

HTML Parameter Reference

Norman Walsh

Copyright © 1999, 2000, 2001 Norman Walsh

COLLABORATORS

	TITLE : HTML Parameter Reference		
ACTION	NAME	DATE	SIGNATURE
WRITTEN BY	Norman Walsh	March 28, 2025	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	Admonitions	1
1.1	admon.graphics.extension	1
1.2	admon.graphics.path	1
1.3	admon.graphics	1
1.4	admon.style	2
2	Callouts	3
2.1	callout.defaultcolumn	3
2.2	callout.graphics.extension	3
2.3	callout.graphics.number.limit	3
2.4	callout.graphics.path	4
2.5	callout.graphics	4
2.6	callout.list.table	4
2.7	callout.unicode.number.limit	5
2.8	callout.unicode.start.character	5
2.9	callout.unicode	5
2.10	callouts.extension	5
3	EBNF	7
3.1	ebnf.table.bgcolor	7
3.2	ebnf.table.border	7
4	ToC/LoT/Index Generation	8
4.1	annotate.toc	8
4.2	autotoc.label.separator	8
4.3	process.source.toc	8
4.4	process.empty.source.toc	9
4.5	bridgehead.in.toc	9
4.6	manual.toc	9
4.7	toc.list.type	9
4.8	toc.section.depth	10
4.9	generate.toc	10
4.10	generate.section.toc.level	11
4.11	generate.index	11

5	Extensions	12
5.1	linenumbering.everyNth	12
5.2	linenumbering.extension	12
5.3	linenumbering.separator	12
5.4	linenumbering.width	13
5.5	tablecolumns.extension	13
5.6	textinsert.extension	13
5.7	use.extensions	13
6	Automatic labelling	15
6.1	chapter.autolabel	15
6.2	appendix.autolabel	15
6.3	part.autolabel	15
6.4	preface.autolabel	16
6.5	qandadiv.autolabel	16
6.6	section.autolabel	16
6.7	section.label.includes.component.label	16
6.8	label.from.part	17
7	HTML	18
7.1	html.base	18
7.2	html.stylesheet.type	18
7.3	html.stylesheet	18
7.4	use.id.as.filename	19
7.5	css.decoration	19
7.6	spacing.paras	19
7.7	emphasis.propagates.style	20
7.8	para.propagates.style	20
7.9	phrase.propagates.style	20
7.10	html.longdesc	20
7.11	html.longdesc.link	21
7.12	make.valid.html	21
7.13	html.cleanup	21
7.14	draft.watermark.image	22
7.15	generate.id.attributes	22
7.16	generate.meta.abstract	22

8	XSLT Processing	24
8.1	rootid	24
8.2	suppress.navigation	24
8.3	suppress.header.navigation	24
8.4	suppress.footer.navigation	25
8.5	header.rule	25
8.6	footer.rule	25
9	Meta/*Info	26
9.1	inherit.keywords	26
9.2	make.single.year.ranges	26
9.3	make.year.ranges	26
9.4	author.othername.in.middle	27
9.5	generate.legalnotice.link	27
10	Reference Pages	28
10.1	funcsynopsis.decoration	28
10.2	funcsynopsis.style	28
10.3	function.parens	28
10.4	refentry.generate.name	29
10.5	refentry.generate.title	29
10.6	refentry.xref.manvolnum	29
10.7	citerefentry.link	29
10.8	refentry.separator	30
11	Tables	31
11.1	default.table.width	31
11.2	nominal.table.width	31
11.3	table.borders.with.css	31
11.4	table.border.style	32
11.5	table.border.thickness	32
11.6	table.border.color	32
11.7	html.cellspacing	32
11.8	html.cellpadding	33
12	QAndASet	34
12.1	qanda.defaultlabel	34
12.2	qanda.inherit.numeration	34

13 Linking	35
13.1 target.database.document	35
13.2 targets.filename	35
13.3 collect.xref.targets	36
13.4 olink.base.uri	36
13.5 use.local.olink.style	36
13.6 current.docid	37
13.7 olink.doctitle	37
13.8 link.mailto.url	37
13.9 ulink.target	38
13.10olink.fragid	38
13.11olink.outline.ext	38
13.12olink.pubid	38
13.13olink.sysid	39
13.14olink.resolver	39
14 Bibliography	40
14.1 biblioentry.item.separator	40
14.2 bibliography.collection	40
14.3 bibliography.numbered	41
15 Glossary	42
15.1 glossterm.auto.link	42
15.2 firstterm.only.link	42
15.3 glossary.collection	42
16 Miscellaneous	46
16.1 formal.procedures	46
16.2 formal.title.placement	46
16.3 runinhead.default.title.end.punct	46
16.4 runinhead.title.end.punct	47
16.5 show.comments	47
16.6 show.revisionflag	47
16.7 shade.verbatim	48
16.8 shade.verbatim.style	48
16.9 punct.honorific	48
16.10segmentedlist.as.table	48
16.11variablelist.as.table	49
16.12tex.math.in.alt	49
16.13tex.math.file	50

16.14	tex.math.delims	50
16.15	pixels.per.inch	50
16.16	points.per.em	50
16.17	use.svg	51
16.18	use.role.as.xrefstyle	51
16.19	menuchoice.separator	51
16.20	menuchoice.menu.separator	51
17	Graphics	53
17.1	graphic.default.extension	53
17.2	nominal.image.width	53
17.3	nominal.image.depth	54
17.4	use.embed.for.svg	54
17.5	make.graphic.viewport	54
18	Chunking	55
18.1	chunker.output.cdata-section-elements	55
18.2	chunker.output.doctype-public	55
18.3	chunker.output.doctype-system	56
18.4	chunker.output.encoding	56
18.5	chunker.output.indent	56
18.6	chunker.output.media-type	57
18.7	chunker.output.method	57
18.8	chunker.output.omit-xml-declaration	57
18.9	chunker.output.standalone	58
18.10	saxon.character.representation	58
18.11	html.ext	59
18.12	html.extra.head.links	59
18.13	root.filename	59
18.14	base.dir	59
18.15	generate.manifest	60
18.16	manifest	60
18.17	chunk.toc	60
18.18	chunk.tocs.and.lots	60
18.19	chunk.section.depth	61
18.20	chunk.first.sections	61
18.21	chunk.quietly	61
18.22	navig.graphics	62
18.23	navig.graphics.extension	62
18.24	navig.graphics.path	62
18.25	navig.showtitles	62

19 Profiling	64
19.1 profile.arch	64
19.2 profile.condition	64
19.3 profile.conformance	65
19.4 profile.lang	65
19.5 profile.os	65
19.6 profile.revision	66
19.7 profile.revisionflag	66
19.8 profile.role	66
19.9 profile.security	67
19.10profile.userlevel	67
19.11profile.vendor	67
19.12profile.attribute	68
19.13profile.value	68
19.14profile.separator	68
20 HTML Help	69
20.1 htmlhelp.encoding	69
20.2 htmlhelp.autolabel	69
20.3 htmlhelp.chm	69
20.4 htmlhelp.default.topic	70
20.5 htmlhelp.hhp	70
20.6 htmlhelp.hhc	70
20.7 htmlhelp.hhk	71
20.8 htmlhelp.hhp.tail	71
20.9 htmlhelp.hhp.window	71
20.10htmlhelp.enumerate.images	71
20.11htmlhelp.force.map.and.alias	72
20.12htmlhelp.map.file	72
20.13htmlhelp.alias.file	72
20.14htmlhelp.hhc.section.depth	72
20.15htmlhelp.hhc.show.root	73
20.16htmlhelp.hhc.folders.instead.books	73
20.17htmlhelp.hhc.binary	73
20.18htmlhelp.title	74
20.19htmlhelp.show.menu	74
20.20htmlhelp.show.advanced.search	74
20.21htmlhelp.show.favorites	74
20.22htmlhelp.button.hideshow	75

20.23htmlhelp.button.back	75
20.24htmlhelp.button.forward	75
20.25htmlhelp.button.stop	75
20.26htmlhelp.button.refresh	76
20.27htmlhelp.button.home	76
20.28htmlhelp.button.home.url	76
20.29htmlhelp.button.options	77
20.30htmlhelp.button.print	77
20.31htmlhelp.button.locate	77
20.32htmlhelp.button.jump1	77
20.33htmlhelp.button.jump1.url	78
20.34htmlhelp.button.jump1.title	78
20.35htmlhelp.button.jump2	78
20.36htmlhelp.button.jump2.url	78
20.37htmlhelp.button.jump2.title	79
20.38htmlhelp.button.next	79
20.39htmlhelp.button.prev	79
20.40htmlhelp.button.zoom	80
20.41htmlhelp.use.hhk	80
20.42htmlhelp.only	80

A The Stylesheet	81
-------------------------	-----------

Introduction

This is technical reference documentation for the DocBook XSL Stylesheets; it documents (some of) the parameters, templates, and other elements of the stylesheets.

This reference describes each of the HTML Stylesheet parameters. These are the “easily customizable” parts of the stylesheet. If you want to specify an alternate value for one or more of these parameters, you can do so in a “driver” stylesheet.

For example, if you want to change the `html.stylesheet` to `reference.css`, you might create a driver stylesheet like this:

```
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
               version='1.0'>

  <xsl:import href="http://docbook.sourceforge.net/release/xsl/snapshot/html/docbook.xsl"/>

  <xsl:param name="html.stylesheet">reference.css</xsl:param>

</xsl:stylesheet>
```

Naturally, you have to change the `href` attribute on `<xsl:import>` to point to `docbook.xsl` on your system. (Or `chunk.xsl`, if you’re using chunking.)

This is not intended to be “user” documentation. It is provided for developers writing customization layers for the stylesheets, and for anyone who’s interested in “how it works”.

Although I am trying to be thorough, this documentation is known to be incomplete. Don’t forget to read the source, too :-)

Chapter 1

Admonitions

1.1 admon.graphics.extension

admon.graphics.extension — Extension for admonition graphics

Synopsis

```
<xsl:param name="admon.graphics.extension" select="'.png'"/>
```

Description

Sets the extension to use on admonition graphics.

1.2 admon.graphics.path

admon.graphics.path — Path to admonition graphics

Synopsis

```
<xsl:param name="admon.graphics.path">images/</xsl:param>
```

Description

Sets the path, probably relative to the directory where the HTML files are created, to the admonition graphics.

1.3 admon.graphics

admon.graphics — Use graphics in admonitions?

Synopsis

```
<xsl:param name="admon.graphics" select="0"/>
```

Description

If true (non-zero), admonitions are presented in an alternate style that uses a graphic. Default graphics are provided in the distribution.

1.4 admon.style

admon.style — CSS style attributes for admonitions

Synopsis

```
<xsl:param name="admon.style"> <xsl:text>margin-left: 0.5in; margin-right: 0.5in;</xsl:text> </xsl:param>
```

Description

Specifies the value of the `STYLE` attribute that should be added to admonitions.

Chapter 2

Callouts

2.1 callout.defaultcolumn

callout.defaultcolumn — Indicates what column callouts appear in by default

Synopsis

```
<xsl:param name="callout.defaultcolumn" select="'60'"/>
```

Description

If a callout does not identify a column (for example, if it uses the `linerange` unit), it will appear in the default column.

2.2 callout.graphics.extension

callout.graphics.extension — Extension for callout graphics

Synopsis

```
<xsl:param name="callout.graphics.extension" select="'.png'"/>
```

Description

Sets the extension to use on callout graphics.

2.3 callout.graphics.number.limit

callout.graphics.number.limit — Number of the largest callout graphic

Synopsis

```
<xsl:param name="callout.graphics.number.limit" select="'10'"/>
```

Description

If *callout.graphics* is non-zero, graphics are used to represent callout numbers. The value of *callout.graphics.number.limit* is the largest number for which a graphic exists. If the callout number exceeds this limit, the default presentation "(nnn)" will always be used.

2.4 callout.graphics.path

callout.graphics.path — Path to callout graphics

Synopsis

```
<xsl:param name="callout.graphics.path" select="'images/callouts/'"/>
```

Description

Sets the path, probably relative to the directory where the HTML files are created, to the callout graphics.

2.5 callout.graphics

callout.graphics — Use graphics for callouts?

Synopsis

```
<xsl:param name="callout.graphics" select="'1'"/>
```

Description

If non-zero, callouts are presented with graphics (e.g., reverse-video circled numbers instead of "(1)", "(2)", etc.). Default graphics are provided in the distribution.

2.6 callout.list.table

callout.list.table — Present callout lists using a table?

Synopsis

```
<xsl:param name="callout.list.table" select="'1'"/>
```

Description

The default presentation of *CalloutLists* uses an HTML DL. Some browsers don't align DLs very well if *callout.graphics* are used. With this option turned on, *CalloutLists* are presented in an HTML TABLE, which usually results in better alignment of the callout number with the callout description.

2.7 callout.unicode.number.limit

callout.unicode.number.limit — Number of the largest callout graphic

Synopsis

```
<xsl:param name="callout.unicode.number.limit" select="'10'"/>
```

Description

If *callout.unicode* is non-zero, unicode characters are used to represent callout numbers. The value of *callout.unicode.number.limit* is the largest number for which a unicode character exists. If the callout number exceeds this limit, the default presentation "(nnn)" will always be used.

2.8 callout.unicode.start.character

callout.unicode.start.character — First Unicode character to use, decimal value.

Synopsis

```
<xsl:param name="callout.unicode.start.character" select="10102"/>
```

Description

If *callout.graphics* is zero and *callout.unicode* is non-zero, unicode characters are used to represent callout numbers. The value of *callout.unicode.start.character* is the decimal unicode value used for callout number one. Currently, only 10102 is supported in the stylesheets for this parameter.

2.9 callout.unicode

callout.unicode — Use Unicode characters rather than images for callouts.

Synopsis

```
<xsl:param name="callout.unicode" select="0"/>
```

Description

The stylesheets can use either an image of the numbers one to ten, or the single Unicode character which represents the numeral, in white on a black background. Use this to select the Unicode character option.

2.10 callouts.extension

callouts.extension — Enable the callout extension

Synopsis

```
<xsl:param name="callouts.extension" select="''1''"/>
```

Description

The callouts extension processes `areaset` elements in `ProgramListingCO` and other text-based callout elements.

Chapter 3

EBNF

3.1 `ebnf.table.bgcolor`

`ebnf.table.bgcolor` — Background color for EBNF tables

Synopsis

```
<xsl:param name="ebnf.table.bgcolor" select="'#F5DCB3'"/>
```

Description

Sets the background color for EBNF tables. No `bgcolor` attribute is output if `ebnf.table.bgcolor` is set to the null string. The default value matches the value used in recent online versions of the W3C's XML Spec productions.

3.2 `ebnf.table.border`

`ebnf.table.border` — Selects border on EBNF tables

Synopsis

```
<xsl:param name="ebnf.table.border" select="1"/>
```

Description

Selects the border on EBNF tables. If non-zero, the tables have borders, otherwise they don't.

Chapter 4

ToC/LoT/Index Generation

4.1 `annotate.toc`

`annotate.toc` — Annotate the Table of Contents?

Synopsis

```
<xsl:param name="annotate.toc" select="1"/>
```

Description

If true, TOCs will be annotated. At present, this just means that the `RefPurpose` of `RefEntry` TOC entries will be displayed.

4.2 `autotoc.label.separator`

`autotoc.label.separator` — Separator between labels and titles in the ToC

Synopsis

```
<xsl:param name="autotoc.label.separator" select="". '"/>
```

Description

String to use to separate labels and title in a table of contents.

4.3 `process.source.toc`

`process.source.toc` — FIXME:

Synopsis

```
<xsl:param name="process.source.toc" select="0"/>
```

Description

FIXME:

4.4 process.empty.source.toc

process.empty.source.toc — FIXME:

Synopsis

```
<xsl:param name="process.empty.source.toc" select="0"/>
```

Description

FIXME:

4.5 bridgehead.in.toc

bridgehead.in.toc — Should bridgehead elements appear in the TOC?

Synopsis

```
<xsl:param name="bridgehead.in.toc" select="0"/>
```

Description

If non-zero, bridgeheads appear in the TOC. Note that this option is not fully supported and may be removed in a future version of the stylesheets.

4.6 manual.toc

manual.toc — An explicit TOC to be used for the TOC

Synopsis

```
<xsl:param name="manual.toc" select=""/>
```

Description

The *manual.toc* identifies an explicit TOC that will be used for building the printed TOC.

4.7 toc.list.type

toc.list.type — Type of HTML list element to use for Tables of Contents

Synopsis

```
<xsl:param name="toc.list.type">dl</xsl:param>
```

Description

When an automatically generated Table of Contents (or List of Titles) is produced, this HTML element will be used to make the list.

4.8 toc.section.depth

toc.section.depth — How deep should recursive sections appear in the TOC?

Synopsis

```
<xsl:param name="toc.section.depth">2</xsl:param>
```

Description

Specifies the depth to which recursive sections should appear in the TOC.

4.9 generate.toc

generate.toc — Control generation of ToCs and LoTs

Synopsis

```
<xsl:param name="generate.toc"> appendix toc article toc book toc,figure,table,example,equation chapter toc part toc preface  
toc qandadiv toc qandaset toc reference toc sect1 toc sect2 toc sect3 toc sect4 toc sect5 toc section toc set toc </xsl:param>
```

Description

This parameter has a structured value. It is a table of space-delimited path/value pairs. Each path identifies some element in the source document using a restricted subset of XPath (only the implicit child axis, no wildcards, no predicates). Paths can be either relative or absolute.

When processing a particular element, the stylesheets consult this table to determine if a ToC (or LoT(s)) should be generated.

For example, consider the entry:

```
book toc,figure
```

This indicates that whenever a `book` is formatted, a Table Of Contents and a List of Figures should be generated. Similarly,

```
/chapter toc
```

indicates that whenever a document *that has a root of* `chapter` is formatted, a Table of Contents should be generated. The entry `chapter` would match all chapters, but `/chapter` matches only `chapter` document elements.

Generally, the longest match wins. So, for example, if you want to distinguish articles in books from articles in parts, you could use these two entries:

```
book/article toc,figure  
part/article toc
```

Note that an article in a part can never match a `book/article`, so if you want nothing to be generated for articles in parts, you can simply leave that rule out.

If you want to leave the rule in, to make it explicit that you're turning something off, use the value "nop". For example, the following entry disables ToCs and LoTs for articles:

```
article nop
```

Do not simply leave the word "article" in the file without a matching value. That'd be just begging the silly little path/value parser to get confused.

Section ToCs are further controlled by the `generate.section.toc.level` parameter. For a given section level to have a ToC, it must have both an entry in `generate.toc` and be within the range enabled by `generate.section.toc.level`.

4.10 generate.section.toc.level

`generate.section.toc.level` — Control depth of TOC generation in sections

Synopsis

```
<xsl:param name="generate.section.toc.level" select="0"/>
```

Description

The `generate.section.toc.level` parameter controls the depth of section in which TOCs will be generated. Note that this is related to, but not the same as `toc.section.depth`, which controls the depth to which TOC entries will be generated in a given TOC.

If, for example, `generate.section.toc.level` is 3, TOCs will be generated in first, second, and third level sections, but not in fourth level sections.

4.11 generate.index

`generate.index` — Do you want an index?

Synopsis

```
<xsl:param name="generate.index" select="1"/>
```

Description

Specify if an index should be generated.

Chapter 5

Extensions

5.1 `linenumbering.everyNth`

`linenumbering.everyNth` — Indicate which lines should be numbered

Synopsis

```
<xsl:param name="linenumbering.everyNth" select="'5'"/>
```

Description

If line numbering is enabled, everyNth line will be numbered.

5.2 `linenumbering.extension`

`linenumbering.extension` — Enable the line numbering extension

Synopsis

```
<xsl:param name="linenumbering.extension" select="'1'"/>
```

Description

If true, verbatim environments (elements that have the `format='linespecific'` notation attribute: `address`, `literallayout`, `program-listing`, `screen`, `synopsis`) that specify line numbering will have, surprise, line numbers.

5.3 `linenumbering.separator`

`linenumbering.separator` — Specify a separator between line numbers and lines

Synopsis

```
<xsl:param name="linenumbering.separator" select="''"/>
```

Description

The separator is inserted between line numbers and lines in the verbatim environment.

5.4 `linenumbering.width`

`linenumbering.width` — Indicates the width of line numbers

Synopsis

```
<xsl:param name="linenumbering.width" select="'3'"/>
```

Description

If line numbering is enabled, line numbers will appear right justified in a field "width" characters wide.

5.5 `tablecolumns.extension`

`tablecolumns.extension` — Enable the table columns extension function

Synopsis

```
<xsl:param name="tablecolumns.extension" select="'1'"/>
```

Description

The table columns extension function adjusts the widths of table columns in the HTML result to more accurately reflect the specifications in the CALS table.

5.6 `textinsert.extension`

`textinsert.extension` — Enable the textinsert extension element

Synopsis

```
<xsl:param name="textinsert.extension" select="'1'"/>
```

Description

The textinsert extension element inserts the contents of a file into the result tree (as text).

5.7 `use.extensions`

`use.extensions` — Enable extensions

Synopsis

```
<xsl:param name="use.extensions" select="''0'"/>
```

Description

If non-zero, extensions may be used. Each extension is further controlled by its own parameter. But if *use.extensions* is zero, no extensions will be used.

Chapter 6

Automatic labelling

6.1 chapter.autolabel

chapter.autolabel — Are chapters automatically enumerated?

Synopsis

```
<xsl:param name="chapter.autolabel" select="1"/>
```

Description

If true (non-zero), unlabeled chapters will be enumerated.

6.2 appendix.autolabel

appendix.autolabel — Are Appendixes automatically enumerated?

Synopsis

```
<xsl:param name="appendix.autolabel" select="1"/>
```

Description

If true (non-zero), unlabeled appendixes will be enumerated.

6.3 part.autolabel

part.autolabel — Are parts and references enumerated?

Synopsis

```
<xsl:param name="part.autolabel" select="1"/>
```

Description

If true (non-zero), unlabeled parts and references will be enumerated.

6.4 preface.autolabel

preface.autolabel — Are prefaces enumerated?

Synopsis

```
<xsl:param name="preface.autolabel" select="0"/>
```

Description

If true (non-zero), unlabeled prefaces will be enumerated.

6.5 qandadiv.autolabel

qandadiv.autolabel — Are divisions in QAndASets enumerated?

Synopsis

```
<xsl:param name="qandadiv.autolabel" select="1"/>
```

Description

If true (non-zero), unlabeled qandadivs will be enumerated.

6.6 section.autolabel

section.autolabel — Are sections enumerated?

Synopsis

```
<xsl:param name="section.autolabel" select="0"/>
```

Description

If true (non-zero), unlabeled sections will be enumerated.

6.7 section.label.includes.component.label

section.label.includes.component.label — Do section labels include the component label?

Synopsis

```
<xsl:param name="section.label.includes.component.label" select="0"/>
```

Description

If true (non-zero), section labels are prefixed with the label of the component that contains them.

6.8 label.from.part

label.from.part — Renumber chapters in each part?

Synopsis

```
<xsl:param name="label.from.part" select="0"/>
```

Description

If *label.from.part* is non-zero, components (chapters, appendixes, etc.) will be numbered from 1 in each part. Otherwise, they will be numbered monotonically throughout each book.

Chapter 7

HTML

7.1 html.base

html.base — An HTML base URI

Synopsis

```
<xsl:param name="html.base"/>
```

Description

If `html.base` is set, it is used for the `BASE` element in the `HEAD` of the HTML documents. This is useful for dynamically served HTML where the base URI needs to be shifted.

7.2 html.stylesheet.type

html.stylesheet.type — The type of the stylesheet used in the generated HTML

Synopsis

```
<xsl:param name="html.stylesheet.type">text/css</xsl:param>
```

Description

The type of the stylesheet to place in the HTML `link` tag.

7.3 html.stylesheet

html.stylesheet — Name of the stylesheet(s) to use in the generated HTML

Synopsis

```
<xsl:param name="html.stylesheet" select=""/>
```

Description

The *html.stylesheet* parameter is either empty, indicating that no stylesheet `LINK` tag should be generated in the HTML output, or it is a list of one or more stylesheets.

Multiple stylesheets are space-delimited. If you need to reference a stylesheet URI that includes a space, encode it with `%20`. A separate HTML `LINK` element will be generated for each stylesheet in the order they are listed in the parameter.

7.4 use.id.as.filename

`use.id.as.filename` — Use ID value of chunk elements as the filename?

Synopsis

```
<xsl:param name="use.id.as.filename" select="''0''"/>
```

Description

If *use.id.as.filename* is non-zero, the filename of chunk elements that have IDs will be derived from the ID value.

7.5 css.decoration

`css.decoration` — Enable CSS decoration of elements

Synopsis

```
<xsl:param name="css.decoration">1</xsl:param>
```

Description

If *css.decoration* is turned on, then HTML elements produced by the stylesheet may be decorated with `STYLE` attributes. For example, the `LI` tags produced for list items may include a fragment of CSS in the `STYLE` attribute which sets the CSS property `"list-style-type"`.

7.6 spacing.paras

`spacing.paras` — Insert additional `<p>` elements for spacing?

Synopsis

```
<xsl:param name="spacing.paras" select="''0''"/>
```

Description

When non-zero, additional, empty paragraphs are inserted in several contexts (for example, around informal figures), to create a more pleasing visual appearance in many browsers.

7.7 **emphasis.propagates.style**

emphasis.propagates.style — Pass emphasis role attribute through to HTML?

Synopsis

```
<xsl:param name="emphasis.propagates.style" select="1"/>
```

Description

If true, the role attribute of `emphasis` elements will be passed through to the HTML as a class attribute on a `span` that surrounds the emphasis.

7.8 **para.propagates.style**

para.propagates.style — Pass para role attribute through to HTML?

Synopsis

```
<xsl:param name="para.propagates.style" select="1"/>
```

Description

If true, the role attribute of `para` elements will be passed through to the HTML as a class attribute on the `p` generated for the paragraph.

7.9 **phrase.propagates.style**

phrase.propagates.style — Pass phrase role attribute through to HTML?

Synopsis

```
<xsl:param name="phrase.propagates.style" select="1"/>
```

Description

If true, the role attribute of `phrase` elements will be passed through to the HTML as a class attribute on a `span` that surrounds the phrase.

7.10 **html.longdesc**

html.longdesc — Should longdesc URIs be created?

Synopsis

```
<xsl:param name="html.longdesc" select="1"/>
```

Description

If non-zero, HTML files will be created for the `longdesc` attribute. These files are created from the `textobjects` in `mediaobjects` and `inlinemediaobject`.

7.11 `html.longdesc.link`

`html.longdesc.link` — Should a link to the `longdesc` be included in the HTML?

Synopsis

```
<xsl:param name="html.longdesc.link" select="$html.longdesc"/>
```

Description

If non-zero, links will be created to the HTML files created for the `longdesc` attribute. It makes no sense to turn enable this option without also enabling the `$html.longdesc` parameter.

The `longdesc.link` named template is called to construct the link.

7.12 `make.valid.html`

`make.valid.html` — Attempt to make sure the HTML output is valid HTML

Synopsis

```
<xsl:param name="make.valid.html" select="0"/>
```

Description

If `make.valid.html` is true, the stylesheets take extra effort to ensure that the resulting HTML is valid. This may mean that some `para` tags are translated into HTML `divs` or that other substitutions occur.

This parameter is different from `html.cleanup` because it changes the resulting markup; it does not use extension functions to manipulate result-tree-fragments and is therefore applicable to any XSLT processor.

7.13 `html.cleanup`

`html.cleanup` — Attempt to clean up the resulting HTML?

Synopsis

```
<xsl:param name="html.cleanup" select="1"/>
```

Description

If non-zero, and if the **EXSLT** extensions are supported by your processor, the resulting HTML will be “cleaned up”. This improves the chances that the resulting HTML will be valid. It may also improve the formatting of some elements.

This parameter is different from `make.valid.html` because it uses extension functions to manipulate result-tree-fragments.

7.14 draft.watermark.image

draft.watermark.image — The URI of the image to be used for draft watermarks

Synopsis

```
<xsl:param name="draft.watermark.image" select="'http://docbook.sourceforge.net/release/images/draft.png'"/>
```

Description

The image to be used for draft watermarks.

7.15 generate.id.attributes

generate.id.attributes —

Synopsis

```
<xsl:param name="generate.id.attributes" select="0"/>
```

Description

If non-zero, the HTML stylesheet will generate ID attributes on containers. For example, the markup:

```
<section id="foo"><title>Some Title</title>
<para>Some para.</para>
</section>
```

might produce:

```
<div class="section" id="foo">
<h2>Some Title</h2>
<p>Some para.</p>
</div>
```

The alternative is to generate anchors:

```
<div class="section">
<h2><a name="foo"></a>Some Title</h2>
<p>Some para.</p>
</div>
```

Because the `name` attribute of the `a` element and the `id` attribute of other tags are both of type “ID”, producing both generates invalid documents.

As of version 1.50, you can use this switch to control which type of identifier is generated. For backwards-compatibility, generating `a` anchors is preferred.

Note: at present, this switch is incompletely implemented. Disabling ID attributes will suppress them, but enabling ID attributes will not suppress the anchors.

7.16 generate.meta.abstract

generate.meta.abstract — Generate HTML `META` element from `abstract`?

Synopsis

```
<xsl:param name="generate.meta.abstract" select="1"/>
```

Description

If non-zero, document abstracts will be reproduced in the HTML HEAD with `<meta name="description" content="...">`

Chapter 8

XSLT Processing

8.1 rootid

rootid — Specify the root element to format

Synopsis

```
<xsl:param name="rootid" select=""/>
```

Description

If *rootid* is specified, it must be the value of an ID that occurs in the document being formatted. The entire document will be loaded and parsed, but formatting will begin at the element identified, rather than at the root. For example, this allows you to process only chapter 4 of a book.

Because the entire document is available to the processor, automatic numbering, cross references, and other dependencies are correctly resolved.

8.2 suppress.navigation

suppress.navigation — Disable header and footer navigation

Synopsis

```
<xsl:param name="suppress.navigation">0</xsl:param>
```

Description

If `suppress.navigation` is turned on, header and footer navigation will be suppressed.

8.3 suppress.header.navigation

suppress.header.navigation — Disable header navigation

Synopsis

```
<xsl:param name="suppress.header.navigation">0</xsl:param>
```

Description

If `suppress.header.navigation` is turned on, header navigation will be suppressed.

8.4 suppress.footer.navigation

`suppress.footer.navigation` — Disable footer navigation

Synopsis

```
<xsl:param name="suppress.footer.navigation">0</xsl:param>
```

Description

If `suppress.footer.navigation` is turned on, footer navigation will be suppressed.

8.5 header.rule

`header.rule` — Rule under headers?

Synopsis

```
<xsl:param name="header.rule" select="1"/>
```

Description

If non-zero, a rule will be drawn below the page headers.

8.6 footer.rule

`footer.rule` — Rule over footers?

Synopsis

```
<xsl:param name="footer.rule" select="1"/>
```

Description

If non-zero, a rule will be drawn above the page footers.

Chapter 9

Meta/*Info

9.1 inherit.keywords

inherit.keywords — Inherit keywords from ancestor elements?

Synopsis

```
<xsl:param name="inherit.keywords" select="'1'"/>
```

Description

If *inherit.keywords* is non-zero, the keyword META for each HTML HEAD element will include all of the keywords from ancestral elements. Otherwise, only the keywords from the current section will be used.

9.2 make.single.year.ranges

make.single.year.ranges — Print single-year ranges (e.g., 1998-1999)

Synopsis

```
<xsl:param name="make.single.year.ranges" select="0"/>
```

Description

If non-zero, year ranges that span a single year will be printed in range notation (1998-1999) instead of discrete notation (1998, 1999).

9.3 make.year.ranges

make.year.ranges — Collate copyright years into ranges?

Synopsis

```
<xsl:param name="make.year.ranges" select="0"/>
```

Description

If non-zero, copyright years will be collated into ranges.

9.4 `author.othername.in.middle`

`author.othername.in.middle` — Is `othername` in `author` a middle name?

Synopsis

```
<xsl:param name="author.othername.in.middle" select="1"/>
```

Description

If true (non-zero), the `othername` of an `author` appears between the `firstname` and `surname`. Otherwise, `othername` is suppressed.

9.5 `generate.legalnotice.link`

`generate.legalnotice.link` — TBD

Synopsis

```
<xsl:param name="generate.legalnotice.link" select="0"/>
```

Description

TBD

Chapter 10

Reference Pages

10.1 funcsynopsis.decoration

funcsynopsis.decoration — Decorate elements of a FuncSynopsis?

Synopsis

```
<xsl:param name="funcsynopsis.decoration" select="1"/>
```

Description

If true (non-zero), elements of the FuncSynopsis will be decorated (e.g. bold or italic). The decoration is controlled by functions that can be redefined in a customization layer.

10.2 funcsynopsis.style

funcsynopsis.style — What style of 'FuncSynopsis' should be generated?

Synopsis

```
<xsl:param name="funcsynopsis.style">kr</xsl:param>
```

Description

If `funcsynopsis.style` is `ansi`, ANSI-style function synopses are generated for a `funcsynopsis`, otherwise K&R-style function synopses are generated.

10.3 function.parens

function.parens — Generate parens after a function?

Synopsis

```
<xsl:param name="function.parens">0</xsl:param>
```

Description

If not 0, the formatting of a `<function>` element will include generated parenthesis.

10.4 refentry.generate.name

refentry.generate.name — Output NAME header before 'RefName'(s)?

Synopsis

```
<xsl:param name="refentry.generate.name" select="1"/>
```

Description

If true (non-zero), a "NAME" section title is output before the list of 'RefName's. This parameter and *refentry.generate.title* are mutually exclusive. This means that if you change this parameter to zero, you should set *refentry.generate.title* to 1 unless you want get quite strange output.

10.5 refentry.generate.title

refentry.generate.title — Output title before 'RefName'(s)?

Synopsis

```
<xsl:param name="refentry.generate.title" select="0"/>
```

Description

If true (non-zero), the reference page title or first name is output before the list of 'RefName's. This parameter and *refentry.generate.name* are mutually exclusive. This means that if you change this parameter to 1, you should set *refentry.generate.name* to 0 unless you want get quite strange output.

10.6 refentry.xref.manvolnum

refentry.xref.manvolnum — Output manvolnum as part of refentry cross-reference?

Synopsis

```
<xsl:param name="refentry.xref.manvolnum" select="1"/>
```

Description

if true (non-zero), the manvolnum is used when cross-referencing refentrys, either with xref or citerefentry.

10.7 citerefentry.link

citerefentry.link — Generate URL links when cross-referencing RefEntrys?

Synopsis

```
<xsl:param name="citerefentry.link" select="''0''"/>
```

Description

If true, a web link will be generated, presumably to an online man->HTML gateway. The text of the link is generated by the `generate.citerefentry.link` template.

10.8 refentry.separator

`refentry.separator` — Generate a separator between consecutive `RefEntry` elements?

Synopsis

```
<xsl:param name="refentry.separator" select="''1''"/>
```

Description

If true, a separator will be generated between consecutive reference pages.

Chapter 11

Tables

11.1 default.table.width

default.table.width — The default width of tables

Synopsis

```
<xsl:param name="default.table.width" select=""/>
```

Description

If specified, this value will be used for the WIDTH attribute on tables that do not specify an alternate width (with the dbhtml processing instruction).

11.2 nominal.table.width

nominal.table.width — The (absolute) nominal width of tables

Synopsis

```
<xsl:param name="nominal.table.width" select="'6in'"/>
```

Description

In order to convert CALS column widths into HTML column widths, it is sometimes necessary to have an absolute table width to use for conversion of mixed absolute and relative widths. This value must be an absolute length (not a percentag).

11.3 table.borders.with.css

table.borders.with.css — Use CSS to specify table, row, and cell borders?

Synopsis

```
<xsl:param name="table.borders.with.css" select="0"/>
```

Description

If true (non-zero), CSS will be used to draw table borders.

11.4 table.border.style

table.border.style —

Synopsis

```
<xsl:param name="table.border.style" select="''solid''"/>
```

Description

FIXME:

11.5 table.border.thickness

table.border.thickness —

Synopsis

```
<xsl:param name="table.border.thickness" select="''0.5pt''"/>
```

Description

FIXME:

11.6 table.border.color

table.border.color —

Synopsis

```
<xsl:param name="table.border.color" select="''''''"/>
```

Description

FIXME:

11.7 html.cellspacing

html.cellspacing — Default value for cellspacing in HTML tables

Synopsis

```
<xsl:param name="html.cellspacing" select=""/>
```

Description

If specified, this value will be used as the default cellspacing value in HTML tables.

11.8 html.cellpadding

html.cellpadding — Default value for cellpadding in HTML tables

Synopsis

```
<xsl:param name="html.cellpadding" select=""/>
```

Description

If specified, this value will be used as the default cellpadding value in HTML tables.

Chapter 12

QAndASet

12.1 qanda.defaultlabel

qanda.defaultlabel — Sets the default for defaultlabel on QandASet.

Synopsis

```
<xsl:param name="qanda.defaultlabel">number</xsl:param>
```

Description

If no defaultlabel attribute is specified on a QandASet, this value is used. It must be one of the legal values for the defaultlabel attribute.

12.2 qanda.inherit.numeration

qanda.inherit.numeration — Does enumeration of QandASet components inherit the numeration of parent elements?

Synopsis

```
<xsl:param name="qanda.inherit.numeration" select="1"/>
```

Description

If true (non-zero), numbered QandADiv elements and Questions and Answers inherit the numeration of the ancestors of the QandASet.

Chapter 13

Linking

13.1 target.database.document

target.database.document — Name of master database file for resolving olinks

Synopsis

```
<xsl:param name="target.database.document" select=""/>
```

Description

To resolve olinks between documents, the stylesheets use a master database document that identifies the target datafiles for all the documents within the scope of the olinks. This parameter value is the URI of the master document to be read during processing to resolve olinks. The default value is `olinkdb.xml`.

The data structure of the file is defined in the `targetdatabase.dtd` DTD. The database file provides the high level elements to record the identifiers, locations, and relationships of documents. The cross reference data for individual documents is generally pulled into the database using system entity references or XIncludes. See also *targets.filename*.

13.2 targets.filename

targets.filename — Name of cross reference targets data file

Synopsis

```
<xsl:param name="targets.filename" select="'target.db'"/>
```

Description

In order to resolve olinks efficiently, the stylesheets can generate an external data file containing information about all potential cross reference endpoints in a document. This parameter lets you change the name of the generated file from the default name `target.db`. The name must agree with that used in the target database used to resolve olinks during processing. See also *target.database.document*.

13.3 collect.xref.targets

collect.xref.targets — Controls whether cross reference data is collected

Synopsis

```
<xsl:param name="collect.xref.targets" select="'no'"/>
```

Description

In order to resolve olinks efficiently, the stylesheets can generate an external data file containing information about all potential cross reference endpoints in a document. This parameter determines whether the collection process is run when the document is processed by the stylesheet. The default value is `no`, which means the data file is not generated during processing. The other choices are `yes`, which means the data file is created and the document is processed for output, and `only`, which means the data file is created but the document is not processed for output. See also *targets.filename*.

13.4 olink.base.uri

olink.base.uri — Base URI used in olink hrefs

Synopsis

```
<xsl:param name="olink.base.uri" select="''"/>
```

Description

When cross reference data is collected for resolving olinks, it may be necessary to prepend a base URI to each target's href. This parameter lets you set that base URI when cross reference data is collected. This feature is needed when you want to link to a document that is processed without chunking. The output filename for such a document is not known to the XSL stylesheet; the only target information consists of fragment identifiers such as `#idref`. To enable the resolution of olinks between documents, you should pass the name of the HTML output file as the value of this parameter. Then the hrefs recorded in the cross reference data collection look like `outfile.html#idref`, which can be reached as links from other documents.

13.5 use.local.olink.style

use.local.olink.style — Process olinks using xref style of current document

Synopsis

```
<xsl:param name="use.local.olink.style" select="0"/>
```

Description

When cross reference data is collected for use by olinks, the data for each potential target includes one field containing a completely assembled cross reference string, as if it were an xref generated in that document. Other fields record the separate title, number, and element name of each target. When an olink is formed to a target from another document, the olink resolves to that preassembled string by default. If the *use.local.olink.style* parameter is set to non-zero, then instead the cross reference string is formed again from the target title, number, and element name, using the stylesheet processing the targeting document. Then olinks will match the xref style in the targeting document rather than in the target document. If both documents are processed with the same stylesheet, then the results will be the same.

13.6 current.docid

current.docid — targetdoc identifier for the document being processed

Synopsis

```
<xsl:param name="current.docid" select=""/>
```

Description

When olinks between documents are resolved for HTML output, the stylesheet can compute the relative path between the current document and the target document. The stylesheet needs to know the `targetdoc` identifiers for both documents, as they appear in the `target.database.document` database file. This parameter passes to the stylesheet the targetdoc identifier of the current document, since that identifier does not appear in the document itself.

This parameter can also be used for print output. If an olink's `targetdoc` id differs from the `current.docid`, then the stylesheet can append the target document's title to the generated olink text. That identifies to the reader that the link is to a different document, not the current document. See also `olink.doctype` to enable that feature.

13.7 olink.doctype

olink.doctype — show the document title for external olinks?

Synopsis

```
<xsl:param name="olink.doctype" select="0"/>
```

Description

When olinks between documents are resolved for print output, the generated text may not make it clear that the reference is to another document. It is possible for the stylesheets to append the other document's title to external olinks. For this to happen, two parameters must be set. The `olink.doctype` parameter should be set to nonzero to enable this feature. And you should set the `current.docid` parameter to the document id for the document currently being processed for output. If an olink's `targetdoc` id differs from the `current.docid`, then the stylesheet can append the target document's title to the generated olink text.

13.8 link.mailto.url

link.mailto.url — Mailto URL for the LINK REL=made HTML HEAD element

Synopsis

```
<xsl:param name="link.mailto.url"/>
```

Description

If not the empty string, this address will be used for the REL=made LINK element in the HTML HEAD.

13.9 ulink.target

ulink.target — The HTML anchor target for ULinks

Synopsis

```
<xsl:param name="ulink.target" select="'_top'"/>
```

Description

If *ulink.target* is set, its value will be used for the `target` attribute on anchors generated for `ulinks`.

13.10 olink.fragid

olink.fragid — Names the fragment identifier portion of an OLink resolver query

Synopsis

```
<xsl:param name="olink.fragid" select="''fragid='"/>
```

Description

FIXME:

13.11 olink.outline.ext

olink.outline.ext — The extension of OLink outline files

Synopsis

```
<xsl:param name="olink.outline.ext" select="'.olink'"/>
```

Description

FIXME:

13.12 olink.pubid

olink.pubid — Names the public identifier portion of an OLink resolver query

Synopsis

```
<xsl:param name="olink.pubid" select="''pubid='"/>
```

Description

FIXME:

13.13 olink.sysid

olink.sysid — Names the system identifier portion of an OLink resolver query

Synopsis

```
<xsl:param name="olink.sysid" select="'sysid='"/>
```

Description

FIXME:

13.14 olink.resolver

olink.resolver — The root name of the OLink resolver (usually a script)

Synopsis

```
<xsl:param name="olink.resolver" select="'/cgi-bin/olink'"/>
```

Description

FIXME:

Chapter 14

Bibliography

14.1 biblioentry.item.separator

biblioentry.item.separator — Text to separate bibliography entries

Synopsis

```
<xsl:param name="biblioentry.item.separator">. </xsl:param>
```

Description

Text to separate bibliography entries

14.2 bibliography.collection

bibliography.collection — Name of the bibliography collection file

Synopsis

```
<xsl:param name="bibliography.collection" select="'http://docbook.sourceforge.net/release/bibliography/bibliography.xml'"/>
```

Description

Maintaining bibliography entries across a set of documents is tedious, time consuming, and error prone. It makes much more sense, usually, to store all of the bibliography entries in a single place and simply “extract” the ones you need in each document.

That’s the purpose of the *bibliography.collection* parameter. To setup a global bibliography “database”, follow these steps:

First, create a stand-alone bibliography document that contains all of the documents that you wish to reference. Make sure that each bibliography entry (whether you use *biblioentry* or *bibliomixed*) has an ID.

My global bibliography, *~/bibliography.xml* begins like this:

```

<!DOCTYPE bibliography
  PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
    "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<bibliography><title>References</title>

<bibliomixed id="xml-rec"><abbrev>XML 1.0</abbrev>Tim Bray,
Jean Paoli, C. M. Sperberg-McQueen, and Eve Maler, editors.
<citetitle><ulink url="http://www.w3.org/TR/REC-xml">Extensible Markup
Language (XML) 1.0 Second Edition</ulink></citetitle>.
World Wide Web Consortium, 2000.
</bibliomixed>

<bibliomixed id="xml-names"><abbrev>Namespaces</abbrev>Tim Bray,
Dave Hollander,
and Andrew Layman, editors.
<citetitle><ulink url="http://www.w3.org/TR/REC-xml-names/">Namespaces in
XML</ulink></citetitle>.
World Wide Web Consortium, 1999.
</bibliomixed>

<!-- ... -->
</bibliography>

```

When you create a bibliography in your document, simply provide *empty* `bibliomixed` entries for each document that you wish to cite. Make sure that these elements have the same ID as the corresponding “real” entry in your global bibliography.

For example:

```

<bibliography><title>Bibliography</title>

<bibliomixed id="xml-rec"/>
<bibliomixed id="xml-names"/>
<bibliomixed id="DKnuth86">Donald E. Knuth. <citetitle>Computers and
Typesetting: Volume B, TeX: The Program</citetitle>. Addison-Wesley,
1986. ISBN 0-201-13437-3.
</bibliomixed>
<bibliomixed id="relaxng"/>

</bibliography>

```

Note that it’s perfectly acceptable to mix entries from your global bibliography with “normal” entries. You can use `xref` or other elements to cross-reference your bibliography entries in exactly the same way you do now.

Finally, when you are ready to format your document, simply set the `bibliography.collection` parameter (in either a customization layer or directly through your processor’s interface) to point to your global bibliography.

The stylesheets will format the bibliography in your document as if all of the entries referenced appeared there literally.

14.3 bibliography.numbered

`bibliography.numbered` — Should bibliography entries be numbered?

Synopsis

```
<xsl:param name="bibliography.numbered" select="0"/>
```

Description

If non-zero bibliography entries will be numbered

Chapter 15

Glossary

15.1 glossterm.auto.link

`glossterm.auto.link` — Generate links from `glossterm` to `glossentry` automatically?

Synopsis

```
<xsl:param name="glossterm.auto.link" select="'0'"/>
```

Description

If true, a link will be automatically created from `glossterm` to `glossentry` for that glossary term. This is usefull when your `glossterm` names are consistent and you don't want to add links manually.

If there is `linkend` on `glossterm` then is used instead of autogeneration of link.

15.2 firstterm.only.link

`firstterm.only.link` — Does automatic `glossterm` linking only apply to `firstterms`?

Synopsis

```
<xsl:param name="firstterm.only.link" select="0"/>
```

Description

If true, only `firstterms` will be automatically linked to the glossary. If glossary linking is not enabled, this parameter has no effect.

15.3 glossary.collection

`glossary.collection` — Name of the glossary collection file

Synopsis

```
<xsl:param name="glossary.collection" select=""/>
```

Description

Glossaries maintained independently across a set of documents are likely to become inconsistent unless considerable effort is expended to keep them in sync. It makes much more sense, usually, to store all of the glossary entries in a single place and simply “extract” the ones you need in each document.

That’s the purpose of the *glossary.collection* parameter. To setup a global glossary “database”, follow these steps:

Setting Up the Glossary Database

First, create a stand-alone glossary document that contains all of the entries that you wish to reference. Make sure that each glossary entry has an ID.

Here’s an example glossary:

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE glossary
  PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
  "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<glossary>
<glossaryinfo>
<editor><firstname>Eric</firstname><surname>Raymond</surname></editor>
<title>Jargon File 4.2.3 (abridged)</title>
<releaseinfo>Just some test data</releaseinfo>
</glossaryinfo>

<glossdiv><title>0</title>

<glossentry>
<glossterm>0</glossterm>
<glossdef>
<para>Numeric zero, as opposed to the letter 'O' (the 15th letter of
the English alphabet). In their unmodified forms they look a lot
alike, and various kluges invented to make them visually distinct have
compounded the confusion. If your zero is center-dotted and letter-O
is not, or if letter-O looks almost rectangular but zero looks more
like an American football stood on end (or the reverse), you’re
probably looking at a modern character display (though the dotted zero
seems to have originated as an option on IBM 3270 controllers). If
your zero is slashed but letter-O is not, you’re probably looking at
an old-style ASCII graphic set descended from the default typewheel on
the venerable ASR-33 Teletype (Scandinavians, for whom /O is a letter,
curse this arrangement). (Interestingly, the slashed zero long
predates computers; Florian Cajori’s monumental "A History of
Mathematical Notations" notes that it was used in the twelfth and
thirteenth centuries.) If letter-O has a slash across it and the zero
does not, your display is tuned for a very old convention used at IBM
and a few other early mainframe makers (Scandinavians curse <emphasis>this</emphasis>
arrangement even more, because it means two of their letters collide).
Some Burroughs/Unisys equipment displays a zero with a <emphasis>reversed</emphasis>
slash. Old CDC computers rendered letter O as an unbroken oval and 0
as an oval broken at upper right and lower left. And yet another
convention common on early line printers left zero unornamented but
added a tail or hook to the letter-O so that it resembled an inverted
Q or cursive capital letter-O (this was endorsed by a draft ANSI
standard for how to draw ASCII characters, but the final standard
```

```

changed the distinguisher to a tick-mark in the upper-left corner).
Are we sufficiently confused yet?</para>
</glossdef>
</glossentry>

<glossentry>
<glossterm>1TBS</glossterm>
<glossdef>
<para role="accidence">
<phrase role="pronounce"></phrase>
<phrase role="partsofspeech">n</phrase>
</para>
<para>The "One True Brace Style"</para>
<glossseealso>indent style</glossseealso>
</glossdef>
</glossentry>

<!-- ... -->

</glossdiv>

<!-- ... -->

</glossary>

```

Marking Up Glossary Terms

That takes care of the glossary database, now you have to get the entries into your document. Unlike bibliography entries, which can be empty, creating “placeholder” glossary entries would be very tedious. So instead, support for *glossary.collection* relies on implicit linking.

In your source document, simply use *firstterm* and *glossterm* to identify the terms you wish to have included in the glossary. The stylesheets assume that you will either set the *baseform* attribute correctly, or that the content of the element exactly matches a term in your glossary.

If you’re using a *glossary.collection*, don’t make explicit links on the terms in your document.

So, in your document, you might write things like this:

```

<para>This is dummy text, without any real meaning.
The point is simply to reference glossary terms like <glossterm>0</glossterm>
and the <firstterm baseform="1TBS">One True Brace Style (1TBS)</firstterm>.
The <glossterm>1TBS</glossterm>, as you can probably imagine, is a nearly
religious issue.</para>

```

If you set the *firstterm.only.link* parameter, only the terms marked with *firstterm* will be links. Otherwise, all the terms will be linked.

Marking Up the Glossary

The glossary itself has to be identified for the stylesheets. For lack of a better choice, the *role* is used. To identify the glossary as the target for automatic processing, set the *role* to “auto”. The title of this glossary (and any other information from the *glossaryinfo* that’s rendered by your stylesheet) will be displayed, but the entries will come from the database.

Unfortunately, the glossary can’t be empty, so you must put in at least one *glossentry*. The content of this entry is irrelevant, it will not be rendered:

```

<glossary role="auto">
<glossentry>
<glossterm>Irrelevant</glossterm>

```

```
<glossdef>
<para>If you can see this, the document was processed incorrectly. Use
the <parameter>glossary.collection</parameter> parameter.</para>
</glossdef>
</glossentry>
</glossary>
```

What about glossary divisions? If your glossary database has glossary divisions *and* your automatic glossary contains at least one `glossdiv`, the automatic glossary will have divisions. If the `glossdiv` is missing from either location, no divisions will be rendered.

Glossary entries (and divisions, if appropriate) in the glossary will occur in precisely the order they occur in your database.

Formatting the Document

Finally, when you are ready to format your document, simply set the `glossary.collection` parameter (in either a customization layer or directly through your processor's interface) to point to your global glossary.

The stylesheets will format the glossary in your document as if all of the entries implicitly referenced appeared there literally.

Limitations

Glossary cross-references *within the glossary* are not supported. For example, this *will not* work:

```
<glossentry>
<glossterm>gloss-1</glossterm>
<glossdef><para>A description that references <glossterm>gloss-2</glossterm>.</para>
<glossseealso>gloss-2</glossseealso>
</glossdef>
</glossentry>
```

If you put glossary cross-references in your glossary that way, you'll get the cryptic error: Warning: `glossary.collection` specified, but there are 0 automatic glossaries.

Instead, you must do two things:

1. Markup your glossary using `glossseealso`:

```
<glossentry>
<glossterm>gloss-1</glossterm>
<glossdef><para>A description that references <glossterm>gloss-2</glossterm>.</para>
<glossseealso>gloss-2</glossseealso>
</glossdef>
</glossentry>
```

2. Make sure there is at least one `glossterm` reference to *gloss-2* in your document. The easiest way to do that is probably within a remark in your automatic glossary:

```
<glossary role="auto">
<remark>Make sure there's a reference to <glossterm>gloss-2</glossterm>.</remark>
<glossentry>
<glossterm>Irrelevant</glossterm>
<glossdef>
<para>If you can see this, the document was processed incorrectly. Use
the <parameter>glossary.collection</parameter> parameter.</para>
</glossdef>
</glossentry>
</glossary>
```


Chapter 16

Miscellaneous

16.1 formal.procedures

formal.procedures — Selects formal or informal procedures

Synopsis

```
<xsl:param name="formal.procedures" select="1"/>
```

Description

Formal procedures are numbered and always have a title.

16.2 formal.title.placement

formal.title.placement —

Synopsis

```
<xsl:param name="formal.title.placement"> figure before example before equation before table before procedure before </xsl:param>
```

Description

Specifies where formal object titles should occur. For each formal object type (figure, example, equation, table, and procedure) you can specify either the keyword “before” or “after”.

16.3 runinhead.default.title.end.punct

runinhead.default.title.end.punct — Default punctuation character on a run-in-head

Synopsis

```
<xsl:param name="runinhead.default.title.end.punct" select=".'"/>
```

Description

FIXME:

16.4 runinhead.title.end.punct

runinhead.title.end.punct — Characters that count as punctuation on a run-in-head

Synopsis

```
<xsl:param name="runinhead.title.end.punct" select="".!?:'"/>
```

Description

FIXME:

16.5 show.comments

show.comments — Display `comment` elements?

Synopsis

```
<xsl:param name="show.comments">1</xsl:param>
```

Description

If true (non-zero), comments will be displayed, otherwise they are suppressed. Comments here refers to the `comment` element, which will be renamed `remark` in DocBook V4.0, not XML comments (`<!--` like this `-->`) which are unavailable.

16.6 show.revisionflag

show.revisionflag — Enable decoration of elements that have a revisionflag

Synopsis

```
<xsl:param name="show.revisionflag">0</xsl:param>
```

Description

If `show.revisionflag` is turned on, then the stylesheets may produce additional markup designed to allow a CSS stylesheet to highlight elements that have specific revisionflag settings.

The markup inserted will be usually be either a `` or `<div>` with an appropriate `class` attribute. (The value of the class attribute will be the same as the value of the revisionflag attribute). In some contexts, for example tables, where extra markup would be structurally illegal, the class attribute will be added to the appropriate container element.

In general, the stylesheets only test for revisionflag in contexts where an importing stylesheet would have to redefine whole templates. Most of the revisionflag processing is expected to be done by another stylesheet, for example `changebars.xsl`.

16.7 shade.verbatim

shade.verbatim — Should verbatim environments be shaded?

Synopsis

```
<xsl:param name="shade.verbatim" select="0"/>
```

Description

FIXME:

16.8 shade.verbatim.style

shade.verbatim.style — Properties that specify the style of shaded verbatim listings

Synopsis

```
<xsl:attribute-set name="shade.verbatim.style"> <xsl:attribute name="border">0</xsl:attribute> <xsl:attribute name="bgcolor">#E0E0E0</xsl:attribute-set>
```

Description

FIXME:

16.9 punct.honorific

punct.honorific — Punctuation after an honorific in a personal name.

Synopsis

```
<xsl:param name="punct.honorific" select="".?"/>
```

Description

This parameter specifies the punctuation that should be added after an honorific in a personal name.

16.10 segmentedlist.as.table

segmentedlist.as.table — Format segmented lists as tables?

Synopsis

```
<xsl:param name="segmentedlist.as.table" select="0"/>
```

Description

If non-zero, `segmentedlists` will be formatted as tables.

16.11 `variablelist.as.table`

`variablelist.as.table` — Format `variablelists` as tables?

Synopsis

```
<xsl:param name="variablelist.as.table" select="0"/>
```

Description

If non-zero, `variablelists` will be formatted as tables.

This parameter only applies to the HTML transformations. In the FO case, proper list markup is robust enough to handle the formatting. But see also `variablelist.as.blocks`.

16.12 `tex.math.in.alt`

`tex.math.in.alt` — TeX notation used for equations

Synopsis

```
<xsl:param name="tex.math.in.alt" select=""/>
```

Description

If you want type math directly in TeX notation in equations, this parameter specifies notation used. Currently are supported two values -- `plain` and `latex`. Empty value means that you are not using TeX math at all.

Preferred way for including TeX alternative of math is inside of `textobject` element. Eg.:

```
<inlineequation>
<inlinemediaobject>
<imageobject>
<imagedata fileref="eq1.gif"/>
</imageobject>
<textobject><phrase>E=mc squared</phrase></textobject>
<textobject role="tex"><phrase>E=mc^2</phrase></textobject>
</inlinemediaobject>
</inlineequation>
```

If you are using `graphic` element, you can store TeX inside `alt` element:

```
<inlineequation>
<alt role="tex">a^2+b^2=c^2</alt>
<graphic fileref="a2b2c2.gif"/>
</inlineequation>
```

If you want use this feature, you should process your FO with PassiveTeX, which only supports TeX math notation. When calling `stylsheet`, don't forget to specify also `passivetex.extensions=1`.

If you want equations in HTML, just process generated file `tex-math-equations.tex` by TeX or LaTeX. Then run `dvi2bitmap` program on result DVI file. You will get images for equations in your document.

16.13 tex.math.file

tex.math.file — Name of temporary file for generating images from equations

Synopsis

```
<xsl:param name="tex.math.file" select="'tex-math-equations.tex'"/>
```

Description

Name of auxiliary file for TeX equations. This file can be processed by dvi2bitmap to get bitmap versions of equations for HTML output.

16.14 tex.math.delims

tex.math.delims — Should be equations outputted for processing by TeX automatically surrounded by math mode delimiters

Synopsis

```
<xsl:param name="tex.math.delims" select="'1'"/>
```

Description

For compatibility with DSSSL based DBTeXMath from Allin Cottrell you should set this parameter to 0.

16.15 pixels.per.inch

pixels.per.inch — How many pixels are there per inch?

Synopsis

```
<xsl:param name="pixels.per.inch" select="90"/>
```

Description

When lengths are converted to pixels, this value is used to determine the size of a pixel. The default value is taken from the [XSL Recommendation](#).

16.16 points.per.em

points.per.em — Specify the nominal size of an em-space in points

Synopsis

```
<xsl:param name="points.per.em" select="10"/>
```

Description

FIXME:

16.17 use.svg

use.svg — Allow SVG in the result tree?

Synopsis

```
<xsl:param name="use.svg" select="1"/>
```

Description

If non-zero, SVG will be considered an acceptable image format. SVG is passed through to the result tree, so correct rendering of the resulting diagram depends on the formatter (FO processor or web browser) that is used to process the output from the stylesheet.

16.18 use.role.as.xrefstyle

use.role.as.xrefstyle — Use `role` attribute for `xrefstyle` on `xref`?

Synopsis

```
<xsl:param name="use.role.as.xrefstyle" select="1"/>
```

Description

If non-zero, the `role` attribute on `xref` will be used to select the cross reference style. It is likely that the [DocBook Technical Committee](#) will eventually add an `xrefstyle` attribute for this purpose. Until then, this flag allows `role` to serve that purpose.

16.19 menuchoice.separator

menuchoice.separator —

Synopsis

```
<xsl:param name="menuchoice.separator" select="'+'"/>
```

Description

Separator used to connect items of a `menuchoice` other than `guimenuitem` and `guisubmenu`. The latter elements are linked with `menuchoice.menu.separator`.

16.20 menuchoice.menu.separator

menuchoice.menu.separator —

Synopsis

```
<xsl:param name="menuchoice.menu.separator" select="'->'"/>
```

Description

Separator used to connect items of a `menuchoice` with `guimenuitem` or `guisubmenu`. Other elements are linked with *menuchoice.separator*.

Chapter 17

Graphics

17.1 graphic.default.extension

graphic.default.extension — Default extension for graphic filenames

Synopsis

```
<xsl:param name="graphic.default.extension"/>
```

Description

If a `graphic` or `mediaobject` includes a reference to a filename that does not include an extension, and the `format` attribute is *unspecified*, the default extension will be used.

17.2 nominal.image.width

nominal.image.width — The nominal image width

Synopsis

```
<xsl:param name="nominal.image.width" select="6 * $pixels.per.inch"/>
```

Description

Graphic widths expressed as a percentage are problematic. In the following discussion, we speak of width and contentwidth, but the same issues apply to depth and contentdepth.

A width of 50% means "half of the available space for the image." That's fine. But note that in HTML, this is a dynamic property and the image size will vary if the browser window is resized.

A contentwidth of 50% means "half of the actual image width". But what does that mean if the stylesheets cannot assess the image's actual size? Treating this as a width of 50% is one possibility, but it produces behavior (dynamic scaling) that seems entirely out of character with the meaning.

Instead, the stylesheets define a `nominal.image.width` and convert percentages to actual values based on that nominal size.

17.3 nominal.image.depth

nominal.image.depth — Nominal image depth

Synopsis

```
<xsl:param name="nominal.image.depth" select="4 * $pixels.per.inch"/>
```

Description

See *nominal.image.width*.

17.4 use.embed.for.svg

use.embed.for.svg — Use HTML `embed` for SVG?

Synopsis

```
<xsl:param name="use.embed.for.svg" select="0"/>
```

Description

If non-zero, an `embed` element will be created for SVG figures. An `object` is *always* created, this parameter merely controls whether or not an additional `embed` is generated inside the `object`.

On the plus side, this may be more portable among browsers and plug-ins. On the minus side, it isn't valid HTML.

17.5 make.graphic.viewport

make.graphic.viewport — Use tables in HTML to make viewports for graphics

Synopsis

```
<xsl:param name="make.graphic.viewport" select="1"/>
```

Description

The HTML `img` element only supports the notion of content-area scaling; it doesn't support the distinction between a content-area and a viewport-area, so we have to make some compromises.

If *make.graphic.viewport* is non-zero, a table will be used to frame the image. This creates an effective viewport-area.

Tables and alignment don't work together, so this parameter is ignored if alignment is specified on an image.

Chapter 18

Chunking

18.1 chunker.output.cdata-section-elements

chunker.output.cdata-section-elements — List of elements to escape with CDATA sections

Synopsis

```
<xsl:param name="chunker.output.cdata-section-elements" select=""/>
```

Description

This parameter specifies the list of elements that should be escaped as CDATA sections by the chunking stylesheet. Not all processors support specification of this parameter.

Note

This parameter is documented here, but the declaration is actually in the `chunker.xsl` stylesheet module.

18.2 chunker.output.doctype-public

chunker.output.doctype-public — Public identifier to use in the document type of generated pages

Synopsis

```
<xsl:param name="chunker.output.doctype-public" select=""/>
```

Description

This parameter specifies the public identifier that should be used by the chunking stylesheet in the document type declaration of chunked pages. Not all processors support specification of this parameter.

Note

This parameter is documented here, but the declaration is actually in the `chunker.xsl` stylesheet module.

18.3 chunker.output.doctype-system

chunker.output.doctype-system — System identifier to use for the document type in generated pages

Synopsis

```
<xsl:param name="chunker.output.doctype-system" select=""/>
```

Description

This parameter specifies the system identifier that should be used by the chunking stylesheet in the document type declaration of chunked pages. Not all processors support specification of this parameter.

Note

This parameter is documented here, but the declaration is actually in the `chunker.xsl` stylesheet module.

18.4 chunker.output.encoding

chunker.output.encoding — Encoding used in generated pages

Synopsis

```
<xsl:param name="chunker.output.encoding" select="'ISO-8859-1'"/>
```

Description

This parameter specifies the encoding to be used in files generated by the chunking stylesheet. Not all processors support specification of this parameter.

This parameter used to be named `default.encoding`.

Note

This parameter is documented here, but the declaration is actually in the `chunker.xsl` stylesheet module.

18.5 chunker.output.indent

chunker.output.indent — Specification of indentation on generated pages

Synopsis

```
<xsl:param name="chunker.output.indent" select="'no'"/>
```

Description

This parameter specifies the value of the indent specification for generated pages. Not all processors support specification of this parameter.

Note

This parameter is documented here, but the declaration is actually in the `chunker.xsl` stylesheet module.

18.6 chunker.output.media-type

chunker.output.media-type — Media type to use in generated pages

Synopsis

```
<xsl:param name="chunker.output.media-type" select=""/>
```

Description

This parameter specifies the media type that should be used by the chunking stylesheet. Not all processors support specification of this parameter.

Note

This parameter is documented here, but the declaration is actually in the `chunker.xsl` stylesheet module.

18.7 chunker.output.method

chunker.output.method — Method used in generated pages

Synopsis

```
<xsl:param name="chunker.output.method" select="'html'"/>
```

Description

This parameter specifies the output method to be used in files generated by the chunking stylesheet.

This parameter used to be named `output.method`.

Note

This parameter is documented here, but the declaration is actually in the `chunker.xsl` stylesheet module.

18.8 chunker.output.omit-xml-declaration

chunker.output.omit-xml-declaration — Omit-xml-declaration for generated pages

Synopsis

```
<xsl:param name="chunker.output.omit-xml-declaration" select="'no'"/>
```

Description

This parameter specifies the value of the omit-xml-declaration specification for generated pages. Not all processors support specification of this parameter.

Note

This parameter is documented here, but the declaration is actually in the `chunker.xsl` stylesheet module.

18.9 chunker.output.standalone

chunker.output.standalone — Standalone declaration for generated pages

Synopsis

```
<xsl:param name="chunker.output.standalone" select="'no'"/>
```

Description

This parameter specifies the value of the standalone specification for generated pages. Not all processors support specification of this parameter.

Note

This parameter is documented here, but the declaration is actually in the `chunker.xsl` stylesheet module.

18.10 saxon.character.representation

saxon.character.representation — Saxon character representation used in generated HTML pages

Synopsis

```
<xsl:param name="saxon.character.representation" select="'entity;decimal'"/>
```

Description

This character representation is used in files generated by chunking stylesheet. If you want to suppress entity references for characters with direct representation in `default.encoding`, set this parameter to value `native`.

Note

This parameter is documented here, but the declaration is actually in the `chunker.xsl` stylesheet module.

18.11 `html.ext`

`html.ext` — Identifies the extension of generated HTML files

Synopsis

```
<xsl:param name="html.ext" select="".html"/>
```

Description

The extension identified by `html.ext` will be used as the filename extension for chunks created by this stylesheet.

18.12 `html.extra.head.links`

`html.extra.head.links` — Toggle extra HTML head link information

Synopsis

```
<xsl:param name="html.extra.head.links" select="0"/>
```

Description

If non-zero, extra `link` elements will be generated in the `head` of chunked HTML files. These extra links point to chapters, appendixes, sections, etc. as supported by the “Site Navigation Bar” in Mozilla 1.0 (as of CR1, at least).

18.13 `root.filename`

`root.filename` — Identifies the name of the root HTML file when chunking

Synopsis

```
<xsl:param name="root.filename" select="index"/>
```

Description

The `root.filename` is the base filename for the chunk created for the root of each document processed.

18.14 `base.dir`

`base.dir` — The base directory of chunks

Synopsis

```
<xsl:param name="base.dir" select=""/>
```

Description

If specified, the `base.dir` identifies the output directory for chunks. (If not specified, the output directory is system dependent.)

18.15 generate.manifest

`generate.manifest` — Generate a manifest file?

Synopsis

```
<xsl:param name="generate.manifest" select="0"/>
```

Description

If non-zero, a list of HTML files generated by the stylesheet transformation is written to the file named by the *manifest* parameter.

18.16 manifest

`manifest` — Name of manifest file

Synopsis

```
<xsl:param name="manifest" select="'HTML.manifest'"/>
```

Description

The name of the file to which a manifest is written (if the value of the *generate.manifest* parameter is non-zero).

18.17 chunk.toc

`chunk.toc` — An explicit TOC to be used for chunking

Synopsis

```
<xsl:param name="chunk.toc" select=""/>
```

Description

The *chunk.toc* identifies an explicit TOC that will be used for chunking. This parameter is only used by the `chunktoc.xsl` stylesheet (and customization layers built from it).

18.18 chunk.tocs.and.lots

`chunk.tocs.and.lots` — Should ToC and LoTs be in separate chunks?

Synopsis

```
<xsl:param name="chunk.tocs.and.lots" select="0"/>
```

Description

If non-zero, ToC and LoT (List of Examples, List of Figures, etc.) will be put in a separate chunk. At the moment, this chunk is not in the normal forward/backward navigation list. Instead, a new link is added to the navigation footer.

This feature is still somewhat experimental. Feedback welcome.

18.19 chunk.section.depth

chunk.section.depth — Depth to which sections should be chunked

Synopsis

```
<xsl:param name="chunk.section.depth" select="1"/>
```

Description

This parameter sets the depth of section chunking.

18.20 chunk.first.sections

chunk.first.sections — Chunk the first top-level section?

Synopsis

```
<xsl:param name="chunk.first.sections" select="0"/>
```

Description

If non-zero, a chunk will be created for the first top-level `sect1` or `section` elements in each component. Otherwise, that section will be part of the chunk for its parent.

18.21 chunk.quietly

chunk.quietly — Omit the chunked filename messages.

Synopsis

```
<xsl:param name="chunk.quietly" select="0"/>
```

Description

If zero (the default), the XSL processor emits a message naming each separate chunk filename as it is being output. If nonzero, then the messages are suppressed.

18.22 `navig.graphics`

`navig.graphics` — Use graphics in navigational headers and footers?

Synopsis

```
<xsl:param name="navig.graphics" select="0"/>
```

Description

If true (non-zero), the navigational headers and footers in chunked HTML are presented in an alternate style that uses graphical icons for Next, Previous, Up, and Home. Default graphics are provided in the distribution.

18.23 `navig.graphics.extension`

`navig.graphics.extension` — Extension for navigational graphics

Synopsis

```
<xsl:param name="navig.graphics.extension" select="".gif"/>
```

Description

Sets the filename extension to use on navigational graphics used in the headers and footers of chunked HTML.

18.24 `navig.graphics.path`

`navig.graphics.path` — Path to navigational graphics

Synopsis

```
<xsl:param name="navig.graphics.path">images/</xsl:param>
```

Description

Sets the path, probably relative to the directory where the HTML files are created, to the navigational graphics used in the headers and footers of chunked HTML.

18.25 `navig.showtitles`

`navig.showtitles` — Display titles in HTML headers and footers?

Synopsis

```
<xsl:param name="navig.showtitles">1</xsl:param>
```

Description

If true (non-zero), the headers and footers of chunked HTML display the titles of the next and previous chunks, along with the words 'Next' and 'Previous' (or the equivalent graphical icons if `navig.graphics` is true). If false (zero), then only the words 'Next' and 'Previous' (or the icons) are displayed.

Chapter 19

Profiling

Following parameters can be used for attribute value based profiling of your document. For more info about profiling look at <http://docbook.sourceforge.net/projects/xsl/doc/tools/profiling.html>.

19.1 profile.arch

profile.arch — Target profile for arch attribute

Synopsis

```
<xsl:param name="profile.arch" select=""/>
```

Description

Value of this parameter specifies profiles which should be included in the output. You can specify multiple profiles by separating them by semicolon. You can change separator character by [profile.separator](#) parameter.

This parameter has effect only when you are using profiling stylesheets (`profile-docbook.xsl`, `profile-chunk.xsl`, ...) instead of normal ones (`docbook.xsl`, `chunk.xsl`, ...).

19.2 profile.condition

profile.condition — Target profile for condition attribute

Synopsis

```
<xsl:param name="profile.condition" select=""/>
```

Description

Value of this parameter specifies profiles which should be included in the output. You can specify multiple profiles by separating them by semicolon. You can change separator character by [profile.separator](#) parameter.

This parameter has effect only when you are using profiling stylesheets (`profile-docbook.xsl`, `profile-chunk.xsl`, ...) instead of normal ones (`docbook.xsl`, `chunk.xsl`, ...).

19.3 profile.conformance

profile.conformance — Target profile for `conformance` attribute

Synopsis

```
<xsl:param name="profile.conformance" select=""/>
```

Description

Value of this parameter specifies profiles which should be included in the output. You can specify multiple profiles by separating them by semicolon. You can change separator character by [profile.separator](#) parameter.

This parameter has effect only when you are using profiling stylesheets (`profile-docbook.xml`, `profile-chunk.xml`, ...) instead of normal ones (`docbook.xml`, `chunk.xml`, ...).

19.4 profile.lang

profile.lang — Target profile for `lang` attribute

Synopsis

```
<xsl:param name="profile.lang" select=""/>
```

Description

Value of this parameter specifies profiles which should be included in the output. You can specify multiple profiles by separating them by semicolon. You can change separator character by [profile.separator](#) parameter.

This parameter has effect only when you are using profiling stylesheets (`profile-docbook.xml`, `profile-chunk.xml`, ...) instead of normal ones (`docbook.xml`, `chunk.xml`, ...).

19.5 profile.os

profile.os — Target profile for `os` attribute

Synopsis

```
<xsl:param name="profile.os" select=""/>
```

Description

Value of this parameter specifies profiles which should be included in the output. You can specify multiple profiles by separating them by semicolon. You can change separator character by [profile.separator](#) parameter.

This parameter has effect only when you are using profiling stylesheets (`profile-docbook.xml`, `profile-chunk.xml`, ...) instead of normal ones (`docbook.xml`, `chunk.xml`, ...).

19.6 profile.revision

profile.revision — Target profile for revision attribute

Synopsis

```
<xsl:param name="profile.revision" select=""/>
```

Description

Value of this parameter specifies profiles which should be included in the output. You can specify multiple profiles by separating them by semicolon. You can change separator character by [profile.separator](#) parameter.

This parameter has effect only when you are using profiling stylesheets (`profile-docbook.xsl`, `profile-chunk.xsl`, ...) instead of normal ones (`docbook.xsl`, `chunk.xsl`, ...).

19.7 profile.revisionflag

profile.revisionflag — Target profile for revisionflag attribute

Synopsis

```
<xsl:param name="profile.revisionflag" select=""/>
```

Description

Value of this parameter specifies profiles which should be included in the output. You can specify multiple profiles by separating them by semicolon. You can change separator character by [profile.separator](#) parameter.

This parameter has effect only when you are using profiling stylesheets (`profile-docbook.xsl`, `profile-chunk.xsl`, ...) instead of normal ones (`docbook.xsl`, `chunk.xsl`, ...).

19.8 profile.role

profile.role — Target profile for role attribute

Synopsis

```
<xsl:param name="profile.role" select=""/>
```

Description

Value of this parameter specifies profiles which should be included in the output. You can specify multiple profiles by separating them by semicolon. You can change separator character by [profile.separator](#) parameter.

This parameter has effect only when you are using profiling stylesheets (`profile-docbook.xsl`, `profile-chunk.xsl`, ...) instead of normal ones (`docbook.xsl`, `chunk.xsl`, ...).

19.9 profile.security

profile.security — Target profile for security attribute

Synopsis

```
<xsl:param name="profile.security" select=""/>
```

Description

Value of this parameter specifies profiles which should be included in the output. You can specify multiple profiles by separating them by semicolon. You can change separator character by [profile.separator](#) parameter.

This parameter has effect only when you are using profiling stylesheets (`profile-docbook.xsl`, `profile-chunk.xsl`, ...) instead of normal ones (`docbook.xsl`, `chunk.xsl`, ...).

19.10 profile.userlevel

profile.userlevel — Target profile for userlevel attribute

Synopsis

```
<xsl:param name="profile.userlevel" select=""/>
```

Description

Value of this parameter specifies profiles which should be included in the output. You can specify multiple profiles by separating them by semicolon. You can change separator character by [profile.separator](#) parameter.

This parameter has effect only when you are using profiling stylesheets (`profile-docbook.xsl`, `profile-chunk.xsl`, ...) instead of normal ones (`docbook.xsl`, `chunk.xsl`, ...).

19.11 profile.vendor

profile.vendor — Target profile for vendor attribute

Synopsis

```
<xsl:param name="profile.vendor" select=""/>
```

Description

Value of this parameter specifies profiles which should be included in the output. You can specify multiple profiles by separating them by semicolon. You can change separator character by [profile.separator](#) parameter.

This parameter has effect only when you are using profiling stylesheets (`profile-docbook.xsl`, `profile-chunk.xsl`, ...) instead of normal ones (`docbook.xsl`, `chunk.xsl`, ...).

19.12 profile.attribute

profile.attribute — Name of user-specified profiling attribute

Synopsis

```
<xsl:param name="profile.attribute" select=""/>
```

Description

This parameter is used in conjunction with [profile.value](#).

This parameter has effect only when you are using profiling stylesheets (`profile-docbook.xsl`, `profile-chunk.xsl`, ...) instead of normal ones (`docbook.xsl`, `chunk.xsl`, ...).

19.13 profile.value

profile.value — Target profile for user-specified attribute

Synopsis

```
<xsl:param name="profile.value" select=""/>
```

Description

When you are using this parameter you must also specify name of profiling attribute with parameter [profile.attribute](#).

Value of this parameter specifies profiles which should be included in the output. You can specify multiple profiles by separating them by semicolon. You can change separator character by [profile.separator](#) parameter.

This parameter has effect only when you are using profiling stylesheets (`profile-docbook.xsl`, `profile-chunk.xsl`, ...) instead of normal ones (`docbook.xsl`, `chunk.xsl`, ...).

19.14 profile.separator

profile.separator — Separator character for compound profile values

Synopsis

```
<xsl:param name="profile.separator" select="','"/>
```

Description

Separator character for compound profile values.

Chapter 20

HTML Help

20.1 `htmlhelp.encoding`

`htmlhelp.encoding` — Character encoding to use in files for HTML Help compiler.

Synopsis

```
<xsl:param name="htmlhelp.encoding" select="'iso-8859-1'"/>
```

Description

HTML Help Compiler is not UTF-8 aware, so you should always use appropriate single-byte encoding here.

20.2 `htmlhelp.autolabel`

`htmlhelp.autolabel` — Should tree-like ToC use autonumbering feature?

Synopsis

```
<xsl:param name="htmlhelp.autolabel" select="0"/>
```

Description

If you want to include chapter and section numbers into ToC in the left panel, set this parameter to 1.

20.3 `htmlhelp.chm`

`htmlhelp.chm` — Filename of output HTML Help file.

Synopsis

```
<xsl:param name="htmlhelp.chm" select="'htmlhelp.chm'"/>
```

Description

Change this parameter if you want different name of result CHM file than htmlhelp.chm.

20.4 htmlhelp.default.topic

htmlhelp.default.topic — Name of file with default topic

Synopsis

```
<xsl:param name="htmlhelp.default.topic" select=""/>
```

Description

Normally first chunk of document is displayed when you open HTML Help file. If you want to display another topic, simply set its filename by this parameter.

This is useful especially if you don't generate ToC in front of your document and you also hide root element in ToC. E.g.:

```
<xsl:param name="generate.book.toc" select="0"/>
<xsl:param name="htmlhelp.hhc.show.root" select="0"/>
<xsl:param name="htmlhelp.default.topic" select="'pr01.html'"/>
```

20.5 htmlhelp.hhp

htmlhelp.hhp — Filename of project file.

Synopsis

```
<xsl:param name="htmlhelp.hhp" select="'htmlhelp.hhp'"/>
```

Description

Change this parameter if you want different name of project file than htmlhelp.hhp.

20.6 htmlhelp.hhc

htmlhelp.hhc — Filename of TOC file.

Synopsis

```
<xsl:param name="htmlhelp.hhc" select="'toc.hhc'"/>
```

Description

Change this parameter if you want different name of TOC file than toc.hhc.

20.7 htmlhelp.hhk

htmlhelp.hhk — Filename of index file.

Synopsis

```
<xsl:param name="htmlhelp.hhk" select="'index.hhk'"/>
```

Description

Change this parameter if you want different name of index file than index.hhk.

20.8 htmlhelp.hhp.tail

htmlhelp.hhp.tail — Additional content for project file.

Synopsis

```
<xsl:param name="htmlhelp.hhp.tail"/>
```

Description

If you want to include some additional parameters into project file, store appropriate part of project file into this parameter.

20.9 htmlhelp.hhp.window

htmlhelp.hhp.window — Name of default window.

Synopsis

```
<xsl:param name="htmlhelp.hhp.window" select="'Main'"/>
```

Description

Name of default window. If empty no [WINDOWS] section will be added to project file.

20.10 htmlhelp.enumerate.images

htmlhelp.enumerate.images — Should be paths to all used images added to project file?

Synopsis

```
<xsl:param name="htmlhelp.enumerate.images" select="0"/>
```

Description

You should turn on this flag, if you insert images into your documents as external binary entities or if you are using absolute path in image names.

20.11 `htmlhelp.force.map.and.alias`

`htmlhelp.force.map.and.alias` — Should be [MAP] and [ALIAS] section added to project file unconditionally?

Synopsis

```
<xsl:param name="htmlhelp.force.map.and.alias" select="0"/>
```

Description

You should turn on this flag, if you have your own `alias.h` and `context.h` files and you want include reference to them in project file.

20.12 `htmlhelp.map.file`

`htmlhelp.map.file` — Filename of map file.

Synopsis

```
<xsl:param name="htmlhelp.map.file" select="'context.h'"/>
```

Description

Change this parameter if you want different name of map file than `context.h`.

20.13 `htmlhelp.alias.file`

`htmlhelp.alias.file` — Filename of map file.

Synopsis

```
<xsl:param name="htmlhelp.alias.file" select="'alias.h'"/>
```

Description

Change this parameter if you want different name of map file than `alias.h`.

20.14 `htmlhelp.hhc.section.depth`

`htmlhelp.hhc.section.depth` — Depth of TOC for sections in a left pane.

Synopsis

```
<xsl:param name="htmlhelp.hhc.section.depth" select="5"/>
```

Description

Change this parameter if you want shallower ToC in a left pane of HTML Help viewer.

20.15 htmlhelp.hhc.show.root

htmlhelp.hhc.show.root — Should be entry for root element shown in ToC?

Synopsis

```
<xsl:param name="htmlhelp.hhc.show.root" select="1"/>
```

Description

If set to 0, there will be no entry for root element in ToC. This is useful when you want provide user with expanded ToC as a default.

20.16 htmlhelp.hhc.folders.instead.books

htmlhelp.hhc.folders.instead.books —

Synopsis

```
<xsl:param name="htmlhelp.hhc.folders.instead.books" select="1"/>
```

Description

This parameter controls whether there should be folder-like icons (1) or book-like icons (0) in ToC.

20.17 htmlhelp.hhc.binary

htmlhelp.hhc.binary —

Synopsis

```
<xsl:param name="htmlhelp.hhc.binary" select="1"/>
```

Description

This parametr controls whether binary TOC will be generated. You must create binary TOC if you want to add Prev/Next buttons to toolbar (which is default behaviour). Files with binary TOC can't be merged.

20.18 htmlhelp.title

htmlhelp.title — Title of HTML Help

Synopsis

```
<xsl:param name="htmlhelp.title" select=""/>
```

Description

Content of this parameter will be used as a title for generated HTML Help. If empty, title will be automatically taken from document.

20.19 htmlhelp.show.menu

htmlhelp.show.menu — Should be menu shown?

Synopsis

```
<xsl:param name="htmlhelp.show.menu" select="0"/>
```

Description

If you want application menu in your HTML Help file, turn this parameter to 1.

20.20 htmlhelp.show.advanced.search

htmlhelp.show.advanced.search — Should be advanced search available?

Synopsis

```
<xsl:param name="htmlhelp.show.advanced.search" select="0"/>
```

Description

If you want advanced search features in your help, turn this parameter to 1.

20.21 htmlhelp.show.favorites

htmlhelp.show.favorites — Should be favorites tab shown?

Synopsis

```
<xsl:param name="htmlhelp.show.favorites" select="0"/>
```

Description

If you want favourites tab shown in your help, turn this parameter to 1.

20.22 `htmlhelp.button.hideshow`

`htmlhelp.button.hideshow` — Should be Hide/Show button shown?

Synopsis

```
<xsl:param name="htmlhelp.button.hideshow" select="1"/>
```

Description

If you want Hide/Show button shown on toolbar, turn this parameter to 1.

20.23 `htmlhelp.button.back`

`htmlhelp.button.back` — Should be Back button shown?

Synopsis

```
<xsl:param name="htmlhelp.button.back" select="1"/>
```

Description

If you want Back button shown on toolbar, turn this parameter to 1.

20.24 `htmlhelp.button.forward`

`htmlhelp.button.forward` — Should be Forward button shown?

Synopsis

```
<xsl:param name="htmlhelp.button.forward" select="0"/>
```

Description

If you want Forward button shown on toolbar, turn this parameter to 1.

20.25 `htmlhelp.button.stop`

`htmlhelp.button.stop` — Should be Stop button shown?

Synopsis

```
<xsl:param name="htmlhelp.button.stop" select="0"/>
```

Description

If you want Stop button shown on toolbar, turn this parameter to 1.

20.26 htmlhelp.button.refresh

htmlhelp.button.refresh — Should be Refresh button shown?

Synopsis

```
<xsl:param name="htmlhelp.button.refresh" select="0"/>
```

Description

If you want Refresh button shown on toolbar, turn this parameter to 1.

20.27 htmlhelp.button.home

htmlhelp.button.home — Should be Home button shown?

Synopsis

```
<xsl:param name="htmlhelp.button.home" select="0"/>
```

Description

If you want Home button shown on toolbar, turn this parameter to 1.

20.28 htmlhelp.button.home.url

htmlhelp.button.home.url — URL address of page accessible by Home button

Synopsis

```
<xsl:param name="htmlhelp.button.home.url"/>
```

Description

URL address of page accessible by Home button.

20.29 htmlhelp.button.options

htmlhelp.button.options — Should be Options button shown?

Synopsis

```
<xsl:param name="htmlhelp.button.options" select="1"/>
```

Description

If you want Options button shown on toolbar, turn this parameter to 1.

20.30 htmlhelp.button.print

htmlhelp.button.print — Should be Print button shown?

Synopsis

```
<xsl:param name="htmlhelp.button.print" select="1"/>
```

Description

If you want Print button shown on toolbar, turn this parameter to 1.

20.31 htmlhelp.button.locate

htmlhelp.button.locate — Should be Locate button shown?

Synopsis

```
<xsl:param name="htmlhelp.button.locate" select="0"/>
```

Description

If you want Locate button shown on toolbar, turn this parameter to 1.

20.32 htmlhelp.button.jump1

htmlhelp.button.jump1 — Should be Jump1 button shown?

Synopsis

```
<xsl:param name="htmlhelp.button.jump1" select="0"/>
```

Description

If you want Jump1 button shown on toolbar, turn this parameter to 1.

20.33 `htmlhelp.button.jump1.url`

`htmlhelp.button.jump1.url` — URL address of page accessible by Jump1 button

Synopsis

```
<xsl:param name="htmlhelp.button.jump1.url"/>
```

Description

URL address of page accessible by Jump1 button.

20.34 `htmlhelp.button.jump1.title`

`htmlhelp.button.jump1.title` — Title of Jump1 button

Synopsis

```
<xsl:param name="htmlhelp.button.jump1.title" select="'User1'"/>
```

Description

Title of Jump1 button.

20.35 `htmlhelp.button.jump2`

`htmlhelp.button.jump2` — Should be Jump2 button shown?

Synopsis

```
<xsl:param name="htmlhelp.button.jump2" select="0"/>
```

Description

If you want Jump2 button shown on toolbar, turn this parameter to 1.

20.36 `htmlhelp.button.jump2.url`

`htmlhelp.button.jump2.url` — URL address of page accessible by Jump2 button

Synopsis

```
<xsl:param name="htmlhelp.button.jump2.url"/>
```

Description

URL address of page accessible by Jump2 button.

20.37 htmlhelp.button.jump2.title

htmlhelp.button.jump2.title — Title of Jump2 button

Synopsis

```
<xsl:param name="htmlhelp.button.jump2.title" select="'User2'"/>
```

Description

Title of Jump2 button.

20.38 htmlhelp.button.next

htmlhelp.button.next — Should be Next button shown?

Synopsis

```
<xsl:param name="htmlhelp.button.next" select="1"/>
```

Description

If you want Next button shown on toolbar, turn this parameter to 1.

20.39 htmlhelp.button.prev

htmlhelp.button.prev — Should be Prev button shown?

Synopsis

```
<xsl:param name="htmlhelp.button.prev" select="1"/>
```

Description

If you want Prev button shown on toolbar, turn this parameter to 1.

20.40 htmlhelp.button.zoom

htmlhelp.button.zoom — Should be Zoom button shown?

Synopsis

```
<xsl:param name="htmlhelp.button.zoom" select="0"/>
```

Description

If you want Zoom button shown on toolbar, turn this parameter to 1.

20.41 htmlhelp.use.hhk

htmlhelp.use.hhk — Should be index built using HHK file?

Synopsis

```
<xsl:param name="htmlhelp.use.hhk" select="0"/>
```

Description

If non-zero, index is created using HHK file. This provides some new features.

20.42 htmlhelp.only

htmlhelp.only — Should be only project files generated?

Synopsis

```
<xsl:param name="htmlhelp.only" select="0"/>
```

Description

If you want to play with various HTML Help parameters and you don't need to regenerate all HTML files, you can set this parameter to 1. This setting will not process whole document, only project files (hhp, hhc, hhk,...) will be generated.

Appendix A

The Stylesheet

The `param.xml` stylesheet is just a wrapper around all these parameters. <!-- This file is generated from param.xweb; do not edit this file! --> <xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" exclude-result-prefixes="src" version="1.0">
<!-- ***** \$Id: param.xml,v 1.1.2.1 2002/09/04 13:50:47 jdj Exp \$ ***** This file is part of the XSL DocBook Stylesheet distribution. See ../README or http://nwalsh.com/docbook/xsl/ for copyright and other information. ***** --> </xsl:stylesheet>