DSSSL Online Application Profile

Last revised 1996.08.16

Recent additions: keyword feature; lambda, #!key, let (letrec, let*, named let)

1. Introduction

This draft describes an application profile of DSSSL designed for the formatting specification requirements of online SGML browsers and editors (including HTML browsers and editors). This subset, DSSSL Online (informally "dsssl-o"), supports the basic features needed to provide publisher-oriented formatting control of online displays and a minimum set of page-oriented features needed to provide utility printouts from browsers and editors.

Note that this draft is based on *committee drafts of the full DSSSL Standard* and may not accurately represent the final DSSSL Standard as published. Also, this document is not intended as an introduction to DSSSL, but rather as a description of conformance levels for users and implementors who have already read the standard. A PostScript copy of the DSSSL standard as published is available by anonymous FTP from ftp.ornl.gov in the directory /sgml/wg8/dsssl.

2. DSSSL Online in context

DSSSL (Document Style Semantics and Specification Language) is an International Standard, ISO/IEC 10179:1996, for specifying document transformation and formatting in a platform- and vendor-neutral manner. DSSSL can be used with any document format for which a property set can be defined according to the Property Set Definition Requirements of ISO/IEC 10744. In particular, it can be used to specify the presentation of documents marked up according to ISO 8879:1986, Standard Generalized Markup Language (SGML).

DSSSL consists of two main components: a transformation language and a style language. The transformation language is used to specify structural transformations on SGML source files. For example, a telephone directory structured as a series of entries ordered by last name could, by applying a transformation spec, be rendered as a series of entries sorted by first name instead. The transformation language can also be used to specify the merging of two or more documents, the generation of indexes and tables of contents, and other operations. While the transformation language is a powerful tool for gaining the maximum use from document databases, the focus in early DSSSL implementations will be on the style language component.

Within the style language, it is possible to identify a number of capabilities that for one reason or another should be considered optional for early implementations. Recognizing this, the designers of DSSSL designated certain features of the style language as optional and created a Core Query Language and a Core Expression Language specifically in order to make more limited implementations possible. However, they did not define any particular subset of the style language component within the standard itself, but rather left that task to industry organizations and standards bodies. This application profile is intended to be one such specification.

The current document grew out of discussions on the former DSSSL-Lite mailing list during the period from September through November 1995. These discussions culminated in a December 9, 1995 meeting of key SGML and DSSSL implementors in Boston hosted by Jon Bosak of Novell and chaired by the late Yuri Rubinsky of SoftQuad. The application profile resulting from that meeting was published on the Internet on December 12, 1995 as the document do951212.htm and announced at a workshop of the Fourth International World Wide Web Conference the same day.

In August 1996, the application profile was updated by Jon Bosak (now an employee of Sun Microsystems) to correct a number of discrepancies between the summary descriptions of flow object characteristics, which had been based on the September 1995 committee draft of the DSSSL specification, and the final DSSSL standard published in April 1996, and also to add certain optional features that are by consensus of the active DSSSL implementors now considered to be part of a minimal DSSSL implementation. These features include lambda, keywords, and let (including letrec, let*, and named let). This document is archived at sunsite.unc.edu in the directory /pub/sun-info/standards/dsssl/dssslo, from which it can be obtained by anonymous FTP.

3. DSSSL Online summary

This section summarizes the features and flow object classes that must be supported by a minimally conformant DSSSL Online application. Note that the appearance of a flow object class in this section does not imply that the minimal dsssl-o application supports all standard values of all characteristics associated with each flow object class; the extent of this support is detailed in the tables that appear later. The section numbers refer to sections in the DSSSL standard.

3.1 Core DSSSL components

The following items are mandatory in the DSSSL standard and therefore in DSSSL Online.

Core query language, core expression language

Basic flow object classes
12.6.1 Sequence
12.6.2 Display group
12.6.6 Paragraph
12.6.7 Paragraph break
12.6.8 Line field
12.6.9 Sideline
12.6.11 Character
12.6.12 Leader
12.6.14 Rule
12.6.15 External graphic
12.6.17 Score
12.6.18 Box
12.6.22 Alignment point
12.6.23 Aligned column

3.2 DSSSL options required in DSSSL Online

The following flow object classes and features are optional in the full DSSSL Standard but are required in DSSSL Online.

Features: table, table-auto-width, online, simple-page, alignment, keyword

Syntactic keywords: lambda, #!key, letrec, let*, let (including named let)

Simple page flow object class
12.6.3 Simple page sequence

Table flow object classes
12.6.27.1 Table
12.6.27.2 Table part
12.6.27.3 Table column
12.6.27.5 Table row
12.6.27.6 Table cell
12.6.27.7 Table border

Online display flow object classes
12.6.28.1 Scroll
12.6.28.2 Multi-mode
12.6.28.3 Link
12.6.28.4 Marginalia

3.3 DSSSL Options not required in DSSSL Online

For reference purposes, this section lists those optional components of the full DSSSL Standard that need not be supported in applications claiming to minimally conform to the DSSSL Online application profile. The categorization of features and flow object classes used here does not appear in the standard itself but is provided as a possibly useful editorial device.

3.3.1 Complex typography options

Features: page, multi-column, nested-column-set, combine-char, general-indirect, font-info, included-container, actual-characteristic

Complex typography flow object classes
12.6.4 Page sequence
12.6.5 Column set sequence
12.6.10 Anchor
12.6.16 Included container
12.6.19 Side-by-side
12.6.20 Side-by-side item

3.3.2 Bidi option

Features: bidi

Bidi flow object class
12.6.13 Embedded text

3.3.3 Asian language options

Features: vertical, inline-note, glyph-annotation, emphasizing-mark

Asian language flow object classes
12.6.21 Glyph annotation
12.6.24 Multi-line inline note
12.6.25 Emphasizing mark

3.3.4 Mathematics options

Features: math

Mathematics flow object classes
12.6.26.1 Math sequence
12.6.26.2 Unmath
12.6.26.3 Subscript
12.6.26.4 Superscript
12.6.26.5 Script
12.6.26.6 Mark
12.6.26.7 Fence
12.6.26.8 Fraction
12.6.26.9 Radical
12.6.26.10 Math operator
12.6.26.11 Grid
12.6.26.12 Grid-cell

3.3.5 Complex SGML options

Full expression language
Full query language (SDQL)

Features: query, keyword

4. DSSSL Online flow object class synopses

This section describes each flow object class in the DSSSL Online application profile and specifies minimum support for the characteristics of each flow object class.

4.1 Key to the tables of flow object characteristics

In the tables below, the column headed **Inh.** indicates whether the characteristic is inherited (I) or non-inherited (N); the column headed **Standard Values** shows the values that are defined for each class in the full DSSSL standard; the column headed **Standard Init/Default** shows the initial or default value, if any, specified in the DSSSL standard; and the column headed **Dsssl-o** shows the minimum level of support to be expected in applications purporting to follow this application profile.

Dsssl-o support in the last column of each table is described as **Standard**, indicating that the characteristic is to be treated exactly as specified in the DSSSL Standard; **Default**, indicating that a minimal DSSSL Online application can, if necessary, substitute the default value specified in the Standard for any other legal value provided in a DSSSL style specification; **Ignore**, indicating that any value given for the characteristic can be ignored by a minimal DSSSL Online application; or **Partial**, indicating that some of the standard values must be supported, but not all of them. In the case of characteristics in the **Partial** category, the values for which support is not required are marked with asterisks (*). A minimal DSSSL Online application is allowed to substitute the standard initial or default value for any characteristic marked with an asterisk. However, every legal value must be propagated to inheriting flow objects when support for that value is required in the inheriting context.

4.2 Sequence flow object class [12.6.1]

A sequence flow object class is formatted to produce the concatenation of the areas produced by each of its children. It has a single principal port. Its children may be inlined or displayed.

NOTE: *A sequence flow object is useful for specifying inherited characteristics. For example, a sequence flow object with a specification of a* `font-posture:` *characteristic may be constructed for an emphasized phrase element in a paragraph.*

A port of a flow object shall accept a sequence flow object if and only if it would accept each of the flow objects in that sequence.

4.3 Display group flow object class [12.6.2]

A display-group flow object class is formatted to produce the concatenation of the areas produced by each of its children. It has a single principal port. Its children shall all be displayed and it is itself displayed.

NOTE: *It will, therefore, cause a line break in a paragraph even if the display-group has no content.*

Characteristic	Inh.	Standard Values	Standard Init/Default	Dsssl-o
coalesce-id	N	#f, *string*	#f	Ignore
position-preference	N	#f, top, bottom	#f	Ignore
space-before	N	*display space*	no space before	Standard
space-after	N	*display space*	no space after	Standard
keep-with-previous?	N	#f, #t	#f	Standard
keep-with-next?	N	#f, #t	#f	Standard
break-before	N	#f, page, page-region*, column*, column-set*	#f	Partial
break-after	N	#f, page, page-region*, column*, column-set*	#f	Partial
keep	N	#f, page, column-set*, column*, #t*	#f	Partial
may-violate-keep-before?	N	#f, #t	#f	Default
may-violate-keep-after?	N	#f, #t	#f	Default

Table 1. Display group flow object class characteristics

4.4 Simple page sequence flow object class [12.6.3]

A simple-page-sequence flow object class is formatted to produce a sequence of page areas. A simple-page-sequence flow object has a single principal port that accepts any displayed flow object.

NOTE: *The simple-page-sequence flow object is intended for systems that wish to provide a very simple page layout facility. More complex page layouts can be obtained with the page-sequence and column-set-sequence flow object classes.*

A simple-page-sequence flow object shall not be allowed within the content of any other flow object class.

A simple-page-sequence may have a single line header and footer containing text that is constant except for a page number.

NOTE: *A document can contain multiple simple-page-sequences. For example, each chapter of a document could be a separate simple-page-sequence; this would allow the chapter title within a header or footer line.*

The page shall be filled from top to bottom. The display size for the contents of the simple-page-sequence shall be the value of the `page-width:` less the value of the `left-margin:` and `right-margin:` characteristics.

Characteristic	**Inh.**	**Standard Values**	**Standard Init/Default**	**Dsssl-o**
page-width	I	*length*	system-dependent	Standard
page-height	I	*length*	system-dependent	Standard
left-margin	I	*length*	0pt	Standard
right-margin	I	*length*	0pt	Standard
top-margin	I	*length*	0pt	Standard
bottom-margin	I	*length*	0pt	Standard
header-margin	I	*length*	0pt	Standard
footer-margin	I	*length*	0pt	Standard
left-header	N	*unlabeled sosofo*	empty sosofo	Standard
center-header	N	*unlabeled sosofo*	empty sosofo	Standard
right-header	N	*unlabeled sosofo*	empty sosofo	Standard
left-footer	N	*unlabeled sosofo*	empty sosofo	Standard
center-footer	N	*unlabeled sosofo*	empty sosofo	Standard
right-footer	N	*unlabeled sosofo*	empty sosofo	Standard
writing-mode	I	left-to-right, right-to-left, top-to-bottom	left-to-right	Default

Table 2. Simple page sequence flow object class characteristics

Associated procedures:

`(page-number-sosofo)`

Returns an indirect-sosofo whose content is a sequence of character flow objects representing the page number of the page on which the first area resulting from the indirect flow object specified by the indirect-sosofo occurs.

`(current-node-page-number-sosofo)`

Returns an indirect-sosofo whose content is a sequence of character flow objects representing the page number of the primary flow object of the current node.

NOTE: *This is intended to handle cross references in conjunction with* `process-referent`.

4.5 Paragraph flow object class [12.6.6]

A paragraph flow object represents a paragraph. It has a single principal port. The contents of this port may be either inlined or displayed. Inline flow objects are formatted to produce line areas. Displayed flow objects implicitly specify a break, and their areas shall be added to the resulting sequence of areas. A paragraph flow object may only be displayed.

NOTE: *Typically, a break implies that a new line is to be started.*

Characteristic	Inh.	Standard Values	Standard Init/Default	Dsssl-o
lines	I	wrap, asis, asis-wrap*, asis-truncate*, none	wrap	Partial
asis-truncate-char	I	#f, *char object*	#f	Ignore
asis-wrap-char	I	#f, *char object*	#f	Ignore
asis-wrap-indent	I	*length-spec*	#f	Ignore
first-line-align	I	#f, *char object*	#f	Ignore
alignment-point-offset	I	#f, *number between 0 and 100*	50	Ignore
ignore-record-end?	I	#f, #t	#f	Standard
expand-tabs?	I	#f, *integer greater than zero*	8	Standard
line-spacing	I	*length-spec*	12pt	Standard
line-spacing-priority	I	force, *integer*	0	Default
min-pre-line-spacing	I	#f, *length-spec*	#f	Default
min-post-line-spacing	I	#f, *length-spec*	#f	Default
min-leading	I	#f, *length-spec*	#f	Standard
first-line-start-indent	I	*length-spec*	0pt	Standard
last-line-end-indent	I	*length-spec*	0pt	Default
hyphenation-char	I	*character*	#\- (the hyphen character)	Ignore
hyphenation-method	I	#f, *public identifier*	#f	Default
hyphenation-ladder-count	I	#f, *integer greater than zero*	#f	Default
hyphenation-remain-char-count	I	*positive integer*	2	Ignore
hyphenation-push-char-count	I	*positive integer*	2	Ignore
hyphenation-keep	I	#f, spread, page, column	#f	Default
hyphenation-exceptions	I	*list of strings*	the empty list	Default
line-breaking-method	I	#f, *public identifier*	#f	Default
line-composition-method	I	#f, *public identifier*	#f	Default
implicit-bidi-method	I	#f, *public identifier*		Default
glyph-alignment-mode	I	base, center, top, bottom, font	font	Default
font-family-name	I	#f, *string*	iso-serif	Standard

Characteristic	Inh.	Standard Values	Standard Init/Default	Dsssl-o
font-weight	I	#f, not-applicable*, ultra-light*, extra-light*, light, semi-light*, medium, semi-bold*, bold, extra-bold*, ultra-bold*	medium	Partial
font-posture	I	#f, not-applicable*, upright, oblique, back-slanted-oblique*, italic, back-slanted-italic*	upright	Partial
font-structure	I	#f, not-applicable, solid, outline	solid	Default
font-proportionate-width	I	#f, not-applicable*, ultra-condensed*, extra-condensed*, condensed, semi-condensed*, medium, semi-expanded*, expanded, extra-expanded*, ultra-expanded*	medium	Partial
font-name	I	#f, *public identifier*	#f	Standard
font-size	I	*length*	10pt	Standard
numbered-lines?	I	#f, #t	#t	Ignore
line-number	I	#f, *unlabeled sosofo*	#f	Ignore
line-number-side	I	start, end, spread-inside, spread-outside, page-inside, page-outside		Ignore
line-number-sep	I	*length-spec*		Ignore
quadding	I	start, end, spread-inside*, spread-outside*, page-inside*, page-outside*, center, justify*	start	Partial
last-line-quadding	I	relative, start, end, spread-inside, spread-outside, page-inside, page-outside, center, justify	relative	Default
last-line-justify-limit	I	*length-spec*	0	Ignore
justify-glyph-space-max-add	I	*length-spec*	0pt	Ignore
justify-glyph-space-max-remove	I	*length-spec*	0pt	Ignore
hanging-punct?	I	#f, #t	#f	Default
widow-count	I	*positive integer*	2	Default
orphan-count	I	*positive integer*	2	Default
language	I	#f, *ISO 639 language code*	#f	Standard
country	I	#f, *ISO 3166 country code*	#f	Standard
position-preference	N	#f, top, bottom	#f	Ignore

Characteristic	**Inh.**	**Standard Values**	**Standard Init/Default**	**Dsssl-o**
writing-mode	I	left-to-right, right-to-left, top-to-bottom	left-to-right	Default
start-indent	I	*length-spec*	0pt	Standard
end-indent	I	*length-spec*	0pt	Standard
span	I	*integer greater than zero*	1	Default
span-weak?	I	#f, #t	#f	Default
space-before	N	*display space*	no space before	Standard
space-after	N	*display space*	no space after	Standard
keep-with-previous?	N	#f, #t	#f	Standard
keep-with-next?	N	#f, #t	#f	Standard
break-before	N	#f, page, page-region*, column*, column-set*	#f	Partial
break-after	N	#f, page, page-region*, column*, column-set*	#f	Partial
keep	N	#f, page, column-set*, column*, #t*	#f	Partial
may-violate-keep-before?	N	#f, #t	#f	Default
may-violate-keep-after?	N	#f, #t	#f	Default

Table 3. Paragraph flow object class characteristics

4.6 Paragraph break flow object class [12.6.7]

In conjunction with the paragraph-break flow object class, a paragraph flow object can be used to represent a sequence of paragraphs. The paragraphs are separated by paragraph-break flow objects, which are atomic. Paragraph-break flow objects are allowed only in paragraph flow objects. All the characteristics that are applicable to a paragraph flow object are also applicable to a paragraph-break flow object. The characteristics of a paragraph-break flow object determine how the portion of the content of the paragraph flow object following that paragraph-break flow object up to the next paragraph-break flow object, if any, is formatted.

NOTE: *The paragraph-break flow object inherits from its containing paragraph flow object in the usual way.*

The `first-line-start-indent:` is applicable to the line following a paragraph-break flow object, and the `last-line-end-indent:` is applicable to the line preceding a paragraph-break flow object.

NOTE: *It is recommended that paragraph-break flow objects are used only if there is no other way of specifying the desired formatting.*

Characteristics: See Paragraph flow object class.

4.7 Line field flow object class [12.6.8]

The line-field flow object class is inlined and has inline content. It produces a single inline area. The width of this area is equal to the value of the `field-width:` characteristic. If the content of line-field area cannot fit in this width, then the area will grow to accommodate the content and, if the line-field occurs in the paragraph, there shall be a break after the line-field.

It has a single principal port.

Characteristic	Inh.	Standard Values	Standard Init/Default	Dsssl-o
field-width	I	*length-spec*	0pt	Standard
field-align	I	start, end, center	start	Default
writing-mode	I	left-to-right, right-to-left, top-to-bottom	left-to-right	Default
inhibit-line-breaks?	I	#f, #t	#f	Standard
break-before-priority	N	*integer*	0	Ignore
break-after-priority	N	*integer*	0	Ignore

Table 4. Line field flow object class characteristics

4.8 Sideline flow object class [12.6.9]

A sideline flow object is used to contain flow objects that have an attachment area consisting of a line parallel to the placement direction. A sideline flow object has a single principal port which can contain both inlined and displayed flow objects. For each display area produced by its content, the sideline flow object adds an attachment. For each inline area produced by its content, the sideline flow object annotates that area so as to cause the paragraph in which the flow object occurs to add an attachment area to the line in which that inline area occurs.

NOTE: *Sidelines are often used to mark changes.*

Sidelines on consecutive areas in a single area container which have no space between them should be drawn as a single line.

Characteristic	Inh.	Standard Values	Standard Init/Default	Dsssl-o
sideline-side	I	start, end, both, spread-inside*, spread-outside*, page-inside*, page-outside*		Partial
sideline-sep	I	*length-spec*		Standard
color	I	*color*	default color in Device Gray color space	Standard
layer	I	*integer*	0	Default
line-cap	I	butt, round, square	butt	Default
line-dash	I	*list of lengths*	list containing the length 0pt	Default
line-thickness	I	*length*	1pt	Standard
line-repeat	I	*integer*	1	Default
line-sep	I	*length*	1pt	Ignore

Table 5. Sideline flow object class characteristics

4.9 Character flow object class [12.6.11]

A character flow object is atomic. Flow objects of this class can only be inlined.

A character flow object is formatted to produce a single inline area. This may be merged with adjacent inline areas if the `ligature?:` property is true. The position point of the inline area is the position point property of the glyph specified in the font resource for the specified writing-mode. The escapement direction is the direction between the position point and escapement points as specified in the font resource for the specified writing-mode. The size of the area in the escapement direction is the distance between the position and escapement points. The size of the area before and after the placement path in the line-progression direction is the smallest that will enclose the extent of the glyph in those directions as specified in the font resource for the specified writing-mode. If the nominal alignment mode of the font resource for the character flow object's writing-mode is not the same as the paragraph's alignment mode, then the glyph area is automatically adjusted as specified by the alignment mode property in the font resource for the specified writing-mode.

Characteristic	**Inh.**	**Standard Values**	**Standard Init/Default**	**Dsssl-o**
char	N	*character*	value of the current node's char property, if there is one, else the value of applying the char-map characteristic, if there is one	Standard
char-map	I	#f, *procedure*	#f	Default
glyph-id	N	#f, *glyph-id*	glyph-id of the nominal glyph specified using the standard-chars architectural form, else #f	Standard
glyph-subst-table	I	#f, *glyph-subst-table*, *list of glyph-subst-table*	#f	Default
glyph-subst-method	I	#f, *public identifier*, *list of public identifiers*	#f	Default
glyph-reorder-method	I	#f, *public identifier*, *list of public identifiers*	#f	Default
writing-mode	I	left-to-right, right-to-left, top-to-bottom	left-to-right	Default
font-family-name	I	#f, *string*	iso-serif	Standard
font-weight	I	#f, not-applicable*, ultra-light*, extra-light*, light, semi-light*, medium, semi-bold*, bold, extra-bold*, ultra-bold*	medium	Partial
font-posture	I	#f, not-applicable*, upright, oblique, back-slanted-oblique*, italic, back-slanted-italic*	upright	Partial
math-font-posture	N	#f, not-applicable*, upright, oblique, back-slanted-oblique*, italic, back-slanted-italic*	the value of the math-font-posture character property of the char: characteristic	Partial

Characteristic	Inh.	Standard Values	Standard Init/Default	Dsssl-o
font-structure	I	#f, not-applicable, solid, outline	solid	Default
font-proportionate-width	I	#f, not-applicable*, ultra-condensed*, extra-condensed*, condensed, semi-condensed*, medium, semi-expanded*, expanded, extra-expanded*, ultra-expanded*	medium	Partial
font-name	I	#f, *public identifier*	#f	Standard
font-size	I	*length*	10pt	Standard
stretch-factor	N	*number*	1	Default
hyphenate?	I	#f, #t	#f	Default
hyphenation-method	I	#f, *public identifier*	#f	Default
kern?	I	#f, #t	#f	Default
kern-mode	I	loose, normal, kern, tight, touch	normal	Default
ligature?	I	#f, #t	#f	Default
allowed-ligatures	I	*list of glyph-ids or chars*	the empty list	Default
space?	N	#f, #t	the value of the space? character property of the char: characteristic	Standard
inline-space-space	I	*inline-space*		Ignore
escapement-space-before	I	*inline-space*	inline-space of 0pt	Ignore
escapement-space-after	I	*inline-space*	inline-space of 0pt	Ignore
record-end?	N	#f, #t	the value of the record-end? character property of the char: characteristic	Standard
input-tab?	N	#f, #t	the value of the input-tab? character property of the char: characteristic if the char: characteristic was not explicitly specified, and otherwise #f	Standard
input-whitespace-treatment	I	preserve, collapse, ignore	preserve	Standard
input-whitespace?	N	#f, #t	the value of the input-white-space? character property of the char: characteristic if the char: characteristic was not explicitly specified, and otherwise #f	Default
punct?	N	#f, #t	the value of the punct? character property of the char: characteristic	Ignore

Characteristic	Inh.	Standard Values	Standard Init/Default	Dsssl-o
break-before-priority	N	*integer*	the value of the break-before-priority character property of the char: characteristic	Ignore
break-after-priority	N	*integer*	the value of the break-after-priority character property of the char: characteristic	Ignore
drop-after-line-break?	N	#f, #t	the value of the drop-after-line-break? character property of the char: characteristic	Ignore
drop-unless-before-line-break?	N	#f, #t	the value of the drop-unless-before-line-break? character property of the char: characteristic	Ignore
math-class	N	ordinary, operator, binary, relation, opening, closing, punctuation, inner, space	the value of the math-class character property of the char: characteristic	Ignore
script	N	#f, *formal public identifier*	the value of the script character property of the char: characteristic	Ignore
position-point-shift	I	*length-spec*	0pt	Standard
language	I	#f, *ISO 639 language code*	#f	Standard
country	I	#f, *ISO 3166 country code*	#f	Standard
color	I	*color*	default color in Device Gray color space	Standard
layer	I	*integer*	0	Default
inhibit-line-breaks?	I	#f, #t	#f	Standard

Table 6. Character flow object class characteristics

4.10 Leader flow object class [12.6.12]

A leader flow object can only be inlined. A leader flow object has a single principal port containing the inline flow objects to be repeated.

Characteristic	Inh.	Standard Values	Standard Init/Default	Dsssl-o
length	N	*length-spec*	determined by context	Standard
truncate-leader?	I	#f, #t	#f	Standard
align-leader?	I	#f, #t	#t	Standard
min-leader-repeat	I	*integer greater than 0*	1	Standard
inhibit-line-breaks?	I	#f, #t	#f	Standard
break-before-priority	N	*integer*	0	Ignore
break-after-priority	N	*integer*	0	Ignore

Table 7. Leader flow object class characteristics

4.11 Rule flow object class [12.6.14]

A rule is used to specify a straight line. Rules may be inlined or displayed.

Characteristic	**Inh.**	**Standard Values**	**Standard Init/Default**	**Dsssl-o**
orientation	N	horizontal, vertical, escapement, line-progression	shall be specified	Standard
length	N	*length*	determined by context	Standard
color	I	*color*	default color in Device Gray color space	Standard
layer	I	*integer*	0	Default
line-cap	I	butt, round, square	butt	Default
line-dash	I	*list of lengths*	list containing the length 0pt	Default
line-thickness	I	*length*	1pt	Standard
line-repeat	I	*integer*	1	Default
line-sep	I	*length*	1pt	Ignore
position-point-shift	I	*length-spec*	0pt	Standard
inhibit-line-breaks?	I	#f, #t	#f	Standard
break-before-priority	N	*integer*	0	Ignore
break-after-priority	N	*integer*	0	Ignore
position-preference	N	#f, top, bottom	#f	Ignore
display-alignment	I	start, center, end, inside*, outside*	start	Partial
start-indent	I	*length-spec*	0pt	Standard
end-indent	I	*length-spec*	0pt	Standard
writing-mode	I	left-to-right, right-to-left, top-to-bottom	left-to-right	Default
span	I	*integer greater than zero*	1	Default
span-weak?	I	#f, #t	#f	Default
space-before	N	*display space*	no space before	Standard
space-after	N	*display space*	no space after	Standard
keep-with-previous?	N	#f, #t	#f	Standard
keep-with-next?	N	#f, #t	#f	Standard
break-before	N	#f, page, page-region*, column*, column-set*	#f	Partial
break-after	N	#f, page, page-region*, column*, column-set*	#f	Partial
keep	N	#f, page, column-set*, column*, #t*	#f	Partial
may-violate-keep-before?	N	#f, #t	#f	Default
may-violate-keep-after?	N	#f, #t	#f	Default

Table 8. Rule flow object class characteristics

4.12 External graphic flow object class [12.6.15]

The external-graphic flow object class is used for graphics contained in an external entity. Flow objects of this class may be inlined or displayed. This flow object is atomic.

Characteristic	**Inh.**	**Standard Values**	**Standard Init/Default**	**Dsssl-o**
display?	N	#f, #t	#f	Standard
scale	N	*number*, *list of two numbers*, max, max-uniform	max-uniform	Standard
max-width	N	*length-spec*		Standard
max-height	N	*length-spec*		Standard
entity-system-id	N	#f, *system identifier*	shall be specified	Standard
notation-system-id	N	*system identifier*	shall be specified	Standard
color	I	*color*	default color in Device Gray color space	Standard
layer	I	*integer*	0	Default
position-preference	N	#f, top, bottom	#f	Ignore
display-alignment	I	start, center, end, inside*, outside*	start	Partial
start-indent	I	*length-spec*	0pt	Standard
end-indent	I	*length-spec*	0pt	Standard
writing-mode	I	left-to-right, right-to-left, top-to-bottom	left-to-right	Default
span	I	*integer greater than zero*	1	Default
span-weak?	I	#f, #t	#f	Default
space-before	N	*display space*	no space before	Standard
space-after	N	*display space*	no space after	Standard
keep-with-previous?	N	#f, #t	#f	Standard
keep-with-next?	N	#f, #t	#f	Standard
break-before	N	#f, page, page-region*, column*, column-set*	#f	Partial
break-after	N	#f, page, page-region*, column*, column-set*	#f	Partial
keep	N	#f, page, column-set*, column*, #t*	#f	Partial
may-violate-keep-before?	N	#f, #t	#f	Default
may-violate-keep-after?	N	#f, #t	#f	Default
position-point-x	N	*length-spec*	0 if writing-mode is left-to-right or right-to-left	Standard
position-point-y	N	*length-spec*	0 if writing-mode is top-to-bottom	Standard
escapement-direction	N	top-to-bottom, left-to-right, right-to-left	the value of writing-mode	Default
inhibit-line-breaks?	I	#f, #t	#f	Standard
break-before-priority	N	*integer*	0	Ignore
break-after-priority	N	*integer*	0	Ignore

Table 9. External graphic flow object class characteristics

4.13 Score flow object class [12.6.17]

The score flow object has a single principal port. The content of this port is scored. The port can contain only inlined flow objects.

***NOTE:** Kendot scoring in Japanese is achieved using the emphasizing-mark flow object class of full DSSSL.*

Characteristic	Inh.	Standard Values	Standard Init/Default	Dsssl-o
type	N	before, through, after, *length-spec*, *character*	shall be specified	Standard
score-spaces?	I	#f, #t	#t	Standard
color	I	*color*	default color in Device Gray color space	Standard
layer	I	*integer*	0	Default
line-cap	I	butt, round, square	butt	Default
line-dash	I	*list of lengths*	list containing the length 0pt	Default
line-thickness	I	*length*	1pt	Standard
line-repeat	I	*integer*	1	Default
line-sep	I	*length*	1pt	Ignore
inhibit-line-breaks?	I	#f, #t	#f	Standard

Table 10. Score flow object class characteristics

4.14 Box flow object class [12.6.18]

The box flow object may be used to put a box around a sequence of flow objects. The box flow object is either displayed or inlined depending on the value of the `display?:` characteristic. The box flow object has a single principal port. If the box is displayed, then the port shall accept any displayed flow objects. If the box is inlined, then the port shall accept any inline flow objects.

The box flow object may result in more than one area. In this case, the border of the box adjacent to the break may be omitted if the `box-open-end?:` characteristic is true.

If the box is inlined, then this border shall be perpendicular to the writing-mode. If the box is displayed, then this border shall be parallel to the writing-mode.

When the box is displayed, the size of the box (that is, the distance between the positions of the borders) in the direction determined by the writing-mode shall be equal to the display size of the box less the start and end indents. The display size for the content of the box shall be equal to the size of the box.

***NOTE:** Thus, the `start-indent:` and `end-indent:` characteristics for the content of the box shall be set to give the desired separation between the border of the box and its content. There is no automatic separation to take account of the thickness of border.*

Characteristic	Inh.	Standard Values	Standard Init/Default	Dsssl-o
display?	N	#f, #t	#f	Standard
box-type	I	border, background, both	border	Standard
box-open-end?	I	#f, #t	#f	Standard
background-color	I	*color*	#f	Standard
background-layer	I	*integer*	-1	Default
box-corner-rounded	I	#f, #t, *list of corner specs*	#f	Default
box-corner-radius	I	*length-spec*	3pt	Ignore
box-border-alignment	I	center, outside, inside	outside	Default
box-size-before	I	*length*	8pt	Standard
box-size-after	I	*length*	4pt	Standard
color	I	*color*	default color in Device Gray color space	Standard
layer	I	*integer*	0	Default
line-cap	I	butt, round, square	butt	Default
line-dash	I	*list of lengths*	list containing the length 0pt	Default
line-thickness	I	*length*	1pt	Standard
line-repeat	I	*integer*	1	Default
line-sep	I	*length*	1pt	Ignore
line-miter-limit	I	*number*	10	Default
line-join	I	miter, round, bevel	miter	Default
writing-mode	I	left-to-right, right-to-left, top-to-bottom	left-to-right	Default
position-preference	N	#f, top, bottom	#f	Ignore
inhibit-line-breaks?	I	#f, #t	#f	Standard
break-before-priority	N	*integer*	0	Ignore
break-after-priority	N	*integer*	0	Ignore
start-indent	I	*length-spec*	0pt	Standard
end-indent	I	*length-spec*	0pt	Standard
span	I	*integer greater than zero*	1	Default
span-weak?	I	#f, #t	#f	Default
space-before	N	*display space*	no space before	Standard
space-after	N	*display space*	no space after	Standard
keep-with-previous?	N	#f, #t	#f	Standard
keep-with-next?	N	#f, #t	#f	Standard
break-before	N	#f, page, page-region*, column*, column-set*	#f	Partial
break-after	N	#f, page, page-region*, column*, column-set*	#f	Partial
keep	N	#f, page, column-set*, column*, #t*	#f	Partial
may-violate-keep-before?	N	#f, #t	#f	Default
may-violate-keep-after?	N	#f, #t	#f	Default

Table 11. Box flow object class characteristics

4.15 Alignment point flow object class [12.6.22]

An alignment-point flow object class specifies an explicit alignment point for paragraphs with a `first-line-align:` characteristic equal to #t. It is atomic and inlined.

4.16 Aligned column flow object class [12.6.23]

An aligned-column flow object is used for grouping together externally aligned paragraphs. An aligned-column is displayed. It has a single principal port that may contain any displayed flow objects. Displayed flow objects in the port that are not externally aligned paragraphs shall be formatted normally. The externally aligned paragraphs in the content or in side-by-side flow objects in the content are aligned in the direction of the writing-mode so that their alignment points lie on a line in the placement direction. The resulting group of aligned lines is then positioned according to the `display-alignment:`, `start-indent:`, and `end-indent:` characteristics.

Characteristic	Inh.	Standard Values	Standard Init/Default	Dsssl-o
display-alignment	I	start, center, end, inside*, outside*	start	Partial
start-indent	I	*length-spec*	0pt	Standard
end-indent	I	*length-spec*	0pt	Standard
writing-mode	I	left-to-right, right-to-left, top-to-bottom	left-to-right	Default
position-preference	N	#f, top, bottom	#f	Ignore
space-before	N	*display space*	no space before	Standard
space-after	N	*display space*	no space after	Standard
keep-with-previous?	N	#f, #t	#f	Standard
keep-with-next?	N	#f, #t	#f	Standard
break-before	N	#f, page, page-region*, column*, column-set*	#f	Partial
break-after	N	#f, page, page-region*, column*, column-set*	#f	Partial
keep	N	#f, page, column-set*, column*, #t*	#f	Partial
may-violate-keep-before?	N	#f, #t	#f	Default
may-violate-keep-after?	N	#f, #t	#f	Default

Table 12. Aligned column flow object class characteristics

4.17 Table flow object class [12.6.27.1]

A table flow object has a single principal port. The contents of this port shall be either:

- all of class table-part, or
- all of class table-column, table-row, or table-cell

If it contains flow objects of class table-column, they shall occur before all flow objects of other classes. A table flow object can only be displayed.

A table has two directions associated with it, a row-progression direction and a column-progression direction. The row-progression direction is equal to the placement direction of the table flow object. The column-progression direction is given by the value of the `writing-mode:` characteristic of the table flow object. These are required to be perpendicular.

Characteristic	**Inh.**	**Standard Values**	**Standard Init/Default**	**Dsssl-o**
table-width	N	#f, *length-spec*	display-size less applicable indent	Standard
table-auto-width-method	I	#f, *public identifier*	#f	Standard
table-border	I	#f, #t, *unlabeled sosofo*	#f	Standard
before-row-border	N	#f, #t, *unlabeled sosofo*	value of the table-border characteristic	Standard
after-row-border	N	#f, #t, *unlabeled sosofo*	value of the table-border characteristic	Standard
before-column-border	N	#f, #t, *unlabeled sosofo*	value of the table-border characteristic	Standard
after-column-border	N	#f, #t, *unlabeled sosofo*	value of the table-border characteristic	Standard
table-corner-rounded	I	#f, #t, *list of x-y pairs*	#f	Default
table-corner-radius	I	*length-spec*	3pt	Ignore
position-preference	N	#f, top, bottom	#f	Ignore
display-alignment	I	start, center, end, inside*, outside*	start	Partial
start-indent	I	*length-spec*	0pt	Standard
end-indent	I	*length-spec*	0pt	Standard
writing-mode	I	left-to-right, right-to-left, top-to-bottom	left-to-right	Default
span	I	*integer greater than zero*	1	Default
span-weak?	I	#f, #t	#f	Default
space-before	N	*display space*	no space before	Standard
space-after	N	*display space*	no space after	Standard
keep-with-previous?	N	#f, #t	#f	Standard
keep-with-next?	N	#f, #t	#f	Standard
break-before	N	#f, page, page-region*, column*, column-set*	#f	Partial
break-after	N	#f, page, page-region*, column*, column-set*	#f	Partial
keep	N	#f, page, column-set*, column*, #t*	#f	Partial
may-violate-keep-before?	N	#f, #t	#f	Default
may-violate-keep-after?	N	#f, #t	#f	Default

Table 13. Table flow object class characteristics

4.18 Table part flow object class [12.6.27.2]

A table-part flow object is allowed only within a table flow object. A table-part flow object has three ports:

- the principal port is used for the table body.
- `header` is used for the table header.
- `footer` is used for the table footer.

The only flow objects that are allowed in the ports of a table-part flow object shall be only those explicitly specified herein. Flow objects of class table-column are allowed in the principal port; they shall occur before flow objects of any other class. All ports shall accept flow objects of class table-row and table-cell.

The result of formatting a table-part flow object is a sequence of areas. Each area consists of the content of the header port (unless omitted because of the `table-part-omit-middle-header?:` characteristic), followed by some portion of the content of the principal port, followed by the content of the footer port (unless omitted because of the `table-part-omit-middle-footer?:` characteristic). Each row in the principal port occurs exactly once and the order of the rows shall be preserved. The rows in the header and footer ports shall be replicated for each result area.

All table-parts in a table have the same width.

Characteristic	**Inh.**	**Standard Values**	**Standard Init/Default**	**Dsssl-o**
table-part-omit-middle-header?	I	#f, #t	#f	Default
table-part-omit-middle-footer?	I	#f, #t	#f	Default
space-before	N	*display space*	no space before	Standard
space-after	N	*display space*	no space after	Standard
keep-with-previous?	N	#f, #t	#f	Standard
keep-with-next?	N	#f, #t	#f	Standard
break-before	N	#f, page, page-region*, column*, column-set*	#f	Partial
break-after	N	#f, page, page-region*, column*, column-set*	#f	Partial
keep	N	#f, page, column-set*, column*, #t*	#f	Partial
may-violate-keep-before?	N	#f, #t	#f	Default
may-violate-keep-after?	N	#f, #t	#f	Default

Table 14. Table part flow object class characteristics

4.19 Table column flow object class [12.6.27.3]

A table-column flow object is an atomic flow object that specifies characteristics applicable to table cells that have the same column and span.

Characteristic	Inh.	Standard Values	Standard Init/Default	Dsssl-o
column-number	N	*integer greater than zero*	1 plus the column-number of the previous table column flow object, if any, else 1	Standard
n-columns-spanned	N	*integer greater than zero*	1	Standard
width	N	*length-spec*	shall be specified unless table-auto-width feature is present	Standard
display-alignment	I	start, center, end, inside*, outside*	start	Partial
start-indent	I	*length-spec*	0pt	Standard
end-indent	I	*length-spec*	0pt	Standard

Table 15. Table column flow object class characteristics

Associated procedure: `(table-unit` *`k`*`)`

Returns a length-spec that specifies *k* units of proportional measure. This can be used in the value of the `width:` characteristic. The value of a unit of proportional measure for a particular table is chosen so that the total width of the columns is equal to the specified width of the table.

NOTE: *This allows the width of a column to be specified proportionally.*

4.19.1 Automatic table width computation [12.6.27.4]

The width of a column for which no width is specified shall be at least as great as the maximum of the minimum possible widths of the content of any table cells that span exactly that column.

NOTE: *When a table-cell includes a paragraph for which the* `lines:` *characteristic has a value of* `wrap`*, the minimum possible width of that paragraph is determined in a system-dependent manner. It might, for example, be the length of the longest word.*

If a table cell spans more than one column, then the sum of the widths of the columns that it spans shall be at least as great as the width of the content of the table cell.

If a length-spec is specified for a column's width, then the column shall be exactly that wide.

If a length-spec is specified for a column-width, then that length shall be used as the display size for any table-cells which span exactly that column.

Other aspects of the width computation method can be controlled with the `table-auto-width-method:` characteristic.

4.20 Table row flow object class [12.6.27.5]

A table-row flow object serves to group table cells into rows: all table cells in a table-row start in the same geometric row.

A table-row has a single principal port, which accepts flow objects of class table-cell.

A table-row flow object can only occur as the child of a table-part or table flow object.

When flow objects of class table-cell occur directly in a table-part or table, then cells are grouped into rows using the `starts-row?:` and `ends-row?:` characteristics.

4.21 Table cell flow object class [12.6.27.6]

A table-cell has a single principal port. It shall accept any flow object that can be displayed. A table-cell flow object shall only occur as the child of a table-row, table-part, or table flow object.

NOTE: *The alignment of the content of a table-cell in the column-progression direction (e.g., horizontal alignment for left-to-right, top-to-bottom text) shall be controlled by the* `display-alignment:` *or* `quadding:` *characteristics of the content of the table-cell.*

Characteristic	Inh.	Standard Values	Standard Init/Default	Dsssl-o
column-number	N	*integer greater than zero*	current column number	Standard
n-columns-spanned	N	*integer greater than zero*	1	Standard
n-rows-spanned	N	*integer greater than zero*	1	Standard
cell-before-row-margin	I	*length*	0pt	Standard
cell-after-row-margin	I	*length*	0pt	Standard
cell-before-column-margin	I	*length*	0pt	Standard
cell-after-column-margin	I	*length*	0pt	Standard
cell-row-alignment	I	start, end, center	start	Standard
cell-background?	I	#f, #t	#f	Standard
background-color	I	*color*	#f	Standard
background-layer	I	*integer*	-1	Default
cell-before-row-border	I	#f, #t, *unlabeled sosofo*	#f	Standard
cell-after-row-border	I	#f, #t, *unlabeled sosofo*	#f	Standard
cell-before-column-border	I	#f, #t, *unlabeled sosofo*	#f	Standard
cell-after-column-border	I	#f, #t, *unlabeled sosofo*	#f	Standard
starts-row?	N	#f, #t	#f	Standard
ends-row?	N	#f, #t	#f	Standard
cell-crossed	I	#f, with, across, both	#f	Default
line-cap	I	butt, round, square	butt	Default
line-dash	I	*list of lengths*	list containing the length 0pt	Default
line-thickness	I	*length*	1pt	Standard
line-repeat	I	*integer*	1	Default
line-sep	I	*length*	1pt	Ignore
float-out-sidelines?	I	#f, #t	#f	Default
float-out-marginalia?	I	#f, #t	#f	Default
float-out-line-numbers?	I	#f, #t	#f	Default

Table 16. Table cell flow object class characteristics

4.22 Table border flow object class [12.6.27.7]

A table-border flow object is an atomic flow object used to specify the border of a table cell or of the table as a whole. A table-border flow object is not allowed in the content of any flow object.

The width of borders does not affect the width of cells, nor the positioning of the contents of cells, nor the width of the table, nor the size of the area produced by the table. In particular the width of the table, as specified by the `table-width:` characteristic on the table, is equal to the sum of the widths of the cells.

Characteristic	Inh.	Standard Values	Standard Init/Default	Dsssl-o
border-priority	I	*integer*	0	Default
border-alignment	I	center, start, end, outside, inside	center	Default
border-present?	I	#f, #t	#t	Standard
border-omit-at-break?	I	#f, #t	#f	Standard
color	I	*color*	default color in Device Gray color space	Standard
layer	I	*integer*	0	Default
line-cap	I	butt, round, square	butt	Default
line-dash	I	*list of lengths*	list containing the length 0pt	Default
line-thickness	I	*length*	1pt	Standard
line-repeat	I	*integer*	1	Default
line-sep	I	*length*	1pt	Ignore
line-miter-limit	I	*number*	10	Default
line-join	I	miter, round, bevel	miter	Default

Table 17. Table-border flow object class characteristics

4.23 Scroll flow object class [12.6.28.1]

A scroll flow object class is used as the top-level flow object for online display that does not divide output into pages. It has a single principal port, which accepts displayed flow objects. The size of the flow object in the direction perpendicular to the filling-direction is determined by the viewing environment.

Characteristic	Inh.	Standard Values	Standard Init/Default	Dsssl-o
filling-direction	I	top-to-bottom, left-to-right, right-to-left	top-to-bottom	Default
writing-mode	I	left-to-right, right-to-left, top-to-bottom	left-to-right	Default
background-color	I	*color*	#f	Standard
background-layer	I	*integer*	-1	Default
background-tile	I	#f, *external graphic*	#f	Standard
start-margin	I	*length-spec*	0pt	Standard
end-margin	I	*length-spec*	0pt	Standard

Table 18. Scroll flow object class characteristics

4.24 Multi-mode flow object class [12.6.28.2]

A multi-mode flow object is a flow object with two or more modes of presentation. The flow object can be switched between these modes of presentation in a system-dependent way.

NOTE: *An implementation might present a menu of the different modes. Alternatively, clicking on the formatted flow object might cycle through the modes.*

This flow object is inlined or displayed according to its content and mode of presentation.

The flow object has one port for each mode. The content of that port specifies the presentation in the corresponding mode. Initially, the flow object shall be displayed using the principal mode.

NOTE: *For example, an icon which when clicked causes a window to be popped up could be represented by a multi-mode flow object with two ports, the first containing an icon flow object and the second containing a vertical-scroll flow object. In this case, the* `principal-mode-simultaneous?:` *characteristic would be true.*

Characteristic	Inh.	Standard Values	Standard Init/Default	Dsssl-o
multi-modes	N	*list*	must be specified	Standard
principal-mode-simultaneous?	I	#f, #t	#f	Standard

Table 19. Multi-mode flow object class characteristics

4.25 Link flow object class [12.6.28.3]

A link flow object represents a hypertext link that can be interactively traversed typically by clicking on the areas representing the flow object and its content. A link has a single principal port, which can contain both inlined and displayed flow objects. Link flow objects can be nested and the innermost link is effective.

Characteristic	Inh.	Standard Values	Standard Init/Default	Dsssl-o
destination	N	#f, *address*, *list of addresses*	shall be specified	Standard

Table 20. Link flow object class characteristics

4.26 Marginalia flow object class [12.6.28.4]

The marginalia flow object class is used to contain flow objects whose resulting areas shall be attachment areas for the line in which the marginalia flow object occurs. The marginalia flow object has a single principal port which shall contain only inlined flow objects. A marginalia flow object shall have an ancestor flow object that is of class paragraph.

The behavior when there is more than one marginalia area attached to a single line is system-dependent.

Characteristic	Inh.	Standard Values	Standard Init/Default	Dsssl-o
marginalia-sep	I	*length-spec*	0pt	Standard
marginalia-side	I	start, end	start	Standard
marginalia-keep-with-previous?	I	#f, #t	#f	Standard

Table 21. Marginalia flow object class characteristics

5. Summary tables of characteristics

The following tables show standard values and minimum dsssl-o support for all characteristics that can be specified for the flow object classes in the DSSSL Online subset.

5.1 Inherited characteristics of dsssl-o flow object classes

Characteristic	Flow object classes	Standard Values	Std. Initial Value	Dsssl-o
align-leader?	leader	#f, #t	#t	Standard
alignment-point-offset	paragraph	#f, *number between 0 and 100*	50	Ignore
allowed-ligatures	character	*list of glyph-ids or chars*	the empty list	Default
asis-truncate-char	paragraph	#f, *char object*	#f	Ignore
asis-wrap-char	paragraph	#f, *char object*	#f	Ignore
asis-wrap-indent	paragraph	*length-spec*	#f	Ignore
background-color	box scroll table-cell	*color*	#f	Standard
background-layer	box scroll table-cell	*integer*	-1	Default
background-tile	scroll	#f, *external graphic*	#f	Standard
border-alignment	table-border	center, start, end, outside, inside	center	Default
border-omit-at-break?	table-border	#f, #t	#f	Standard
border-present?	table-border	#f, #t	#t	Standard
border-priority	table-border	*integer*	0	Default
bottom-margin	simple-page-sequence	*length*	0pt	Standard
box-border-alignment	box	center, outside, inside	outside	Default
box-corner-radius	box	*length-spec*	3pt	Ignore
box-corner-rounded	box	#f, #t, *list of corner specs*	#f	Default
box-open-end?	box	#f, #t	#f	Standard
box-size-after	box	*length*	4pt	Standard
box-size-before	box	*length*	8pt	Standard
box-type	box	border, background, both	border	Standard
cell-after-column-border	table-cell	#f, #t, *unlabeled sosofo*	#f	Standard
cell-after-column-margin	table-cell	*length*	0pt	Standard
cell-after-row-border	table-cell	#f, #t, *unlabeled sosofo*	#f	Standard

Characteristic	**Flow object classes**	**Standard Values**	**Std. Initial Value**	**Dsssl-o**
cell-after-row-margin	table-cell	*length*	0pt	Standard
cell-background?	table-cell	#f, #t	#f	Standard
cell-before-column-border	table-cell	#f, #t, *unlabeled sosofo*	#f	Standard
cell-before-column-margin	table-cell	*length*	0pt	Standard
cell-before-row-border	table-cell	#f, #t, *unlabeled sosofo*	#f	Standard
cell-before-row-margin	table-cell	*length*	0pt	Standard
cell-crossed	table-cell	#f, with, across, both	#f	Default
cell-row-alignment	table-cell	start, end, center	start	Standard
char-map	character	#f, *procedure*	#f	Default
color	box character external-graphic rule score sideline table-border	*color*	default color in Device Gray color space	Standard
country	character paragraph	#f, *ISO 3166 country code*	#f	Standard
display-alignment	aligned-column external-graphic rule table table-column	start, center, end, inside*, outside*	start	Partial
end-indent	aligned-column box external-graphic paragraph rule table table-column	*length-spec*	0pt	Standard
end-margin	scroll	*length-spec*	0pt	Standard
escapement-space-after	character	*inline-space*	inline-space of 0pt	Ignore
escapement-space-before	character	*inline-space*	inline-space of 0pt	Ignore
expand-tabs?	paragraph	#f, *integer greater than zero*	8	Standard
field-align	line-field	start, end, center	start	Default
field-width	line-field	*length-spec*	0pt	Standard
filling-direction	scroll	top-to-bottom, left-to-right, right-to-left	top-to-bottom	Default
first-line-align	paragraph	#f, *char object*	#f	Ignore
first-line-start-indent	paragraph	*length-spec*	0pt	Standard

Characteristic	Flow object classes	Standard Values	Std. Initial Value	Dsssl-o
float-out-line-numbers?	table-cell	#f, #t	#f	Default
float-out-marginalia?	table-cell	#f, #t	#f	Default
float-out-sidelines?	table-cell	#f, #t	#f	Default
font-family-name	character paragraph	#f, *string*	iso-serif	Standard
font-name	character paragraph	#f, *public identifier*	#f	Standard
font-posture	character paragraph	#f, not-applicable*, upright, oblique, back-slanted-oblique*, italic, back-slanted-italic*	upright	Partial
font-proportionate-width	character paragraph	#f, not-applicable*, ultra-condensed*, extra-condensed*, condensed, semi-condensed*, medium, semi-expanded*, expanded, extra-expanded*, ultra-expanded*	medium	Partial
font-size	character paragraph	*length*	10pt	Standard
font-structure	character paragraph	#f, not-applicable, solid, outline	solid	Default
font-weight	character paragraph	#f, not-applicable*, ultra-light*, extra-light*, light, semi-light*, medium, semi-bold*, bold, extra-bold*, ultra-bold*	medium	Partial
footer-margin	simple-page-sequence	*length*	0pt	Standard
glyph-alignment-mode	paragraph	base, center, top, bottom, font	font	Default
glyph-reorder-method	character	#f, *public identifier*, *list of public identifiers*	#f	Default
glyph-subst-method	character	#f, *public identifier*, *list of public identifiers*	#f	Default
glyph-subst-table	character	#f, *glyph-subst-table*, *list of glyph-subst-table*	#f	Default
hanging-punct?	paragraph	#f, #t	#f	Default
header-margin	simple-page-sequence	*length*	0pt	Standard
hyphenate?	character	#f, #t	#f	Default

Characteristic	Flow object classes	Standard Values	Std. Initial Value	Dsssl-o
hyphenation-char	paragraph	*character*	#\- (the hyphen character)	Ignore
hyphenation-exceptions	paragraph	*list of strings*	the empty list	Default
hyphenation-keep	paragraph	#f, spread, page, column	#f	Default
hyphenation-ladder-count	paragraph	#f, *integer greater than zero*	#f	Default
hyphenation-method	character paragraph	#f, *public identifier*	#f	Default
hyphenation-push-char-count	paragraph	*positive integer*	2	Ignore
hyphenation-remain-char-count	paragraph	*positive integer*	2	Ignore
ignore-record-end?	paragraph	#f, #t	#f	Standard
implicit-bidi-method	paragraph	#f, *public identifier*		Default
inhibit-line-breaks?	box character external-graphic leader line-field rule score	#f, #t	#f	Standard
inline-space-space	character	*inline-space*		Ignore
input-whitespace-treatment	character	preserve, collapse, ignore	preserve	Standard
justify-glyph-space-max-add	paragraph	*length-spec*	0pt	Ignore
justify-glyph-space-max-remove	paragraph	*length-spec*	0pt	Ignore
kern-mode	character	loose, normal, kern, tight, touch	normal	Default
kern?	character	#f, #t	#f	Default
language	character paragraph	#f, *ISO 639 language code*	#f	Standard
last-line-end-indent	paragraph	*length-spec*	0pt	Default
last-line-justify-limit	paragraph	*length-spec*	0	Ignore
last-line-quadding	paragraph	relative, start, end, spread-inside, spread-outside, page-inside, page-outside, center, justify	relative	Default

Characteristic	Flow object classes	Standard Values	Std. Initial Value	Dsssl-o
layer	box character external-graphic rule score sideline table-border	*integer*	0	Default
left-margin	simple-page-sequence	*length*	0pt	Standard
ligature?	character	#f, #t	#f	Default
line-breaking-method	paragraph	#f, *public identifier*	#f	Default
line-cap	box rule score sideline table-border table-cell	butt, round, square	butt	Default
line-composition-method	paragraph	#f, *public identifier*	#f	Default
line-dash	box rule score sideline table-border table-cell	*list of lengths*	list containing the length 0pt	Default
line-join	box table-border	miter, round, bevel	miter	Default
line-miter-limit	box table-border	*number*	10	Default
line-number-sep	paragraph	*length-spec*		Ignore
line-number-side	paragraph	start, end, spread-inside, spread-outside, page-inside, page-outside		Ignore
line-number	paragraph	#f, *unlabeled sosofo*	#f	Ignore
line-repeat	box rule score sideline table-border table-cell	*integer*	1	Default
line-sep	box rule score sideline table-border table-cell	*length*	1pt	Ignore
line-spacing-priority	paragraph	force, *integer*	0	Default
line-spacing	paragraph	*length-spec*	12pt	Standard

Characteristic	**Flow object classes**	**Standard Values**	**Std. Initial Value**	**Dsssl-o**
line-thickness	box rule score sideline table-border table-cell	*length*	1pt	Standard
lines	paragraph	wrap, asis, asis-wrap*, asis-truncate*, none	wrap	Partial
marginalia-keep-with-previous?	marginalia	#f, #t	#f	Standard
marginalia-sep	marginalia	*length-spec*	0pt	Standard
marginalia-side	marginalia	start, end	start	Standard
min-leader-repeat	leader	*integer greater than 0*	1	Standard
min-leading	paragraph	#f, *length-spec*	#f	Standard
min-post-line-spacing	paragraph	#f, *length-spec*	#f	Default
min-pre-line-spacing	paragraph	#f, *length-spec*	#f	Default
numbered-lines?	paragraph	#f, #t	#t	Ignore
orphan-count	paragraph	*positive integer*	2	Default
page-height	simple-page-sequence	*length*	system-dependent	Standard
page-width	simple-page-sequence	*length*	system-dependent	Standard
position-point-shift	character rule	*length-spec*	0pt	Standard
principal-mode-simultaneous?	multi-mode	#f, #t	#f	Standard
quadding	paragraph	start, end, spread-inside*, spread-outside*, page-inside*, page-outside*, center, justify*	start	Partial
right-margin	simple-page-sequence	*length*	0pt	Standard
score-spaces?	score	#f, #t	#t	Standard
sideline-sep	sideline	*length-spec*		Standard
sideline-side	sideline	start, end, both, spread-inside*, spread-outside*, page-inside*, page-outside*		Partial
span-weak?	box external-graphic paragraph rule table	#f, #t	#f	Default

Characteristic	Flow object classes	Standard Values	Std. Initial Value	Dsssl-o
span	box external-graphic paragraph rule table	*integer greater than zero*	1	Default
start-indent	aligned-column box external-graphic paragraph rule table table-column	*length-spec*	0pt	Standard
start-margin	scroll	*length-spec*	0pt	Standard
table-auto-width-method	table	#f, *public identifier*	#f	Standard
table-border	table	#f, #t, *unlabeled sosofo*	#f	Standard
table-corner-radius	table	*length-spec*	3pt	Ignore
table-corner-rounded	table	#f, #t, *list of x-y pairs*	#f	Default
table-part-omit-middle-footer?	table-part	#f, #t	#f	Default
table-part-omit-middle-header?	table-part	#f, #t	#f	Default
top-margin	simple-page-sequence	*length*	0pt	Standard
truncate-leader?	leader	#f, #t	#f	Standard
widow-count	paragraph	*positive integer*	2	Default
writing-mode	aligned-column box character external-graphic line-field paragraph rule scroll simple-page-sequence table	left-to-right, right-to-left, top-to-bottom	left-to-right	Default

Table 22. Inherited characteristics

5.2 Non-inherited characteristics of dsssl-o flow object classes

Characteristic	**Flow object classes**	**Standard Values**	**Standard Default**	**Dsssl-o**
after-column-border	table	#f, #t, *unlabeled sosofo*	value of the table-border characteristic	Standard
after-row-border	table	#f, #t, *unlabeled sosofo*	value of the table-border characteristic	Standard
before-column-border	table	#f, #t, *unlabeled sosofo*	value of the table-border characteristic	Standard
before-row-border	table	#f, #t, *unlabeled sosofo*	value of the table-border characteristic	Standard
break-after-priority	box external-graphic leader line-field rule	*integer*	0	Ignore
break-after-priority	character	*integer*	the value of the break-after-priority character property of the char: characteristic	Ignore
break-after	aligned-column box display-group external-graphic paragraph rule table table-part	#f, page, page-region*, column*, column-set*	#f	Partial
break-before-priority	box external-graphic leader line-field rule	*integer*	0	Ignore
break-before-priority	character	*integer*	the value of the break-before-priority character property of the char: characteristic	Ignore
break-before	aligned-column box display-group external-graphic paragraph rule table table-part	#f, page, page-region*, column*, column-set*	#f	Partial
center-footer	simple-page-sequence	*unlabeled sosofo*	empty sosofo	Standard

Characteristic	**Flow object classes**	**Standard Values**	**Standard Default**	**Dsssl-o**
center-header	simple-page-sequence	*unlabeled sosofo*	empty sosofo	Standard
char	character	*character*	value of the current node's char property, if there is one, else the value of applying the char-map characteristic, if there is one	Standard
coalesce-id	display-group	#f, *string*	#f	Ignore
column-number	table-cell	*integer greater than zero*	current column number	Standard
column-number	table-column	*integer greater than zero*	1 plus the column-number of the previous table column flow object, if any, else 1	Standard
destination	link	#f, *address*, *list of addresses*	shall be specified	Standard
display?	box external-graphic	#f, #t	#f	Standard
drop-after-line-break?	character	#f, #t	the value of the drop-after-line-break? character property of the char: characteristic	Ignore
drop-unless-before-line-break?	character	#f, #t	the value of the drop-unless-before-line-break? character property of the char: characteristic	Ignore
ends-row?	table-cell	#f, #t	#f	Standard
entity-system-id	external-graphic	#f, *system identifier*	shall be specified	Standard
escapement-direction	external-graphic	top-to-bottom, left-to-right, right-to-left	the value of writing-mode	Default
glyph-id	character	#f, *glyph-id*	glyph-id of the nominal glyph specified using the standard-chars architectural form, else #f	Standard
input-tab?	character	#f, #t	the value of the input-tab? character property of the char: characteristic if the char: characteristic was not explicitly specified, and otherwise #f	Standard

Characteristic	**Flow object classes**	**Standard Values**	**Standard Default**	**Dsssl-o**
input-whitespace?	character	#f, #t	the value of the input-white-space? character property of the char: characteristic if the char: characteristic was not explicitly specified, and otherwise #f	Default
keep-with-next?	aligned-column box display-group external-graphic paragraph rule table table-part	#f, #t	#f	Standard
keep-with-previous?	aligned-column box display-group external-graphic paragraph rule table table-part	#f, #t	#f	Standard
keep	aligned-column box display-group external-graphic paragraph rule table table-part	#f, page, column-set*, column*, #t*	#f	Partial
left-footer	simple-page-sequence	*unlabeled sosofo*	empty sosofo	Standard
left-header	simple-page-sequence	*unlabeled sosofo*	empty sosofo	Standard
length	leader rule	*length-spec*	determined by context	Standard
math-class	character	ordinary, operator, binary, relation, opening, closing, punctuation, inner, space	the value of the math-class character property of the char: characteristic	Ignore
math-font-posture	character	#f, not-applicable*, upright, oblique, back-slanted-oblique*, italic, back-slanted-italic*	the value of the math-font-posture character property of the char: characteristic	Partial
max-height	external-graphic	*length-spec*		Standard
max-width	external-graphic	*length-spec*		Standard

Characteristic	Flow object classes	Standard Values	Standard Default	Dsssl-o
may-violate-keep-after?	aligned-column box display-group external-graphic paragraph rule table table-part	#f, #t	#f	Default
may-violate-keep-before?	aligned-column box display-group external-graphic paragraph rule table table-part	#f, #t	#f	Default
multi-modes	multi-mode	*list*	must be specified	Standard
n-columns-spanned	table-cell table-column	*integer greater than zero*	1	Standard
n-rows-spanned	table-cell	*integer greater than zero*	1	Standard
notation-system-id	external-graphic	*system identifier*	shall be specified	Standard
orientation	rule	horizontal, vertical, escapement, line-progression	shall be specified	Standard
position-point-x	external-graphic	*length-spec*	0 if writing-mode is left-to-right or right-to-left	Standard
position-point-y	external-graphic	*length-spec*	0 if writing-mode is top-to-bottom	Standard
position-preference	aligned-column box display-group external-graphic paragraph rule table	#f, top, bottom	#f	Ignore
punct?	character	#f, #t	the value of the punct? character property of the char: characteristic	Ignore
record-end?	character	#f, #t	the value of the record-end? character property of the char: characteristic	Standard
right-footer	simple-page-sequence	*unlabeled sosofo*	empty sosofo	Standard
right-header	simple-page-sequence	*unlabeled sosofo*	empty sosofo	Standard

Characteristic	Flow object classes	Standard Values	Standard Default	Dsssl-o
scale	external-graphic	*number*, *list of two numbers*, max, max-uniform	max-uniform	Standard
script	character	#f, *formal public identifier*	the value of the script character property of the char: characteristic	Ignore
space-after	aligned-column box display-group external-graphic paragraph rule table table-part	*display space*	no space after	Standard
space-before	aligned-column box display-group external-graphic paragraph rule table table-part	*display space*	no space before	Standard
space?	character	#f, #t	the value of the space? character property of the char: characteristic	Standard
starts-row?	table-cell	#f, #t	#f	Standard
stretch-factor	character	*number*	1	Default
table-width	table	#f, *length-spec*	display-size less applicable indent	Standard
type	score	before, through, after, *length-spec*, *character*	shall be specified	Standard
width	table-column	*length-spec*	shall be specified unless table-auto-width feature is present	Standard

Table 23. Non-inherited characteristics

6. Core expression language

6.1 Syntax

[120] expression = *primitive-expression* | *derived-expression*

[121] primitive-expression = *variable-reference* | *literal* | *procedure-call* | *conditional*

[122] variable-reference = *variable*

[123] variable = *identifier*

[124] literal = *quotation* | *self-evaluating*

[125] quotation = `"'"`*datum* | `"(""quote"` *datum*`")"`

[126] datum = *simple-datum* | *list*

[127] simple-datum = *boolean* | *number* | *character* | *string* | *symbol* | *keyword* | *glyph-identifier*

[128] list = `"("`*datum**`")"` | `"'"`*datum*

[129] self-evaluating = *boolean* | *number* | *character* | *string* | *keyword* | *glyph-identifier*

[130] procedure-call = `"("`*operator operand**`")"`

[131] operator = *expression*

[132] operand = *expression*

[133] conditional = `"(""if"` *test consequent alternate*`")"`

[134] test = *expression*

[135] consequent = *expression*

[136] alternate = *expression*

[137] derived-expression = *cond-expression* | *case-expression* | *and-expression* | *or-expression*

[138] cond-expression = `"(""cond"` *cond-clause*+`")"` | `"(""cond"` *cond-clause** `"(""else"` *expression*`")"")"`

[139] cond-clause = `"("`*test expression*`")"`

[140] case-expression = `"(""case"` *key case-clause*+`")"` | `"(""case"` *key case-clause** `"(""else"` *expression*`")"")"`

[141] key = *expression*

[142] case-clause = `"(""("`*datum**`")"` *expression*`")"`

[143] and-expression = `"(""and"` *test**`")"`

[144] or-expression = `"(""or"` *test**`")"`

[145] definition = `"(""define"` *variable expression*`")"`

6.2 Procedures

(`not` *obj*)

(`boolean?` *obj*)

(`equal?` obj_1 obj_2)

(`null?` *obj*)

(`list?` *obj*)

(`list` *obj* ...)

(`length` *list*)

(append *list* ...)

(reverse *list*)

(list-tail *list k*)

(list-ref *list k*)

(member *obj list*)

(symbol? *obj*)

(keyword? *obj*)

(quantity? *obj*)

(number? *obj*)

(real? *obj*)

(integer? *obj*)

(= q_1 q_2 q_3 ...)

(< q_1 q_2 q_3 ...)

(> q_1 q_2 q_3 ...)

(<= q_1 q_2 q_3 ...)

(>= q_1 q_2 q_3 ...)

(max q_1 q_2 ...)

(min q_1 q_2 ...)

(+ q_1 ...)

(* q_1 ...)

(- q_1 q_2)

(- *q*)

(/ q_1 q_2)

(/ *q*)

(abs *q*)

(quotient n_1 n_2)

(remainder n_1 n_2)

(modulo n_1 n_2)

(floor *x*)

(ceiling *x*)

(truncate *x*)

(round *x*)

(number->string *number*)

(number->string *number radix*)

(string->number *string*)

(string->number *string radix*)

(char? *obj*)

(char=? $char_1$ $char_2$)

(char-property *symbol char*)

(char-property *symbol char obj*)

`(string?` *`obj`*`)`

`(string` *`char`* ...`)`

`(string-length` *`string`*`)`

`(string-ref` *`string k`*`)`

`(string=?` $string_1$ $string_2$`)`

`(substring` *`string start end`*`)`

`(string-append` *`string`* ...`)`

`(procedure?` *`obj`*`)`

`(apply` *`proc args`*`)`

`(external-procedure` *`string`*`)`

`(time)`

`(time->string` *`k`*`)`

`(time->string` *`k boolean`*`)`

`(error` *`string`*`)`

6.3 DSSSL Online extensions

In addition to the Core Expression Language, DSSSL Online is now considered to include lambda, #!key, and let (including letrec, let*, and named let). See the full DSSSL standard for details.

7. Core query language

The core query language is a subset of SDQL, the Standard Document Query Language. In addition to the procedures defined in the core query language subclause, the current-node, node-list-empty?, node-list?, parent, and node-list-error procedures are allowed in the subset. This subset is designed so that a node-list never contains more than one node, and so that any node that it does contain is always of type element.

In the following procedures, the argument that is of type node-list can be omitted, and defaults to `(current-node)`. *`osnl`* (optional singleton node-list) denotes an argument that must be a node-list containing zero or one nodes.

7.1 Navigation

`(ancestor` *`string osnl`*`)`

Returns a node-list containing the nearest ancestor of *`osnl`* with gi equal to *`string`*, or an empty node-list if there is no such ancestor or if *`osnl`* is empty.

`(gi` *`osnl`*`)`

Returns the value of the `gi` property of the node contained in *`osnl`* or #f if *`osnl`* is empty or if *`osnl`* has no `gi` property or a null `gi` property.

`(id` *`osnl`*`)`

Returns the value of the `id` property of the node contained in *`osnl`* or #f if *`osnl`* is empty or if *`osnl`* has no `id` property or a null `id` property.

`(element` *`string snl`*`)`

Returns a singleton node-list returning the element in the same grove as *`snl`* whose unique identifier is *`string`*, if there is such an element, and otherwise returns the empty node-list. This procedures requires the `random-access` feature.

7.2 Counting

`(child-number` *`snl`*`)`

Returns the child number of *`snl`*. The **child number** of an element is the number of element siblings of the current element that are before or equal in tree order to the current element and that have the same generic identifier as the current element.

`(ancestor-child-number` *`string snl`*`)`

Returns the child number of the nearest ancestor of *`snl`* whose generic identifier is *`string`*, or #f if there is no such ancestor.

`(hierarchical-number` *`list snl`*`)`

Returns a list of non-negative integers with the same number of members as *`list`*. *`list`* shall be a list of strings. The last member is the child number of the nearest ancestor of *`snl`* whose generic identifier is equal to the last member of *`list`*; the next to last member is the child number of the nearest ancestor of that element whose generic identifier is equal to the next to last member; and so on for each member of *`list`*.

`(hierarchical-number-recursive` *`string snl`*`)`

Returns a list of non-negative integers. The last member of list is the child number of the nearest ancestor of *`snl`* element whose generic identifier is equal to *`string`*, the next to last member is the child number of the nearest ancestor of that element whose generic identifier is equal to *`string`*, and so on for each ancestor of the current element with generic identifier equal to *`string`*. Note that the length of this list is the nesting level of *`string`*.

`(element-number` *`snl`*`)`

Returns the number of elements before or equal to *`snl`* with the same gi as *`snl`*.

`(element-number-list` *`list snl`*`)`

Returns a list of non-negative integers, one for each member of *`list`*, which shall be a list of strings, where the `i`-th integer is the number of elements that:

- are before or equal to *`snl`*
- have a generic identifier equal to the `i`-th member of *`list`*, and
- if `i` is greater than 1, are after the last element before *`snl`* whose generic identifier is equal to the `i-1`-th member of *`list`*.

NOTE: *In effect the counter for each argument is reset at the start of the element referred to by the previous argument.*

NOTE: *An element is considered to be after its parent.*

NOTE: *This procedure could be used to number footnotes sequentially within a chapter (by using the last number in the list). It could also be used to number headings in a DTD which lacks container elements.*

7.3 Accessing attribute values

In the following procedures attribute values are represented as strings by applying the `data` procedure to the `attribute-assignment` node.

`(attribute-string` *`string osnl`*`)`

Returns a string representation of the attribute with name equal to *`string`* of *`osnl`*, or #f if *`osnl`* has no such attribute or the attribute is implied or *`osnl`* is empty.

`(inherited-attribute-string` *`string osnl`*`)`

Returns a string representation of the attribute with name equal to *`string`* of *`osnl`* or of the nearest ancestor of *`osnl`* for which this attribute is present and not implied, or #f if there is no such element or *`osnl`* is empty.

`(inherited-element-attribute-string` *`string1 string2 osnl`* `)`

Returns a string representation of the attribute with name equal to *`string2`* of the nearest ancestor of *`osnl`* whose generic identifier is equal to *`string1`* and for which this attribute is present and not implied, or #f if there is no such element or *`osnl`* is empty.

7.4 Testing current location

`(first-sibling?` *`snl`*`)`

Returns #t if *`snl`* has no preceding sibling that is an element with the same generic identifier as itself, and otherwise returns #f.

`(absolute-first-sibling?` *`snl`*`)`

Returns #t if *`snl`* has no preceding sibling that is an element, and otherwise returns #f.

`(last-sibling?` *`snl`*`)`

Returns #t if *`snl`* has no following sibling that is an element with the same generic identifier as itself, and otherwise returns #f.

`(absolute-last-sibling?` *`snl`*`)`

Returns #t if *`snl`* has no following sibling that is an element, and otherwise returns #f.

`(have-ancestor?` *`obj snl`*`)`

`obj` shall be either a string or a list of strings. If *`obj`* is a string, then `have-ancestor?` returns #t if *`snl`* has an ancestor with a generic identifier that matches that string, and otherwise returns #f. If *`obj`* is a list of strings, then `have-ancestor?` returns #t if *`snl`* has an ancestor with generic identifier equal to the last member of *`obj`*, which itself has an ancestor with generic identifier equal to the next to last member of *`obj`*, and so on for each member, and otherwise returns #f.

7.5 Entities and notations

`snl` here determines the document in which to find the entity.

`(entity-public-id` *`string snl`*`)`

Returns the value of the public-id property of the value of the external-id property of the general entity whose name is *`string`* in the governing document type of the same grove as *`snl`*, or #f if there is no such entity or the entity has a null value for the external-id property, or the external-id has a null value for the public-id property.

`(entity-system-id` *`string snl`*`)`

Returns the value of the system-id property of the value of the external-id property of the general entity whose name is *`string`* in the governing document type of the same grove as *`snl`*, or #f if there is no such entity or the entity has a null value for the external-id property, or the external-id has a null value for the system-id property.

`(entity-generated-system-id` *`string snl`*`)`

Returns the value of the generated-system-id property of the value of the external-id property of the general entity whose name is *`string`* in the governing document type of the same grove as *snl*, or #f if there is no such entity or the entity has a null value for the external-id property, or the external-id has a null value for the generated-system-id property.

`(entity-text` *string snl*`)`

Returns the value of the text property of the general entity whose name is *string* in the governing document type of the same grove as *snl*, or #f if there is no such entity or the entity has a null value for the text property.

`(entity-notation` *string snl*`)`

Returns the value of the notation-name property of the general entity whose name is *string* in the governing document type of the same grove as *snl*, or #f if there is no such entity or the entity has a null value for the notation-name property.

`(entity-attribute-string` *string1 string2 snl*`)`

Returns a string representation of the value of the attribute named *string2* of the general entity whose name is *string1* in the governing document type of the same grove as *snl*, or #f if there is no such entity or the entity has no such attribute or the attribute is implied.

`(entity-type` *string snl*`)`

Returns the value of the entity-type property of the general entity whose name is *string* in the governing document type of the same grove as *snl*, or #f if there is no such entity or the entity has a null value for the entity-type property.

`(notation-public-id` *string snl*`)`

Returns the value of the public-id property of the value of the external-id property of the general notation whose name is *string* in the governing document type of the same grove as *snl*, or #f if there is no such notation or the external-id has a null value for the public-id property.

`(notation-system-id` *string snl*`)`

Returns the value of the system-id property of the value of the external-id property of the general notation whose name is *string* in the governing document type of the same grove as *snl*, or #f if there is no such notation or the external-id has a null value for the system-id property.

`(notation-generated-system-id` *string snl*`)`

Returns the value of the generated-system-id property of the value of the external-id property of the general notation whose name is *string* in the governing document type of the same grove as *snl*, or #f if there is no such notation or the external-id has a null value for the generated-system-id property.

7.6 Name normalization

`(general-name-normalize` *string snl*`)`

Returns *string* transformed using the general namecase substitution string normalization rule of the grove in which *snl* occurs. This could be defined as follows:

```
(define (general-name-normalize string snl)
  (named-node-list-normalize string
        (node-property 'elements (grove-root snl))))
```

`(entity-name-normalize` *string snl*`)`

Returns *string* transformed using the entity namecase substitution string normalization rule of the grove in which *snl* occurs. This could be defined as follows:

```
(define (entity-name-normalize string snl)
  (named-node-list-normalize string
        (node-property 'entities (grove-root snl))))
```

Compilation by Jon Bosak, Sun Microsystems