.45
IsNeNom 1.2, 2.61, 2.62, 2.90, 2.91, 2.92, 2.98, 2.107, 2.108, 2.109, 2.119, 2.120, 2.127, 2.128, 2.131, 2.133, 2.134, 4.2, 4.10, 4.24, 4.25, 4.27, 4.29, 4.33, 4.34, 4.35, 4.36, 4.37, 4.41, 4.44, 4.47, 4.50, 4.51, 4.52, 4.57, 4.58, 4.60, 4.62, 4.63, 4.66, 4.67, 4.69, 4.75, 4.77, 4.81, 4.83, 4.84, 4.85, 4.86, 4.87, 4.89, 4.91, 4.93, 4.94, 4.95, 4.96, 6.14, 6.35, 7.10, 7.14, 7.16, 7.17, 7.18, 8.8, 8.50
IsNnInt 1.7, 1.8, 1.22, 1.23, 2.91, 2.93, 2.107, 2.109, 2.127, 2.128, 2.131, 2.133, 2.147, 4.2, 4.29, 4.81, 4.89, 6.34, 6.35, 6.36, 6.42, 6.99, 6.103, 7.16, 7.23, 7.24, 7.28, 7.31, 8.9, 8.14, 8.45, 8.50
IsNnReal 1.24, 8.15
IsNnVDCValue 8.15, 8.22, 8.57
IsNom 1.2, 1.18, 2.91, 2.92, 2.93, 4.33, 4.82, 6.2, 6.3, 6.16, 6.36, 6.42, 6.46, 7.2, 7.3, 8.2, 8.3, 8.9, 8.11, 8.13, 8.18, 8.20, 8.22, 8.24, 8.27, 8.30, 8.31, 8.33, 8.35, 8.38, 8.39, 8.41, 8.43, 8.45, 8.46, 8.47, 8.48, 8.50, 9.23, 9.24, 9.36, 9.37, 9.180, 9.182
IsNumberOfDiscardedPelsValue 7.1, 7.9, 7.10, 7.24
IsNumberOfLinesPerTileValue 7.1, 7.9, 7.10, 7.26
IsNumberOfLinesValue 7.9, 7.10, 7.22
IsNumberOfPelsPerLineValue 7.1, 7.9, 7.10, 7.23
IsNumberOfPelsPerTileLineValue 7.1, 7.9, 7.10, 7.27
IsNumericExpression 2.63, 2.65, 2.67, 2.67
IsNumericFunction 2.67, 2.68
IsObjectClassDescription 2.9, 2.35, 2.135, 2.136, 2.139, 2.148, 2.157, 9.11, 9.15
IsObjectClassId 2.61, 2.71, 2.75, 2.82, 2.83, 2.86, 4.8, 4.65
IsObjectDescription 2.13, 2.53, 2.135, 2.136, 2.140, 2.149, 2.152, 2.157
IsObjectId 2.48, 2.61, 2.67, 2.71, 2.75, 2.79, 2.89, 4.65

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

IsObjectOrObjectClassIdExpression 2.61, 2.63, 2.67, 2.69, 2.69, 2.74, 2.75, 2.108, 2.134
IsObjectOrObjectClassSelectionFunction 2.69, 2.70
IsObjectTypeValue 2.61, 2.75, 2.76
IsOctetString 1.10, 2.48, 2.61, 2.65, 2.89, 4.54, 4.57, 4.63, 4.67, 7.25, 8.53
IsODASecurityLabelValue 4.2, 4.57
IsODAVersionValue 4.2, 4.12
IsOffsetValue 2.61, 2.107
IsOrientationVectorPair 8.22, 8.28
IsOrphanSizeValue 6.12, 6.37
IsOwnersValue 4.2, 4.35
IsPageClassDescription 2.158
IsPageDescription 2.15, 2.45
IsPageDimensionsValue 4.2, 4.13
IsPagePositionsValue 4.2, 4.22
IsPageSetClassDescription 2.25, 2.32
IsPageSetDescription 2.15, 2.42, 2.51
IsPairOfNnInt 1.8, 2.61, 2.100, 2.110, 2.122, 4.10, 4.22, 4.56, 4.59, 4.61, 4.70, 7.11
IsPairOfPosInt 1.9, 2.98, 4.2, 4.13
IsPairwiseKerningValue 6.12, 6.39
IsParameterizedLayoutControlFn 6.100, 6.103
IsParameterizedLayoutControlFnName 6.102, 6.103
IsParameterizedLogicalControlFn 6.104, 6.107
IsParameterizedLogicalControlFnName 6.106, 6.107
IsParameterizedSharedControlFn 6.96, 6.99
IsParameterizedSharedControlFnName 6.98, 6.99
IsParameterlessLayoutControlFn 6.100, 6.101
IsParameterlessLogicalControlFn 6.104, 6.105
IsParameterlessSharedControlFn 6.96, 6.97
IsPatternSizeValue 8.31, 8.38
IsPatternTableSpecification 8.34, 8.35
IsPelPathValue 7.1, 7.8, 7.13
IsPelSpacingValue 7.1, 7.8, 7.17
IsPelTransmissionDensityValue 7.1, 7.8, 7.15
IsPersonalName 4.34, 4.35, 4.36, 4.41, 4.45, 4.46, 4.47, 4.67
IsPictureDimensionsValue 8.6, 8.50
IsPictureOrientationValue 8.6, 8.49
IsPlaceholder 1.19, 2.20, 2.21, 2.22, 2.24, 2.25, 2.26, 2.27, 2.28, 2.29, 2.37, 2.38, 2.39, 2.40, 2.41, 2.42, 2.43, 2.44, 2.46, 2.47, 2.48, 2.55, 2.56, 2.57, 2.58, 2.59, 2.60, 2.61, 2.91, 2.92, 2.98, 2.107, 2.109, 2.120, 2.127, 2.128, 2.130, 2.131, 2.133, 6.12, 6.14, 7.5, 7.8, 7.9, 7.10, 8.6, 8.9, 8.11, 8.18, 8.22, 8.31, 8.39, 8.43, 8.46, 8.47, 9.42, 9.43, 9.44, 9.46, 9.47, 9.49, 9.51, 9.52, 9.64, 9.65, 9.66, 9.70, 9.71, 9.72, 9.74, 9.80, 9.82, 9.85, 9.86, 9.87, 9.94, 9.95, 9.96, 9.97, 9.98, 9.99, 9.100, 9.101, 9.102, 9.103, 9.104, 9.105, 9.106, 9.107, 9.109, 9.111, 9.112, 9.114, 9.115, 9.116, 9.117, 9.118, 9.119, 9.120, 9.121, 9.122, 9.123, 9.128, 9.129, 9.130, 9.131, 9.132, 9.133, 9.134, 9.135, 9.137, 9.138, 9.139, 9.140, 9.141, 9.142, 9.143, 9.145, 9.146, 9.147, 9.148, 9.149, 9.150, 9.151, 9.152, 9.153, 9.154, 9.155, 9.162, 9.165, 9.169, 9.173, 9.176, 9.177, 9.179
IsPositionValue 2.61, 2.91
IsPosReal 1.25, 2.130, 8.22, 8.27
IsPostEncipheredDocumentBodyPartDescription 2.18, 2.60
IsPostEncipheredDocumentBodyPartId 4.60, 4.61
IsPostEncipheredDocumentBodyPartsValue 4.2, 4.60
IsPostureValue 6.27, 6.29
IsPreEncipheredDocumentBodyPartDescription 2.18, 2.59
IsPreEncipheredDocumentBodyPartId 4.58, 4.59
IsPreEncipheredDocumentBodyPartsValue 4.2, 4.58
IsPreparersValue 4.2, 4.34
IsPresentationAttributesValue 2.61, 2.88
IsPresentationFeaturesValue 4.2, 4.25

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

IsPresentationStyle 2.17, 2.56, 2.157, 9.7, 9.9, 9.18
IsPresentationStyleId 2.56, 2.61, 2.110, 4.65
IsPresentationStyleSet 2.5, 2.6, 2.7, 2.8, 2.17
IsPrivilegedRecipientInformationValue 4.50, 4.51, 4.58, 4.60
IsProcessable 2.4, 2.5
IsProcessableCharacterContent 6.5, 6.6
IsProcessableCharacterDescription 6.4, 6.5
IsProcessableContentElement 6.51, 6.54
IsProfCharSeq 4.2, 4.3, 4.4, 4.6, 4.12, 4.33, 4.34, 4.35, 4.36, 4.41, 4.44, 4.45, 4.47, 4.81, 4.85
IsProfileAttributeSet 4.1, 4.2
IsProfileAttributeSetPart2 2.62
IsProfileCharacter 4.4, 4.5
IsProfileCodingAttributesValue 4.2, 4.24
IsProportionalLineSpacingValue 6.12, 6.41
IsProtectedDocumentDescription 2.1, 2.3
IsProtectedPart 2.3, 2.18
IsProtectedPartId 2.57, 2.58, 2.59, 2.60, 2.61, 2.122, 2.123
IsProtectionsValue 4.2, 4.16
IsProtectionValue 2.61, 2.99, 4.16
IsRasterGraphicsCodingSpecification 4.24, 7.2
IsRasterGraphicsContentArchitectureClassValue 2.87, 7.19
IsRasterGraphicsContentCodingAttribute 2.116, 7.2, 7.9
IsRasterGraphicsContentInformationValue 2.114, 7.10, 7.25
IsRasterGraphicsContentPortionAttributeSet 7.7, 7.10
IsRasterGraphicsContentPortionDescription 2.54, 7.5, 7.6, 7.7, 9.95
IsRasterGraphicsContentPresentationAttribute 2.88, 7.3, 7.8
IsRasterGraphicsContentTypeOfCodingValue 2.113, 7.1, 7.9, 7.10, 7.20
IsRasterGraphicsInitialOffsetValue 7.1, 7.8, 7.14
IsRasterGraphicsLineProgressionValue 7.1, 7.8, 7.12
IsRasterGraphicsPresentationFeature 4.25, 7.3
IsReal 1.24, 1.25, 2.129, 4.86, 4.89, 4.92, 4.94, 4.95, 4.96, 8.22, 8.27, 8.30
IsReferencePropertiesValue 4.28, 4.29
IsReferenceWhiteValue 4.83, 4.84, 4.85, 4.86
IsRegionOfInterestSpecificationValue 8.6, 8.48
IsRegisteredDesignationSequenceTail 8.24, 8.26
IsRegisteredEdgeType 8.42, 8.61
IsRegisteredHatchIndex 8.37, 8.60
IsRegisteredLineType 8.17, 8.58
IsRegisteredMarkerType 8.21, 8.59
IsResourcesValue 4.2, 4.8
IsRevisionHistoryValue 4.2, 4.32
IsRevisionValue 4.32, 4.33
IsRGBCalibrationData 4.82, 4.84
IsRootClassDescription 2.9, 2.10, 2.19
IsRootDescription 2.13, 2.36
IsSameLayoutObjectValue 2.61, 2.108
IsScaleAndOffsetValue 4.93, 4.94
IsSealedDocumentBodyPartsValue 4.2, 4.62
IsSealedDocumentProfileDescription 2.18, 2.57
IsSealedDocumentProfileId 4.69, 4.70
IsSealedDocumentProfileInformationValue 2.61, 2.121
IsSealedDocumentProfilesValue 4.2, 4.69
IsSealedInformationValue 4.63, 4.67
IsSealedValue 2.61, 2.120
IsSealId 2.124, 2.125
IsSealMethodValue 4.63, 4.66

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

IsSeparationValue 2.61, 2.109
IsSeqClassIdImSubIn 2.136, 2.145
IsSeqOfConstituentIds 4.62, 4.64
IsSeqOfDiffNnInt 1.22, 2.61, 2.119
IsSeqOfDirectColourValue 8.35, 8.36
IsSeqOfNnInt 1.23, 2.77, 2.78, 2.80, 2.81, 2.161, 2.162, 8.35
IsSeqOfObjectClassId 2.83, 2.146
IsSetOfColourTableEntries 2.131, 2.132
IsSetOfColourTableSpecifications 8.43, 8.44
IsSetOfDatesValue 4.37, 4.38
IsSetOfEdgeBundleSpecifications 8.39, 8.40
IsSetOfExternalReferences 4.2, 4.43
IsSetOfFillBundleSpecifications 8.31, 8.32
IsSetOfLayoutCatNames 2.61, 2.106
IsSetOfLineBundleSpecifications 8.11, 8.12
IsSetOfMarkerBundleSpecifications 8.18, 8.19
IsSetOfPatternTableSpecifications 8.31, 8.34
IsSetOfPersonalNames 4.33, 4.46, 4.51, 4.62, 4.69
IsSetOfProfCharSeq 4.2, 4.3, 4.37
IsSetOfReferenceProperties 4.27, 4.28
IsSetOfSealIds 2.120, 2.124
IsSetOfTextBundleSpecifications 8.22, 8.25
IsSharedControlFn 6.56, 6.96
IsSingle 1.11, 1.12
IsSpaceControlFn 6.56, 6.109
IsSpacingRatioValue 7.1, 7.8, 7.18
IsSpecificationModeValue 8.11, 8.16, 8.18, 8.39
IsSpecificDescription 2.13, 2.14, 2.15
IsSpecificId 2.163
IsSpecificLayoutDescription 2.6, 2.7, 2.8, 2.15, 2.135, 2.141
IsSpecificLogicalDescription 2.5, 2.6, 2.8, 2.14, 2.135, 2.141
IsSpotset 1.17
IsStringExpression 2.61, 2.63, 2.64
IsStringFunction 2.65, 2.66
IsTabReferenceString 6.42, 6.43
IsTextAlignmentSpecification 8.22, 8.30
IsTextBundleSpecification 8.25, 8.27
IsTextRenditionValue 8.6, 8.22
IsTileEncodingValue 7.29, 7.30
IsTileTypesValue 7.1, 7.9, 7.10, 7.29
IsTilingOffsetValue 7.1, 7.9, 7.10, 7.28
IsTransformationSpecificationValue 8.6, 8.47
IsTransparenciesValue 4.2, 4.19
IsTransparencySpecificationValue 8.6, 8.46
IsTransparencyValue 2.61, 2.96, 4.10, 4.19
IsTypeOfCodingValue 2.61, 2.113, 4.10, 4.23
IsTypesOfCodingValue 4.2, 4.23
IsUnderliningValue 6.27, 6.30
IsValidCommonContentSemantics 6.57, 6.58, 6.59, 6.60
IsValidDelimiterControlSemantics 6.59, 6.62
IsValidFormattedContentSemantics 6.47, 6.57
IsValidFormattedContentSyntax 6.47, 6.50, 6.50
IsValidFormattedProcessableContentSemantics 6.49, 6.59
IsValidFormattedProcessableContentSyntax 6.49, 6.52, 6.52
IsValidGCC0 6.63, 6.64, 6.64
IsValidGCC12 6.63, 6.65, 6.65, 6.66

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

IsValidGCC12End 6.65, 6.66, 6.66
IsValidLF 6.63, 6.67
IsValidLogicalControlSemantics 6.58, 6.59, 6.61
IsValidNextLF0 6.67, 6.68, 6.68, 6.69
IsValidNextLF1 6.67, 6.68, 6.69, 6.69
IsValidPLUD 6.63, 6.70, 6.70, 6.71, 6.72
IsValidPLUDInSRS 6.63, 6.76, 6.76, 6.77
IsValidPLUDInSRSEnd 6.76, 6.77, 6.77, 6.78, 6.79
IsValidProcessableContentSemantics 6.48, 6.58
IsValidProcessableContentSyntax 6.48, 6.51, 6.51
IsValidPTX 6.61, 6.82, 6.82, 6.84
IsValidPTXEnd 6.83, 6.84, 6.84
IsValidPTXNext 6.82, 6.83, 6.83
IsValidSharedControlSemantics 6.60, 6.63
IsValidSOS 6.62, 6.85, 6.85, 6.86
IsValidSOSEnd 6.85, 6.86, 6.86
IsValidSOSNest 6.86, 6.87, 6.87
IsValidSRS 6.63, 6.73, 6.73, 6.74
IsValidSRSEnd 6.73, 6.74, 6.74
IsValidSRSNest 6.74, 6.75, 6.75
IsValidVP 6.63, 6.80
IsValidVPInSRS 6.63, 6.88
IsValidVPSum 6.80, 6.81, 6.81
IsValidVPSumInSRS 6.88, 6.89, 6.89
IsValidVPSumInSRSEnd 6.89, 6.90, 6.90
IsVDCPair 8.28, 8.31, 8.47, 8.48, 8.55
IsVDCValue 8.38, 8.55, 8.56
IsWeightValue 6.27, 6.28
IsWidowSizeValue 6.12, 6.40
IsWidthValue 8.11, 8.13, 8.15, 8.18, 8.20, 8.39, 8.41
itemization 6.12, 9.116
ITEMIZATION_VALUE 9.116

JFY 6.102, 6.103
justified 6.15

K value 4.95
kerning offset 6.12, 9.117
KERNING_OFFSET_VALUE 9.117
key information 4.51
key method 4.52
keywords 2.121, 4.1, 4.2

languages 2.121, 4.1, 4.2
LASTC 1.14, 7.7, 9.16, 9.160, 9.161
layout category 2.8, 2.55, 2.61, 9.62
layout object class 2.55, 2.61, 9.63
layout object 2.8, 2.108, 2.134, 9.59, 9.60
layout path 2.28, 2.46, 2.61, 2.90, 4.10, 9.64
layout paths 2.121, 4.1, 4.2
layout stream categories 2.25, 2.26, 2.27, 2.28, 2.29, 2.42, 2.43, 2.44, 2.46, 2.47, 2.61, 9.65
layout stream sub-categories 2.25, 2.26, 2.27, 2.28, 2.29, 2.42, 2.43, 2.44, 2.46, 2.47, 2.61, 9.66
layout style identifier 2.8, 2.16, 2.55, 2.61, 9.6, 9.8, 9.20
layout style 2.8, 2.20, 2.21, 2.22, 2.37, 2.38, 2.39, 2.61, 2.90, 9.2, 9.6, 9.13, 9.19, 9.20, 9.24, 9.27, 9.30, 9.32, 9.37
layout styles 2.5, 2.6, 2.7, 2.62, 2.121, 4.1, 4.2
LAYOUT_CATEGORY_VALUE 9.62
LAYOUT_OBJECT_CLASS_VALUE 9.63

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

LAYOUT_PATH_VALUE 9.64
LAYOUT_STREAM_CATEGORIES_VALUE 9.65
LAYOUT_STREAM_SUBCATEGORIES_VALUE 9.66
leading offset 2.91, 2.107, 9.75, 9.76
leading-edge 2.91, 2.93, 2.109, 9.89, 9.90
LEADING_EDGE_VALUE 9.89, 9.90
LEADING_OFFSET_VALUE 9.75, 9.76
left-hand aligned 2.91, 2.101
left-hand offset 2.91, 2.107, 9.75, 9.78
left-hand-edge 2.93
left 8.22, 8.30
LEFTHAND_OFFSET_VALUE 9.75, 9.78
LENGTH 1.11, 1.12, 1.16, 2.94, 2.111, 2.112, 2.129, 2.156, 6.43, 6.65
length 7.17, 9.139
LF 6.67, 6.68, 6.69, 6.71, 6.72, 6.74, 6.75, 6.81, 6.97
line aspect source flags 8.11, 9.155
Line Bundle Index 8.11, 8.13, 9.155
line bundle representation 8.13
line bundle specifications 8.11, 9.155
line colour asf 8.11, 9.155
Line Colour 8.11, 8.13, 9.155
line layout table 6.12, 9.118
line progression 6.12, 7.1, 7.8, 9.119, 9.133
line rendition 8.6, 9.155
line spacing value 7.18, 9.140
line spacing 6.12, 9.120
line type asf 8.11, 9.155
Line Type 8.11, 8.13, 9.155
line width asf 8.11, 9.155
Line Width Specification Mode 8.11, 9.155, 9.158
Line Width 8.11, 8.13, 9.155
line 7.32
LINE_LAYOUT_TABLE_VALUE 9.118
LINE_RENDITION_VALUE 9.155, 9.158
LINE_SPACING_VALUE 9.120
LINE_WIDTH_DEFAULT 9.155, 9.157
LINE_WIDTH_SPECIFICATION_MODE 9.157, 9.158
local colour precision 8.35
local file references 2.121, 4.1, 4.2
local filing date and time 2.121, 4.1, 4.2
location of the document 4.44
location 4.67
logical object 2.8, 2.108, 2.134, 9.59, 9.60, 9.88
logical source 2.8, 2.28, 2.61
logical stream category 2.55, 2.61, 9.67
logical stream sub-category 2.55, 2.61, 9.68
LOGICAL_STREAM_CATEGORY_VALUE 9.67
LOGICAL_STREAM_SUBCATEGORY_VALUE 9.68
LONGEST_VDC_EXTENT 9.157, 9.159, 9.163, 9.166, 9.174
look-up table 4.78
LOWER-ALPHA 2.66
LOWER-ROMAN 2.66

M value 4.95
m 4.87
MAKE-STRING 2.66

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

marker aspect source flags 8.18, 9.162
Marker Bundle Index 8.18, 8.20, 9.162
marker bundle representation 8.20
marker bundle specifications 8.18, 9.162
marker colour asf 8.18, 9.162
Marker Colour 8.18, 8.20, 9.162
marker rendition 8.6, 9.162
marker size asf 8.18, 9.162
Marker Size Specification Mode 8.18, 9.162, 9.164
Marker Size 8.18, 8.20, 9.162
marker type asf 8.18, 9.162
Marker Type 8.18, 8.20, 9.162
MARKER_RENDITION_VALUE 9.162, 9.164
MARKER_SIZE_DEFAULT 9.162, 9.163
MARKER_SIZE_SPECIFICATION_MODE 9.163, 9.164
matrices and look-up table 4.78
matrices 4.78
matrix1 4.84
matrix2 4.84
maximum CMY(K) grid size 4.75
Maximum Colour Index 8.9, 9.154
maximum dimension 2.92
maximum number of colour table entries 4.75
maximum number of look-up table entries 4.75
maximum size 2.92
MAX_PELS_PER_LINE 9.143, 9.144
medium type 2.26, 2.27, 2.43, 2.44, 2.61, 2.90, 4.10, 9.70, 9.71, 9.72
medium types 2.121, 4.1, 4.2
MEDIUM_TYPE_VALUE 9.69
method information 4.51
minimum colour tolerance 4.75
minimum dimension 2.92
minimum height 7.16, 8.50
minimum width 7.16, 8.50
MINIMUM_EDGE_LOSS 9.80, 9.81

n 4.87
names 4.33
NAMS 1.18, 2.4, 2.5, 2.6, 2.7, 2.8, 2.10, 2.11, 2.12, 2.14, 2.15, 2.20, 2.21, 2.22, 2.24, 2.25, 2.26, 2.27, 2.28, 2.29, 2.37, 2.38, 2.39, 2.41, 2.42, 2.43, 2.44, 2.46, 2.47, 2.54, 2.55, 2.56, 2.57, 2.58, 2.59, 2.60, 2.90, 2.91, 2.92, 2.93, 2.98, 2.107, 2.108, 2.109, 2.119, 2.120, 2.127, 2.128, 2.131, 2.133, 2.134, 2.135, 2.136, 2.138, 2.139, 2.154, 2.156, 2.159, 2.160, 2.165, 4.1, 4.10, 4.24, 4.25, 4.27, 4.29, 4.33, 4.34, 4.35, 4.36, 4.37, 4.41, 4.44, 4.47, 4.50, 4.51, 4.52, 4.57, 4.58, 4.60, 4.62, 4.63, 4.66, 4.67, 4.69, 4.75, 4.77, 4.81, 4.82, 4.83, 4.84, 4.85, 4.86, 4.87, 4.89, 4.91, 4.93, 4.94, 4.95, 4.96, 6.4, 6.16, 6.35, 6.36, 6.42, 6.46, 7.5, 7.6, 7.7, 7.14, 7.16, 7.17, 7.18, 8.9, 8.11, 8.13, 8.18, 8.20, 8.22, 8.24, 8.27, 8.30, 8.31, 8.33, 8.35, 8.38, 8.39, 8.41, 8.43, 8.45, 8.46, 8.47, 8.48, 8.50, 9.2, 9.3, 9.4, 9.5, 9.10, 9.12, 9.13, 9.14, 9.17, 9.19, 9.22, 9.23, 9.24, 9.27, 9.28, 9.29, 9.30, 9.31, 9.32, 9.35, 9.36, 9.37, 9.39, 9.40, 9.42, 9.43, 9.44, 9.46, 9.47, 9.49, 9.51, 9.52, 9.64, 9.65, 9.66, 9.70, 9.71, 9.72, 9.74, 9.80, 9.82, 9.85, 9.86, 9.87, 9.94, 9.95, 9.96, 9.97, 9.98, 9.99, 9.100, 9.101, 9.102, 9.103, 9.104, 9.105, 9.106, 9.107, 9.108, 9.109, 9.111, 9.112, 9.114, 9.115, 9.116, 9.117, 9.118, 9.119, 9.120, 9.121, 9.122, 9.123, 9.128, 9.129, 9.130, 9.131, 9.132, 9.133, 9.134, 9.135, 9.137, 9.138, 9.139, 9.140, 9.141, 9.142, 9.143, 9.145, 9.146, 9.147, 9.148, 9.149, 9.150, 9.151, 9.152, 9.153, 9.154, 9.155, 9.162, 9.165, 9.169, 9.173, 9.176, 9.177, 9.179
NBH 6.105
new layout object 2.55, 2.61, 9.73
NEW_LAYOUT_OBJECT_VALUE 9.73
ninth alternative font 6.16
no calibration 4.78

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

no itemization 6.46, 9.116
no 2.120, 6.18, 6.39, 6.41, 9.121, 9.122, 9 130
nominal page size 2.98, 9.69, 9.70
NOMINAL_PAGE_SIZE_VALUE 9.69, 9.70
non-concatenated 2.102, 9.48
none 2.119
normal horizontal 8.30, 9.165
normal order 2.91, 2.103, 9.57
normal vertical 8.30, 9.165
null background 7.30
null foreground 7.30
null 2.8, 2.24, 2.25, 2.26, 2.28, 2.41, 2.42, 2.43, 2.46, 2.61, 2.93, 2.94, 2.101, 2.104, 2.108, 2.134, 7.16, 7.17, 8.48, 8.50, 9.43, 9.59, 9.60, 9.61, 9.62, 9.63, 9.67, 9.68, 9.73, 9.82, 9.88, 9.93, 9.151, 9.153
number of discarded pels 7.1, 7.5, 7.7, 7.9, 7.10, 9.143
number of entries 4.87
number of lines per tile 7.1, 7.6, 7.7, 7.9, 7.10, 9.145
number of lines 7.6, 7.7, 7.9, 7.10, 9.132
number of objects per page 2.121, 4.1, 4.2
number of pages 2.121, 4.1, 4.2
number of pels per line 7.1, 7.5, 7.7, 7.9, 7.10, 9.142
number of pels per tile line 7.1, 7.6, 7.7, 7.9, 7.10, 9.146
NUMBER_OF_DISCARDED_PELS_VALUE 9.143
NUMBER_OF_LINES_PER_TILE_VALUE 9.145
NUMBER_OF_PELS_PER_LINE_VALUE 9.132, 9.142, 9.143
NUMBER_OF_PELS_PER_TILE_LINE_VALUE 9.146
nx 8.35
ny 8.35

object class identifier 2.8, 2.9, 2.20, 2.21, 2.22, 2.24, 2.25, 2.26, 2.27, 2.28, 2.29, 2.61, 2.135, 2.136, 2.138, 2.139, 2.144, 2.150, 9.11, 9.15, 9.18, 9.20
object class 2.37, 2.38, 2.39, 2.41, 2.42, 2.43, 2.44, 2.46, 2.47, 2.61, 2.135, 2.136, 2.156, 9.10, 9.11, 9.12, 9.13, 9.14, 9.17, 9.19, 9.22, 9.29, 9.30, 9.31, 9.32, 9.35, 9.39, 9.40
object colour table 2.27, 2.28, 2.29, 2.43, 2.44, 2.46, 2.47, 2.56, 2.61, 2.90, 4.10, 9.99
object colour tables 2.121, 4.1, 4.2
object identifier 2.8, 2.37, 2.38, 2.39, 2.41, 2.42, 2.43, 2.44, 2.46, 2.47, 2.61, 2.138, 2.140, 2.143, 2.144, 2.152, 2.154, 2.156
object type page 2.104, 2.108, 2.134
object type 2.20, 2.21, 2.22, 2.24, 2.25, 2.26, 2.27, 2.28, 2.29, 2.37, 2.38, 2.39, 2.41, 2.42, 2.43, 2.44, 2.46, 2.47, 2.61, 2.135, 9.74
OBJECT_CLASSID_INRES 9.14, 9.16, 9.17, 9.19, 9.25, 9.31, 9.32, 9.38, 9.40
OBJECT_COLOUR_TABLE_VALUE 9.99
OBJECT_TYPE_VALUE 9.45, 9.46, 9.47, 9.48, 9.49, 9.51, 9.52, 9.53, 9.55, 9.57, 9.59, 9.60, 9.61, 9.62, 9.63, 9.64, 9.67, 9.68, 9.70, 9.71, 9.72, 9.73, 9.74, 9.76, 9.77, 9.78, 9.79, 9.80, 9.82, 9.85, 9.86, 9.87, 9.88, 9.90, 9.91, 9.92, 9.93, 9.94, 9.98, 9.99, 9.100, 9.101, 9.102, 9.103, 9.104, 9.105, 9.106, 9.107, 9.109, 9.111, 9.112, 9.114, 9.115, 9.116, 9.117, 9.118, 9.119, 9.120, 9.121, 9.122, 9.123, 9.128, 9.129, 9.130, 9.131, 9.132, 9.133, 9.134, 9.135, 9.137, 9.138, 9.139, 9.140, 9.151, 9.152, 9.153, 9.154, 9.155, 9.162, 9.165, 9.169, 9.173, 9.176, 9.177, 9.179
OccursIn 2.139, 2.142, 2.147
ODA label data 4.57
ODA label text 4.57
ODA security label 2.121, 4.1, 4.2
ODA version 2.121, 4.1, 4.2
off 8.39, 8.46, 8.47, 9.173
offset 2.55, 2.61, 2.91, 9.76, 9.77, 9.78, 9.79
OFFSET_VALUE 9.75
on 8.39, 8.46, 8.47, 9.177, 9.179
opaque 2.96

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

opt 2.86
optrep 2.86
ORDINAL 2.67, 2.68
ORELSE 9.1, 9.2, 9.3, 9.4, 9.5, 9.6, 9.7, 9.11, 9.12, 9.13, 9.15, 9.17, 9.19, 9.21, 9.22, 9.23, 9.27, 9.28, 9.30, 9.32, 9.33, 9.35, 9.36, 9.41, 9.42, 9.43, 9.44, 9.45, 9.46, 9.47, 9.48, 9.49, 9.51, 9.52, 9.57, 9.59, 9.60, 9.61, 9.62, 9.63, 9.64, 9.65, 9.66, 9.67, 9.68, 9.70, 9.71, 9.72, 9.73, 9.76, 9.77, 9.78, 9.79, 9.80, 9.82, 9.85, 9.86, 9.87, 9.88, 9.90, 9.91, 9.92, 9.93, 9.94, 9.95, 9.96, 9.97, 9.98, 9.99, 9.100, 9.101, 9.102, 9.103, 9.104, 9.105, 9.106, 9.107, 9.109, 9.111, 9.112, 9.114, 9.115, 9.116, 9.117, 9.118, 9.119, 9.120, 9.121, 9.122, 9.123, 9.128, 9.129, 9.130, 9.131, 9.132, 9.133, 9.134, 9.135, 9.137, 9.138, 9.139, 9.140, 9.141, 9.142, 9.143, 9.145, 9.146, 9.147, 9.148, 9.149, 9.150, 9.151, 9.152, 9.153, 9.154, 9.155, 9.162, 9.165, 9.169, 9.173, 9.176, 9.177, 9.179
organization 4.33
organizations 2.121, 4.1, 4.2
orphan size 6.4, 6.12, 9.129
ORPHAN_SIZE_VALUE 9.129
owners organization 4.35
owners 2.121, 4.1, 4.2

page dimensions 2.121, 4.1, 4.2
page position 2.26, 2.27, 2.43, 2.44, 2.61, 2.90, 4.10, 9.80
page positions 2.121, 4.1, 4.2
page set 2.25, 2.42, 2.76
PAGE_POSITION_VALUE 9.80
pairwise kerning 6.12, 9.121
PAIRWISE_KERNING_VALUE 9.121
partial generator set 2.7, 2.62, 4.1, 4.2
pattern index asf 8.31, 9.169
Pattern Index 8.31, 8.33, 9.169
Pattern Size 8.31, 9.169
pattern table index 8.35
pattern table specifications 8.31, 9.169
pattern 8.31, 8.33
pel path 7.1, 7.8, 9.134
pel spaces 7.17, 9.139
pel spacing value 7.18, 9.140
pel spacing 7.1, 7.8, 9.139
pel transmission density 7.1, 7.8, 9.137
pel 7.32
PEL_PATH_VALUE 9.134, 9.136
PEL_SPACING_VALUE 9.139
PEL_TRANSMISSION_DENSITY_VALUE 9.137, 9.142
permitted categories 2.8, 2.28, 2.46, 2.61, 2.90, 9.82
PERMITTED_CATEGORIES_VALUE 9.82
personal name of author 4.36
personal name of owner 4.35
personal name of preparer 4.34
personal name of recipient 4.41
PF 1.11, 1.13, 1.15
picture dimensions 8.6, 9.153
picture orientation 8.6, 9.152
PICTURE_DIMENSIONS_VALUE 9.153
PICTURE_ORIENTATION_VALUE 9.152
placeholder for any attribute value 1.19
plane 7.32, 9.150
PLD 6.70, 6.71, 6.72, 6.77, 6.78, 6.97
PLU 6.70, 6.71, 6.72, 6.77, 6.79, 6.97
position 2.28, 2.29, 2.46, 2.47, 2.61, 2.90, 4.33, 9.85, 9.86

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

POSITION_VALUE 9.83
post-enciphered body parts 2.121, 4.1, 4.2
post-enciphered document body parts 2.121, 4.1, 4.2
post-sealed document body parts 2.121, 4.1, 4.2
PR 1.12, 1.14, 2.154
pre-enciphered body parts 2.121, 4.1, 4.2
pre-enciphered document body parts 2.121, 4.1, 4.2
pre-sealed document body parts 2.121, 4.1, 4.2
precedence number 4.29
PRECEDING-OBJECT 2.69, 2.70
PRECEDING 2.74
preferred height 7.16, 8.50
preferred width 7.16, 8.50
preparers organization 4.34
preparers 2.121, 4.1, 4.2
present 2.62, 4.2
presentation attributes 2.27, 2.29, 2.44, 2.47, 2.56, 2.61, 2.90
presentation features 2.121, 4.1, 4.2
presentation style identifier 2.8, 2.17, 2.56, 2.61, 9.7, 9.9, 9.18
presentation style 2.8, 2.22, 2.27, 2.29, 2.39, 2.44, 2.47, 2.61, 2.90, 9.4, 9.7, 9.12, 9.17, 9.18, 9.24, 9.37
presentation styles 2.5, 2.6, 2.7, 2.62, 2.121, 4.1, 4.2
primary font 6.16
primary 2.39, 2.44, 2.47, 2.61
privileged recipient information 4.50, 4.58, 4.60
privileged recipients 4.51, 4.62, 4.69
processable 2.4, 2.5, 2.62, 4.1, 4.2
profile character sets 2.121, 4.1, 4.2
properties 4.29
proportional line spacing 6.4, 6.12, 9.130
PROPORTIONAL_LINE_SPACING_VALUE 9.130
protected document part identifier 4.50, 4.58, 4.60
protected part identifier 2.57, 2.58, 2.59, 2.60, 2.61, 2.119
protected 2.99
protection 2.20, 2.21, 2.22, 2.37, 2.38, 2.39, 2.61, 2.90, 9.87
protections 2.121, 4.1, 4.2
PROTECTION_VALUE 9.87
PTX 6.82, 6.83, 6.84, 6.106, 6.107
purge date and time 2.121, 4.1, 4.2

R value 4.89
raster-graphics coding attributes 4.24
raster-graphics presentation features 4.25
Real Precision 8.9, 9.154
real 8.9
recipients organization 4.41
rectangle 8.48
recto 2.98
reference properties 4.27
reference white 4.83, 4.84, 4.85
references to other documents 2.121, 4.1, 4.2
REF_LAY_STYLE 9.2, 9.6, 9.13, 9.24, 9.27, 9.30, 9.37
REF_OBJECT_CLASS 9.10, 9.11, 9.12, 9.13, 9.14, 9.17, 9.19, 9.22, 9.29, 9.30, 9.31, 9.32, 9.35, 9.39, 9.40
REF_PRES_STYLE 9.4, 9.7, 9.12, 9.24, 9.37
region of interest specification 8.6, 9.151
REGION_OF_INTEREST_SPECIFICATION_VALUE 9.151
release date and time 2.121, 4.1, 4.2

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

rep 2.86
req 2.86
RES_DEFAULT_PAR_VALUE 9.35, 9.38
RES_DEFAULT_VALUE 9.22, 9.25
resource document 2.4, 2.121, 4.1, 4.2
resource 2.4, 2.20, 2.21, 2.22, 2.24, 2.25, 2.26, 2.27, 2.28, 2.29, 2.61, 9.14, 9.16, 9.17, 9.19, 9.22, 9.31, 9.32, 9.35, 9.40
RESOURCE_LAY_STYLE 9.19, 9.20, 9.32
RESOURCE_OBJECT_CLASS 9.14, 9.15, 9.17, 9.19, 9.25, 9.31, 9.32, 9.38, 9.40
RESOURCE_PRES_STYLE 9.17, 9.18
resources 2.121, 4.1, 4.2, 9.16
reverse order 2.91, 2.103
revisers 4.33
revision date and time 4.33
revision history 2.121, 4.1, 4.2
RGB 4.79, 4.82
RGCA_INITIAL_OFFSET_DEFAULT_VALUE 9.135, 9.136
RGCA_INITIAL_OFFSET_VALUE 9.135
RGCA_LINE_PROGRESSION_VALUE 9.133, 9.136
right-hand aligned 2.91, 2.101, 9.45
right-hand offset 2.91, 2.107, 9.75, 9.79
right-hand-edge 2.93
right 8.22, 8.30, 9.165
RIGHTHAND_OFFSET_VALUE 9.75, 9.79
rule A 2.26, 2.27, 2.29, 2.43, 2.44, 2.46, 2.47, 2.92
rule B 2.26, 2.27, 2.29, 2.43, 2.44, 2.46, 2.47, 2.92

SACS 6.102, 6.103
same layout object 2.8, 2.55, 2.61, 9.88
SAME_LAYOUTOBJECT_VALUE 9.88
SatisfiesCharacterContentArchitectureConstraints 2.2, 2.5, 2.6, 2.7, 6.4
SatisfiesGeometricGraphicsContentArchitectureConstraints 2.2, 2.5, 2.6, 2.7, 8.4
SatisfiesRasterGraphicsContentArchitectureConstraints 2.2, 2.5, 2.6, 2.7, 7.4
scaled 8.16, 9.155, 9.157, 9.162, 9.163, 9.173, 9.174
SCS 6.98, 6.99
seal identifier 4.62
seal identifiers 2.120
seal method 4.63
seal originator 4.67
seal 4.63
sealed constituents 4.62
sealed document profile identifier 4.69
sealed document profile information 2.57, 2.61
sealed document profiles 2.121, 4.1, 4.2
sealed information 4.63
sealed profiles 2.121, 4.1, 4.2
sealed status 2.120
sealed 2.20, 2.21, 2.22, 2.24, 2.25, 2.26, 2.27, 2.28, 2.29, 2.37, 2.38, 2.39, 2.41, 2.42, 2.43, 2.44, 2.46, 2.47, 2.55, 2.56, 2.61, 2.90
sealing information 4.66
sealing method 4.66
second alternative font 6.16
second component 4.93
security classification 2.121, 4.1, 4.2
separation 2.55, 2.61, 2.91, 9.90, 9.91, 9.92
SEPARATION_VALUE 9.89

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

SEQCLASSIDIMSUB 2.136, 2.156, 2.156
sequence 2.85
seventh alternative font 6.16
SGR 6.98, 6.99
SHS 6.98, 6.99
side of sheet 2.98, 9.69, 9.71
SIDE_OF_SHEET_VALUE 9.69, 9.71
sixth alternative font 6.16
SLS 6.98, 6.99
solid 2.93, 8.31, 8.33, 9.46
SOS 6.85, 6.86, 6.87, 6.108
SP 6.109
spacing ratio 7.1, 7.8, 9.140
SPACING_RATIO_VALUE 9.140
specific layout structure 2.5, 2.6, 2.7, 2.62, 2.121, 4.1, 4.2
specific logical structure 2.5, 2.6, 2.7, 2.62, 2.121, 4.1, 4.2
specified tolerance 2.130
SRCS 6.102, 6.103
SRS 6.73, 6.74, 6.75, 6.76, 6.77, 6.78, 6.79, 6.89, 6.90, 6.98, 6.99
SSW 6.102, 6.103
ST 6.86, 6.87, 6.108
STAB 6.98, 6.99
start date and time 2.121, 4.1, 4.2
start edge offset 6.36, 9.117
start-aligned 6.15, 6.42, 6.46, 9.103
starting index 8.45
status 2.121, 4.1, 4.2
STEP_BL_PAR_VALUE 9.27, 9.76, 9.77, 9.78, 9.79, 9.90, 9.91, 9.92
STEP_BL_VALUE 9.2, 9.45, 9.48, 9.57, 9.59, 9.60, 9.61, 9.62, 9.63, 9.67, 9.68, 9.73, 9.88, 9.93
STEP_BP_VALUE 9.4, 9.46, 9.47, 9.94, 9.98, 9.99, 9.100, 9.101, 9.102, 9.103, 9.104, 9.105, 9.106, 9.107, 9.109, 9.111, 9.112, 9.114, 9.115, 9.116, 9.117, 9.118, 9.119, 9.120, 9.121, 9.122, 9.123, 9.128, 9.129, 9.130, 9.131, 9.132, 9.133, 9.134, 9.135, 9.137, 9.138, 9.139, 9.140, 9.151, 9.152, 9.153, 9.154, 9.155, 9.162, 9.165, 9.169, 9.173, 9.176, 9.177, 9.179
STEP_C_PAR_VALUE 9.29, 9.51, 9.52, 9.70, 9.71, 9.72
STEP_C_SUBPAR_VALUE 9.39, 9.85, 9.86
STEP_C_VALUE 9.10, 9.42, 9.43, 9.44, 9.46, 9.47, 9.49, 9.64, 9.65, 9.66, 9.74, 9.80, 9.82, 9.87, 9.94, 9.95, 9.96, 9.97, 9.98, 9.99, 9.100, 9.101, 9.102, 9.103, 9.104, 9.105, 9.106, 9.107, 9.109, 9.111, 9.112, 9.114, 9.115, 9.116, 9.117, 9.118, 9.119, 9.120, 9.121, 9.122, 9.123, 9.128, 9.129, 9.130, 9.131, 9.132, 9.133, 9.134, 9.135, 9.137, 9.138, 9.139, 9.140, 9.141, 9.142, 9.143, 9.145, 9.146, 9.147, 9.148, 9.149, 9.150, 9.151, 9.152, 9.153, 9.154, 9.155, 9.162, 9.165, 9.169, 9.173, 9.176, 9.177, 9.179
STEP_DL_PAR_VALUE 9.30, 9.76, 9.77, 9.78, 9.79, 9.90, 9.91, 9.92
STEP_DL_VALUE 9.13, 9.45, 9.48, 9.57, 9.59, 9.60, 9.61, 9.62, 9.63, 9.67, 9.68, 9.73, 9.88, 9.93
STEP_DP_VALUE 9.12, 9.46, 9.47, 9.94, 9.98, 9.99, 9.100, 9.101, 9.102, 9.103, 9.104, 9.105, 9.106, 9.107, 9.109, 9.111, 9.112, 9.114, 9.115, 9.116, 9.117, 9.118, 9.119, 9.120, 9.121, 9.122, 9.123, 9.128, 9.129, 9.130, 9.131, 9.132, 9.133, 9.134, 9.135, 9.137, 9.138, 9.139, 9.140, 9.151, 9.152, 9.153, 9.154, 9.155, 9.162, 9.165, 9.169, 9.173, 9.176, 9.177, 9.179
STEP_E_PAR_VALUE 9.31, 9.51, 9.52, 9.70, 9.71, 9.72
STEP_E_SUBPAR_VALUE 9.40, 9.85, 9.86
STEP_E_VALUE 9.14, 9.42, 9.43, 9.44, 9.46, 9.47, 9.49, 9.64, 9.65, 9.66, 9.80, 9.82, 9.87, 9.94, 9.96, 9.97, 9.98, 9.99, 9.100, 9.101, 9.102, 9.103, 9.104, 9.105, 9.106, 9.107, 9.109, 9.111, 9.112, 9.114, 9.115, 9.116, 9.117, 9.118, 9.119, 9.120, 9.121, 9.122, 9.123, 9.128, 9.129, 9.130, 9.131, 9.132, 9.133, 9.134, 9.135, 9.137, 9.138, 9.139, 9.140, 9.151, 9.152, 9.153, 9.154, 9.155, 9.162, 9.165, 9.169, 9.173, 9.176, 9.177, 9.179
STEP_FL_PAR_VALUE 9.32, 9.76, 9.77, 9.78, 9.79, 9.90, 9.91, 9.92
STEP_FL_VALUE 9.19, 9.45, 9.48, 9.57, 9.59, 9.60, 9.61, 9.62, 9.63, 9.67, 9.68, 9.73, 9.88, 9.93
STEP_FP_VALUE 9.17, 9.46, 9.47, 9.94, 9.98, 9.99, 9.100, 9.101, 9.102, 9.103, 9.104, 9.105, 9.106, 9.107, 9.109, 9.111, 9.112, 9.114, 9.115, 9.116, 9.117, 9.118, 9.119, 9.120, 9.121, 9.122, 9.123, 9.128, 9.129, 9.130, 9.131, 9.132,

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9.133, 9.134, 9.135, 9.137, 9.138, 9.139, 9.140, 9.151, 9.152, 9.153, 9.154, 9.155, 9.162, 9.165, 9.169, 9.173, 9.176, 9.177, 9.179
STEP_G_PAR_VALUE 9.33, 9.33, 9.51, 9.52, 9.70, 9.71, 9.72, 9.76, 9.77, 9.78, 9.79, 9.90, 9.91, 9.92
STEP_G_SUBPAR_VALUE 9.41, 9.41, 9.85, 9.86
STEP_G_VALUE 9.21, 9.21, 9.45, 9.46, 9.47, 9.48, 9.49, 9.57, 9.59, 9.60, 9.61, 9.62, 9.63, 9.64, 9.73, 9.80, 9.82, 9.87, 9.88, 9.93, 9.94, 9.98, 9.99, 9.100, 9.101, 9.102, 9.103, 9.104, 9.105, 9.106, 9.107, 9.109, 9.111, 9.112, 9.114, 9.115, 9.116, 9.117, 9.118, 9.119, 9.120, 9.121, 9.122, 9.123, 9.128, 9.129, 9.130, 9.131, 9.132, 9.133, 9.134, 9.135, 9.137, 9.138, 9.139, 9.140, 9.151, 9.152, 9.153, 9.154, 9.155, 9.162, 9.165, 9.169, 9.173, 9.176, 9.177, 9.179
STEP_H_PAR_VALUE 9.34, 9.51, 9.52, 9.70, 9.71, 9.72
STEP_H_VALUE 9.26, 9.45, 9.46, 9.47, 9.49, 9.64, 9.80, 9.94, 9.95, 9.98, 9.99, 9.100, 9.101, 9.102, 9.103, 9.104, 9.105, 9.106, 9.107, 9.109, 9.111, 9.112, 9.114, 9.115, 9.116, 9.117, 9.118, 9.119, 9.120, 9.121, 9.122, 9.123, 9.128, 9.129, 9.130, 9.131, 9.132, 9.133, 9.134, 9.135, 9.137, 9.138, 9.139, 9.140, 9.141, 9.142, 9.145, 9.146, 9.147, 9.148, 9.149, 9.150, 9.151, 9.152, 9.153, 9.154, 9.155, 9.162, 9.165, 9.169, 9.173, 9.176, 9.177, 9.179
string 8.22, 8.27, 9.165
stroke 8.22, 8.27
SUB 6.97
subject 2.121, 4.1, 4.2
subordinates 2.8, 2.37, 2.38, 2.41, 2.42, 2.43, 2.46, 2.61, 2.136, 2.140, 2.143, 2.152, 2.154, 2.156
SUPERIOR-OBJECT 2.69, 2.70
SUPERIOR 2.74
SUPERIOR_PAR_VALUE 9.33, 9.35, 9.41
SUPERIOR_VALUE 9.21, 9.22
superseded documents 2.121, 4.1, 4.2
SUP_LAY_STYLE 9.3, 9.8, 9.28
SUP_LAY_STYLE_PAR_VALUE 9.27, 9.28, 9.28, 9.30, 9.32
SUP_LAY_STYLE_VALUE 9.2, 9.3, 9.3, 9.13, 9.19
SUP_PRES_STYLE 9.5, 9.9
SUP_PRES_STYLE_VALUE 9.4, 9.5, 9.5, 9.12, 9.17
surname 4.47
SVS 6.98, 6.99
synchronization 2.55, 2.61, 9.93
SYNCHRONIZATION_VALUE 9.93

T4 one dimensional encoded 7.30
T4 two dimensional encoded 7.30
T6 encoded 7.30, 9.148
tab position 6.42
tab reference 6.42
TAIL 1.15, 2.156, 6.50, 6.51, 6.52, 6.64, 6.65, 6.66, 6.67, 6.68, 6.69, 6.70, 6.71, 6.72, 6.73, 6.74, 6.75, 6.76, 6.77, 6.78, 6.79, 6.81, 6.82, 6.83, 6.84, 6.85, 6.86, 6.87, 6.89, 6.90, 9.108
Text Alignment 8.22, 9.165
text aspect source flags 8.22, 9.165
Text Bundle Index 8.22, 8.27, 9.165
text bundle representation 8.27
text bundle specifications 8.22, 9.165
text colour asf 8.22, 9.165
Text Colour 8.22, 8.27, 9.165
text font asf 8.22, 9.165
Text Font Index 8.22, 8.27, 9.165
Text Path 8.22, 9.165
text precision asf 8.22, 9.165
Text Precision 8.22, 8.27, 9.165
text rendition 8.6, 9.165
TEXT_RENDITION_VALUE 9.165
third alternative font 6.16
third component 4.93

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

tile types 7.1, 7.6, 7.7, 7.9, 7.10, 9.148
TILE_TYPES_VALUE 9.148
tiling offset 7.1, 7.6, 7.7, 7.9, 7.10, 9.147
TILING_OFFSET_VALUE 9.147
time 4.67
title 2.121, 4.1, 4.2, 4.47
tolerance space 2.130
tolerance value 2.130
top 8.30
trailing offset 2.91, 2.107, 9.75, 9.77
trailing-edge 2.91, 2.93, 2.109, 9.89, 9.91
TRAILING_EDGE_VALUE 9.89, 9.91
TRAILING_OFFSET_VALUE 9.75, 9.77
transformation specification 8.6, 9.179
TRANSFORMATION_SPECIFICATION_VALUE 9.159, 9.170, 9.171, 9.172, 9.179
transparencies 2.121, 4.1, 4.2
transparency specification 8.6, 9.177
transparency 2.26, 2.27, 2.28, 2.29, 2.43, 2.44, 2.46, 2.47, 2.56, 2.61, 2.90, 4.10, 9.94
Transparency 8.46, 9.177
TRANSPARENCY_SPECIFICATION_VALUE 9.177
TRANSPARENCY_VALUE 9.94
transparent 2.96, 9.94
type of coding 2.54, 2.61, 4.10, 6.13, 6.14, 7.1, 7.7, 7.9, 7.10, 8.7, 9.95
TYPE_OF_CODING_VALUE 9.95
types of coding 2.121, 4.1, 4.2

uncompressed 7.21
UNDEF 1.11, 1.12, 1.13, 1.14, 1.15, 1.16, 1.17, 1.18, 2.154, 2.155, 2.156, 9.1, 9.2, 9.3, 9.4, 9.5, 9.10, 9.12, 9.13, 9.14, 9.17, 9.19, 9.22, 9.27, 9.28, 9.29, 9.30, 9.31, 9.32, 9.35, 9.39, 9.40, 9.51, 9.52, 9.182, 9.183
unit scaling 2.121, 4.1, 4.2, 9.108
unprotected 2.99, 9.87
unspecified tolerance 2.130, 9.98, 9.99, 9.102
unspecified 2.98, 9.71, 9.72
up 8.22
UPPER-ALPHA 2.66
UPPER-ROMAN 2.66
user comments 4.33, 4.44
user-readable comments 2.20, 2.21, 2.22, 2.24, 2.25, 2.26, 2.27, 2.28, 2.29, 2.37, 2.38, 2.39, 2.41, 2.42, 2.43, 2.44, 2.46, 2.47, 2.55, 2.56, 2.61, 4.27, 4.29, 9.96
user-specific codes 2.121, 4.1, 4.2
user-visible name 2.20, 2.21, 2.22, 2.24, 2.25, 2.26, 2.27, 2.28, 2.29, 2.37, 2.38, 2.39, 2.41, 2.42, 2.43, 2.44, 2.46, 2.47, 2.55, 2.56, 2.61, 4.27, 9.97
USER_READABLE_COMMENTS_VALUE 9.96
USER_VISIBLE_NAME_VALUE 9.97

value array 8.35
variable page height 2.26, 2.28, 2.29, 2.43, 2.46, 2.47, 2.92
variable position 2.29, 2.46, 2.47, 2.91
variable 7.16, 8.50
VDC Extent 8.47, 9.159, 9.170, 9.171, 9.172, 9.179
VDC Integer Precision 8.9, 9.154
VDC Real Precision 8.9, 9.154
VDC Type 8.9, 9.154
version number 4.33
version reference 4.33
verso 2.98
vertical alignment 8.30, 9.165

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

vertical coordinate 6.35, 7.14, 9.124, 9.136
vertical dimension 2.92, 9.50, 9.52, 9.126
vertical position 2.91, 9.84, 9.86
VERTICAL_DIMENSION 9.124, 9.126, 9.136
VERTICAL_DIMENSION_VALUE 9.50, 9.52
VERTICAL_FIXED_DIMENSION_VALUE 9.52, 9.55
VERTICAL_POSITION_VALUE 9.84, 9.86
VPB 6.81, 6.90, 6.98, 6.99
VPR 6.81, 6.90, 6.98, 6.99

widow size 6.4, 6.12, 9.131
WIDOW_SIZE_VALUE 9.131
width controlled 7.16, 8.50
width vector x component 8.38, 9.169
width vector y component 8.38, 9.169
WIDTH 9.159, 9.160, 9.172
WIDTH_OF_VDC_EXTENT 9.169, 9.172

X value 4.96
Xn value 4.86

Y value 4.95, 4.96
yes 2.120, 6.18, 6.39, 6.41
Yn value 4.86

Z value 4.96
Zn value 4.86

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Annex A

(informative)

Tutorial on the specification language

A.1 Introduction

This informative annex gives an introduction to the formal description technique which is used for the formal specifications of ISO/IEC 8613. The examples are closely related to ISO/IEC 8613. The description of the specification language (IMCL) is given in clause 4 and is not repeated here in full detail.

ISO/IEC 8613 describes document structures, the document profile and the content architectures in terms of abstract information constructs which are drawn from the following structural categories:

— An ODA construct may be an **atomic** construct, e.g. an attribute name or a natural number within an object identifier.

— An ODA construct may be a **composite** construct, i.e. may consist of other constructs. With respect to their interrelationship, three kinds of composition are distinguished. An ODA construct may be

a) a **set**;
b) a **function** (mapping);
c) a **sequence** (list, string);

of other ODA constructs.

For example, a specific layout description is a set (of constituents), a constituent is a nomination (see below) which is is a function or mapping (from attribute names onto attribute values), and an attribute value of 'subordinates' or of 'object identifier' is a sequence (of atomic natural numbers).

It is these very structures which are captured by the language used for the formal specifications of ISO/IEC 8613. The language used is called IMCL, *Information Modelling by Composition Language*. The semantics of the specification language consist of the following abstract elements:

— the universe which is a non-empty set of **entities** of the following kinds:

a) constructs;
b) spots;
c) spotsets (i.e. sets of spots);
d) the entity UNDEF ("undefined");

— functions from the universe to the universe, that is, operators on entities of the universe;

— predicates in the universe, that is, predicates on entities of the universe.

A **construct** is an information object which is one of the following:

— an atomic construct or **atom**, for short;
— a composite construct or **compound**, for short, which may be:
a) a **collection**, which is an unordered set of component constructs;
b) a **nomination**, which is a function that can be regarded as an unordered set of ordered pairs where each pair consists of a *name* and a *value*;
c) a **catenation**, which is a sequence of component constructs.

The special terminology for composite constructs is to distinguish them from other sets, functions or sequences.

A.2 Atomic constructs

Atomic constructs are either numbers or special atoms with application defined semantics. They are syntactically denoted by character strings, e.g., 314 is a number, 'a' or 'ODA' are special atoms. Character strings which

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

denote special atoms are enclosed by single quotes. Note: 'ODA' is considered atomic; it is *not* a sequence of three characters and cannot be decomposed.

In the formal specifications of ISO/IEC 8613, the following atomic constructs appear:

— numbers (integers and real numbers), for example as values for certain attributes;
— attribute names, parameters names, sub-parameter names, etc., for example, 'protection' or 'object identifier';
— certain non-composite attribute values, parameter values, sub-parameter values, etc., for example 'protected' or 'null';
— certain entities which are referenced but not defined in ISO/IEC 8613, for example a date value which conforms to ISO 8601.

A.3 Composite constructs

For explicitly specifying the composition of composite constructs (collections, catenations and nominations) the following notation is used:

— Collections are denoted by enclosing their components in brackets and separating the elements by a semicolon. For example [314; 'a'; 'ODA'] is a collection with the three (atomic) components 314, 'a' and 'ODA'. An empty collection is denoted by [].
— Catenations are denoted by enclosing their components in brackets and surrounding their elements with an arrow. For example, [→ 'O' → 'D' → 'A' →] is a catenation consisting of the three (atomic) constructs 'O', 'D' and 'A' which could also be written as "ODA". Note the difference between the atom 'ODA' and the catenation "ODA". An empty catenation is denoted by [→].
— Nominations are denoted by enclosing the ordered pairs in brackets and separating the pairs by a semicolon. The *name*-part and the *value*-part of each pair are separated by a colon. For example, ['id': 7; 'std': 'ODA'] is a nomination consisting of two (*name*,*value*)-pairs; the first one has the name 'id' with the value 7 and the second one the name 'std' with the value 'ODA'. An empty nomination is denoted by [:].

It should be noted that composite constructs can be arbitrarily nested using atoms, collections, catenations and nominations. Some examples are given below.

IMCL constructs are sufficient to model *any* valid ODA document and, consequently, all structures in ODA documents. For example, the specific logical structure can be formally described as a collection whose elements are the constituents of the specific logical structure. A constituent, e.g., a basic logical object or a presentation style can be formally described as a nomination whose ordered pairs are the attributes and their respective values. Any attribute value can be formally described as an atom, a collection, catenation or nomination.

The formal notation of a given constituent of type *page*, having the attributes 'object type', 'object identifier' and 'medium type' might be, for instance,

```
['object type': 'composite or basic page'; 'object identifier': [ → 0 → 7 → 8 → ];
 'medium type': ['nominal page size': [ → 200 → 350 → ]; 'side of sheet': 'recto']]
```

(Additional attributes would be necessary for a page according to the specifications of ISO/IEC 8613-2. Since this example is only a generalization, these have been intentionally left out.)

In the above manner, any ODA document could be completely encoded using IMCL, i.e., IMCL can be considered an alternative data format to ODIF for the encoding of ODA documents. The IMCL encoding of an ODA document is both human-readable and machine-processable. Translations from the ODIF encoding to an IMCL encoding are straightforward. An IMCL encoding of an ODA document might be useful if a conformance test for the document is to be performed, using the formal specifications of ISO/IEC 8613 as the basis for conformance testing.

However, the purpose of the formal specifications of ISO/IEC 8613 is primarily not for the formal description of a specific ODA document but rather for the formal description of the general rules which hold for ODA documents, e.g., which value ranges are permitted for attributes, which attributes are mandatory, non-mandatory or defaultable for constituents and which sets of constituents are permitted or required in documents of a given document architecture class.

Therefore, additional formal concepts for expressing these general rules are needed.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

A.4 Spots

In order to be able to address components in constructs of arbitrary compositional structure, the concept of a *spot* is introduced. This concept is an abstract counterpart for the intuitive idea associated with pointing into an information structure at some position and saying "here". However, in general the "here" is not identified uniquely by the component construct as such (e.g. in a word, the same letter may occur several times), but rather by the context in which it appears. To deal with the concept of "here" requires a way to identify contexts.

The concept of a spot allows the distinction to be made between a particular construct and its position within a composite construct of which it is a component. For example, the character string "data" (a catenation) has the component constructs 'd', 'a' and 't'. Whereas 'd' and 't' appear at one spot each, the 'a' appears at two spots, namely at the second and at the fourth position counted from the front end. So, "data" has four component spots, but only three component constructs. If a construct is considered outside any context, it is said to be at its *ownspot*.

Spots are usually identified by selection criteria which are built on operators (see below). However, a selection criterion need not be unique. Thus, it is more natural to deal with sets of spots (*spotsets*) rather than spots. Consequently, there are expressions for singleton spotsets instead of individual spots. (see $\hat{\in}$ and $\hat{}$ below).

A spot can be considered an ordered pair (cxt, g) of its context, cxt, and its construct, g. The construct is called the *component at* the spot. The context is the unique specification of the construct which contains the construct g, but without this construct. For the purpose of this tutorial, the position of g within cxt is indicated by an asterix $(*)$. Note that cxt and $*$ are not entities within IMCL, but are used to illustrate the spot concept.

Examples:

— The spot of the second 'a' in "data" (a catenation) is given by:
("dat*",'a').

— The spot of the 'a' in "gerald" in the collection ["nik";"wolfgang";"gerald"] is given by:
(["nik"; "wolfgang"; "ger*ld"],'a').

— The spot of the author of the book in the nomination ['author': 'richard'; 'title': 'ODA'] is described by:
(['author': *; 'title': 'ODA'], 'richard').

The *ownspot* of a construct is defined by $(*, g)$, where g is the construct of interest. The ownspot considers the construct in its own context only (not that of a construct in which it may reside), but does not mean that the construct has no other context.

Example:

— The ownspot of $[\rightarrow a \rightarrow b \rightarrow c \rightarrow]$ is $(*, [\rightarrow a \rightarrow b \rightarrow c \rightarrow])$.

A *spotset* is an unordered set of spots. It is not syntactically defined within IMCL, except for the empty spotset which is denoted by <>. Spotsets are derived from IMCL constructs using predefined functions (see A.6).

For the purpose of this tutorial, spotsets are described syntactically as lists of ordered pairs separated by semicolons and surrounded by angle brackets.

Example:

— <("dat*", 'a'); ("*ata", 'd')> is a spotset of two spots.

A *singleton spotset* is a spotset of only one element.

Example:

— <("dat*", 'a')> is a singleton spotset.

A.5 Predicates

In IMCL these rules are usually expressed by predicates in first-order predicate logic. From predicate logic, IMCL uses logical connectors *and*, *or*, *xor*, *iff* (if and only if) and *impl* (implies) and the logical quantifiers $\forall$ (for all) and $\exists$ (exists)

Several predicates, yielding either true or false, are already predefined, for example:

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

IsAtom(t) is true, if and only if (iff) t is an atomic construct;
IsInt(t) is true, iff t is an integer;
IsCol(t) is true, iff t is a collection;
IsCat(t) is true, iff t is a catenation;
IsNom(t) is true, iff t is a nomination;
$t_1 = t_2$ is true, iff the entities t_1 and t_2 are identical;
$t_1 \in t_2$ is true, iff t_1 is an element of the collection t_2.

(The complete list of predefined predicates is given in 4.3.)

Based on these predefined predicates (and some additional functions which are described below), additional predicates can be defined for formally expressing the concepts of ISO/IEC 8613. In fact, the formal specifications of ISO/IEC 8613 consist essentially of a comprehensive set of such predicate definitions.

For example, the concept that a basic logical object (as any other constituent in an ODA document) is a set of attributes could be expressed by the following predicate:

$\forall\, cst$
(IsBasicLogicalObject(cst) *iff*
IsNom(cst))

This could be read as: "An entity *cst* is a basic logical object, if and only if it is a nomination." Note that sets of attributes are modelled as nominations in IMCL. (Of course, additional constraints have to be satisfied for a basic logical object according to the specifications in ISO/IEC 8613-2.)

A predicate which yields true if an entity is an ordered pair of integers, could be defined as:

$\forall\, v$
(IsPairOfInt(v) *iff*
$\exists\, l, r$
(v = [$\rightarrow$ l $\rightarrow$ r $\rightarrow$] *and* IsInt(l) *and* IsInt(r)))

This could be read as: "An entity v is a pair of integers, if and only if there exist two entities l and r, v is the catenation with the two components l and r, and both l and r are integers."

A.6 Operators

Several operators (also called functions) are part of IMCL. For numbers the usual arithmetical operators ($+ - * /$) are defined. For collections the set operators $\cup$ (union), $\cap$ (intersection) and $\setminus$ (set difference) identical to those in mathematical set theory are defined. For catenations the concatenation operator // is defined.

Several further operators dealing with names, components and spotsets, are also part of the specification language, such as:

$\hat{}t$ If t denotes a construct, then $\hat{}t$ ("ownspot of t") denotes the singleton spotset containing the ownspot of t.

Example:

— If t is the atom `'gerald'`, then $\hat{}t$ is <(*, `'gerald'`)>.

This operator transforms a construct into a singleton spotset. Usually, it is applied before using further operators, such as N t or C t, which are defined only for spotsets.

$t\bullet$ If t denotes a spotset containing no atom spots (spots with atoms), then $t\bullet$ ("next inwards") denotes the set of all spots which are immediately inward of the spots of the spotset t.

Examples:

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

— If t is a singleton spotset of the ownspot of the catenation [→ 'g' → 'n' → 'w' →], which is <(*, [→ 'g' → 'n' → 'w' →])>, then the value of t• is:
<([→ * → 'n' → 'w' →], 'g'); (['g' → * → 'w' →], 'n'); (['g' → 'n' → * →], 'w')>

This operator expands a spotset of composite components into a spotset of the immediate component spots of the composite component spots. This is most commonly applied to a singleton spotset (as shown by the example above). An example illustrating the more general case is the following:

— If t is the spotset <(*, "can"); (*, "frg"); (*, "uk")> then the value of t• is:
<("*an", 'c'); ("c*n", 'a'); ("ca*", 'n'); ("*rg", 'f'); ("f*g", 'r'); ("fr*", 'g');
("*k", 'u'); ("u*", 'k')>

C t — If t denotes a singleton spotset, then C t denotes the component construct at the spot given by t.

Examples:

— If t is <([→ 'a' → 'b' → * →], 'c')> then C t is 'c'.

— If t is <(['a': 1; 'b': 2; 'c': *], 3)> then C t is 3.

N t — If t denotes a singleton spotset of a spot that is a component of a nomination ("immediately inward" of a nomination is the formal term), then N t denotes the name construct of the component as it is within the nomination.

Examples:

— If t is <(['a': 1; 'b': 2; 'c': *], 3)> then N t is 'c'.

— If t is <([→ ['a': 1; 'b': 2] → ['c': *; 'd': 4] →], 3)> then N t is 'c'.

Note: In the second example, the spot refers to a nomination value for a nomination within a catenation. The value 3 is still immediately inward of a nomination, though.

N t is often used when constraining the values of attributes.

Examples:

— N t = 'abc' *impl* C t = 'xyz'

— N t = 'abc' *impl* IsXYZ(C t)

t•'name' — If t denotes a spotset of a nomination spot then t•'name' returns a spotset which is the set of all immediately inward spots (of the nomination spots) for which the name constructs are 'name'.

Usually, this is used when t is a singleton spotset of a nomination spot, thus, assuming that 'name' is a valid name construct within the nomination, then t•'name' returns a singleton spotset of the value stored under 'name'. In programming languages with record data types (like Pascal or Ada) this is like selecting a field value from a record.

Examples:

— If t is <(*, ['a': 1; 'b': 2; 'c': 3])> then t•'b' is <(['a': 1; 'b': *; 'c': 3], 2)>. Using C t from before, C (t•'b') returns 2.

— As a more general example, if t is <(*, ['a': 1; 'b': 2; 'c': 3]); (*, ['a': 4; 'b': 5; 'c': 6])> then t•'b' is <(['a': 1; 'b': *; 'c': 3], 2); (['a': 4; 'b': *; 'c': 6], 5)>. The operator C t cannot be applied to this result because it can only be used for singleton spotsets.

The notation t•'name' is a syntactic shorthand for t•<xs || N xs = 'name'> which is read as "the union of all xs which are singleton subsets of t• such that N xs = 'name'".

$x \hat{\in} t$ — If t denotes a spotset then $x \hat{\in} t$ is true iff x is a singleton spotset of one of the spots in t.

Examples:

— <(*, 'gk')> $\hat{\in}$ <(*, 'wa'); (*, 'ntl'); (*, 'gk')> is true.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

— <(*, 'rc')> $\hat{\in}$ <(*, 'wa'); (*, 'ntl'); (*, 'gk')> is false.

This predicate is most often used to specify properties of a spotset.

Examples:

— $\forall x \hat{\in} t$ (IsInt(C x))

— $\forall x \hat{\in} t$ (IsNnInt(C x) *and* C $x > 5$)

The details of operators and their semantics are given in 4.4.

Using these operators additional operators can be defined.

Example:

— An operator NAMS (read "name set") which returns all the *name*-parts of the ordered pairs of a nomination, could be defined as follows:

$\forall n$
(NAMS(n) =
IF IsNom(n)
THEN [m | $\exists p \hat{\in}$ ^n• (m = N p)]
ELSE UNDEF)

This could be read as: "If an entity n is a nomination then the operator NAMS returns the collection of those entities m, which are the *name*-parts at the component spots p of the nomination n. Otherwise the value UNDEF is returned."

In studying this definition, assume that n in NAMS(n) is the nomination ['a': 1; 'b': 2; 'c': 3]. The expression [m|$\exists p \hat{\in}$ ^n•(m = N p)] could be read as "the collection of all m such that there exists some singleton spotset p that is a subset ($\hat{\in}$) of the spotset ^n• and m equals N p". Evaluating ^n gives <(*, ['a': 1; 'b': 2; 'c': 3])> and ^n• gives <(['a': *; 'b': 2; 'c': 3],1); (['a': 1; 'b': *; 'c': 3],2); (['a': 1; 'b': 2; 'c': *],3)>. With this value for ^n• the values of p will be <(['a': *; 'b': 2; 'c': 3],1)>, <(['a': 1; 'b': *; 'c': 3],2)> and <(['a': 1; 'b': 2; 'c': *],3)>. The values of m for all values of p are defined by N p which would be 'a', 'b' and 'c', thus the collection of these values would be ['a';'b';'c'].

Example:

— An operator HEAD (read "head component") which returns the first component of a catenation could be defined as:

$\forall cat$
($_0$HEAD(cat) =
IF IsNeCat(cat)
THEN C (^cat•<xs || PF xs = 1>)
ELSE UNDEF$_0$)

This could be read as: "If an entity *cat* is a non-empty catenation the function HEAD applied to this catenation returns that component of the catenation whose position from the front end (PF) is 1, otherwise the function returns UNDEF."

In studying this definition, assume that *cat* in HEAD(*cat*) is the catenation [→ 1 → 2 → 3 →]. The expression (^*cat*•<xs || PF xs = 1>) is read as "the spotset that is the union of all xs which are singleton subsets of ^*cat*• such that the position of xs from the front of its immediately outwards catenation is 1". Since *cat* is a catenation then ^*cat* is <(*, [→ 1 → 2 → 3 →])> and ^*cat*• is
<([→ * → 2 → 3 →],1); ([→ 1 → * → 3 →],2); ([→ 1 → 2 → * →],3)>. The values of xs are <([→ * → 2 → 3 →],1)>, <([→ 1 → * → 3 →],2)> and <([→ 1 → 2 → * →],3)>.

The values of xs for which the component is in the first position from the front of the catenation [→ 1 → 2 → 3 →] is the sole spotset <([→ * → 2 → 3 →],1)>. There will only ever be one result because ^*cat*• is generated from

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

the catenation *cat* and a catenation can have only one first element. The final operator C(...) returns the value 1, i.e., C <([→ * → 2 → 3 →],1)> returns 1.

A.7 Predicates for the formal specifications of ISO/IEC 8613

Based on these concepts the formal specifications of ISO/IEC 8613 consist essentially of the definition of a set of predicates, each predicate usually expressing a technical specification of the natural English parts of the Standard in a rigorous and unambiguous way.

The predicates can be grouped into the following classes:

— predicates describing values or value ranges of attributes;
— predicates describing constituents (the attributes and their values which can or must be specified for a given constituent);
— predicates describing sets of constituents (e.g., the generic logical structure or the specific layout structure);
— predicates describing additional concepts (e.g., that one constituent is immediately subordinate to another constituent).

A.8 Further examples

The following simple examples demonstrate the approach discussed in A.7. Note: Some of the predicates shown below are simplifications of the actual predicates given in the formal specifications since they do not reflect all of the specifications given in ISO/IEC 8613.

In ISO/IEC 8613-2, 9.4.3.5, it is specified that the attribute 'medium type' has the two parameters 'nominal page size' and 'side of sheet'. The value of the parameter 'nominal page size' is a pair of integers and the value of the parameter 'side of sheet' is one of 'recto', 'verso', or 'unspecified'. This could be specified formally as:

$\forall v$
(IsMediumTypeValue(v) *iff*
IsNom(v) *and*
NAMS(v) = ['nominal page size'; 'side of sheet'] *and*
$\forall a \hat{\in} \hat{}v \bullet$
((N a = 'nominal page size' *impl* IsPairOfInt(C a)) *and*
(N a = 'side of sheet' *impl* C $a \in$ ['recto'; 'verso'; 'unspecified'])))

This could be read as follows: "An entity v is a medium type value (a construct which could be used as the value of the attribute medium type), if and only if it is a nomination where the names 'nominal page size' and 'side of sheet' are present." (Note: The value of an attribute which contains parameters is modelled as a nomination.) "For the name 'nominal page size' the *value*-part is a pair of integers. For the name 'side of sheet' the *value*-part is one of 'recto', 'verso' or 'unspecified'."

In ISO/IEC 8613-2, 9.3.1.2, it is specified that the identifier of a layout constituent consists of a sequence of non-negative integers, the first of these integers being 1. This could be specified formally as:

$\forall v$
(IsLayoutObjectId(v) *iff*
IsCat(v) *and*
$\forall a \hat{\in} \hat{}v \bullet$ (IsInt(C a) *and* C $a \geq 0$) *and*
HEAD(v) = 1)

This could be read as follows: "An entity v is a layout object identifier, if and only if it is a catenation where all components of this catenation are integers greater than or equal to 0, and the first component of the catenation is the integer 1."

An (intentionally simplified) description of a constituent of type page could be the following: "A page is a constituent consisting of the attributes 'object type' and 'object identifier' and, optionally, of 'medium type'. The attribute

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

'object type' has the value 'composite or basic page'. The value of the attribute 'object identifier' is an layout object identifier and the value of the attribute 'medium type' is a permitted value for this attribute. This could be specified formally as:

$\forall\, cst$
(IsPageDescription(cst) *iff*
IsNom(cst) *and*
NAMS(cst) $\supseteq$ ['object type'; 'object identifier'] *and*
NAMS(cst) $\subseteq$ ['object type'; 'object identifier'; 'medium type'] *and*
$\forall\, a \in \hat{}cst \bullet$
((N a = 'object type' *impl* C a = 'composite or basic page') *and*
(N a = 'object identifier' *impl* IsLayoutObjectId(C a)) *and*
(N a = 'medium type' *impl* IsMediumTypeValue(C a))))

This could be read as follows: "An entity *cst* is a page description, if and only if it is a nomination whose name set contains at least 'object type' and 'object identifier' and at most 'object type', 'object identifier' and 'medium type'. For the name 'object type' the *value*-part is 'composite or basic page'. For the name 'object identifier' the *value*-part is a layout object identifier. For the name 'medium type' the *value*-part is a medium type value."

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

© ISO/IEC

ICS 35.240.20

Descriptors: data processing, text processing, information interchange, documents, document architecture, logical structure, formats, specifications.

Price based on 361 pages

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS

Not for Resale