
**Information technology — Processing
languages — Document Style Semantics
and Specification Language (DSSSL)**

AMENDMENT 1: Extensions to DSSSL

*Technologies de l'information — Langages de traitement —
Sémantique de présentation de documents et langage de spécifications
(DSSSL)*

AMENDEMENT 1: Extensions de DSSSL



PDF disclaimer

This PDF file may contain embedded typefaces. In accordance with Adobe's licensing policy, this file may be printed or viewed but shall not be edited unless the typefaces which are embedded are licensed to and installed on the computer performing the editing. In downloading this file, parties accept therein the responsibility of not infringing Adobe's licensing policy. The ISO Central Secretariat accepts no liability in this area.

Adobe is a trademark of Adobe Systems Incorporated.

Details of the software products used to create this PDF file can be found in the General Info relative to the file; the PDF-creation parameters were optimized for printing. Every care has been taken to ensure that the file is suitable for use by ISO member bodies. In the unlikely event that a problem relating to it is found, please inform the Central Secretariat at the address given below.

© ISO/IEC 2003

All rights reserved. Unless otherwise specified, 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 either ISO at the address below or ISO's member body in the country of the requester.

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.org
Web www.iso.org

Published in Switzerland

Contents	Page
Foreword.....	iv
Introduction	v
Abstract Application Program Interface to DSSSL Flow Object Tree	1
Formatting Areas	1
Flow Object Classes	2
Annexes	4
Annex B (informative) Types and Symbols.....	5
Annex C (informative) Formal public identifiers and their associated characteristics	9
Annex D (informative) Grove plan and SGML property set	10
Annex E (informative) Flow object classes and characteristics	49
Annex F (informative) Values of Characteristics	93

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

The main task of the joint technical committee is to prepare International Standards. 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.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

Amendment 1 to ISO/IEC 10179:1996 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 34, *Document description and processing languages*.

Introduction

This Amendment specifies additional flow object classes and additional formatting areas, and clarifies some existing definitions in ISO/IEC 10179:1996, as follows.

- a) Additional flow object classes are defined to specify on flow object classes for online display. Additional formatting areas are defined to clarify formatting areas for inline-display area and inter-line attachment area.
- b) Annex B (informative) summarizes and clarifies the existing types and symbols. This annex will be a key to considering future extensions to DSSSL.
- c) Annex C (informative) specifies formal public identifiers and their associated characteristics for various line breaking and line composition methods.
- d) Annex D (informative) summarizes and clarifies the grove plan and SGML property set.
- e) Annex E (informative) summarizes and clarifies the flow object classes and characteristics.
- f) Annex F (informative) summarizes and clarifies the values of characteristics.

Information technology — Processing languages — Document Style Semantics and Specification Language (DSSSL)

AMENDMENT 1: Extensions to DSSSL

Abstract Application Program Interface to DSSSL Flow Object Tree

Add the following subclause after subclause 12.2.

12.2.1 API to Flow Object Tree

A DSSSL processor generates a flow object tree. The flow object tree contains information about the results of applying formatting specifications. DSSSL style editors operate on the flow object tree through an abstract application program interface. This API reports the following information:

- character, glyph and glyph-annotations
- line composition
- paragraph composition
- column-set composition
- page composition
- document composition
- documents' volume composition

The abstract API will include dynamic information relating to the document under construction. DSSSL will define a core interface for page composition. The API architecture is system independent.

Formatting Areas

Add the following subclauses after subclause 12.3.4.

12.3.5 Inline-display areas

DSSSL has the following conceptualized formatting areas:

- Display area
- Inline area
- Attachment area

Display areas may include other display areas as well as inline areas. Inline areas may also include conceptualized display areas known as inline-display areas. DSSSL defines an inline-display area as a "display area included within an inline area".

- display-area := (display-area* | inline-area*)
- inline-area := (inline-display-area* | contents*)
- inline-display-area := (inline-display-area* | contents*)

For examples, DSSSL has the following formatting areas for composition:

- (Main) Display area
 - (Sub) Display area
 - (Subsub) Display area ...
 - Inline area ...
 - Inline-display area ...
 - Inline area ...
 - Inline-display area ...

12.3.6 Inter-line attachment areas

The concept of an attachment creates a link between one of display area or inline area and an attachment area. The attachment area for a display area shall be outside the display area. The attachment area for an inline area shall be within the inter-line area between the current line and the immediately following line, or the border adjacent to the next display area. This amendment, therefore, extends the set of DSSSL formatting areas to the following set:

- Display area
- Inline area
- Inline-display area
- Inter-line area
- Attachment area (for display area)
- Inter-line attachment area

Flow Object Classes

Add the following subclauses after subclause 12.6.28.4.

12.6.28.5 Display-window Flow Object Class

This clause defines an extended-online feature. The display-window flow object class specifies an abstract size for the display frame of an online display. The display-window flow object class may get the top position of current scroll flow object classes as its root class. The display-window flow object class may be a recursive flow object class. It has a single principal port, which accepts any displayed flow objects. This flow object has the following characteristics:

- frame-type: one of the symbols window, dialogue, note, caution, alarm or warning. It specifies the type of window to be displayed. The default value is window.
- frame-size: one of the symbols maximum-size, optimum-size or minimum-size. It specifies the relative size of window to be used for an online display. The default value is optimum-size. This actual values used to represent this characteristic will depend on frame-type characteristics and display devices.

12.6.28.6 Pop-up Flow Object Class

This clause defines an extended-online feature. The pop-up flow object class provides information relating to a pop-up area, or the edge area of the current window frame. It has a single principal port, which accepts any displayed flow objects. This flow object has the following characteristics:

- information-type: one of the symbols anything, warning, error, additional-information, note, origin-of-link, voice-annotation or semantic-annotation. It specifies the type of information to be placed into the pop-up window or the edge area of the current window on online display. The default value is anything.

NOTE This flow object can be used to display any information, including position data relating to the grove structure.

12.6.28.7 Dialogue Flow Object Class

This clause defines an extended-online feature. The dialogue flow object class is used to specify interactive dialogues for online display. The dialogue flow object class shall be placed at the front top of the screen display within a display-window flow object class. Dialogue flow object classes may be treated as an interactive flow object class. It has a single principal port, which accepts any displayed flow objects. This dialogue flow object has the following characteristics:

- dialogue-type: one of the symbols request, acknowledgement, select-objects, select-from-list or interaction. The initial value is acknowledgement.
- dialogue-return-type: one of the symbols yes-no-cancel, yes-no, OK-NO, OK, tokens or phrase. The initial value is OK.

12.6.29 Sound-Voice and Animation Flow Object Classes

This clause defines the sound-voice-and-animation feature. The sound-voice and animation flow object classes is used for sounds, voices and animations stored within an external entity. Flow objects of these classes may be inlined or displayed as an online display. This flow object is atomic.

12.6.29.1 Sound-voice Flow Object Class

This clause defines the sound-voice feature. The sound-voice flow object class is used to specify sound and voice data to be used in conjunction with an online display. The sound-voice flow object class may be an atomic object on the other flow object classes, and it may be recursive on itself, depending on the sound-voice system being used. It has a single principal port, which accepts any flow objects. This flow object has the following characteristics:

- output?: boolean specifying whether the flow object shall be output or not. This characteristics is not inherited. The default value is #t.

Other specifications are system independent.

12.6.29.2 Animation Flow Object Class

This clause defines the animation feature. The animation flow object class is used to specify animation on an online display. Animation flow object class may be an atomic object on other flow object classes. It has a single principal port, which accepts any flow objects. This flow object has the following characteristics:

- output?:boolean specifying whether the flow object shall be output rather than displayed and inlined. This characteristics is not inherited. The default value is #t.

Other specifications are system independent.

Annexes

Add the following annexes after Annex A:

- *Annex B (informative) Types and Symbols*
- *Annex C (informative) Formal public identifiers and their associated characteristics*
- *Annex D (informative) Grove plan and SGML property set*
- *Annex E (informative) Flow object classes and characteristics*
- *Annex F (informative) Values of Characteristics*

Annex B (informative)

Types and Symbols

This annex clarifies the existing data types of specification for DSSSL.

B.1 Expression Language

DSSSL expressions have the following data types:

boolean: logic-type

This type specifies true or false. The value can be defined whether **#t** or **#f**. This type is used in conditional expressions. **#f** means false, **#t** means true, other objects except **#t** and **#f** mean true. (See 8.2.2 True and False, and 8.5.1 Booleans.)

e.g., ((if equal wordflag **#t**) x y)

symbol: symbol-type

The symbol identifies strings for optional semantics. The value is used for specification of DSSSL scripts as parameters. In an expression, the symbol requires a single quotation (') before the string of symbol. (See 8.5.4 Symbols.)

e.g., (font-weight: '**bold**)

keyword: keyword-type

The keyword specifies names of keyword strings. This type is used for specification of a DSSSL script as constant and invariable parameters. In an expression, the keyword requires a single colon (:) after the strings of keywords and some keyword arguments. (See 8.5.5 Keywords.)

e.g., (font-weight: '**bold**)

char: character-type

The char specifies characters and names of characters. In an expression, the char requires a single sharp (#) and backslash (\) before the character or the name of character. (See 8.5.8 Characters.)

e.g.,

#\A: character 'A'

#\space: spacing 'empty character'

#\(: open circle parentheses '('

pair: pair-type

The pair is constructed with two arguments. A list is constructed from a head part and other part of a pair; the head part is called **car**, other part is called **cdr**.

A pair is converted from a list. When a list is converted from a pair, DSSSL processor should generate an empty list as a tail of the list. The empty list is known as ().

e.g., (glyph-subset-table (list (cons *glyph1 glyph2*)))

DSSSL expressions have the following numerical types:

quantity: quantity-type

number: number-type

real: real-number-type

integer: integer-number-type

A quantity-type is represented as the product of a number and the base unit raised to the power of an integer-type. A number-type is the quantity-type without dimension. A real-number-type is a subtype of the number-type. An integer-type is a subtype of the real-number-type and integer-number-type. The quantity-type has both concepts of exactness and inexactness. An inexactness of the quantity-type allows a large size of numerical object to be printed or displayed.

e.g., (even? *tapesize*)

string: string-type

A string-type is similar to a string-type of other programming languages. The string-type is enclosed with double quotation ("").

e.g., (string=? headstr "*Contents*")

procedure: procedure-type

A procedure-type is a name of procedure treated as an object in DSSSL.

e.g., (apply (***format-number*** *stringsizes*))

language: language-type

e.g., (with-language ***french*** (spellcheck-french))

B.2 SDQL

node-list: node-list-type

A node-list is the most basic type of SDQL that is used to specify node lists of a grove. A single node should be specified by node-list with a single node list. In DSSSL transformations, this data type with flag specifies and discriminates the result node-list. (See 10 Standard Document Query Language and 10.1.2 Node Lists.)

e.g., (node-list-first ***firstpage***)

named-node-list: node-list-with-name-type

A named-node-list specifies node lists with name in a grove tree. (See 10 Standard Document Query Language and 10.1.3 Named Node Lists.)

e.g., (named-node *column1* ***page-top***)

B.3 Transformation Language

subgrove-spec: sub-grove-specification-type

A subgrove-spec specifies a sub-grove. All the sub-grove can be specified by specifying the root of the sub-grove.

e.g., (subgrove-spec node | subgrove | class | add | null | remove | children | sublabel sort-children)

create-spec: create-specification-type

e.g.,

(**create-root** *obj sg*)

(**create-sub** *snl sg property label unique*)

(**create-proc** *snl sg label result-path optional unique*)

(**create-follow** *snl sg label result-path optional unique*)

result-node-list: result-node-list-type

e.g., (**select-by-relation** *rnl i proc*)

transform-grove-spec: transform-grove-specification-type

Transform-grove-spec specifies information of a grove transformation.

e.g., (**transform-grove** *snl obj ...*) (**select-grove** *nl obj*)

transliteration-map: transliteration-map-type

e.g., (**define-transliteration-map** *variable transliteration-entry*)

B.4 Style Language

sosofo: sosofo-type

A flow object class of DSSSL generates a specification of sequence of flow object(sosofo) as the result of execution.

e.g., (element p (**make paragraph**))

style: style type

e.g., (make paragraph style: **emphasizing-style**)

generated-object: generated object type

e.g., (asis-indirect-sosofo (**column-number**))

length: length type

e.g., page-height: **15in**

decoration-area: decoration area type

e.g.,

```
(make external-graphic entity-system-id: "sample.gif"
(decoration-area "graphics sample"
placement-point-x: 50
placement-point-y: 250))
```

display-space: display space type

e.g., space-before: (display-space **15pt max: 45pt priority: 1**)

inline-space: inline space type

e.g., escapement-space-before: (inline-space **15pt max:45pt**)

glyph-id: glyph identifier type

e.g., (glyph-subst *gst-eng* **glyph-aacute**)

glyph-subst-table: glyph substitution table type

e.g., (make character glyph-id: *eacute* glyph-subst-table: **gst-eng**)

address: address type

e.g., (make *link* destination: **chapter-2**)

color-space: color space type

(color-space string arg ...)

e.g., (color-space **"ISO/IEC 10179:1996//Color-Space Family::Device RGB"**)

color: color type

(color color-space arg ...)

e.g.,

```
(define *rgb-color-space*
(color-space
"ISO/IEC 10179:1996//Color-Space Family::Device RGB"))
(define *midnight-blue* (color *rgb-color-space* 0.0 0.0 0.5))
...
(make paragraph color: *midnight-blue*)
```

page-model: page-model type

(define-page-model **page-model-name** [[region+ | width | height | filling | decoration]])

e.g., (region [[x | y | width | height | decoration* | filling | header | footer | flow-map?]])

column-set-model: column-set-model type

```
(define-column-set-model variable [[column-subset* | fill-out? | tied-column-subset* | filling-direction? | width?
| height? | decoration*]]) (column-subset [[column+ | flow-map | top-float-space-below? | bottom-float-space-
above? | balance? | justify? | justify-limit? | justify-last-limit? | length-deviation? | length-decrease-order? | align-
lines?]]) (column [[width? | height? | x-origin? | y-origin? | footnote-separator? | header? | footer?]])
```

Annex C

(informative)

Formal public identifiers and their associated characteristics

DSSSL should support to register the formal public identifiers (fpi) for values without description of characteristics. The characteristics are;

- line-breaking-method:
- line-composition-method: .

The characteristics' values should be registered according to ISO/IEC 9070, e.g.,

- "ISO/IEC 10179:1996//LINE COMPOSITION METHOD::KYOTO"
- "ISO/IEC 10179:1996//LINE COMPOSITION METHOD::OXFORD"
- "ISO/IEC 10179:1996//LINE COMPOSITION METHOD::CHICAGO"
- "ISO/IEC 10179:1996//LINE BREAKING METHOD::TRADITIONAL VERTICAL BREAK".

Annex D (informative)

Grove plan and SGML property set

This annex clarifies the relationship among modules of grove plan, and property set. A result tree which was parsed is called a "grove". The grove becomes a source tree for the input of a DSSSL processor for translation and style specification. A property set consists of elements, attributes, and values of information structure of the grove.

NOTE See Annex A "SGML Extended Facilities" in ISO/IEC 10744:1997.

D.1 Grove plan

A grove plan defines the level of pre-processing for information structure given by parsing an SGML instance. The number of modules is 18 in the existing grove plan. A base abstract(baseabs) module is the top of module structure.

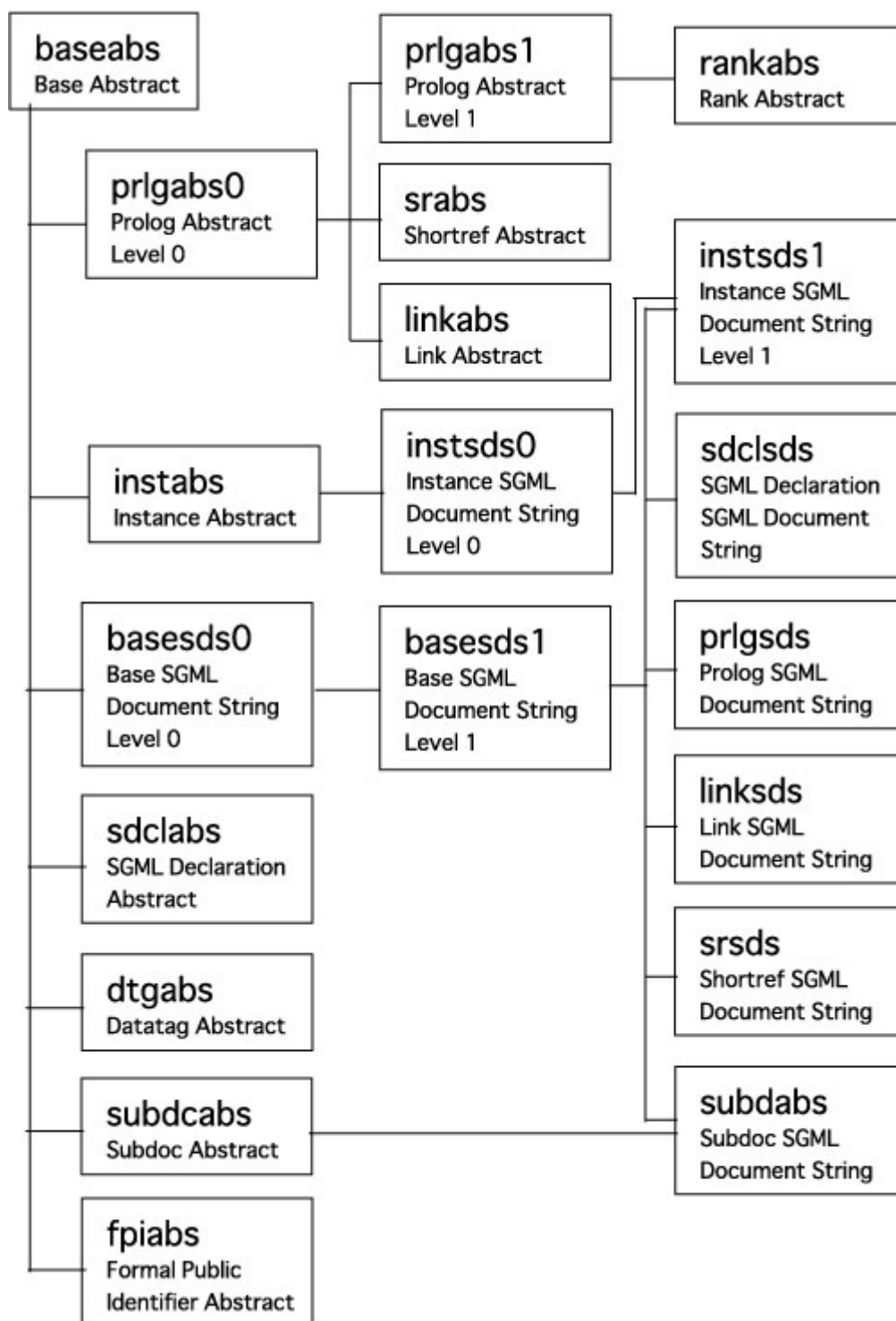


Figure D.1 — Relationship between modules of grove plan

D.2 SGML property set

The property set is an information structure of the grove. It includes elements, structures, and values of the information structure. Table D.1 shows the contents of the property set.

The elements consists of property set modules (psmodule), class definitions (classdef), and property definitions (propdef). Each of elements has some attributes.

Table D.1 — Contents of SGML property set

element			attribute															
	ps module	classdef	profdef	Node related specification	fullnm	appnm	clause	data type	ac	strlex	cn	conprop	dsepprop	strmnorm	sd	dependon	acnmprop	mayadd
1	baseabs	-	-	-	"base abstract"	-	-	-										
2		sgmldoc	-	-	-	"sgml document"	"62001"											
3			sgmlcsts	subnode	-	"sgml constants"	"41170 41180"	node	sgmlcsts									
4			appinfo	-	"application information"	"application info"	"d6001"	string		min data								
5	sgmlcsts	prolog	prolog	subnode	-	-	"71001"	odelist	"doctpdcl lktpdcl comdcl pi ssep"		sgml doc							
6			epilog	subnode	-	-	"71002"	odelist	"comdcl pi ssep"		sgml doc							
7			-	-	-	"sgml constants"	"b6004 c2101"	-										
8		attasgn	-	-	-	-	"attribute assignment"	"79002"	-				value	tokensep				
9	sgmlcsts	value	value	subnode	-	-	"79401"	odelist	"att valtk data chars data intgnch entstart entend"									
10			name	-	-	-	"93001"	string		name					general			
11			implied	-	-	-	-	"b3407"	boolean									

Table D.1 (continued)

element			attribute															
Element No.	ps module	classdef	profdef	Node related specification	fullnm	appnm	clause	data type	ac	strlex	cn	conprop	dsepprop	strmnorm	sd	dependon	acnmprop	mayadd
12			token sep	-	"token separator"	"token sep"	"79400"	char										
13		attvaltk	-	-	-	"attribute value token"	"79305											
14			token	-	-	-	"93003"	string		nm token								
15		datachar	-	-	"data character"	"data char"	"92002"	-				char						
16			char	-	-	character	-	"92003"	char									
17		sdata	-	-	"Internal specific character data entity reference result"	-	"92101"	-				char						
18			sysdata	-	-	-	"system data"	"43041"	string									
19			char	-	-	character	-	-	char							DSSSL		
20		pi	-	-	"processing instruction"	-	"80000"	-										
21			sysdata	-	-	-	"system data"	"80002"	string									
22	prig abs()	-	-	-	"prolog abstract level 0"	-	-	-								baseabs		
23		-	govdt	irefnode	-	-	"governing doctype"	"71004"	node	doctype		sgml doc						

Table D.1 (continued)

	element			attribute															
	ps module	classdef	profdef	Node related specification	fullnm	appnm	clause	data type	ac	strlex	cn	conprop	dsepprop	strmnorm	sd	dependon	acmprop	mayadd	
24		-	dtlts	subnode	"document types and link types"	"doctypes and linktypes"	"71001"	nmnd list	"doctype linktype"		sgml doc						"name name"		
		doctype	-	-	"document type"	"b1000"													
		name	-	-	-	"b1002"	string			name				general					
		governing	-	-	-	governing	"71005"	boolean											
		genents	subnode	-	-	"general entities"	"b1004"	nmnd list	entity								name		
		nots	subnode	-	-	notations	"b1005"	nmnd list	notation								name		
		-	-	-	-	-	"60000"	-											
		name	-	-	-	-	"93001"	string		name					entity				
		enttype	-	-	-	-	"entity type"	enum											
		text	-	-	-	"replacement text"	-	"92101"	string										
		extid	subnode	-	-	"external identifier"	"external id"	"a1601"	node	extid									
		atts	subnode	-	-	-	attributes	"b4120"	nmnd list	attasgn								name	
		notname	-	-	-	-	"notation name"	"79408"	string		name				general				
37			notation	irefnode	-	-	-	"b4001"	node	notation									

Table D.1 (continued)

element			attribute															
Element No.	ps module	classdef	profdef	Node related specification	fullnm	apnrm	clause	data type	ac	strlex	cn	conprop	dsepprop	strmnorm	sd	dependon	acnmprop	mayadd
38		notation	-	-	"data content notation"	-	"b4000"	-										
39			name	-	-	-	"79441"	string		name				general				
40			extid	subnode	"external identifier"	"external id"	"a1601"	node	extid									
41		extid	-	-	"external identifier"	"external id"	"a1600"	-										
42			pubid	-	"public identifier"	"public id"	"a1602"	string		min data								
43			sysid	-	"system identifier"	"system id"	"a1603"	string										
44			gen sysid	optional	"generated system identifier"	"generated system id"	-	string										
45	instabs	-	-	-	"Instance abstract"	-	-	-								baseabs		
46		-	dolelem	subnode	"document element"	-	"72003"	node	element		sgml doc							
47		-	elements	irefnode	-	-	"73001"	nmnd list	element		sgml doc						id	
48		-	entities	irefnode	-	-	"94410"	nmnd list	entity		sgml doc						name	
49		-	entities	irefnode	-	-	"94410"	nmnd list	entity		sgml doc						name	
50		-	dfntents	subnode	-	"defaulted entities"	"94412"	nmnd list	entity		sgml doc						name	
51		-	entity	irefnode	-	-	"79401"	node	entity		att valtk							

Table D.1 (continued)

element			attribute																
Element No.	ps module	classdef	profdef	Node related specification	fullnm	appnm	clause	data type	ac	strlex	cn	conprop	dsepprop	strmnorm	sd	dependon	acnmpprop	mayadd	
52		-	referent	irefnod	-	-	"79403"	node	element		att valk								
53		element	-	-	-	-	"73000"	-				content							
54			gi	-	"generic identifier"	-	"78001"	string		name				general					
55			id	derived	"unique identifier"	-	"79403"	string		name					general				
56			atts	subnode	-	attributes	"79301"	nmnd list	attasgn									name	
57			content	subnode	-	-	"76001"	node list	"data chars data element extdata subdoc pi msignch ignrs ignre repos usemap uselink entstart entend ssep comdcl msstart msend ignmrkup"										
58		extdata	-	-	"reference to external data"	"external data"	"a5500"	-											
59																			entname

Table D.1 (continued)

element			attribute																
Element No.	ps module	classdef	profdef	Node related specification	fullnm	appnm	clause	data type	ac	strlex	cn	conprop	dsepprop	strmnorm	sd	dependon	acnmprop	mayadd	
60			entity	irefnode	-	-	"94410"	node	entity										
61	basesds ()	-	-	-	"base SGML document string level 0"	-	-	-								baseabs			
62		-	entname	optional	-	"entity name"	"a5101"	string		name	sdata			entity					
63		-	entity	irefnode	-	-	"94410"	node	entity		sdata								
64		-	entname	-	-	"entity name"	"a5101"	string		name	pi			entity					
65		-	entity	irefnode	-	-	-	"94410"	node	entity		pi							
66		-		dfited	-	-	defaulted	"94412"	boolean			entity							
67	basesds 1	-	-	-	"base SGML document string level 1"	-	-	-								basesds0			
68		-	entref	subnode optional	"entity reference"	"entity ref"	"94401"	node list	"gendelm name ssep entstart entend refendre shortref"		pi								
69		-	open	subnode optional	"open delimiter"	"open delim"	"80001"	node	gendelm		pi								

Table D.1 (continued)

	element			attribute														
Element No.	ps module	classdef	profdef	Node related specification	fulnm	apnrm	clause	data type	ac	strlex	cn	conprop	dsepprop	strmnrm	sd	dependon	acnmprop	mayadd
70		-	close	subnode optional	"close delimiter"	"close delim"	"80001"	node	gendelm		pi							
71		-	attspec	irefnode	"attribute specification"	"attribute spec"	"79002"	node list	"name ssep gendelm literal attvalue"		att asgn							
72		-	attvalsp	irefnode	"attribute value specification"	"attribute value spec"	"79301"	node	"attvalue literal"		att asgn							
73		-	intrplch		"interpretation replaced character"	"interp replaced char"	"a1704"	char			data char							
74		-	name cref	subnode optional	"named character reference"	"named char ref"	"95001"	node list	"gendelm name refendre"		data char							
75		-	numcref	subnode optional	"numeric character reference"	"numeric char ref"	"95001"	node list	"gendelm name crefnum refendre"		data char							
76		-	markup	subnode optional	-	-	"94401"	node list	"gendelm name ssep entstart entend refendre shortref"		sdata							
77		ssep	-	-	"s separator"	"s sep"	"62100"											+
78			char	-	character	-	"92003"	char										

Table D.1 (continued)

element			attribute																
Element No.	ps module	classdef	profdef	Node related specification	fullnm	apnrm	clause	data type	ac	strlex	cn	conprop	dsepprop	strmnorm	sd	dependon	acnmprop	mayadd	
79			name cref	subnode optional	"named character reference"	"named char ref"	"95001"	node list	"gendelm name refendre"										
			-	-	-	-	"a3002"	-											
80		comment		open	subnode optional	"open delimiter"	"open delim"	"a3002"	node	gendelm									
				chars		characters	-	"g2101"	string										
81			comdcl		close	subnode optional	"close delimiter"	"close delim"	"a3002"	node	gendelm								
					-	-	"comment declaration"	"comment decl"	"a3001"	-			markup						+
82				ignmrkup		markup	subnode	-	-	"a3001"	node list	"comment ssep"							
					-	-	"ignored markup"	"77002 94405 c3007"	-			markup							
83		markup																	
84																			
85																			
86																			
87																			
88																			

Table D.1 (continued)

Element No.	element			attribute										
	ps module	classdef	profdef	Node related specification	fulnm	apnrm	clause	data type	ac	strlex	cn	conprop	dsepprop	strmnrm
89			markup	subnode optional	-	-	-	node list	"gendelm name ssep entstart entend refendre shortref"					
90			entname	optional	-	"entity name"		string		name				entity
91			entity	irefnod	-	-	"a5201"	node	entity					
92		entend	-	-	-	"entity end"	"94500"	-						
93		msignch	-	-	"marked section ignored character"	"marked section ignored char"	"a4204"	-						
94			char	-	character	-	"92101"	char						
95		intgnch	-	-	"interpretation ignored char"	"interp ignored char"	"79303 a1704"	-						
96			name cref	subnode optional	"named character reference"	"named char ref"	"95001"	node list	"gendelm name refendre"					
97			char	-	character	-	"92101"	char						
98		gendelm	-	-	"general delimiter"	"general delim"	"FIG30"	-						
99			name cref	subnode optional	"named character reference"	"named char ref"	"95001"	node list	"gendelm name refendre"					

Table D.1 (continued)

element			attribute																
Element No.	ps module	classdef	profdef	Node related specification	fullnm	appnm	clause	data type	ac	strlex	cn	conprop	dsepprop	strmnorm	sd	dependon	acmprop	mayadd	
100			role	-	-	-	"96001 FIG30"	string							rcsgener				
				optional	"original delimiter"	"original delim"	"92102 FIG22"	string											
			name	-	-	-	"93001"	-											
					optional	-	"original name"	"93005"	string										
			rname	-	-	"reserved name"	"d4701"	-											
						optional	-	"reference name"	"ref name"	"93005"	string				rcsgener				
				orig name	optional	-	"original name"	"93005"	string										
			literal	-	-	-	"a1201 79302 a1701 a1603"	-					value						
					subnode optional	"open delimiter"	"open delim"	"96100 FIG30"	node	gendelm									
				value	subnode	-	-	"a1202 91001 a1702 80002"	node list	"entstart entend datachar sdata intignch"									
				close	subnode optional	"close delimiter"	"close delim"	"96100 FIG30"	node	gendelm									
			number	-	-	-	-	"93002"	-										
				digits	-			"93002"	string		number								

Table D.1 (continued)

Element No.	element			attribute										
	ps module	classdef	profdef	Node related specification	fulnm	apnrm	clause	data type	ac	strlex	cn	conprop	dsepprop	strmnrm
113		crefcnum	-	-	"character reference character number"	"char ref char number"	"95001"	-						
114			ndigits	optional	"number of digits"	"n digits"	"95003 93002"	integer						
115		refendre	-	-	"reference end RE"	"ref end re"	"94502"	-						
116		attvalue	-	-	-	"attribute value"	"79400"	-						
117			value	-	-		"93005"	string						
118		nmtoken	-	-	-	"name token"	"93003"	-						
119			orig name	-	-	"original name token"	"93005"	string						
120		msstart	-	-	"marked section declaration start"	"marked section start"	"a4002"	-				markup		
121			markup	subnode optional	-	-	"a4002"	node list	"gendelm name ssep entstart entend comment ignmrkup"					
122			status	-	-	-	"a4201"	enum						
123		msend	-	-	-	"marked section end"	"a4003"	-				markup		

Table D.1 (continued)

element			attribute															
	ps module	classdef	profdef	Node related speci- fication	fullnm	appnm	clause	data type	ac	strlex	cn	conprop	dsepprop	strmnorm	sd	dependon	acnmprop	mayadd
Element No.																		
124			markup	subnode optional	-	-	"FIG3e FIG3h"	node list	gendelm									
125	sdclabs	-	-	-	"sgml declaration abstract"	-	-	-								baseabs		
126		-	sgmlver	-	-	"sgml version"	"d0002"	string		min data	sgml doc							
127		-	doc chset	subnode	"document character set"	"document char set"	"d1001"	node	charset		sgml doc							
128		-	capset	subnode		"capacity set"	"d2001"	node	capset		sgml doc							
129		-	syn scope		"concrete syntax scope"	"syntax scope"	"d3002"	enum			sgml doc							
130		-	dcisyn	subnode	"declared concrete syntax"	"decl syntax"	"d4001"	node	syntax		sgml doc							
131		-	refsyn	subnode	"reference concrete syntax"	"ref syntax"	"d4002 e0001 FIG70"	node	syntax		sgml doc							
132		-	prosyn	irefnode	"prolog concrete syntax"	"prolog syntax"	"d4001"	node	syntax		sgml doc							
133		-	features	subnode	"feature use"	-	"d5001"	node	features		sgml doc							
134		charset	-	-	"character set"	"char set"	"d1000"	-				chdescs						
135			chdescs	subnode	"character descriptions"	"char descs"	"d1101"	node list	chardesc									

Table D.1 (continued)

element			attribute															
	ps module	classdef	profdef	Node related specification	fullnm	appnm	clause	data type	ac	strlex	cn	conprop	dsepprop	strmnorm	sd	dependon	acnmprop	mayadd
136		chardesc	-	-	"character description"	"char desc"	"d1122"	-										
			desc num	-	"described set character number"	"desc set number"	"d1123"	integer										
			nchars	-	"number of characters"	"n chars"	"d1125"	integer										
137					"base set character number"	"base set number"	"d1124"	integer										
138					"base character set"	"base char set"	"d1111"	string		min data								
139					"description literal"	"desc literal"	"a1701"	string		min data								
140					"concrete syntax"		"d4000"	-										
141					-	shuncha r controls"	"d4204"	boolean										
142					shun char	"shunned character numbers"	"d4201"	intlist										
143					syn chset	"syntax-reference character set"	"d4301"	node	charset									
144					re	"record end"	-	char										
145					rs	"record start"	-	char										

Table D.1 (continued)

element			attribute															
	ps module	classdef	profdef	Node related specification	fullnm	apnrm	clause	data type	ac	strlex	cn	conprop	dsepprop	strmnorm	sd	dependon	acnmprop	mayadd
148			space	-	-	-	"d4401"	char										
			addfuns	subnode	"added function characters"	"added function chars"	"d4401"	nmnd list	addfun								name	
150			lcnmstrt	-	-	-	"d4503"	string										
			ucnmstrt	-	-	-	"d4504"	string										
152			lcnm char	-	-	-	"d4505"	string										
			ucnm char	-	-	-	"d4506"	string										
154			subst gen	-	"substitute general names"	"subst general names"	"d4507"	boolean										
			substnt	-	"substitute entity names"	"subst entity names"	"d4507"	boolean										
156			gdsns	subnode	"general delimiter role associations"	"general delim assoc"	"d4611"	nmnd list	dlimrlas								role	
			srdelms	-	"short reference delimiters"	"shortref delims"	"d4621"	strlist										
158			slitasns	subnode	"syntax literal associations"	"syntax literal assoc"	"d4701"	nmnd list	synlrlas								resname	
	159		attont	-	-	-	"FIG41"	integer										

Table D.1 (continued)

	element			attribute															
Element No.	ps module	classdef	profdef	Node related speci- fication	fullnm	apnrm	clause	data type	ac	strlex	cn	conprop	dsepprop	strmnorm	sd	dependon	acnmprop	mayadd	
160			attisplen	-	-	-	"FIG42"	integer											
161			bseqlen	-	-	-	-	"FIG43"	integer										
162			diaglen	-	-	-	-	"FIG44"	integer										
163			dtempnen	-	-	-	-	"FIG45"	integer										
164			entlvi	-	-	-	-	"FIG46"	integer										
165			grpcont	-	-	-	-	-	"FIG47"	integer									
166			grpgtcont	-	-	-	-	-	"FIG48"	integer									
167			grplvi	-	-	-	-	-	"FIG49"	integer									
168			litten	-	-	-	-	-	"FIG4a"	integer									
169			namelen	-	-	-	-	-	"FIG4b"	integer									
170			normsep	-	-	-	-	-	"FIG4c"	integer									
171			pilen	-	-	-	-	-	"FIG4d"	integer									
172	taglen	-	-	-	-	-	"FIG4e"	integer											
173	taglvi	-	-	-	-	-	"FIG4f"	integer											
174		addfun	-	-	"added function character"	"added function char"	"d4400"	-											

Table D.1 (continued)

	element			attribute														
Element No.	ps module	classdef	profdef	Node related specifi- cation	fullnm	apnrm	clause	data type	ac	strlex	cn	conprop	dsepprop	strmnorm	sd	dependon	acnmprop	mayadd
175			name	-	-	-	"d4402"	string		name				general				
176			class	-	"function class"	-	"d4403"	enum										
177			char	-	character	-	"95003"	char										
178		dlmrlas	-	-	"delimiter role association"	"delim role assoc"	"d4610"	-										
179			role	-	-	-	"d4612"	string						rcsgener				
180			delm	-	delimiter	delim	"d4611"	string							general			
181		synlitas	-	-	"syntactic literal association"	"syntactic literal assoc"	"d4700"	-										
182			synlilt	-	-	-	"syntactic literal"	"d4702"	string						rcsgener			
183			resname	-	-	-	"reserved name"	"d4702"	string		name				general			
184		capset	-	-	-	"capacity set"	"d2000"	-										
185			totalcap	-	-	-	-	"FIG51"	integer									
186			entcap	-	-	-	-	"FIG52"	integer									
187		entchcap	-	-	-	-	"FIG53"	integer										
188		elemcap	-	-	-	-	"FIG54"	integer										