DocBook XSL Stylesheets: Reference DocumentationTable of ContentsDocBook XSL Stylesheets User Reference: Parame-
tersPart I. HTML Parameter ReferenceAdmonitionsadmon.graphics.extensionSynopsisDescriptionadmon.graphics.pathSynopsisDescrip
Generationannotate.tocSynopsisDescriptionautotoc.label.separatorSynopsisDescriptionautotoc.label.in.hyperlinkSynopsisDescriptionpr
Extensionslinenumbering.everyNthSynopsisDescriptionlinenumbering.extensionSynopsisDescriptionlinenumbering.separatorSynopsis
AlsoMore informationtextdata.default.encodingSynopsisDescriptiongraphicsize.extensionSynopsisDescriptiongraphicsize.use.img.src.p
labellingchapter.autolabelSynopsisDescriptionappendix.autolabelSynopsisDescriptionpart.autolabelSynopsisDescriptionreference.autol
ProcessingrootidSynopsisDescriptionsuppress.navigationSynopsisDescriptionsuppress.header.navigationSynopsisDescriptionsuppress.f
and Titlepagesinherit.keywordsSynopsisDescriptionmake.single.year.rangesSynopsisDescriptionmake.year.rangesSynopsisDescriptiona
the default valuehtml.head.legalnotice.link.multipleSynopsisDescriptionReference Pagesfuncsynopsis.decorationSynopsisDescriptionfu
Referencescollect.xref.targetsSynopsisDescriptioninsert.xref.page.numberSynopsisDescriptionuse.role.as.xrefstyleSynopsisDescription
page.separatorSynopsisDescriptionxref.label-title.separatorSynopsisDescriptionxref.title-page.separatorSynopsisDescriptionListssegme
Up the Glossary DatabaseMarking Up Glossary TermsMarking Up the GlossaryFormatting the DocumentLimitationsglossary.sortSynops
Parameterstex.math.fileSynopsisDescriptionRelated ParametersMore informationtex.math.delimsSynopsisDescriptionRelated Pa-
rametersSee Alsopixels.per.inchSynopsisDescriptionpoints.per.emSynopsisDescriptionuse.svgSynopsisDescriptionmenuchoice.separato
section-elementsSynopsisDescriptionchunker.output.doctype-publicSynopsisDescriptionchunker.output.doctype-systemSynopsisDescri
typeSynopsisDescriptionchunker.output.methodSynopsisDescriptionchunker.output.omit-xml-declarationSynopsisDescriptionchunker.o
Helphtmlhelp.encodingSynopsisDescriptionhtmlhelp.autolabelSynopsisDescriptionhtmlhelp.chmSynopsisDescriptionhtmlhelp.default.
Help Platformeclipse.autolabelSynopsisDescriptioneclipse.plugin.nameSynopsisDescriptioneclipse.plugin.idSynopsisDescriptioneclips
Parameter ReferenceAdmonitionsadmon.graphicsSynopsisDescriptionadmon.graphics.extensionSynopsisDescriptionadmon.graphics.pa
Generationautotoc.label.separatorSynopsisDescriptionprocess.empty.source.tocSynopsisDescriptionprocess.source.tocSynopsisDescrip
Extensionsarbortext.extensionsSynopsisDescriptionaxf.extensionsSynopsisDescriptionfop.extensionsSynopsisDescriptionfop1.extensio
Parameterstex.math.delimsSynopsisDescriptionRelated ParametersSee Alsoxep.extensionsSynopsisDescriptionStylesheet Exten-
sionslinenumbering.everyNthSynopsisDescriptionlinenumbering.extensionSynopsisDescriptionlinenumbering.separatorSynopsisDescri
AlsoMore informationtextdata.default.encodingSynopsisDescriptionuse.extensionsSynopsisDescriptionAutomatic labellingappendix.au
ProcessingrootidSynopsisDescriptionMeta/*Infomake.single.year.rangesSynopsisDescriptionmake.year.rangesSynopsisDescriptionauth
Pagesfuncsynopsis.decorationSynopsisDescriptionfuncsynopsis.styleSynopsisDescriptionfunction.parensSynopsisDescriptionrefentry.g
Referencesinsert.xref.page.numberSynopsisDescriptionxref.propertiesSynopsisDescriptionxref.label-title.separatorSynopsisDescription
page.separatorSynopsisDescriptionxref.title-page.separatorSynopsisDescriptioninsert.link.page.numberSynopsisDescriptionListscompa
Up the Glossary DatabaseMarking Up Glossary TermsMarking Up the GlossaryFormatting the DocumentLimitationsglossary.as.blocksS
and General StylesUnderstanding XSL FO Marginspage.heightSynopsisDescriptionpage.height.portraitSynopsisDescriptionpage.margin
heightSynopsisDescriptioncolumn.count.backSynopsisDescriptioncolumn.count.bodySynopsisDescriptioncolumn.count.frontSynopsis
Familiesbody.font.familySynopsisDescriptiondingbat.font.familySynopsisDescriptionmonospace.font.familySynopsisDescriptionsans.f
Setsformal.object.propertiesSynopsisDescriptionformal.title.propertiesSynopsisDescriptioninformal.object.propertiesSynopsisDescripti
pages Parameter ReferenceHyphenation, justification, and breakingman.hyphenateSynopsisDescriptionman.hyphenate.urlsSynopsisDes
Parametersman.font.table.headingsSynopsisDescriptionman.font.table.titleSynopsisDescriptionSYNOPSIS sectionman.funcsynopsis.st
and COPYRIGHT sectionsman.authors.section.enabledSynopsisDescriptionman.copyright.section.enabledSynopsisDescriptionEndnote
and link handlingman.endnotes.list.enabledSynopsisDescriptionman.endnotes.list.headingSynopsisDescriptionman.endnotes.are.numbe
Parametersman.base.url.for.relative.linksSynopsisDescriptionRelated ParametersListsman.segtitle.suppressSynopsisDescriptionCharact
substitutionman.charmap.enabledSynopsisDescriptionDetailsman.charmap.uriSynopsisDescriptionman.charmap.use.subsetSynopsisDe
Unicode block names and "class" valuesman.charmap.subset.profile.englishSynopsisDescriptionman.string.subst.map.local.preSynopsis
of the substitution mapman.string.subst.map.local.postSynopsisDescriptionRefentry metadata gatheringrefentry.meta.get.quietlySynopsi
header/footerman.th.extra1.suppressSynopsisDescriptionman.th.extra2.suppressSynopsisDescriptionman.th.extra3.suppressSynopsisDe
Parameter Referencewordml.templateSynopsisDescriptionpages.templateSynopsisDescriptionPart V. Slides Parameter ReferenceX-
HTML: General Paramsdisable.collapsibleSynopsisDescriptiondisable.incrementalSynopsisDescriptiongenerate.copyrightSynopsisDes
S5 Paramss5.controlsSynopsisDescriptions5.defaultviewSynopsisDescriptions5.opera.cssSynopsisDescriptions5.outline.cssSynopsisDe
Slidy Paramsslidy.durationSynopsisDescriptionslidy.path.prefixSynopsisDescriptionslidy.slidy.cssSynopsisDescriptionslidy.slidy.jsSyn
General Paramsfoil.title.masterSynopsisDescriptionfoil.title.sizeSynopsisDescriptiongenerate.copyrightSynopsisDescriptiongenerate.fo
Property Setsfoil.header.propertiesSynopsisDescriptionfoil.master.propertiesSynopsisDescriptionfoil.page-sequence.propertiesSynopsis
after.propertiesSynopsisDescriptionfoil.region-before.propertiesSynopsisDescriptionfoil.region-body.propertiesSynopsisDescriptionfoil
body.propertiesSynopsisDescriptionslides.titlepage.corpauthor.propertiesSynopsisDescriptionslides.titlepage.title.propertiesSynopsisDe
site Parameter ReferenceGeneral Parametersautolayout-fileSynopsisDescriptionbody.attributesSynopsisDescriptioncurrentpage.markerS
runSynopsisDescriptionfeedback.hrefSynopsisDescriptionfeedback.link.textSynopsisDescriptionfeedback.with.idsSynopsisDescription
prefixSynopsisDescriptionfooter.hrSynopsisDescriptionheader.hrSynopsisDescriptionoutput-rootSynopsisDescriptionrebuild-allSynops
Parametersbanner.before.navigationSynopsisDescriptionnavbgcolorSynopsisDescriptionnavbodywidthSynopsisDescriptionnav.table.su
Parameterstoc.blank.graphicSynopsisDescriptiontoc.blank.imageSynopsisDescriptiontoc.blank.textSynopsisDescriptiontoc.pointer.grap
XSL Stylesheets User Reference: PIsPart I. HTML Processing Instruction Referencedbhtml_background-colorSynopsisDescriptionPara
Information in DocBook XSL: The Complete Guidedbhtml_bgcolorSynopsisDescriptionParametersRelated Information in Doc-

PDF BY DBLATEX

Book XSL: The Complete Guidedbhtml_cellpaddingSynopsisDescriptionParametersRelated Global ParametersRelated Informa-
tion in DocBook XSL: The Complete Guidedbhtml_cellspacingSynopsisDescriptionParametersRelated Global ParametersRe-
lated Information in DocBook XSL: The Complete Guidedbhtml_classSynopsisDescriptionParametersRelated Information in
DocBook XSL: The Complete Guidedbhtml_dirSynopsisDescriptionParametersRelated Global ParametersRelated Information
in DocBook XSL: The Complete Guidedbhtml_filenameSynopsisDescriptionParametersRelated Global ParametersRelated In-
formation in DocBook XSL: The Complete Guidedbhtml_funcsynopsis-styleSynopsisDescriptionParametersRelated Global Pa-
rametersdbhtml_img.src.pathSynopsisDescriptionParametersRelated Global ParametersRelated Information in DocBook XSL:
The Complete Guidedbhtml_label-widthSynopsisDescriptionParametersRelated Information in DocBook XSL: The Complete
Guidedbhtml_linenumbering.everyNthSynopsisDescriptionParametersRelated Global ParametersRelated Information in Doc-
Book XSL: The Complete Guidedbhtml_linenumbering.separatorSynopsisDescriptionParametersRelated Global ParametersRe-
lated Information in DocBook XSL: The Complete Guidedbhtml_linenumbering.widthSynopsisDescriptionParametersRelated
Global ParametersRelated Information in DocBook XSL: The Complete Guidedbhtml_list-presentationSynopsisDescriptionParametersR
Global ParametersRelated Information in DocBook XSL: The Complete Guidedbhtml_list-widthSynopsisDescriptionParametersRelated
Information in DocBook XSL: The Complete Guidedbhtml_row-heightSynopsisDescriptionParametersRelated Information in
DocBook XSL: The Complete Guidedbhtml_startSynopsisDescriptionParametersRelated Information in DocBook XSL: The
Complete Guidedbhtml_stop-chunkingSynopsisDescriptionRelated Information in DocBook XSL: The Complete Guidedbhtml_table-
summarySynopsisDescriptionParametersRelated Information in DocBook XSL: The Complete Guidedbhtml_table-widthSynopsisDescr
Global ParametersRelated Information in DocBook XSL: The Complete Guidedbhtml_term-presentationSynopsisDescriptionParameters
Information in DocBook XSL: The Complete Guidedbhtml_term-separatorSynopsisDescriptionParametersRelated Global Pa-
rametersRelated Information in DocBook XSL: The Complete Guidedbhtml_term-widthSynopsisDescriptionParametersRelated
Information in DocBook XSL: The Complete Guidedbhtml_tocSynopsisDescriptionParametersRelated Information in DocBook
XSL: The Complete GuidedbcmdlistSynopsisDescriptionParametersdbfunclistSynopsisDescriptionParametersdbhtml-include_hrefSyno
Global ParametersRelated Information in DocBook XSL: The Complete GuidedbhhSynopsisDescriptionParametersRelated In-
formation in DocBook XSL: The Complete GuidePart II. FO Processing Instruction Referencedbfo_background-colorSynopsisDescriptio
Information in DocBook XSL: The Complete Guidedbfo_bgcolorSynopsisDescriptionParametersRelated Information in Doc-
Book XSL: The Complete Guidedbfo_float-typeSynopsisDescriptionParametersRelated Global ParametersRelated Information
in DocBook XSL: The Complete Guidedbfo_funcsynopsis-styleSynopsisDescriptionParametersRelated Global Parametersdbfo_glossary
presentationSynopsisDescriptionParametersRelated Global ParametersRelated Information in DocBook XSL: The Complete
Guidedbfo_glosslist-presentationSynopsisDescriptionParametersRelated Global ParametersRelated Information in DocBook XSL:
The Complete Guidedbfo_glossterm-widthSynopsisDescriptionParametersRelated Global ParametersRelated Information in Doc-
Book XSL: The Complete Guidedbfo_keep-togetherSynopsisDescriptionParametersRelated Global ParametersRelated Informa-
tion in DocBook XSL: The Complete Guidedbfo_label-widthSynopsisDescriptionParametersRelated Information in DocBook
XSL: The Complete Guidedbfo_linenumbering.everyNthSynopsisDescriptionParametersRelated Global ParametersRelated In-
formation in DocBook XSL: The Complete Guidedbfo_linenumbering.separatorSynopsisDescriptionParametersRelated Global
ParametersRelated Information in DocBook XSL: The Complete Guidedbfo_linenumbering.widthSynopsisDescriptionParametersRelate
Global ParametersRelated Information in DocBook XSL: The Complete Guidedbfo_list-presentationSynopsisDescriptionParametersRel
Global ParametersRelated Information in DocBook XSL: The Complete Guidedbfo_list-widthSynopsisDescriptionParametersdbfo_orie
Global Parametersdbfo_rotated-widthSynopsisDescriptionParametersdbfo_sidebar-widthSynopsisDescriptionParametersRelated
Global ParametersRelated Information in DocBook XSL: The Complete Guidedbfo_startSynopsisDescriptionParametersRelated
Information in DocBook XSL: The Complete Guidedbfo_table-widthSynopsisDescriptionParametersRelated Information in Doc-
Book XSL: The Complete Guidedbfo_term-widthSynopsisDescriptionParametersRelated Information in DocBook XSL: The
Complete Guidedbfo_tocSynopsisDescriptionParametersRelated Information in DocBook XSL: The Complete Guidedbfo-needSynopsis
Information in DocBook XSL: The Complete Guidedbfo_row-heightSynopsisDescriptionParametersRelated Information in Doc-
Book XSL: The Complete GuidePart III. manpages Processing Instruction Referencedbman_funcsynopsis-styleSynopsisDescriptionPara
Global ParametersPart IV. Common Processing Instruction Referencedbchoice_choiceSynopsisDescriptionParametersdbtimestampSyno
Global ParametersDocBook XSL Stylesheets Developer ReferencePart I. XSL Library Template ReferenceGeneral Library
Templatesdot.countDescriptioncopy-stringDescriptionstring.substDescriptionxpointer.idrefDescriptionidrefDescriptionlength-magnitud
unitsDescriptionlength-specDescriptionlength-in-pointsDescriptionpi-attributeDescriptionlookup.keyDescriptionxpath.locationDescript
escape-stringDescriptioncomment-escape-string.recursiveDescriptiontrim.textDescriptionstr.tokenize.keep.delimitersDescriptionapply-
string-subst-mapDescriptionRelative URI Functionscount.uri.path.depthDescriptiontrim.common.uri.pathsDescriptionresolve.pathDesc
mon Template ReferenceCommon » Base Template Referenceis.componentSynopsisDescriptionParametersReturnsis.sectionSynopsisDe
typeSynopsisDescriptionCommon » Refentry Metadata Template Referenceget.refentry.metadataSynopsisDescriptionParametersReturn
» Utility Template Referencelog.messageSynopsisDescriptionParametersReturnsget.doc.titleSynopsisDescriptionReturnspad-stringSyno
» Character-Map Template Referenceapply-character-mapSynopsisDescriptionParametersread-character-mapSynopsisDescriptionPara
matting Object Table Referencecalc.column.widthSynopsisDescriptionParametersReturnsPart IV. Titlepage Template Stylesheet
Referencet:templatesSynopsisDescriptionxsl:*SynopsisDescriptiont:titlepageSynopsisDescription@* (in copy.literal.atts mode)Synopsi
contentSynopsisDescriptiont:titlepage-separatorSynopsisDescriptiont:titlepage-beforeSynopsisDescription* (in copy mode)SynopsisDe

PDF BY DBLATEX

(in copy mode)SynopsisDescription* (in document.order mode)SynopsisDescription* (in document.order mode)SynopsisDescription* (in titlepage.specialrules mode)SynopsisDescription* (in titlepage.subrules mode)SynopsisDescriptiont:orSynopsisDescriptiont:or (in titlepage.subrules mode)SynopsisDescriptionelement-or-listSynopsisDescriptionDocBook XSL Stylesheets: Reference DocumentationNorman Walsh The DocBook Project $Id: reference.xml 9625 2012-10-20 23:12:33Z dcramer $ Copyright © 1999-2007 Norman WalshCopyright © 2003 Jiří KosekCopyright © 2004-2007 Steve BallCopyright © 2001-2007 The DocBook ProjectLicense Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions: The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software. Except as contained in this notice, the names of individuals credited with contribution to this software shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization from the individuals in question. Any stylesheet derived from this Software that is publically distributed will be identified with a different name and the version strings in any derived Software will be changed so that no possibility of confusion between the derived package and this Software will exist. Warranty THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL NORMAN WALSH OR ANY OTHER CONTRIBUTOR BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE. About this documentThis is generated reference documentation for the DocBook XSL stylesheets. It is available in the following formats: •HTML11 http://docbook.sourceforge.net/release/xsl/current/doc/reference.html, PDF22 http://docbook.sourceforge.net/release/xsl/current/doc/reference.pdf, plain text33 http://docbook.sourceforge.net/release/xsl/current/doc This is primarily documentation on the parameters and processing instructions you can use to control the behavior of the stylesheets. NoteThis is purely reference documentation – not how-to documentation. For a thorough step-by-step how-to guide to publishing content using the DocBook XSL stylesheets, see Bob Stayton's DocBook XSL: The Complete Guide44 http://www.sagehill.net/book-description.html, available online at http://www.sagehill.net/docbookxsl/index.html This document is divided into three sets of references: the first two sets provides user documentation; the third, developer documentation.DocBook XSL Stylesheets: Reference DocumentationDocBook XSL Stylesheets: Reference DocumentationTable of ContentsDocBook XSL Stylesheets User Reference: ParametersI. HTML Parameter ReferenceI. Admonitionsadmon.graphics.extensionadmon.graphics.pat Calloutscallout.defaultcolumncallout.graphics.extensioncallout.graphics.number.limitcallout.graphics.pathcallout.graphicscallout.list.ta EBNFebnf.table.bgcolorebnf.table.borderebnf.assignmentebnf.statement.terminatorIV. ToC/LoT/Index Generationannotate.tocautotoc.l Stylesheet Extensionslinenumbering.everyNthlinenumbering.extensionlinenumbering.separatorlinenumbering.widthtablecolumns.exten Automatic labellingchapter.autolabelappendix.autolabelpart.autolabelreference.autolabelpreface.autolabelqandadiv.autolabelsection.aut HTMLhtml.basehtml.stylesheethtml.stylesheet.typecss.decorationhtml.scripthtml.script.typespacing.parasemphasis.propagates.stylepar XSLT Processingrootidsuppress.navigationsuppress.header.navigationsuppress.footer.navigationheader.rulefooter.ruleid.warningsIX. Meta/*Info and Titlepagesinherit.keywordsmake.single.year.rangesmake.year.rangesauthor.othername.in.middleblurb.on.titlepage.enabl Reference Pagesfuncsynopsis.decorationfuncsynopsis.stylefunction.parensrefentry.generate.namerefentry.generate.titlerefentry.xref.mar Tablesdefault.table.widthnominal.table.widthtable.borders.with.csstable.cell.border.styletable.cell.border.thicknesstable.cell.border.colo QAndASetqanda.defaultlabelqanda.inherit.numerationqanda.in.tocqanda.nested.in.tocXIII. Linkingtarget.database.documenttargets.file Cross Referencescollect.xref.targetsinsert.xref.page.numberuse.role.as.xrefstylexref.with.number.and.titlexref.label-page.separatorxref. title.separatorxref.title-page.separatorXV. Listssegmentedlist.as.tablevariablelist.as.tablevariablelist.term.separatorvariablelist.term.brea Bibliographybibliography.stylebiblioentry.item.separatorbibliography.collectionbibliography.numberedXVII. Glossaryglossterm.auto.li Miscellaneousformal.proceduresformal.title.placementruninhead.default.title.end.punctruninhead.title.end.punctshow.commentsshow.re Annotationsannotation.supportannotation.jsannotation.cssannotation.graphic.openannotation.graphic.closeXX. Graphicsimg.src.pathke Chunkingchunker.output.cdata-section-elementschunker.output.doctype-publicchunker.output.doctype-systemchunker.output.encoding typechunker.output.methodchunker.output.omit-xml-declarationchunker.output.standalonesaxon.character.representationhtml.extuse.id. Profilingprofile.archprofile.audienceprofile.conditionprofile.conformanceprofile.langprofile.osprofile.revisionprofile.revisionflagprofile. HTML Helphtmlhelp.encodinghtmlhelp.autolabelhtmlhelp.chmhtmlhelp.default.topichtmlhelp.display.progresshtmlhelp.hhphtmlhelp.h Eclipse Help Platformeclipse.autolabeleclipse.plugin.nameeclipse.plugin.ideclipse.plugin.providerXXV. WebHelpwebhelp.autolabelwe JavaHelpjavahelp.encodingXXVII. Localizationl10n.gentext.languagel10n.gentext.default.languagel10n.gentext.use.xref.languagel10n. FO Parameter ReferenceXXVIII. Admonitionsadmon.graphicsadmon.graphics.extensionadmon.graphics.pathadmon.textlabeladmonitio Calloutscalloutlist.propertiescallout.propertiescallout.defaultcolumncallout.graphicscallout.graphics.extensioncallout.graphics.number. ToC/LoT/Index Generationautotoc.label.separatorprocess.empty.source.tocprocess.source.tocgenerate.tocgenerate.indexmake.index.ma Processor Extensionsarbortext.extensionsaxf.extensionsfop.extensionsfop1.extensionspassivetex.extensionstex.math.in.alttex.math.delin Stylesheet Extensionslinenumbering.everyNthlinenumbering.extensionlinenumbering.separatorlinenumbering.widthtablecolumns.exten Automatic labellingappendix.autolabelchapter.autolabelpart.autolabelreference.autolabelpreface.autolabelsection.autolabelsection.autol XSLT ProcessingrootidXXXV. Meta/*Infomake.single.year.rangesmake.year.rangesauthor.othername.in.middleXXXVI. Refer-

PDF BY DBLATEX

ence Pagesfuncsynopsis.decorationfuncsynopsis.stylefunction.parensrefentry.generate.namerefentry.generate.titlerefentry.pagebreakrefe
Tablesdefault.table.widthnominal.table.widthdefault.table.framedefault.table.rulestable.cell.paddingtable.frame.border.thicknesstable.fr
Linkingcurrent.docidactivate.external.olinkscollect.xref.targetsinsert.olink.page.numberinsert.olink.pdf.fragolink.base.uriolink.debugol
Cross Referencesinsert.xref.page.numberxref.propertiesxref.label-title.separatorxref.label-page.separatorxref.title-page.separatorinsert.l
Listscompact.list.item.spacingitemizedlist.propertiesitemizedlist.label.propertiesitemizedlist.label.widthlist.block.propertieslist.block.sp
QAndASetqandadiv.autolabelqanda.inherit.numerationqanda.defaultlabelqanda.in.tocqanda.nested.in.tocXLII. Bibliographybib-
liography.stylebiblioentry.item.separatorbibliography.collectionbibliography.numberedbiblioentry.propertiesXLIII. Glossaryglossterm.a
Miscellaneousformal.proceduresformal.title.placementruninhead.default.title.end.punctruninhead.title.end.punctshow.commentspunct.h
Graphicsgraphic.default.extensiondefault.image.widthpreferred.mediaobject.roleuse.role.for.mediaobjectignore.image.scalingimg.src.pa
Pagination and General Stylespage.heightpage.height.portraitpage.margin.bottompage.margin.innerpage.margin.outerpage.margin.toppa
heightcolumn.count.backcolumn.count.bodycolumn.count.frontcolumn.count.indexcolumn.count.lotcolumn.count.titlepagecolumn.gap
Font Familiesbody.font.familydingbat.font.familymonospace.font.familysans.font.familytitle.font.familysymbol.font.familyXLVIII.
Property Setsformal.object.propertiesformal.title.propertiesinformal.object.propertiesmonospace.propertiesverbatim.propertiesmonospa
Profilingprofile.archprofile.audienceprofile.conditionprofile.conformanceprofile.langprofile.osprofile.revisionprofile.revisionflagprofile.
Localizationl10n.gentext.languagel10n.gentext.default.languagel10n.gentext.use.xref.languagel10n.lang.value.rfc.compliantwriting.mo
EBNFebnf.assignmentebnf.statement.terminatorLII. Prepresscrop.markscrop.mark.widthcrop.mark.offsetcrop.mark.bleedIII. Man-
pages Parameter ReferenceLIII. Hyphenation, justification, and breakingman.hyphenateman.hyphenate.urlsman.hyphenate.filenamesma
Indentationman.indent.widthman.indent.refsectman.indent.blurbsman.indent.listsman.indent.verbatimsLV. Fontsman.font.funcprototype
SYNOPSIS sectionman.funcsynopsis.styleLVII. AUTHORS and COPYRIGHT sectionsman.authors.section.enabledman.copyright.sect
Endnotes and link handlingman.endnotes.list.enabledman.endnotes.list.headingman.endnotes.are.numberedman.base.url.for.relative.link
Listsman.segtitle.suppressLX. Character/string substitutionman.charmap.enabledman.charmap.uriman.charmap.use.subsetman.charmap
Refentry metadata gatheringrefentry.meta.get.quietlyrefentry.date.profilerefentry.date.profile.enabledrefentry.manual.profilerefentry.man
Page header/footerman.th.extra1.suppressman.th.extra2.suppressman.th.extra3.suppressman.th.title.max.lengthman.th.extra2.max.lengt
Outputman.output.manifest.enabledman.output.manifest.filenameman.output.in.separate.dirman.output.lang.in.name.enabledman.outpu
Otherman.table.footnotes.dividerman.subheading.divider.enabledman.subheading.dividerIV. Roundtrip Parameter Referencewordml.ten
Slides Parameter ReferenceLXV. XHTML: General Paramsdisable.collapsibledisable.incrementalgenerate.copyrightgenerate.foilgroup.n
XHTML: S5 Paramss5.controlss5.defaultviews5.opera.csss5.outline.csss5.path.prefixs5.print.csss5.slides.csss5.slides.jsLXVII.
XHTML: Slidy Paramsslidy.durationslidy.path.prefixslidy.slidy.cssslidy.slidy.jsslidy.user.cssLXVIII. FO: General Paramsfoil.title.maste
FO: Property Setsfoil.header.propertiesfoil.master.propertiesfoil.page-sequence.propertiesfoil.propertiesfoil.region-after.propertiesfoil.re
before.propertiesfoil.region-body.propertiesfoil.subtitle.propertiesfoil.title.propertiesfoil.footer.propertieshandoutnotes.propertiesslides.
body.propertiesslides.titlepage.corpauthor.propertiesslides.titlepage.title.propertiesslides.titlepage.subtitle.propertiesslides.titlepage.auth
Website Parameter ReferenceLXX. General Parametersautolayout-filebody.attributescurrentpage.markerdry-runfeedback.hreffeedback.l
prefixfooter.hrheader.hroutput-rootrebuild-allsequential.linkssuppress.homepage.titletable.spacer.imageLXXI. Navigation Param-
etersbanner.before.navigationnavbgcolornavbodywidthnav.table.summarynavtocwidthtextbgcolorLXXII. ToC Parameterstoc.blank.grap
XSL Stylesheets User Reference: PIsI. HTML Processing Instruction Referencedbhtml_background-colordbhtml_bgcolordbhtml_cellpa
styledbhtml_img.src.pathdbhtml_label-widthdbhtml_linenumbering.everyNthdbhtml_linenumbering.separatordbhtml_linenumbering.w
presentationdbhtml_list-widthdbhtml_row-heightdbhtml_startdbhtml_stop-chunkingdbhtml_table-summarydbhtml_table-widthdbhtml_
presentationdbhtml_term-separatordbhtml_term-widthdbhtml_tocdbcmdlistdbfunclistdbhtml-include_hrefdbhhII. FO Processing
Instruction Referencedbfo_background-colordbfo_bgcolordbfo_float-typedbfo_funcsynopsis-styledbfo_glossary-presentationdbfo_glos
presentationdbfo_glossterm-widthdbfo_keep-togetherdbfo_label-widthdbfo_linenumbering.everyNthdbfo_linenumbering.separatordbfo
presentationdbfo_list-widthdbfo_orientationdbfo_pgwidedbfo_rotated-widthdbfo_sidebar-widthdbfo_startdbfo_table-widthdbfo_term-
widthdbfo_tocdbfo-needdbfo_row-heightIII. manpages Processing Instruction Referencedbman_funcsynopsis-styleIV. Common
Processing Instruction Referencedbchoice_choicedbtimestampdbtex_delimsDocBook XSL Stylesheets Developer ReferenceI.
XSL Library Template ReferenceI. General Library Templatesdot.countcopy-stringstring.substxpointer.idrefidreflength-magnitudelength
unitslength-speclength-in-pointspi-attributelookup.keyxpath.locationcomment-escape-stringcomment-escape-string.recursivetrim.textst
string-subst-mapII. Relative URI Functionscount.uri.path.depthtrim.common.uri.pathsresolve.pathdirnameII. Common Template Ref-
erenceIII. Common » Base Template Referenceis.componentis.sectionsection.levelqanda.section.levelselect.mediaobjectselect.mediaobj
typeIV. Common » Refentry Metadata Template Referenceget.refentry.metadataget.refentry.titleget.refentry.sectionget.refentry.dateget.re
Common » Utility Template Referencelog.messageget.doc.titlepad-stringVI. Common » Character-Map Template Referenceapply-
character-mapread-character-mapIII. Formatting Object Table Referencecalc.column.widthIV. Titlepage Template Stylesheet
Referencet:templatesxsl:*t:titlepage@* (in copy.literal.atts mode)t:titlepage-contentt:titlepage-separatort:titlepage-before* (in copy
mode)@* (in copy mode)* (in document.order mode)* (in document.order mode)* (in titlepage.specialrules mode)* (in ti-
tlepage.subrules mode)t:ort:or (in titlepage.subrules mode)element-or-listDocBook XSL Stylesheets User Reference: Param-
etersDocBook XSL Stylesheets User Reference: ParametersAbstractThis is generated reference documentation for all user-
configurable parameters in the DocBook XSL stylesheets. NoteThis is purely reference documentation – not how-to documenta-
tion. For a thorough step-by-step how-to guide to publishing content using the DocBook XSL stylesheets, see Bob Stayton's Doc-
Book XSL: The Complete Guide55 http://www.sagehill.net/book-description.html, available online at http://www.sagehill.net/docbookx

PDF BY DBLATEX

DocBook XSL Stylesheets User Reference: ParametersDocBook XSL Stylesheets User Reference: ParametersTable of ContentsI.
HTML Parameter ReferenceI. Admonitionsadmon.graphics.extensionadmon.graphics.pathadmon.graphicsadmon.textlabeladmon.styleI
Calloutscallout.defaultcolumncallout.graphics.extensioncallout.graphics.number.limitcallout.graphics.pathcallout.graphicscallout.list.ta
EBNFebnf.table.bgcolorebnf.table.borderebnf.assignmentebnf.statement.terminatorIV. ToC/LoT/Index Generationannotate.tocautotoc.l
Stylesheet Extensionslinenumbering.everyNthlinenumbering.extensionlinenumbering.separatorlinenumbering.widthtablecolumns.exten
Automatic labellingchapter.autolabelappendix.autolabelpart.autolabelreference.autolabelpreface.autolabelqandadiv.autolabelsection.aut
HTMLhtml.basehtml.stylesheethtml.stylesheet.typecss.decorationhtml.scripthtml.script.typespacing.parasemphasis.propagates.stylepar
XSLT Processingrootidsuppress.navigationsuppress.header.navigationsuppress.footer.navigationheader.rulefooter.ruleid.warningsIX.
Meta/*Info and Titlepagesinherit.keywordsmake.single.year.rangesmake.year.rangesauthor.othername.in.middleblurb.on.titlepage.enabl
Reference Pagesfuncsynopsis.decorationfuncsynopsis.stylefunction.parensrefentry.generate.namerefentry.generate.titlerefentry.xref.mar
Tablesdefault.table.widthnominal.table.widthtable.borders.with.csstable.cell.border.styletable.cell.border.thicknesstable.cell.border.colo
QAndASetqanda.defaultlabelqanda.inherit.numerationqanda.in.tocqanda.nested.in.tocXIII. Linkingtarget.database.documenttargets.file
Cross Referencescollect.xref.targetsinsert.xref.page.numberuse.role.as.xrefstylexref.with.number.and.titlexref.label-page.separatorxref.
title.separatorxref.title-page.separatorXV. Listssegmentedlist.as.tablevariablelist.as.tablevariablelist.term.separatorvariablelist.term.brea
Bibliographybibliography.stylebiblioentry.item.separatorbibliography.collectionbibliography.numberedXVII. Glossaryglossterm.auto.li
Miscellaneousformal.proceduresformal.title.placementruninhead.default.title.end.punctruninhead.title.end.punctshow.commentsshow.re
Annotationsannotation.supportannotation.jsannotation.cssannotation.graphic.openannotation.graphic.closeXX. Graphicsimg.src.pathke
Chunkingchunker.output.cdata-section-elementschunker.output.doctype-publicchunker.output.doctype-systemchunker.output.encoding
typechunker.output.methodchunker.output.omit-xml-declarationchunker.output.standalonesaxon.character.representationhtml.extuse.id.
Profilingprofile.archprofile.audienceprofile.conditionprofile.conformanceprofile.langprofile.osprofile.revisionprofile.revisionflagprofile.
HTML Helphtmlhelp.encodinghtmlhelp.autolabelhtmlhelp.chmhtmlhelp.default.topichtmlhelp.display.progresshtmlhelp.hhphtmlhelp.h
Eclipse Help Platformeclipse.autolabeleclipse.plugin.nameeclipse.plugin.ideclipse.plugin.providerXXV. WebHelpwebhelp.autolabelwe
JavaHelpjavahelp.encodingXXVII. Localizationl10n.gentext.languagel10n.gentext.default.languagel10n.gentext.use.xref.languagel10n.
FO Parameter ReferenceXXVIII. Admonitionsadmon.graphicsadmon.graphics.extensionadmon.graphics.pathadmon.textlabeladmonitio
Calloutscalloutlist.propertiescallout.propertiescallout.defaultcolumncallout.graphicscallout.graphics.extensioncallout.graphics.number.l
ToC/LoT/Index Generationautotoc.label.separatorprocess.empty.source.tocprocess.source.tocgenerate.tocgenerate.indexmake.index.ma
Processor Extensionsarbortext.extensionsaxf.extensionsfop.extensionsfop1.extensionspassivetex.extensionstex.math.in.alttex.math.delin
Stylesheet Extensionslinenumbering.everyNthlinenumbering.extensionlinenumbering.separatorlinenumbering.widthtablecolumns.exten
Automatic labellingappendix.autolabelchapter.autolabelpart.autolabelreference.autolabelpreface.autolabelsection.autolabelsection.autol
XSLT ProcessingrootidXXXV. Meta/*Infomake.single.year.rangesmake.year.rangesauthor.othername.in.middleXXXVI. Refer-
ence Pagesfuncsynopsis.decorationfuncsynopsis.stylefunction.parensrefentry.generate.namerefentry.generate.titlerefentry.pagebreakrefe
Tablesdefault.table.widthnominal.table.widthdefault.table.framedefault.table.rulestable.cell.paddingtable.frame.border.thicknesstable.fr
Linkingcurrent.docidactivate.external.olinkscollect.xref.targetsinsert.olink.page.numberinsert.olink.pdf.fragolink.base.uriolink.debugol
Cross Referencesinsert.xref.page.numberxref.propertiesxref.label-title.separatorxref.label-page.separatorxref.title-page.separatorinsert.l
Listscompact.list.item.spacingitemizedlist.propertiesitemizedlist.label.propertiesitemizedlist.label.widthlist.block.propertieslist.block.sp
QAndASetqandadiv.autolabelqanda.inherit.numerationqanda.defaultlabelqanda.in.tocqanda.nested.in.tocXLII. Bibliographybib-
liography.stylebiblioentry.item.separatorbibliography.collectionbibliography.numberedbiblioentry.propertiesXLIII. Glossaryglossterm.a
Miscellaneousformal.proceduresformal.title.placementruninhead.default.title.end.punctruninhead.title.end.punctshow.commentspunct.h
Graphicsgraphic.default.extensiondefault.image.widthpreferred.mediaobject.roleuse.role.for.mediaobjectignore.image.scalingimg.src.pa
Pagination and General Stylespage.heightpage.height.portraitpage.margin.bottompage.margin.innerpage.margin.outerpage.margin.toppa
heightcolumn.count.backcolumn.count.bodycolumn.count.frontcolumn.count.indexcolumn.count.lotcolumn.count.titlepagecolumn.gap
Font Familiesbody.font.familydingbat.font.familymonospace.font.familysans.font.familytitle.font.familysymbol.font.familyXLVIII.
Property Setsformal.object.propertiesformal.title.propertiesinformal.object.propertiesmonospace.propertiesverbatim.propertiesmonospa
Profilingprofile.archprofile.audienceprofile.conditionprofile.conformanceprofile.langprofile.osprofile.revisionprofile.revisionflagprofile.
Localizationl10n.gentext.languagel10n.gentext.default.languagel10n.gentext.use.xref.languagel10n.lang.value.rfc.compliantwriting.mo
EBNFebnf.assignmentebnf.statement.terminatorLII. Prepresscrop.markscrop.mark.widthcrop.mark.offsetcrop.mark.bleedIII. Man-
pages Parameter ReferenceLIII. Hyphenation, justification, and breakingman.hyphenateman.hyphenate.urlsman.hyphenate.filenamesma
Indentationman.indent.widthman.indent.refsectman.indent.blurbsman.indent.listsman.indent.verbatimsLV. Fontsman.font.funcprototype
SYNOPSIS sectionman.funcsynopsis.styleLVII. AUTHORS and COPYRIGHT sectionsman.authors.section.enabledman.copyright.sect
Endnotes and link handlingman.endnotes.list.enabledman.endnotes.list.headingman.endnotes.are.numberedman.base.url.for.relative.link
Listsman.segtitle.suppressLX. Character/string substitutionman.charmap.enabledman.charmap.uriman.charmap.use.subsetman.charmap
Refentry metadata gatheringrefentry.meta.get.quietlyrefentry.date.profilerefentry.date.profile.enabledrefentry.manual.profilerefentry.mar
Page header/footerman.th.extra1.suppressman.th.extra2.suppressman.th.extra3.suppressman.th.title.max.lengthman.th.extra2.max.lengt
Outputman.output.manifest.enabledman.output.manifest.filenameman.output.in.separate.dirman.output.lang.in.name.enabledman.outpu
Otherman.table.footnotes.dividerman.subheading.divider.enabledman.subheading.dividerIV. Roundtrip Parameter Referencewordml.ten
Slides Parameter ReferenceLXV. XHTML: General Paramsdisable.collapsibledisable.incrementalgenerate.copyrightgenerate.foilgroup.
XHTML: S5 Paramss5.controlss5.defaultviews5.opera.csss5.outline.csss5.path.prefixs5.print.csss5.slides.csss5.slides.jsLXVII.

PDF BY DBLATEX

XHTML: Slidy Paramsslidy.durationslidy.path.prefixslidy.slidy.cssslidy.slidy.jsslidy.user.cssLXVIII. FO: General Paramsfoil.title.maste
FO: Property Setsfoil.header.propertiesfoil.master.propertiesfoil.page-sequence.propertiesfoil.propertiesfoil.region-after.propertiesfoil.r
before.propertiesfoil.region-body.propertiesfoil.subtitle.propertiesfoil.title.propertiesfoil.footer.propertieshandoutnotes.propertiesslides.
body.propertiesslides.titlepage.corpauthor.propertiesslides.titlepage.title.propertiesslides.titlepage.subtitle.propertiesslides.titlepage.auth
Website Parameter ReferenceLXX. General Parametersautolayout-filebody.attributescurrentpage.markerdry-runfeedback.hreffeedback.l
prefixfooter.hrheader.hroutput-rootrebuild-allsequential.linkssuppress.homepage.titletable.spacer.imageLXXI. Navigation Param-
etersbanner.before.navigationnavbgcolornavbodywidthnav.table.summarynavtocwidthtextbgcolorLXXII. ToC Parameterstoc.blank.grap
XSL Stylesheets User Reference: ParametersDocBook XSL Stylesheets User Reference: ParametersList of Figures1. Page Mod-
elPart I. HTML Parameter Reference This is reference documentation for all user-configurable parameters in the DocBook XSL
HTML stylesheets (for generating HTML output). HTML Parameter ReferenceHTML Parameter ReferenceTable of ContentsI.
Admonitionsadmon.graphics.extensionadmon.graphics.pathadmon.graphicsadmon.textlabeladmon.styleII. Calloutscallout.defaultcolum
EBNFebnf.table.bgcolorebnf.table.borderebnf.assignmentebnf.statement.terminatorIV. ToC/LoT/Index Generationannotate.tocautotoc.l
Stylesheet Extensionslinenumbering.everyNthlinenumbering.extensionlinenumbering.separatorlinenumbering.widthtablecolumns.exten
Automatic labellingchapter.autolabelappendix.autolabelpart.autolabelreference.autolabelpreface.autolabelqandadiv.autolabelsection.aut
HTMLhtml.basehtml.stylesheethtml.stylesheet.typecss.decorationhtml.scripthtml.script.typespacing.parasemphasis.propagates.stylepar
XSLT Processingrootidsuppress.navigationsuppress.header.navigationsuppress.footer.navigationheader.rulefooter.ruleid.warningsIX.
Meta/*Info and Titlepagesinherit.keywordsmake.single.year.rangesmake.year.rangesauthor.othername.in.middleblurb.on.titlepage.enabl
Reference Pagesfuncsynopsis.decorationfuncsynopsis.stylefunction.parensrefentry.generate.namerefentry.generate.titlerefentry.xref.mar
Tablesdefault.table.widthnominal.table.widthtable.borders.with.csstable.cell.border.styletable.cell.border.thicknesstable.cell.border.colo
QAndASetqanda.defaultlabelqanda.inherit.numerationqanda.in.tocqanda.nested.in.tocXIII. Linkingtarget.database.documenttargets.file
Cross Referencescollect.xref.targetsinsert.xref.page.numberuse.role.as.xrefstylexref.with.number.and.titlexref.label-page.separatorxref.
title.separatorxref.title-page.separatorXV. Listssegmentedlist.as.tablevariablelist.as.tablevariablelist.term.separatorvariablelist.term.brea
Bibliographybibliography.stylebiblioentry.item.separatorbibliography.collectionbibliography.numberedXVII. Glossaryglossterm.auto.li
Miscellaneousformal.proceduresformal.title.placementruninhead.default.title.end.punctruninhead.title.end.punctshow.commentsshow.re
Annotationsannotation.supportannotation.jsannotation.cssannotation.graphic.openannotation.graphic.closeXX. Graphicsimg.src.pathke
Chunkingchunker.output.cdata-section-elementschunker.output.doctype-publicchunker.output.doctype-systemchunker.output.encoding
typechunker.output.methodchunker.output.omit-xml-declarationchunker.output.standalonesaxon.character.representationhtml.extuse.id.
Profilingprofile.archprofile.audienceprofile.conditionprofile.conformanceprofile.langprofile.osprofile.revisionprofile.revisionflagprofile.
HTML Helphtmlhelp.encodinghtmlhelp.autolabelhtmlhelp.chmhtmlhelp.default.topichtmlhelp.display.progresshtmlhelp.hhphtmlhelp.h
Eclipse Help Platformeclipse.autolabeleclipse.plugin.nameeclipse.plugin.ideclipse.plugin.providerXXV. WebHelpwebhelp.autolabelwe
JavaHelpjavahelp.encodingXXVII. Localizationl10n.gentext.languagel10n.gentext.default.languagel10n.gentext.use.xref.languagel10n.
of Contentsadmon.graphics.extensionadmon.graphics.pathadmon.graphicsadmon.textlabeladmon.styleNameadmon.graphics.extension
— Filename extension for admonition graphics Synopsis<xsl:param name="admon.graphics.extension">.png</xsl:param> De-
scription Sets the filename extension to use on admonition graphics. The DocBook XSL distribution provides admonition graph-
ics in the following formats:•GIF (extension: .gif)•PNG (extension: .png)•SVG (extension: .svg)•TIFF (extension: .tif) Namead-
mon.graphics.path — Path to admonition graphics Synopsis<xsl:param name="admon.graphics.path">images/</xsl:param> De-
scription Sets the path to the directory containing the admonition graphics (caution.png, important.png etc). This location is
normally relative to the output html directory. See base.dir Nameadmon.graphics — Use graphics in admonitions? Synop-
sis<xsl:param name="admon.graphics" select="0"></xsl:param> Description If true (non-zero), admonitions are presented in an
alternate style that uses a graphic. Default graphics are provided in the distribution. Nameadmon.textlabel — Use text label
in admonitions? Synopsis<xsl:param name="admon.textlabel" select="1"></xsl:param> Description If true (non-zero), admo-
nitions are presented with a generated text label such as Note or Warning in the appropriate language. If zero, such labels
are turned off, but any title child of the admonition element are still output. The default value is 1. Nameadmon.style —
Specifies the CSS style attribute that should be added to admonitions. Synopsis<xsl:param name="admon.style"> <xsl:value-
of select="concat('margin-', $direction.align.start, ': 0.5in; margin-', $direction.align.end, ': 0.5in;')"></xsl:value-of>
</xsl:param> Description Specifies the value of the CSS style attribute that should be added to admonitions. CalloutsCall-
outsCalloutsTable of Contentscallout.defaultcolumncallout.graphics.extensioncallout.graphics.number.limitcallout.graphics.pathcallout.
— Indicates what column callouts appear in by default Synopsis<xsl:param name="callout.defaultcolumn">60</xsl:param> De-
scription If a callout does not identify a column (for example, if it uses the linerange unit), it will appear in the default column.
Namecallout.graphics.extension — Filename extension for callout graphics Synopsis<xsl:param name="callout.graphics.extension">.pn
Description Sets the filename extension to use on callout graphics. The Docbook XSL distribution provides callout graphics in
the following formats:•SVG (extension: .svg)•PNG (extension: .png)•GIF (extension: .gif) Namecallout.graphics.number.limit
— Number of the largest callout graphic Synopsis<xsl:param name="callout.graphics.number.limit">15</xsl:param> Descrip-
tion If callout.graphics is non-zero, graphics are used to represent callout numbers instead of plain text. The value of call-
out.graphics.number.limit is the largest number for which a graphic exists. If the callout number exceeds this limit, the default
presentation "(plain text instead of a graphic)" will be used. Namecallout.graphics.path — Path to callout graphics Synop-
sis<xsl:param name="callout.graphics.path">images/callouts/</xsl:param> Description Sets the path to the directory holding

PDF BY DBLATEX

the callout graphics. his location is normally relative to the output html directory. see base.dir. Always terminate the directory with / since the graphic file is appended to this string, hence needs the separator. Namecallout.graphics — Use graphics for callouts? Synopsis<xsl:param name="callout.graphics" select="1"></xsl:param> 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. Namecallout.list.table — Present callout lists using a table? Synopsis<xsl:param name="callout.list.table" -select="1"></xsl:param> Description The default presentation of calloutlists uses an HTML DL element. Some browsers don't align DLs very well if callout.graphics is 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. Namecallout.unicode.number.limit — Number of the largest unicode callout character Synopsis<xsl:param name="callout.unicode.number.limit">10</xsl:param> 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. Namecallout.unicode.start.character — First Unicode character to use, decimal value. Synopsis<xsl:param -name="callout.unicode.start.character">10102</xsl:param> 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 values 9312 and 10102 are supported in the stylesheets for this parameter. Namecallout.unicode — Use Unicode characters rather than images for callouts. Synopsis<xsl:param name="callout.unicode" -select="0"></xsl:param> 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. Namecallouts.extension — Enable the callout extension Synopsis<xsl:param name="callouts.extension" select="1"></xsl:param> Description The callouts extension processes areaset elements in programlistingco and other text-based callout elements. EBNFEBNFEBNFTable of Contentsebnf.table.bgcolorebnf.table.borderebnf.assignmentebnf.statement.terminatorNameebnf.table.bgcolor — Background color for EBNF tables Synopsis<xsl:param name="ebnf.table.bgcolor">#F5DCB3</xsl:param> Description Sets the background color for EBNF tables (a pale brown). No bgcolor attribute is output if ebnf.table.bgcolor is set to the null string. Nameebnf.table.border — Selects border on EBNF tables Synopsis<xsl:param name="ebnf.table.border" select="1"></xsl:param> Description Selects the border on EBNF tables. If non-zero, the tables have borders, otherwise they don't. Nameebnf.assignment — The EBNF production assignment operator Synopsis<xsl:param name="ebnf.assignment"> <code>::=</code> </xsl:param> Description The ebnf.assignment parameter determines what text is used to show "assignment" in productions in productionsets. While "::=" is common, so are several other operators. Nameebnf.statement.terminator — Punctuation that ends an EBNF statement. Synopsis<xsl:param name="ebnf.statement.terminator"></xsl:param> Description The ebnf.statement.terminator parameter determines what text is used to terminate each production in productionset. Some notations end each statement with a period. ToC/LoT/Index GenerationToC/LoT/Index GenerationToC/LoT/Index GenerationTable of Contentsannotate.tocautotoc.label.separatorau — Annotate the Table of Contents? Synopsis<xsl:param name="annotate.toc" select="1"></xsl:param> Description If true, TOCs will be annotated. At present, this just means that the refpurpose of refentry TOC entries will be displayed. Nameautotoc.label.separator — Separator between labels and titles in the ToC Synopsis<xsl:param name="autotoc.label.separator">. -</xsl:param> Description String used to separate labels and titles in a table of contents. Nameautotoc.label.in.hyperlink — Include label in hyperlinked titles in TOC? Synopsis<xsl:param name="autotoc.label.in.hyperlink" select="1"></xsl:param> Description If the value of autotoc.label.in.hyperlink is non-zero, labels are included in hyperlinked titles in the TOC. If it is instead zero, labels are still displayed prior to the hyperlinked titles, but are not hyperlinked along with the titles. Nameprocess.source.toc — Process a non-empty toc element if it occurs in a source document? Synopsis<xsl:param name="process.source.toc" -select="0"></xsl:param> Description Specifies that the contents of a non-empty "hard-coded" toc element in a source document are processed to generate a TOC in output. NoteThis parameter has no effect on automated generation of TOCs. An automated TOC may still be generated along with the "hard-coded" TOC. To suppress automated TOC generation, adjust the value of the generate.toc paramameter. The process.source.toc parameter also has no effect if the toc element is empty; handling for empty toc is controlled by the process.empty.source.toc parameter. Nameprocess.empty.source.toc — Generate automated TOC if toc element occurs in a source document? Synopsis<xsl:param name="process.empty.source.toc" -select="0"></xsl:param> Description Specifies that if an empty toc element is found in a source document, an automated TOC is generated at this point in the document. NoteDepending on what the value of the generate.toc parameter is, setting this parameter to 1 could result in generation of duplicate automated TOCs. So the process.empty.source.toc is primarily useful as an "override": by placing an empty toc in your document and setting this parameter to 1, you can force a TOC to be generated even if generate.toc says not to. Namebridgehead.in.toc — Should bridgehead elements appear in the TOC? Synopsis<xsl:param name="bridgehead.in.toc" select="0"></xsl:param> 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. Namesimplesect.in.toc — Should simplesect elements appear in the TOC? Synopsis<xsl:param name="simplesect.in.toc" select="0"></xsl:param> Description If non-zero, simplesects will be included in the TOC. Namemanual.toc — An explicit TOC to be used for the TOC Synopsis<xsl:param name="manual.toc"></xsl:param> Description The manual.toc identifies an explicit TOC that will be used for building the printed TOC. Nametoc.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. Nametoc.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. Nametoc.max.depth — How many levels should be created for each TOC? Synopsis<xsl:param name="toc.max.depth">8</xsl:param> Description Specifies the maximal depth of TOC on all levels. Namegenerate.toc — Control generation of ToCs and LoTs Synopsis<xsl:param name="generate.toc"> appendix toc,title article/appendix nop article toc,title book toc,title,figure,table,example,equation chapter toc,title part toc,title preface - toc,title qandadiv toc qandaset toc reference toc,title sect1 toc sect2 toc sect3 toc sect4 toc sect5 toc section toc set toc,title </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. Namegenerate.section.toc.level — Control depth of TOC generation in sections Synopsis<xsl:param name="generate.section.toc.level" select="0"></xsl:param> 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. Namegenerate.index — Do you want an index? Synopsis<xsl:param name="generate.index" select="1"></xsl:param> Description Specify if an index should be generated. Nameindex.method — Select method used to group index entries in an index Synopsis<xsl:param name="index.method">basic</xsl:param> Description This parameter lets you select which method to use for sorting and grouping index entries in an index. Indexes in Latin-based languages that have accented characters typically sort together accented words and unaccented words. Thus "Á" (U+00C1 LATIN CAPITAL LETTER A WITH ACUTE) would sort together with "A" (U+0041 LATIN CAPITAL LETTER A), so both would appear in the "A" section of the index. Languages using other alphabets (such as Russian, which is written in the Cyrillic alphabet) and languages using ideographic chararacters (such as Japanese) require grouping specific to the languages and alphabets. The default indexing method is limited. It can group accented characters in Latin-based languages only. It cannot handle non-Latin alphabets or ideographic languages. The other indexing methods require extensions of one type or another, and do not work with all XSLT processors, which is why they are not used by default. The three choices for indexing method are: basic(default) Sort and groups words based only on the Latin alphabet. Words with accented Latin letters will group and sort with their respective primary letter, but words in non-Latin alphabets will be put in the "Symbols" section of the index. kosekThis method sorts and groups words based on letter groups configured in the DocBook locale file for the given language. See, for example, the French locale file common/fr.xml. This method requires that the XSLT processor supports the EXSLT extensions (most do). It also requires support for using user-defined functions in xsl:key (xsltproc does not). This method is suitable for any language for which you can list all the individual characters that should appear in each letter group in an index. It is probably not practical to use it for ideographic languages such as Chinese that have hundreds or thousands of characters. To use the kosek method, you must: 1.Use a processor that supports its extensions, such as Saxon 6 or Xalan (xsltproc and Saxon 8 do not). 2.Set the index.method parameter's value to "kosek". 3.Import the appropriate index extensions stylesheet module fo/autoidx-kosek.xsl or html/autoidx-kosek.xsl into your customization. kimberThis method uses extensions to the Saxon processor to implement sophisticated indexing processes. It uses its own configuration file, which can include information for any number of languages. Each language's configuration can group words using one of two processes. In the enumerated process similar to that used in the kosek method, you indicate the groupings character-by-character. In the between-key process, you specify the break-points in the sort order that should start a new group. The latter configuration is useful for ideographic languages such as Chinese, Japanese, and Korean. You can also define your own collation algorithms and how you want mixed Latin-alphabet words sorted. •For a whitepaper describing the extensions, see: http://www.innodata-isogen.com/knowledge_center/white_papers/back_of_book_for_xsl_fo.pdf. •To download the extension library, see http://www.innodata-isogen.com/knowledge_center/tools_downloads/i18nsupport. To use the kimber method, you must: 1.Use Saxon (version 6 or 8) as your XSLT processor. 2.Install and configure the Innodata Isogen library, using the documentation that comes with it. 3.Set the index.method parameter's value to "kimber". 4.Import the appropriate index extensions stylesheet module fo/autoidx-kimber.xsl or html/autoidx-kimber.xsl into your customization. Nameindex.on.type — Select indexterms based on type attribute value Synopsis<xsl:param name="index.on.type" select="0"></xsl:param> De-

PDF BY DBLATEX

scription If non-zero, then an index element that has a type attribute value will contain only those indexterm elements with a matching type attribute value. If an index has no type attribute or it is blank, then the index will contain all indexterms in the current scope. If index.on.type is zero, then the type attribute has no effect on selecting indexterms for an index. For those using DocBook version 4.2 or earlier, the type attribute is not available for index terms. However, you can achieve the same effect by using the role attribute in the same manner on indexterm and index, and setting the stylesheet parameter index.on.role to a nonzero value. Nameindex.on.role — Select indexterms based on role value Synopsis<xsl:param name="index.on.role" -select="0"></xsl:param> Description If non-zero, then an index element that has a role attribute value will contain only those indexterm elements with a matching role value. If an index has no role attribute or it is blank, then the index will contain all indexterms in the current scope. If index.on.role is zero, then the role attribute has no effect on selecting indexterms for an index. If you are using DocBook version 4.3 or later, you should use the type attribute instead of role on indexterm and index, and set the index.on.type to a nonzero value. Nameindex.links.to.section — HTML index entries link to container section title Synopsis<xsl:param name="index.links.to.section" select="1"></xsl:param> Description If zero, then an index entry in an index links directly to the location of the generated anchor that is output for the indexterm. If two identical indexterm elements exist in the same section, then both entries appear in the index with the same title but link to different locations. If non-zero, then an index entry in an index links to the section title containing the indexterm, rather than directly to the anchor output for the indexterm. Duplicate indexterm entries in the same section are dropped. The default value is 1, so index entries link to section titles by default. In both cases, the link text in an index entry is the title of the section containing the indexterm. That is because HTML does not have numbered pages. It also provides the reader with context information for each link. This parameter lets you choose which style of index linking you want. •When set to 0, an index entry takes you to the precise location of its corresponding indexterm. However, if you have a lot of duplicate entries in sections, then you have a lot of duplicate titles in the index, which makes it more cluttered. The reader may not recognize why duplicate titles appear until they follow the links. Also, the links may land the reader in the middle of a section where the section title is not visible, which may also be confusing to the reader. •When set to 1, an index entry link is less precise, but duplicate titles in the index entries are eliminated. Landing on the section title location may confirm the reader's expectation that a link that shows a section title will take them to that section title, not a location within the section. Nameindex.prefer.titleabbrev — Should abbreviated titles be used as back references? Synopsis<xsl:param -name="index.prefer.titleabbrev" select="0"></xsl:param> Description If non-zero, and if a titleabbrev is defined, the abbreviated title is used as the link text of a back reference in the index. Nameindex.term.separator — Override for punctuation separating an index term from its list of page references in an index Synopsis<xsl:param name="index.term.separator"></xsl:param> Description This parameter permits you to override the text to insert between the end of an index term and its list of page references. Typically that might be a comma and a space. Because this text may be locale dependent, this parameter's value is normally taken from a gentext template named 'term-separator' in the context 'index' in the stylesheet locale file for the language of the current document. This parameter can be used to override the gentext string, and would typically be used on the command line. This parameter would apply to all languages. So this text string can be customized in two ways. You can reset the default gentext string using the local.l10n.xml parameter, or you can fill in the content for this normally empty override parameter. The content can be a simple string, or it can be something more complex such as a call-template. For fo output, it could be an fo:leader element to provide space of a specific length, or a dot leader. Nameindex.number.separator — Override for punctuation separating page numbers in index Synopsis<xsl:param name="index.number.separator"></xsl:param> Description This parameter permits you to override the text to insert between page references in a formatted index entry. Typically that would be a comma and a space. Because this text may be locale dependent, this parameter's value is normally taken from a gentext template named 'number-separator' in the context 'index' in the stylesheet locale file for the language of the current document. This parameter can be used to override the gentext string, and would typically be used on the command line. This parameter would apply to all languages. So this text string can be customized in two ways. You can reset the default gentext string using the local.l10n.xml parameter, or you can override the gentext with the content of this parameter. The content can be a simple string, or it can be something more complex such as a call-template. In HTML index output, section title references are used instead of page number references. This punctuation appears between such section titles in an HTML index. Nameindex.range.separator — Override for punctuation separating the two numbers in a page range in index Synopsis<xsl:param name="index.range.separator"></xsl:param> Description This parameter permits you to override the text to insert between the two numbers of a page range in an index. This parameter is only used by those XSL-FO processors that support an extension for generating such page ranges (such as XEP). Because this text may be locale dependent, this parameter's value is normally taken from a gentext template named 'range-separator' in the context 'index' in the stylesheet locale file for the language of the current document. This parameter can be used to override the gentext string, and would typically be used on the command line. This parameter would apply to all languages. So this text string can be customized in two ways. You can reset the default gentext string using the local.l10n.xml parameter, or you can override the gentext with the content of this parameter. The content can be a simple string, or it can be something more complex such as a call-template. In HTML index output, section title references are used instead of page number references. So there are no page ranges and this parameter has no effect. Stylesheet ExtensionsStylesheet ExtensionsStylesheet ExtensionsTable of Contentslinenumbering.everyNthlinenumbering.extensionlinenumbering.separatorlinenumbering.widthtablecolumns.extensiontextinsert.extensiontextdata.c — Indicate which lines should be numbered Synopsis<xsl:param name="linenumbering.everyNth">5</xsl:param> Description

PDF BY DBLATEX

If line numbering is enabled, everyNth line will be numbered. Note that numbering is one based, not zero based. See also linenumbering.extension, linenumbering.separator, linenumbering.width and use.extensions Namelinenumbering.extension — Enable the line numbering extension Synopsis<xsl:param name="linenumbering.extension" select="1"></xsl:param> Description If non-zero, verbatim environments (address, literallayout, programlisting, screen, synopsis) that specify line numbering will have line numbers. Namelinenumbering.separator — Specify a separator between line numbers and lines Synopsis<xsl:param name="linenumbering.separator"><xsl:text> </xsl:text></xsl:param> Description The separator is inserted between line numbers and lines in the verbatim environment. The default value is a single white space. Note the interaction with linenumbering.width Namelinenumbering.width — Indicates the width of line numbers Synopsis<xsl:param name="linenumbering.width">3</xsl:p Description If line numbering is enabled, line numbers will appear right justified in a field "width" characters wide. Nametablecolumns.ex — Enable the table columns extension function Synopsis<xsl:param name="tablecolumns.extension" select="1"></xsl:param> 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. Nametextinsert.extension — Enables the textinsert extension element Synopsis <xsl:param name="textinsert.extension" select="1"></xsl:param> Description The textinsert extension element inserts the contents of a file into the result tree (as text). NoteTo use the textinsert extension element, you must use either Saxon or Xalan as your XSLT processor (it doesn't work with xsltproc), along with either the DocBook Saxon extensions or DocBook Xalan extensions (for more information about those extensions, see DocBook XSL: TCG, DocBook Saxon Extensions11 http://www.sagehill.net/docbookxsl/InstallingAProcessor.html#SaxonExtensions and DocBook XSL: TCG, DocBook Xalan Extensions22 http://www.sagehill.net/docbookxsl/InstallingAProcessor.html#XalanExtensions), and you must set both the use.extensions and textinsert.extension parameters to 1. As an alternative to using the textinsert element, consider using an Xinclude element with the parse="text" attribute and value specified, as detailed in DocBook XSL: TCG, Using XInclude for text inclusions33 http://www.sagehill.net/docbookxsl/ExternalCode.html#XIncludeCode. See Also You can also use the <?dbhtmlinclude href?> processing instruction to insert external files — both files containing plain text and files with markup content (including HTML content). More information For how-to documentation on inserting contents of external code files and other text files into output, see DocBook XSL: TCG, External code files44 http://www.sagehill.net/docbookxsl/ExternalCode.html. For guidelines on inserting contents of HTML files into output, see DocBook XSL: TCG, Inserting external HTML code55 http://www.sagehill.net/docbookxsl/InsertExtHtml.html. Nametextdata.default.encoding — Default encoding of external text files which are included using textdata element Synopsis<xsl:param name="textdata.default.encoding"></xsl:param> Description Specifies the encoding of any external text files included using textdata element. This value is used only when you do not specify encoding by the appropriate attribute directly on textdata. An empty string is interpreted as the system default encoding. Namegraphicsize.extension — Enable the getWidth()/getDepth() extension functions Synopsis<xsl:param name="graphicsize.extension" select="1"></xsl:param> Description If non-zero (and if use.extensions is non-zero and if you're using a processor that supports extension functions), the getWidth and getDepth functions will be used to extract image sizes from graphics. The main supported image formats are GIF, JPEG, and PNG. Somewhat cruder support for EPS and PDF images is also available. Namegraphicsize.use.img.src.path — Prepend img.src.path before filenames passed to extension functions Synopsis<xsl:param name="graphicsize.use.img.src.path" select="0"></xsl:param> Description If non-zero img.src.path parameter will be appended before filenames passed to extension functions for measuring image dimensions. Nameuse.extensions — Enable extensions Synopsis<xsl:param name="use.extensions" select="0"></xsl:param> 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. Automatic labellingAutomatic labellingAutomatic labellingTable of Contentschapter.autolabelappendix.autolabelpart.autolabelreference.auto — Specifies the labeling format for Chapter titles Synopsis<xsl:param name="chapter.autolabel" select="1"></xsl:param> Description If non-zero, then chapters will be numbered using the parameter value as the number format if the value matches one of the following: 1 or arabicArabic numeration (1, 2, 3 ...). A or upperalphaUppercase letter numeration (A, B, C ...). a or loweralphaLowercase letter numeration (a, b, c ...). I or upperromanUppercase roman numeration (I, II, III ...). i or lowerromanLowercase roman letter numeration (i, ii, iii ...). Any nonzero value other than the above will generate the default number format (arabic). Nameappendix.autolabel — Specifies the labeling format for Appendix titles Synopsis<xsl:param name="appendix.autolabel">A</xsl:param> Description If non-zero, then appendices will be numbered using the parameter value as the number format if the value matches one of the following: 1 or arabicArabic numeration (1, 2, 3 ...). A or upperalphaUppercase letter numeration (A, B, C ...). a or loweralphaLowercase letter numeration (a, b, c ...). I or upperromanUppercase roman numeration (I, II, III ...). i or lowerromanLowercase roman letter numeration (i, ii, iii ...). Any nonzero value other than the above will generate the default number format (upperalpha). Namepart.autolabel — Specifies the labeling format for Part titles Synopsis<xsl:param name="part.autolabel">I</xsl:param> Description If non-zero, then parts will be numbered using the parameter value as the number format if the value matches one of the following: 1 or arabicArabic numeration (1, 2, 3 ...). A or upperalphaUppercase letter numeration (A, B, C ...). a or loweralphaLowercase letter numeration (a, b, c ...). I or upperromanUppercase roman numeration (I, II, III ...). i or lowerromanLowercase roman letter numeration (i, ii, iii ...). Any nonzero value other than the above will generate the default number format (upperroman). Namereference.autolabel — Specifies the labeling format for Reference titles Synopsis <xsl:param name="reference.autolabel">I</xsl:param> Description If non-zero, references will be numbered using the parameter value as the number format if the value matches one of the following: 1 or arabicArabic numera-

tion (1, 2, 3 ...). A or upperalphaUppercase letter numeration (A, B, C ...). a or loweralphaLowercase letter numeration (a, b, c ...). I or upperromanUppercase roman numeration (I, II, III ...). i or lowerromanLowercase roman letter numeration (i, ii, iii ...). Any non-zero value other than the above will generate the default number format (upperroman). Namepreface.autolabel — Specifices the labeling format for Preface titles Synopsis<xsl:param name="preface.autolabel" select="0"></xsl:param> Description If non-zero then prefaces will be numbered using the parameter value as the number format if the value matches one of the following: 1 or arabicArabic numeration (1, 2, 3 ...). A or upperalphaUppercase letter numeration (A, B, C ...). a or loweralphaLowercase letter numeration (a, b, c ...). I or upperromanUppercase roman numeration (I, II, III ...). i or lowerromanLowercase roman letter numeration (i, ii, iii ...). Any nonzero value other than the above will generate the default number format (arabic). Nameqandadiv.autolabel — Are divisions in QAndASets enumerated? Synopsis<xsl:param name="qandadiv.autolabel" select="1"></xsl:param> Description If non-zero, unlabeled qandadivs will be enumerated. Namesection.autolabel — Are sections enumerated? Synopsis<xsl:param name="section.autolabel" select="0"></xsl:param> Description If true (non-zero), unlabeled sections will be enumerated. Namesection.autolabel.max.depth — The deepest level of sections that are numbered. Synopsis<xsl:param name="section.autolabel.max.depth">8</xsl:param> Description When section numbering is turned on by the section.autolabel parameter, then this parameter controls the depth of section nesting that is numbered. Sections nested to a level deeper than this value will not be numbered. Namesection.label.includes.component.label — Do section labels include the component label? Synopsis<xsl:param name="section.label.includes.component.label" select="0"></xsl:param> Description If non-zero, section labels are prefixed with the label of the component that contains them. Namelabel.from.part — Renumber components in each part? Synopsis<xsl:param name="label.from.part" select="0"></xsl:param> Description If label.from.part is non-zero, then numbering of components — preface, chapter, appendix, and reference (when reference occurs at the component level) — is re-started within each part. If label.from.part is zero (the default), numbering of components is not re-started within each part; instead, components are numbered sequentially throughout each book, regardless of whether or not they occur within part instances. Namecomponent.label.includes.part.label — Do component labels include the part label? Synopsis<xsl:param name="component.label.includes.part.label" select="0"></xsl:param> Description If non-zero, number labels for chapter, appendix, and other component elements are prefixed with the label of the part element that contains them. So you might see Chapter II.3 instead of Chapter 3. Also, the labels for formal elements such as table and figure will include the part label. If there is no part element container, then no prefix is generated. This feature is most useful when the label.from.part parameter is turned on. In that case, there would be more than one chapter “1”, and the extra part label prefix will identify each chapter unambiguously. HTMLHTMLHTMLTable of Contentshtml.basehtml.stylesheethtml.stylesheet.typecss.decorationhtml.scripthtml.script.typespacing.parasempl — An HTML base URI Synopsis<xsl:param name="html.base"></xsl:param> Description If html.base is set, it is used for the base element in the head of the html documents. The parameter specifies the base URL for all relative URLs in the document. This is useful for dynamically served html where the base URI needs to be shifted. Namehtml.stylesheet — Name of the stylesheet(s) to use in the generated HTML Synopsis<xsl:param name="html.stylesheet"></xsl:param> 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 stylesheet files. 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. Namehtml.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. Namecss.decoration — Enable CSS decoration of elements Synopsis<xsl:param name="css.decoration" select="1"></xsl:param> Description If non-zero, 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". Namehtml.script — Name of the script(s) to use in the generated HTML Synopsis<xsl:param name="html.script"></xsl:param> Description The html.script parameter is either empty (default), indicating that no script element should be generated in the html output, or it is a list of one or more script locations. Multiple script locations are space-delimited. If you need to reference a script URI that includes a space, encode it with %20. A separate html script element will be generated for each script in the order they are listed in the parameter. Namehtml.script.type — The type of script used in the generated HTML Synopsis<xsl:param name="html.script.type">text/javascript</xsl:param> Description The type of script to place in the HTML script element. Specifically, the value of the script element’s type attribute. The default value is text/javascript. This param is used only when the stylesheet parameter html.script is non-blank and specifies the location of a script. Namespacing.paras — Insert additional <p> elements for spacing? Synopsis<xsl:param name="spacing.paras" select="0"></xsl:param> 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. Nameemphasis.propagates.style — Pass emphasis role attribute through to HTML? Synopsis<xsl:param name="emphasis.propagates.style" select="1"></xsl:param> Description If non-zero, the role attribute of emphasis elements will be passed through to the HTML as a class attribute on a span that surrounds the emphasis. Namepara.propagates.style — Pass para role attribute through to HTML? Synopsis<xsl:param name="para.propagates.style" select="1"></xsl:param> 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. Namephrase.propagates.style — Pass phrase role attribute through to HTML? Synopsis<xsl:param name="phrase.propagates.style" select="1"></xsl:param>

PDF BY DBLATEX

Description If non-zero, the role attribute of phrase elements will be passed through to the HTML as a class attribute on a span that surrounds the phrase. Nameentry.propagates.style — Pass entry role attribute through to HTML? Synopsis<xsl:param name="entry.propagates.style" select="1"></xsl:param> Description If true, the role attribute of entry elements will be passed through to the HTML as a class attribute on the td or th generated for the table cell. Namehtml.longdesc — Should longdesc URIs be created? Synopsis<xsl:param name="html.longdesc" select="1"></xsl:param> Description If non-zero, HTML files will be created for the longdesc attribute. These files are created from the textobjects in mediaobjects and inlinemediaobject. Namehtml.longdesc.link — Should a link to the longdesc be included in the HTML? Synopsis<xsl:param name="html.longdesc.link" select="$html.longdesc"></xsl:param> Description If non-zero, links will be created to the HTML files created for the longdesc attribute. It makes no sense to enable this option without also enabling the html.longdesc parameter. Namemake.valid.html — Attempt to make sure the HTML output is valid HTML Synopsis<xsl:param name="make.valid.html" select="0"></xsl:param> 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. Namehtml.cleanup — Attempt to clean up the resulting HTML? Synopsis<xsl:param name="html.cleanup" select="1"></xsl:param> Description If non-zero, and if the EXSLT11 http://www.exslt.org/ 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. Namehtml.append — Specifies content to append to HTML output Synopsis<xsl:param name="html.append"></xsl:param> Description Specifies content to append to the end of HTML files output by the html/docbook.xsl stylesheet, after the closing <html> tag. You probably don’t want to set any value for this parameter; but if you do, the only value it should ever be set to is a newline character:
 or
 Namedraft.mode — Select draft mode Synopsis<xsl:param name="draft.mode">no</xsl:param> Description Selects draft mode. If draft.mode is “yes”, the entire document will be treated as a draft. If it is “no”, the entire document will be treated as a final copy. If it is “maybe”, individual sections will be treated as draft or final independently, depending on how their status attribute is set. Namedraft.watermark.image — The URI of the image to be used for draft watermarks Synopsis<xsl:param name="draft.watermark.image">images/draft.png</xsl:param> Description The image to be used for draft watermarks. Namegenerate.id.attributes — Generate ID attributes on container elements? Synopsis<xsl:param name="generate.id.attributes" select="0"></xsl: 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: <h2>Some Title</h2> Some para. The alternative is to generate anchors: <h2>Some Title</h2> Some para. 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. Namegenerate.consistent.ids — Generate consistent id values if document is unchanged Synopsis<xsl:param name="generate.consistent.ids" select="0"></xsl:param> Description When the stylesheet assigns an id value to an output element, the generate-id() function may be used. That function may not produce consistent values between runs. Version control systems may misidentify the changing id values as changes to the document. If you set this parameter’s value to 1, then the template named object.id will replace the use of the function generate-id() with <xsl:number level="multiple" count="*"/>. This counts preceding elements to generate a unique number for the id value. CautionThis param does not associate permanent unique id values with particular elements. The id values are consistent only as long as the document structure does not change. If the document structure changes, then the counting of elements changes, and all id values after the first such change may be different, even when there is no change to the element itself or its output. The default value of this parameter is zero, so generate-id() is used by default. Namegenerate.meta.abstract — Generate HTML META element from abstract? Synopsis<xsl:param name="generate.meta.abstract" select="1"></xsl:param> Description If non-zero, document abstracts will be reproduced in the HTML head, with >meta name="description" content="..." Namemake.clean.html — Make HTML conform to modern coding standards Synopsis<xsl:param name="make.clean.html" select="0"></xsl:param> Description If make.clean.html is true, the stylesheets take extra effort to ensure that the resulting HTML is conforms to modern HTML coding standards. In addition to eliminating excessive and noncompliant coding, it moves presentation HTML coding to a CSS stylesheet. The resulting HTML is dependent on CSS for formatting, and so the stylesheet is capable of generating a supporting CSS file. The docbook.css.source and custom.css.source parameters control how a CSS file is generated. If you require your CSS to reside in the HTML head element, then the generate.css.header can be used to do that. The make.clean.html parameter is different from html.cleanup because the former changes the resulting markup; it does not use extension functions like the latter to manipulate result-tree-fragments, and is therefore applicable to any XSLT processor. If make.clean.html is set to zero (the default), then the stylesheet retains its original “old style” HTML formatting features. Namedocbook.css.source — Name of the default CSS input file Synopsis<xsl:param name="docbook.css.source">docbook.css.xml</xsl:param> Description The docbook.css.source parameter specifies the name of the file containing the default DocBook CSS styles. Those styles are necessary when the make.clean.html

parameter is nonzero. The file is a well-formed XML file that must consist of a single style root element that contains CSS styles as its text content. The default value of the parameter (and filename) is docbook.css.xml. The stylesheets ship with the default file. You can substitute your own and specify its path in this parameter. If docbook.css.source is not blank, and make.clean.html is nonzero, then the stylesheet takes the following actions: 1.The stylesheet uses the XSLT document() function to open the file specified by the parameter and load it into a variable. 2.The stylesheet forms an output pathname consisting of the value of the base.dir parameter (if it is set) and the value of docbook.css.source, with the .xml suffix stripped off. 3.The stylesheet removes the style wrapper element and writes just the CSS text content to the output file. 4.The stylesheet adds a link element to the HTML HEAD element to reference the external CSS stylesheet. For example: <link rel="stylesheet" href="docbook.css" type="text/css"> However, if the docbook.css.link parameter is set to zero, then no link is written for the default CSS file. That is useful if a custom CSS file will import the default CSS stylesheet to ensure proper cascading of styles. If the docbook.css.source parameter is changed from its default docbook.css.xml to blank, then no default CSS is generated. Likewise if the make.clean.html parameter is set to zero, then no default CSS is generated. The custom.css.source parameter can be used instead to generate a complete custom CSS file. You can use the generate.css.header parameter to instead write the CSS to each HTML HEAD element in a style tag instead of an external CSS file. Namedocbook.css.link — Insert a link referencing the default CSS stylesheet Synopsis<xsl:param name="docbook.css.link" select="1"></xsl:param> Description The stylesheets are capable of generating a default CSS stylesheet file. The parameters make.clean.html and docbook.css.source control that feature. Normally if a default CSS file is generated, then the stylesheet inserts a link tag in the HTML HEAD element to reference it. However, you can omit that link reference if you set the docbook.css.link to zero (1 is the default). This parameter is useful when you want to import the default CSS into a custom CSS file generated using the custom.css.source parameter. Namecustom.css.source — Name of a custom CSS input file Synopsis<xsl:param name="custom.css.source"></xsl:param> Description The custom.css.source parameter enables you to add CSS styles to DocBook's HTML output. The parameter specifies the name of a file containing custom CSS styles. The file must be a well-formed XML file that consists of a single style root element that contains CSS styles as its text content. For example: <?xml version="1.0"?> <style> h2 { font-weight: bold; color: blue; } ... </style> The filename specified by the parameter should have a .xml filename suffix, although that is not required. The default value of this parameter is blank. If custom.css.source is not blank, then the stylesheet takes the following actions. These actions take place regardless of the value of the make.clean.html parameter. 1.The stylesheet uses the XSLT document() function to open the file specified by the parameter and load it into a variable. 2.The stylesheet forms an output pathname consisting of the value of the base.dir parameter (if it is set) and the value of custom.css.source, with the .xml suffix stripped off. 3.The stylesheet removes the style wrapper element and writes just the CSS text content to the output file. 4.The stylesheet adds a link element to the HTML HEAD element to reference this external CSS stylesheet. For example: <link rel="stylesheet" href="custom.css" type="text/css"> If the make.clean.html parameter is nonzero (the default is zero), and if the docbook.css.source parameter is not blank (the default is not blank), then the stylesheet will also generate a default CSS file and add a link tag to reference it. The link to the custom CSS comes after the link to the default, so it should cascade properly in most browsers. If you do not want two link tags, and instead want your custom CSS to import the default generated CSS file, then do the following: 1.Add a line like the following to your custom CSS source file: @import url("docbook.css") 2.Set the docbook.css.link parameter to zero. This will omit the link tag that references the default CSS file. If you set make.clean.html to nonzero but you do not want the default CSS generated, then also set the docbook.css.source parameter to blank. Then no default CSS will be generated, and so all CSS styles must come from your custom CSS file. You can use the generate.css.header parameter to instead write the CSS to each HTML HEAD element in a style tag instead of an external CSS file. Namegenerate.css.header — Insert generated CSS styles in HEAD element Synopsis<xsl:param name="generate.css.header" select="0"></xsl:param> Description The stylesheets are capable of generating both default and custom CSS stylesheet files. The parameters make.clean.html, docbook.css.source, and custom.css.source control that feature. If you require that CSS styles reside in the HTML HEAD element instead of external CSS files, then set the generate.css.header parameter to nonzero (it is zero by default). Then instead of generating the CSS in external files, they are wrapped in style elements in the HEAD element of each HTML output file. XSLT ProcessingXSLT ProcessingXSLT ProcessingTable of Contentsrootidsuppress.navigationsuppress.header.navigationsuppress.footer.navigationheader.rulefooter.ruleid.warningsNamerootid — Specify the root element to format Synopsis<xsl:param name="rootid"></xsl:param> Description If rootid is not empty, 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. Namesuppress.navigation — Disable header and footer navigation Synopsis<xsl:param name="suppress.navigation" select="0"></xsl:param> Description If non-zero, header and footer navigation will be suppressed. Namesuppress.header.navigation — Disable header navigation Synopsis<xsl:param name="suppress.header.navigation" select="0"></xs Description If non-zero, header navigation will be suppressed. Namesuppress.footer.navigation — Disable footer navigation Synopsis<xsl:param name="suppress.footer.navigation">0</xsl:param> Description If non-zero, footer navigation will be suppressed. Nameheader.rule — Rule under headers? Synopsis<xsl:param name="header.rule" select="1"></xsl:param> Description If non-zero, a rule will be drawn below the page headers. Namefooter.rule — Rule over footers? Synopsis<xsl:param name="footer.rule" select="1"></xsl:param> Description If non-zero, a rule will be drawn above the page footers. Nameid.warnings

PDF BY DBLATEX

— Should warnings be generated for titled elements without IDs? Synopsis<xsl:param name="id.warnings" select="0"></xsl:param> Description If non-zero, the stylesheet will issue a warning for any element (other than the root element) which has a title but does not have an ID. Meta/*Info and TitlepagesMeta/*Info and TitlepagesMeta/*Info and TitlepagesTable of Contentsinherit.keywordsmake.single.year.rangesmake.year.rangesauthor.othername.in.middleblurb.on.titlepage.enabledcontrib.inline.enablededit — Inherit keywords from ancestor elements? Synopsis<xsl:param name="inherit.keywords" select="1"></xsl:param> Description If inherit.keywords is non-zero, the keyword meta for each HTML head element will include all of the keywords from ancestor elements. Otherwise, only the keywords from the current section will be used. Namemake.single.year.ranges — Print single-year ranges (e.g., 1998-1999) Synopsis<xsl:param name="make.single.year.ranges" select="0"></xsl:param> 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). Namemake.year.ranges — Collate copyright years into ranges? Synopsis<xsl:param name="make.year.ranges" - select="0"></xsl:param> Description If non-zero, multiple copyright year elements will be collated into ranges. This works only if each year number is put into a separate year element. The copyright element permits multiple year elements. If a year element contains a dash or a comma, then that year element will not be merged into any range. Nameauthor.othername.in.middle — Is othername in author a middle name? Synopsis<xsl:param name="author.othername.in.middle" select="1"></xsl:param> Description If non-zero, the othername of an author appears between the firstname and surname. Otherwise, othername is suppressed. Nameblurb.on.titlepage.enabled — Display personblurb and authorblurb on title pages? Synopsis<xsl:param - name="blurb.on.titlepage.enabled" select="0"></xsl:param> Description If non-zero, output from authorblurb and personblurb elements is displayed on title pages. If zero (the default), output from those elements is suppressed on title pages (unless you are using a titlepage customization that causes them to be included). Namecontrib.inline.enabled — Display contrib output inline? Synopsis<xsl:param name="contrib.inline.enabled">1</xsl:param> Description If non-zero (the default), output of the contrib element is displayed as inline content rather than as block content. Nameeditedby.enabled — Display “Edited by” heading above editor name? Synopsis<xsl:param name="editedby.enabled">1</xsl:param> Description If non-zero, a localized Edited by heading is displayed above editor names in output of the editor element. Nameabstract.notitle.enabled — Suppress display of abstract titles? Synopsis<xsl:param name="abstract.notitle.enabled" select="0"></xsl:param> Description If non-zero, in output of the abstract element on titlepages, display of the abstract title is suppressed. Nameothercredit.like.author.enabled — Display othercredit in same style as author? Synopsis<xsl:param name="othercredit.like.author.enabled">0</xsl:param> Description If non-zero, output of the othercredit element on titlepages is displayed in the same style as author and editor output. If zero then othercredit output is displayed using a style different than that of author and editor. Namegenerate.legalnotice.link — Write legalnotice to separate chunk and generate link? Synopsis<xsl:param name="generate.legalnotice.link" select="0"></xsl:param> Description If the value of generate.legalnotice.link is non-zero, the stylesheet: •writes the contents of legalnotice to a separate HTML file •inserts a hyperlink to the legalnotice file •adds (in the HTML head) either a single link or element or multiple link elements (depending on the value of the html.head.legalnotice.link.multiple parameter), with the value or values derived from the html.head.legalnotice.link.types parameter Otherwise, if generate.legalnotice.link is zero, legalnotice contents are rendered on the title page. The name of the separate HTML file is computed as follows: 1.If a filename is given by the <?dbhtml filename?> processing instruction, that filename is used. 2.If the legalnotice has an id/xml:id attribute, and if use.id.as.filename != 0, the filename is the concatenation of the id value and the value of the html.ext parameter. 3.If the legalnotice does not have an id/xml:id attribute, or if use.id.as.filename = 0, the filename is the concatenation of "ln-", auto-generated id value, and html.ext value. Namegenerate.revhistory.link — Write revhistory to separate chunk and generate link? Synopsis<xsl:param name="generate.revhistory.link" select="0"></xsl:param> Description If non-zero, the contents of revhistory are written to a separate HTML file and a link to the file is generated. Otherwise, revhistory contents are rendered on the title page. The name of the separate HTML file is computed as follows: 1.If a filename is given by the <?dbhtml filename?> processing instruction, that filename is used. 2.If the revhistory has an id/xml:id attribute, and if use.id.as.filename != 0, the filename is the concatenation of the id value and the value of the html.ext parameter. 3.If the revhistory does not have an id/xml:id attribute, or if use.id.as.filename = 0, the filename is the concatenation of "rh-", auto-generated id value, and html.ext value. Namehtml.head.legalnotice.link.types — Specifies link types for legalnotice link in html head Synopsis<xsl:param - name="html.head.legalnotice.link.types">copyright</xsl:param> Description The value of html.head.legalnotice.link.types is a space-separated list of link types, as described in Section 6.12 of the HTML 4.01 specification11 http://www.w3.org/TR/html401/types.ht links. If the value of the generate.legalnotice.link parameter is non-zero, then the stylesheet generates (in the head section of the HTML source) either a single HTML link element or, if the value of the html.head.legalnotice.link.multiple is non-zero, one link element for each link type specified. Each link has the following attributes: •a rel attribute whose value is derived from the value of html.head.legalnotice.link.types •an href attribute whose value is set to the URL of the file containing the legalnotice •a title attribute whose value is set to the title of the corresponding legalnotice (or a title programatically determined by the stylesheet) For example: <link rel="license" href="ln-id2524073.html" title="Legal Notice"> About the default value In an ideal world, the default value of html.head.legalnotice.link.types would probably be “license”, since the content of the DocBook legalnotice is typically license information, not copyright information. However, the default value is “copyright” for pragmatic reasons: because that’s among the set of “recognized link types” listed in Section 6.12 of the HTML 4.01 specification22 http://www.w3.org/TR/html401/types.html#type-links, and because certain browsers and browser extensions are

PDF BY DBLATEX

preconfigured to recognize that value. Namehtml.head.legalnotice.link.multiple — Generate multiple link instances in html head for legalnotice? Synopsis<xsl:param name="html.head.legalnotice.link.multiple" select="1"></xsl:param> Description If html.head.legalnotice.link.multiple is non-zero and the value of html.head.legalnotice.link.types contains multiple link types, then the stylesheet generates (in the head section of the HTML source) one link element for each link type specified. For example, if the value of html.head.legalnotice.link.types is “copyright license”: <link rel="copyright" href="ln-id2524073.html" title="Legal Notice"> <link rel="license" href="ln-id2524073.html" title="Legal Notice"> Otherwise, the stylesheet generates generates a single link instance; for example: <link rel="copyright license" href="ln-id2524073.html" title="Legal Notice"> Reference PagesReference PagesReference PagesTable of Contentsfuncsynopsis.decorationfuncsynopsis.stylefunction.parensrefentry.generate.nam — Decorate elements of a funcsynopsis? Synopsis<xsl:param name="funcsynopsis.decoration" select="1"></xsl:param> Description If non-zero, elements of the funcsynopsis will be decorated (e.g. rendered as bold or italic text). The decoration is controlled by templates that can be redefined in a customization layer. Namefuncsynopsis.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. Namefunction.parens — Generate parens after a function? Synopsis<xsl:param name="function.parens" select="0"></xsl:param> Description If non-zero, the formatting of a function element will include generated parentheses. Namerefentry.generate.name — Output NAME header before refnames? Synopsis<xsl:param name="refentry.generate.name" select="1"></xsl:param> Description If non-zero, a "NAME" section title is output before the list of refnames. 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 non-zero unless you want get quite strange output. Namerefentry.generate.title — Output title before refnames? Synopsis<xsl:param name="refentry.generate.title" select="0"></xsl:param> Description If non-zero, the reference page title or first name is output before the list of refnames. This parameter and refentry.generate.name are mutually exclusive. This means that if you change this parameter to non-zero, you should set refentry.generate.name to zero unless you want get quite strange output. Namerefentry.xref.manvolnum — Output manvolnum as part of refentry cross-reference? Synopsis<xsl:param name="refentry.xref.manvolnum" select="1"></xsl:param> Description if non-zero, the manvolnum is used when cross-referencing refentrys, either with xref or citerefentry. Nameciterefentry.link — Generate URL links when cross-referencing RefEntrys? Synopsis<xsl:param name="citerefentry.l select="0"></xsl:param> Description If non-zero, 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. Namerefentry.separator — Generate a separator between consecutive RefEntry elements? Synopsis<xsl:param name="refentry.separator" select="1"></xsl:param> Description If true, a separator will be generated between consecutive reference pages. Namerefclass.suppress — Suppress display of refclass contents? Synopsis<xsl:param name="refclass.suppress" select="0"></xsl:param> Description If the value of refclass.suppress is non-zero, then display of refclass contents is suppressed in output. TablesTablesTablesTable of Contentsdefault.table.widthnominal.table.widthtable.borders.with.csstable.cell.border.styletable.cell.border.thicknesstable.cell.border.colortable.fra — The default width of tables Synopsis<xsl:param name="default.table.width"></xsl:param> Description If non-zero, this value will be used for the width attribute on tables that do not specify an alternate width (with the <?dbhtml table-width?> or <?dbfo table-width?> processing instruction). Namenominal.table.width — The (absolute) nominal width of tables Synopsis<xsl:param name="nominal.table.width">6in</xsl:param> 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 percentage). Nametable.borders.with.css — Use CSS to specify table, row, and cell borders? Synopsis<xsl:param name="table.borders.with.css" select="0"></xsl:param> Description If non-zero, CSS will be used to draw table borders. Nametable.cell.border.style — Specifies the border style of table cells Synopsis<xsl:param name="table.cell.border.style">solid</xsl:param> Description Specifies the border style of table cells. NoteTo control properties of cell borders in HTML output, you must also turn on the table.borders.with.css parameter. Nametable.cell.border.thickness — Specifies the thickness of table cell borders Synopsis<xsl:param name="table.cell.border.thickness">0.5pt</xsl:param> Description If non-zero, specifies the thickness of borders on table cells. The units are points. See CSS11 http://www.w3.org/TR/CSS21/box.htm width-properties NoteTo control properties of cell borders in HTML output, you must also turn on the table.borders.with.css parameter. Nametable.cell.border.color — Specifies the border color of table cells Synopsis<xsl:param name="table.cell.border.color"></x Description Set the color of table cell borders. If non-zero, the value is used for the border coloration. See CSS11 http://www.w3.org/TR/ def-color. A color is either a keyword or a numerical RGB specification. Keywords are aqua, black, blue, fuchsia, gray, green, lime, maroon, navy, olive, orange, purple, red, silver, teal, white, and yellow. NoteTo control properties of cell borders in HTML output, you must also turn on the table.borders.with.css parameter. Nametable.frame.border.style — Specifies the border style of table frames Synopsis<xsl:param name="table.frame.border.style">solid</xsl:param> Description Specifies the border style of table frames. Nametable.frame.border.thickness — Specifies the thickness of the frame border Synopsis<xsl:param name="table.frame.border.thickness">0.5pt</xsl:param> Description Specifies the thickness of the border on the table’s frame. Nametable.frame.border.color — Specifies the border color of table frames Synopsis<xsl:param name="table.frame.border.color"></xsl: Description Specifies the border color of table frames. Namedefault.table.frame — The default framing of tables Synopsis<xsl:param name="default.table.frame">all</xsl:param> Description This value will be used when there is no frame attribute on the table. Namehtml.cellspacing — Default value for cellspacing in HTML tables Synopsis<xsl:param name="html.cellspacing"></xs

Description If non-zero, this value will be used as the default cellspacing value in HTML tables. nn for pixels or nn% for percentage length. E.g. 5 or 5% Namehtml.cellpadding — Default value for cellpadding in HTML tables Synopsis<xsl:param name="html.cellpadding"></xsl:param> Description If non-zero, this value will be used as the default cellpadding value in HTML tables. nn for pixels or nn% for percentage length. E.g. 5 or 5% QAndASetQAndASetQAndASetTable of Contentsqanda.defaultlabelqanda.inherit.numerationqanda.in.tocqanda.nested.in.tocNameqanda.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 is generally one of the legal values for the defaultlabel attribute (none, number or qanda), or one of the additional stylesheet-specific values (qnumber or qnumberanda). The default value is 'number'. The values are rendered as follows: qandaquestions are labeled "Q:" and answers are labeled "A:". numberThe questions are enumerated and the answers are not labeled. qnumberThe questions are labeled "Q:" followed by a number, and answers are not labeled. When sections are numbered, adding a label to the number distinguishes the question numbers from the section numbers. This value is not allowed in the defaultlabel attribute of a qandaset element. qnumberandaThe questions are labeled "Q:" followed by a number, and the answers are labeled "A:". When sections are numbered, adding a label to the number distinguishes the question numbers from the section numbers. This value is not allowed in the defaultlabel attribute of a qandaset element. noneNo distinguishing label precedes Questions or Answers. Nameqanda.inherit.numeration — Does enumeration of QandASet components inherit the numeration of parent elements? Synopsis<xsl:param name="qanda.inherit.numeration" select="1"></xsl:param> Description If non-zero, numbered qandadiv elements and question and answer inherit the enumeration of the ancestors of the qandaset. Nameqanda.in.toc — Should qandaentry questions appear in the document table of contents? Synopsis<xsl:param name="qanda.in.toc" select="0"></xsl:param> Description If true (non-zero), then the generated table of contents for a document will include qandaset titles, qandadiv titles, and question elements. The default value (zero) excludes them from the TOC. This parameter does not affect any tables of contents that may be generated inside a qandaset or qandadiv. Nameqanda.nested.in.toc — Should nested answer/qandaentry instances appear in TOC? Synopsis<xsl:param name="qanda.nested.in.toc" select="0"></xsl:param> Description If non-zero, instances of qandaentry that are children of answer elements are shown in the TOC. LinkingLinkingLinkingTable of Contentstarget.database.documenttargets.filenameolink.base.uriuse.local.olink.stylecurrent.docidolink.doctitleactivate.external.olinksolink.debugol — Name of master database file for resolving olinks Synopsis <xsl:param name="target.database.document">olinkdb.xml</xsl:param> 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. Nametargets.filename — Name of cross reference targets data file Synopsis<xsl:param name="targets.filename">target.db</xsl:param> 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. Nameolink.base.uri — Base URI used in olink hrefs Synopsis<xsl:param name="olink.base.uri"></xsl:param> 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. Nameuse.local.olink.style — Process olinks using xref style of current document Synopsis<xsl:param name="use.local.olink.style" select="0"></xsl:param> 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. Namecurrent.docid — targetdoc identifier for the document being processed Synopsis<xsl:param name="current.docid"></xsl:param> 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

PDF BY DBLATEX

is to a different document, not the current document. See also olink.doctitle to enable that feature. Nameolink.doctitle — show the document title for external olinks? Synopsis<xsl:param name="olink.doctitle">no</xsl:param> Description When olinks between documents are resolved, 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. •This olink.doctitle parameter should be set to either yes or maybe to enable this feature. •And you should also set the current.docid parameter to the document id for the document currently being processed for output. Then if an olink's targetdoc id differs from the current.docid value, the stylesheet knows that it is a reference to another document and can append the target document's title to the generated olink text. The text for the target document's title is copied from the olink database from the ttl element of the top-level div for that document. If that ttl element is missing or empty, no title is output. The supported values for olink.doctitle are: yesAlways insert the title to the target document if it is not the current document. noNever insert the title to the target document, even if requested in an xrefstyle attribute. maybeOnly insert the title to the target document, if requested in an xrefstyle attribute. An xrefstyle attribute may override the global setting for individual olinks. The following values are supported in an xrefstyle attribute using the select: syntax: docnameInsert the target document name for this olink using the docname gentext template, but only if the value of olink.doctitle is not no. docnamelongInsert the target document name for this olink using the docnamelong gentext template, but only if the value of olink.doctitle is not no. nodocnameOmit the target document name even if the value of olink.doctitle is yes. Another way of inserting the target document name for a single olink is to employ an xrefstyle attribute using the template: syntax. The %o placeholder (the letter o, not zero) in such a template will be filled in with the target document's title when it is processed. This will occur regardless of the value of olink.doctitle. Note that prior to version 1.66 of the XSL stylesheets, the allowed values for this parameter were 0 and 1. Those values are still supported and mapped to 'no' and 'yes', respectively. Nameactivate.external.olinks — Make external olinks into active links Synopsis<xsl:param name="activate.external.olinks" select="1"></xsl:param> Description If activate.external.olinks is nonzero (the default), then any olinks that reference another document become active links that can be clicked on to follow the link. If the parameter is set to zero, then external olinks will have the appropriate link text generated, but the link is not made active. Olinks to destinations in the current document remain active. To make an external olink active for HTML outputs, the link text is wrapped in an a element with an href attribute. To make an external olink active for FO outputs, the link text is wrapped in an fo:basic-link element with an external-destination attribute. This parameter is useful when you need external olinks to resolve but not be clickable. For example, if documents in a collection are available independently of each other, then having active links between them could lead to unresolved links when a given target document is missing. The epub stylesheets set this parameter to zero by default because there is no standard linking mechanism between Epub documents. If external links are made inactive, you should consider setting the stylesheet parameter olink.doctitle to yes. That will append the external document's title to the link text, making it easier for the user to locate the other document. An olink is considered external when the current.docid stylesheet parameter is set to some value, and the olink's targetdoc attribute has a different value. If the two values match, then the link is considered internal. If the current.docid parameter is blank, or the olink element does not have a targetdoc attribute, then the link is considered to be internal and will become an active link. See also olink.doctitle, prefer.internal.olink. Nameolink.debug — Turn on debugging messages for olinks Synopsis<xsl:param name="olink.debug" -select="0"></xsl:param> Description If non-zero, then each olink will generate several messages about how it is being resolved during processing. This is useful when an olink does not resolve properly and the standard error messages are not sufficient to find the problem. You may need to read through the olink XSL templates to understand the context for some of the debug messages. Nameolink.properties — Properties associated with the cross-reference text of an olink. Synopsis<xsl:attribute-set -name="olink.properties"> <xsl:attribute name="show-destination">replace</xsl:attribute> </xsl:attribute-set> Description This attribute set is applied to the fo:basic-link element of an olink. It is not applied to the optional page number or optional title of the external document. Nameolink.lang.fallback.sequence — look up translated documents if olink not found? Synopsis<xsl:param -name="olink.lang.fallback.sequence"></xsl:param> Description This parameter defines a list of lang values to search among to resolve olinks. Normally an olink tries to resolve to a document in the same language as the olink itself. The language of an olink is determined by its nearest ancestor element with a lang attribute, otherwise the value of the l10n.gentext.default.lang parameter. An olink database can contain target data for the same document in multiple languages. Each set of data has the same value for the targetdoc attribute in the document element in the database, but with a different lang attribute value. When an olink is being resolved, the target is first sought in the document with the same language as the olink. If no match is found there, then this parameter is consulted for additional languages to try. The olink.lang.fallback.sequence must be a whitespace separated list of lang values to try. The first one with a match in the olink database is used. The default value is empty. For example, a document might be written in German and contain an olink with targetdoc="adminguide". When the document is processed, the processor first looks for a target dataset in the olink database starting with: <document targetdoc="adminguide" lang="de">. If there is no such element, then the olink.lang.fallback.sequence parameter is consulted. If its value is, for example, "fr en", then the processor next looks for targetdoc="adminguide" lang="fr", and then for targetdoc="adminguide" lang="en". If there is still no match, it looks for targetdoc="adminguide" with no lang attribute. This parameter is useful when a set of documents is only partially translated, or is in the process of being translated. If a target of an olink has not yet been translated, then this parameter permits the processor to look for the document in other languages. This assumes the reader would rather have a link to a document

PDF BY DBLATEX

in a different language than to have a broken link. Nameinsert.olink.page.number — Turns page numbers in olinks on and off Synopsis<xsl:param name="insert.olink.page.number">no</xsl:param> Description The value of this parameter determines if cross references made between documents with olink will include page number citations. In most cases this is only applicable to references in printed output. The parameter has three possible values. noNo page number references will be generated for olinks. yesPage number references will be generated for all olink references. The style of page reference may be changed if an xrefstyle attribute is used. maybePage number references will not be generated for an olink element unless it has an xrefstyle attribute whose value specifies a page reference. Olinks that point to targets within the same document are treated as xrefs, and controlled by the insert.xref.page.number parameter. Page number references for olinks to external documents can only be inserted if the information exists in the olink database. This means each olink target element (div or obj) must have a page attribute whose value is its page number in the target document. The XSL stylesheets are not able to extract that information during processing because pages have not yet been created in XSLT transformation. Only the XSL-FO processor knows what page each element is placed on. Therefore some postprocessing must take place to populate page numbers in the olink database. Nameinsert.olink.pdf.frag — Add fragment identifiers for links into PDF files Synopsis<xsl:param name="insert.olink.pdf.frag" select="0"></xsl:param> Description The value of this parameter determines whether the cross reference URIs to PDF documents made with olink will include fragment identifiers. When forming a URI to link to a PDF document, a fragment identifier (typically a '#' followed by an id value) appended to the PDF filename can be used by the PDF viewer to open the PDF file to a location within the document instead of the first page. However, not all PDF files have id values embedded in them, and not all PDF viewers can handle fragment identifiers. If insert.olink.pdf.frag is set to a non-zero value, then any olink targeting a PDF file will have the fragment identifier appended to the URI. The URI is formed by concatenating the value of the olink.base.uri parameter, the value of the baseuri attribute from the document element in the olink database with the matching targetdoc value, and the value of the href attribute for the targeted element in the olink database. The href attribute contains the fragment identifier. If insert.olink.pdf.frag is set to zero (the default value), then the href attribute from the olink database is not appended to PDF olinks, so the fragment identifier is left off. A PDF olink is any olink for which the baseuri attribute from the matching document element in the olink database ends with '.pdf'. Any other olinks will still have the fragment identifier added. Nameprefer.internal.olink — Prefer a local olink reference to an external reference Synopsis<xsl:param name="prefer.internal.olink" select="0"></xsl:param> Description If you are re-using XML content modules in multiple documents, you may want to redirect some of your olinks. This parameter permits you to redirect an olink to the current document. For example: you are writing documentation for a product, which includes 3 manuals: a little installation booklet (booklet.xml), a user guide (user.xml), and a reference manual (reference.xml). All 3 documents begin with the same introduction section (intro.xml) that contains a reference to the customization section (custom.xml) which is included in both user.xml and reference.xml documents. How do you write the link to custom.xml in intro.xml so that it is interpreted correctly in all 3 documents? •If you use xref, it will fail in user.xml. •If you use olink (pointing to reference.xml), the reference in user.xml will point to the customization section of the reference manual, while it is actually available in user.xml. If you set the prefer.internal.olink parameter to a non-zero value, then the processor will first look in the olink database for the olink's targetptr attribute value in document matching the current.docid parameter value. If it isn't found there, then it tries the document in the database with the targetdoc value that matches the olink's targetdoc attribute. This feature permits an olink reference to resolve to the current document if there is an element with an id matching the olink's targetptr value. The current document's olink data must be included in the target database for this to work. CautionThere is a potential for incorrect links if the same id attribute value is used for different content in different documents. Some of your olinks may be redirected to the current document when they shouldn't be. It is not possible to control individual olink instances. Namelink.mailto.url — Mailto URL for the LINK REL=made HTML HEAD element Synopsis<xsl:param name="link.mailto.url"></xsl:param> Description If not the empty string, this address will be used for the rel=made link element in the html head Nameulink.target — The HTML anchor target for ULinks Synopsis<xsl:param name="ulink.target">_top</xsl:param> Description If ulink.target is non-zero, its value will be used for the target attribute on anchors generated for ulinks. Cross ReferencesCross ReferencesCross ReferencesTable of Contentscollect.xref.targetsinsert.xref.page.numberuse.role.as.xrefstylexref.with.number.and.titlexref.label-page.separatorxref.label-title.separatorxref.title-page.separatorNamecollect.xref.targets — Controls whether cross reference data is collected Synopsis<xsl:param name="collect.xref.targets">no</xsl:param> 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. Nameinsert.xref.page.number — Turns page numbers in xrefs on and off Synopsis<xsl:param name="insert.xref.page.number">no</xsl:param> Description The value of this parameter determines if cross references (xrefs) in printed output will include page number citations. It has three possible values. noNo page number references will be generated. yesPage number references will be generated for all xref elements. The style of page reference may be changed if an xrefstyle attribute is used. maybePage number references will not be generated for an xref element unless it has an xrefstyle attribute whose value specifies a page reference. Nameuse.role.as.xrefstyle — Use role attribute for xrefstyle on xref? Synopsis<xsl:param name="use.role.as.xrefstyle" select="1"></xsl:param> Description In Doc-

PDF BY DBLATEX

Book documents that conform to a schema older than V4.3, this parameter allows role to serve the purpose of specifying the cross reference style. If non-zero, the role attribute on xref will be used to select the cross reference style. In DocBook V4.3, the xrefstyle attribute was added for this purpose. If the xrefstyle attribute is present, role will be ignored, regardless of the setting of this parameter. Example The following small stylesheet shows how to configure the stylesheets to make use of the cross reference style: <?xml version="1.0"?> <xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" - version="1.0"> <xsl:import href="../xsl/html/docbook.xsl"/> <xsl:output method="html"/> <xsl:param name="local.l10n.xml" - select="document(”)"/> <l:i18n xmlns:l="http://docbook.sourceforge.net/xmlns/l10n/1.0"> <l:l10n xmlns:l="http://docbook.sourceforg language="en"> <l:context name="xref"> <l:template name="chapter" style="title" text="Chapter %n, %t"/> - <l:template name="chapter" text="Chapter %n"/> </l:context> </l:l10n> </l:i18n> </xsl:stylesheet> With this stylesheet, the cross references in the following document: <?xml version="1.0" encoding="utf-8"?> <!DOCTYPE book PUBLIC "- //OASIS//DTD DocBook XML V4.2//EN" "http://www.oasis-open.org/docbook/xml/4.2/docbookx.dtd"> <book - id="book"><title>Book</title> <preface> <title>Preface</title> <para>Normal: <xref linkend="ch1"/>.</para> <para>Title: - <xref xrefstyle="title" linkend="ch1"/>.</para> </preface> <chapter id="ch1"> <title>First Chapter</title> <para>Irrelevant.</para> </chapter> </book> will appear as: Normal: Chapter 1. Title: Chapter 1, First Chapter. Namexref.with.number.and.title — Use number and title in cross references Synopsis<xsl:param name="xref.with.number.and.title" select="1"></xsl:param> Description A cross reference may include the number (for example, the number of an example or figure) and the title which is a required child of some targets. This parameter inserts both the relevant number as well as the title into the link. Namexref.label-page.separator — Punctuation or space separating label from page number in xref Synopsis<xsl:param name="xref.label-page.separator" </xsl:text></xsl:param> Description This parameter allows you to control the punctuation of certain types of generated cross reference text. When cross reference text is generated for an xref or olink element using an xrefstyle attribute that makes use of the select: feature, and the selected components include both label and page but no title, then the value of this parameter is inserted between label and page number in the output. If a title is included, then other separators are used. Namexref.label-title.separator — Punctuation or space separating label from title in xref Synopsis<xsl:param name="xref.label-title.separator">: </xsl:param> Description This parameter allows you to control the punctuation of certain types of generated cross reference text. When cross reference text is generated for an xref or olink element using an xrefstyle attribute that makes use of the select: feature, and the selected components include both label and title, then the value of this parameter is inserted between label and title in the output. Namexref.title-page.separator — Punctuation or space separating title from page number in xref Synopsis<xsl:param - name="xref.title-page.separator"><xsl:text> </xsl:text></xsl:param> Description This parameter allows you to control the punctuation of certain types of generated cross reference text. When cross reference text is generated for an xref or olink element using an xrefstyle attribute that makes use of the select: feature, and the selected components include both title and page number, then the value of this parameter is inserted between title and page number in the output. ListsListsListsTable of Contentssegmentedlist.as.tablevariablelist.as.tablevariablelist.term.separatorvariablelist.term.break.afterNamesegmentedlist.as.table — Format segmented lists as tables? Synopsis<xsl:param name="segmentedlist.as.table" select="0"></xsl:param> Description If non-zero, segmentedlists will be formatted as tables. Namevariablelist.as.table — Format variablelists as tables? Synopsis<xsl:param - name="variablelist.as.table" select="0"></xsl:param> Description If non-zero, variablelists will be formatted as tables. A processing instruction exists to specify a particular width for the column containing the terms: <?dbhtml term-width=".25in"?> You can override this setting with a processing instruction as the child of variablelist: <?dbhtml list-presentation="table"?> or <?dbhtml list-presentation="list"?>. 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. <variablelist> <?dbhtml list-presentation="table"?> <?dbhtml term-width="1.5in"?> <?dbfo list-presentation="list"?> <?dbfo term-width="1in"?> <varlistentry> <term>list</term> <listitem> <para> - Formatted as a table even if variablelist.as.table is set to 0. </para> </listitem> </varlistentry> </variablelist> Namevariablelist.term.separator — Text to separate terms within a multi-term varlistentry Synopsis<xsl:param - name="variablelist.term.separator">, </xsl:param> Description When a varlistentry contains multiple term elements, the string specified in the value of the variablelist.term.separator parameter is placed after each term except the last. NoteTo generate a line break between multiple terms in a varlistentry, set a non-zero value for the variablelist.term.break.after parameter. If you do so, you may also want to set the value of the variablelist.term.separator parameter to an empty string (to suppress rendering of the default comma and space after each term). Namevariablelist.term.break.after — Generate line break after each term within a multi-term varlistentry? Synopsis<xsl:param name="variablelist.term.break.after">0</xsl:param> Description Set a non-zero value for the variablelist.term.break.after parameter to generate a line break between terms in a multi-term varlistentry. NoteIf you set a non-zero value for variablelist.term.break.after, you may also want to set the value of the variablelist.term.separator parameter to an empty string (to suppress rendering of the default comma and space after each term). BibliographyBibliographyBibliographyTable of Contentsbibliography.stylebiblioentry.item.separatorbibliography.collectionbibliography.numberedNamebibliography.style — Style used for formatting of biblioentries. Synopsis<xsl:param name="bibliography.style">normal</xsl:param> Description Currently only normal and iso690 styles are supported. In order to use ISO690 style to the full extent you might need to use additional markup described on the following WiKi page11 http://wiki.docbook.org/topic/ISO690Bibliography. Namebiblioentry.item.separator — Text to separate bibliography entries Synopsis<xsl:param name="biblioentry.item.separator">. -

PDF BY DBLATEX

</xsl:param> Description Text to separate bibliography entries Namebibliography.collection — Name of the bibliography collection file Synopsis<xsl:param name="bibliography.collection">http://docbook.sourceforge.net/release/bibliography/bibliography.xml</x 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. A relative path in the parameter is interpreted in one of two ways: a.If your document contains no links to empty bibliographic elements, then the path is relative to the file containing the first bibliomixed element in the document. b.If your document does contain links to empty bibliographic elements, then the path is relative to the file containing the first such link element in the document. Once the collection file is opened by the first instance described above, it stays open for the current document and the relative path is not reinterpreted again. The stylesheets will format the bibliography in your document as if all of the entries referenced appeared there literally. Namebibliography.numbered — Should bibliography entries be numbered? Synopsis<xsl:param - name="bibliography.numbered" select="0"></xsl:param> Description If non-zero bibliography entries will be numbered GlossaryGlossaryGlossaryTable of Contentsglossterm.auto.linkfirstterm.only.linkglossary.collectionglossary.sortglossentry.show.acronymNa — Generate links from glossterm to glossentry automatically? Synopsis<xsl:param name="glossterm.auto.link" select="0"></xsl:param Description If non-zero, links from inline glossterms to the corresponding glossentry elements in a glossary or glosslist will be automatically generated. This is useful when your glossterms are consistent and you don't want to add links manually. The automatic link generation feature is not used on glossterm elements that have a linkend attribute. Namefirstterm.only.link — Does automatic glossterm linking only apply to firstterms? Synopsis<xsl:param name="firstterm.only.link" select="0"></xsl:param> Description If non-zero, only firstterms will be automatically linked to the glossary. If glossary linking is not enabled, this parameter has no effect. Nameglossary.collection — Name of the glossary collection file Synopsis<xsl:param name="glossary.collection"></xsl:par 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 -

PDF BY DBLATEX

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="partsofspeach">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 automic 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. A relative path in the parameter is interpreted in one of two ways: a.If the parameter glossterm.auto.link is set to zero, then the path is relative to the file containing the empty glossary element in the document. b.If the parameter glossterm.auto.link is set to non-zero, then the path is relative to the file containing the first inline glossterm or firstterm in the document to be linked. Once the collection file is opened by the first instance described above, it stays open for the current document and the relative path is not reinterpreted again. The stylesheets will format the glossary in your document as if all of the entries implicilty 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> Nameglossary.sort — Sort glossentry elements? Synopsis<xsl:param - name="glossary.sort" select="0"></xsl:param> Description If non-zero, then the glossentry elements within a glossary, glossdiv, or glosslist are sorted on the glossterm, using the current lang setting. If zero (the default), then glossentry elements are not sorted and are presented in document order. Nameglossentry.show.acronym — Display glossentry acronyms? Synopsis<xsl:param name="glossentry.show.acronym">no</xsl:param> Description A setting of "yes" means they should be displayed; "no" means they shouldn't. If "primary" is used, then they are shown as the primary text for the entry. NoteThis setting controls both acronym and abbrev elements in the glossentry. MiscellaneousMiscellaneousMiscellaneousTable of Contentsformal.proceduresformal.title.placementruninhead.default.title.end.punctruninhead.title.end.punctshow.commentsshow.revisionflagshade.v — Selects formal or informal procedures Synopsis<xsl:param name="formal.procedures" select="1"></xsl:param> Description Formal procedures are numbered and always have a title. Nameformal.title.placement — Specifies where formal object titles should occur Synopsis<xsl:param name="formal.title.placement"> figure before example before equation before table before procedure before task 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". Nameruninhead.default.title.end.punct — Default punctuation character on a run-in-head Synopsis<xsl:param name="runinhead.default.title.end.pu Description If non-zero, For a formalpara, use the specified string as the separator between the title and following text. The

PDF BY DBLATEX

period is the default value. Nameruninhead.title.end.punct — Characters that count as punctuation on a run-in-head Synopsis<xsl:param name="runinhead.title.end.punct">.!?:</xsl:param> Description Specify which characters are to be counted as punctuation. These characters are checked for a match with the last character of the title. If no match is found, the runinhead.default.title.end.punct contents are inserted. This is to avoid duplicated punctuation in the output. Nameshow.comments — Display remark elements? Synopsis<xsl:param name="show.comments" select="1"></xsl:param> Description If non-zero, comments will be displayed, otherwise they are suppressed. Comments here refers to the remark element (which was called comment prior to DocBook 4.0), not XML comments (<-- like this -->) which are unavailable. Nameshow.revisionflag — Enable decoration of elements that have a revisionflag Synopsis<xsl:param name="show.revisionflag" select="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 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. Nameshade.verbatim — Should verbatim environments be shaded? Synopsis<xsl:param name="shade.verbatim" select="0"></xsl:param> Description In the FO stylesheet, if this parameter is non-zero then the shade.verbatim.style properties will be applied to verbatim environments. In the HTML stylesheet, this parameter is now deprecated. Use CSS instead. Nameshade.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> </xsl:attribute-set> Description Properties that specify the style of shaded verbatim listings. The parameters specified (the border and background color) are added to the styling of the xsl-fo output. A border might be specified as "thin black solid" for example. See xsl-fo11 http://www.w3.org/TR/2004/WD-xsl11-20041216/#border Namepunct.honorific — Punctuation after an honorific in a personal name. Synopsis<xsl:param name="punct.honorific">.</xsl:param> Description This parameter specifies the punctuation that should be added after an honorific in a personal name. Nametex.math.in.alt — TeX notation used for equations Synopsis<xsl:param name="tex.math.in.alt"></xsl:param> 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. WarningThis feature is useful for print/PDF output only if you use the obsolete and now unsupported PassiveTeX XSL-FO engine. Related Parameters tex.math.delims, passivetex.extensions, tex.math.file Nametex.math.file — Name of temporary file for generating images from equations Synopsis<xsl:param name="tex.math.file">tex-math-equations.tex</xsl:param> Description Name of auxiliary file for TeX equations. This file can be processed by dvi2bitmap to get bitmap versions of equations for HTML output. Related Parameters tex.math.in.alt, tex.math.delims, More information For how-to documentation on embedding TeX equations and generating output from them, see DocBook XSL: TCG, DBTeXMath11 http://www.sagehill.net/docbookxsl/TexMath.html. Nametex.math.delims — Should equations output for processing by TeX be surrounded by math mode delimiters? Synopsis<xsl:param name="tex.math.delims" select="1"></xsl:param> Description For compatibility with DSSSL based DBTeXMath from Allin Cottrell you should set this parameter to 0. WarningThis feature is useful for print/PDF output only if you use the obsolete and now unsupported PassiveTeX XSL-FO engine. Related Parameters tex.math.in.alt, passivetex.extensions See Also You can also use the <?dbtex delims?> processing instruction to control whether delimiters are output. Namepixels.per.inch — How many pixels are there per inch? Synopsis<xsl:param name="pixels.per.inch">90</xsl:param> 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 Recommendation11 http://www.w3.org/TR/2004/WD-xsl11-20041216/. Namepoints.per.em — Specify the nominal size of an em-space in points Synopsis<xsl:param name="points.per.em">10</xsl:param> Description The fixed value used for calculations based upon the size of a character. The assumption made is that ten point font is in use. This assumption may not be valid. Nameuse.svg — Allow SVG in the result tree? Synopsis<xsl:param name="use.svg" select="1"></xsl:param> 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. Namemenuchoice.separator — Separator between items of a menuchoice other than guimenuitem and guisubmenu Synopsis<xsl:param name="menuchoice.separator">+</xsl:param> Description Separator used to connect items of a menuchoice other than guimenuitem and guisubmenu. The latter elements are linked with menuchoice.menu.separator. Namemenuchoice.menu.separator — Separator between items of a menuchoice with guimenuitem or guisubmenu Synopsis<xsl:param -

name="menuchoice.menu.separator"> → </xsl:param> Description Separator used to connect items of a menuchoice with guimenuitem or guisubmenu. Other elements are linked with menuchoice.separator. The default value is →, which is the → (right arrow) character entity. The current FOP (0.20.5) requires setting the font-family explicitly. The default value also includes spaces around the arrow, which will allow a line to break. Replace the spaces with   (nonbreaking space) if you don't want those spaces to break. Namedefault.float.class — Specifies the default float class Synopsis<xsl:param name="default.float.class"> <xsl:choose> <xsl:when test="contains($stylesheet.result.type,'html')">left</xsl:when> <xsl:otherwise>before</xsl:otherwise> </xsl:choose> </xsl:param> Description Selects the direction in which a float should be placed. for xsl-fo this is before, for html it is left. For Western texts, the before direction is the top of the page. Namefootnote.number.format — Identifies the format used for footnote numbers Synopsis<xsl:param name="footnote.number.format">1</xsl:para Description The footnote.number.format specifies the format to use for footnote numeration (1, i, I, a, or A). Nametable.footnote.number.f — Identifies the format used for footnote numbers in tables Synopsis<xsl:param name="table.footnote.number.format">a</xsl:param> Description The table.footnote.number.format specifies the format to use for footnote numeration (1, i, I, a, or A) in tables. Namefootnote.number.symbols — Special characters to use as footnote markers Synopsis<xsl:param name="footnote.number.symbols"></xsl: Description If footnote.number.symbols is not the empty string, footnotes will use the characters it contains as footnote symbols. For example, "*†‡◊✠" will identify footnotes with "*", "†", "‡", "◊", and "✠". If there are more footnotes than symbols, the stylesheets will fall back to numbered footnotes using footnote.number.format. The use of symbols for footnotes depends on the ability of your processor (or browser) to render the symbols you select. Not all systems are capable of displaying the full range of Unicode characters. If the quoted characters in the preceding paragraph are not displayed properly, that's a good indicator that you may have trouble using those symbols for footnotes. Nametable.footnote.number.symbols — Special characters to use a footnote markers in tables Synopsis<xsl:param name="table.footnote.number.symbols"></xsl:param> Description If table.footnote.number.symbols is not the empty string, table footnotes will use the characters it contains as footnote symbols. For example, "*†‡◊✠" will identify footnotes with "*", "†", "‡", "◊", and "✠". If there are more footnotes than symbols, the stylesheets will fall back to numbered footnotes using table.footnote.number.format. The use of symbols for footnotes depends on the ability of your processor (or browser) to render the symbols you select. Not all systems are capable of displaying the full range of Unicode characters. If the quoted characters in the preceding paragraph are not displayed properly, that's a good indicator that you may have trouble using those symbols for footnotes. Namehighlight.source — Should the content of programlisting be syntactically highlighted? Synopsis<xsl:param name="highlight.source" select="0"></xsl:param Description When this parameter is non-zero, the stylesheets will try to do syntax highlighting of the content of programlisting elements. You specify the language for each programlisting by using the language attribute. The highlight.default.language parameter can be used to specify the language for programlistings without a language attribute. Syntax highlighting also works for screen and synopsis elements. The actual highlighting work is done by the XSLTHL extension module. This is an external Java library that has to be downloaded separately (see below). In order to use this extension, you must•add xslthl-2.x.x.jar to your Java classpath. The latest version is available from the XSLT syntax highlighting project11 http://sourceforge.net/projects/xslthl at SourceForge. •use a customization layer in which you import one of the following stylesheet modules: •html/highlight.xsl •xhtml/highlight.xsl •xhtml-1_1/highlight.xsl •fo/highlight.xsl •let either the xslthl.config Java system property or the highlight.xslthl.config parameter point to the configuration file for syntax highlighting (using URL syntax). DocBook XSL comes with a ready-to-use configuration file, highlighting/xslthl-config.xml. The extension works with Saxon 6.5.x and Xalan-J. (Saxon 8.5 or later is also supported, but since it is an XSLT 2.0 processor it is not guaranteed to work with DocBook XSL in all circumstances.) The following is an example of a Saxon 6 command adapted for syntax highlighting, to be used on Windows: java -cp c:/Java/saxon.jar;c:/Java/xslthl-2.0.1.jar -Dxslthl.config=file:///c:/docbook-xsl/highlighting/xslthl-config.xml com.icl.saxon.StyleSheet -o test.html test.xml myhtml.xsl Namehighlight.xslthl.config — Location of XSLTHL configuration file Synopsis<xsl:param name="highlight.xslthl.config"></xsl:param> Description This location has precedence over the corresponding Java property. Please note that usually you have to specify location as URL not just as a simple path on the local filesystem. E.g. file:///home/user/xslthl/my-xslthl-config.xml. Namehighlight.default.language — Default language of programlisting Synopsis<xsl:param name="highlight.default.language"></xsl:param> Description This language is used when there is no language attribute on programlisting. Nameemail.delimiters.enabled — Generate delimiters around email addresses? Synopsis<xsl:param name="email.delimiters.enabled" select="1"></xsl:param> Description If non-zero, delimiters 11For delimiters, the stylesheets are currently hard-coded to output angle brackets. are generated around e-mail addresses (the output of the email element). Nameexsl.node.set.available — Is the test function-available('exsl:node-set') true? Synopsis<xsl:param name="exsl.node.set.available"> <xsl:choose> <xsl:when exsl:foo="" test="function-available('exsl:node-set') or contains(system-property('xsl:vendor'), 'Apache Software Foundation')">1</xsl:when> <xsl:otherwise>0</xsl:otherwise> </xsl:choose> </xsl:param> Description If non-zero, then the exsl:node-set() function is available to be used in the stylesheet. If zero, then the function is not available. This param automatically detects the presence of the function and does not normally need to be set manually. This param was created to handle a long-standing bug in the Xalan processor that fails to detect the function even though it is available. AnnotationsAnnotationsAnnotationsTable of Contentsannotation.supportannotation.jsannotation.cssannotation.graphic.openannotation.graphic.closeNameannotation.support — Enable annotations? Synopsis<xsl:param name="annotation.support" select="0"></xsl:param> Description If non-zero, the stylesheets

PDF BY DBLATEX

will attempt to support annotation elements in HTML by including some JavaScript (see annotation.js). Nameannotation.js — URIs identifying JavaScript files with support for annotation popups Synopsis<xsl:param name="annotation.js"> <xsl:text>http://docboo http://docbook.sourceforge.net/release/script/PopupWindow.js</xsl:text></xsl:param> Description If annotation.support is enabled and the document contains annotations, then the URIs listed in this parameter will be included. These JavaScript files are required for popup annotation support. Nameannotation.css — CSS rules for annotations Synopsis<xsl:param name="annotation.css">
/* ========== Annotations */ div.annotation-list { visibility: hidden; } div.annotation-nocss { position: absolute; visibility: hidden; } div.annotation-popup { position: absolute; z-index: 4; visibility: hidden; padding: 0px; margin: 2px; border-style: solid; border-width: 1px; width: 200px; background-color: white; } div.annotation-title { padding: 1px; font-weight: bold; - border-bottom-style: solid; border-bottom-width: 1px; color: white; background-color: black; } div.annotation-body { padding: 2px; } div.annotation-body p { margin-top: 0px; padding-top: 0px; - } div.annotation-close { position: absolute; top: 2px; right: 2px; } </xsl:param> Description If annotation.support is enabled and the document contains annotations, then the CSS in this parameter will be included in the document. Nameannotation.graphic.open — Image for identifying a link that opens an annotation popup Synopsis<xsl:param - name="annotation.graphic.open">http://docbook.sourceforge.net/release/images/annot-open.png</xsl:param> Description This image is used inline to identify the location of annotations. It may be replaced by a user provided graphic. The size should be approximately 10x10 pixels. Nameannotation.graphic.close — Image for identifying a link that closes an annotation popup Synopsis<xsl:param name="annotation.graphic.close"> http://docbook.sourceforge.net/release/images/annot-close.png</xsl:param> Description This image is used on popup annotations as the “x” that the user can click to dismiss the popup. This image is used on popup annotations as the “x” that the user can click to dismiss the popup. It may be replaced by a user provided graphic. The size should be approximately 10x10 pixels. GraphicsGraphicsGraphicsTable of Contentsimg.src.pathkeep.relative.image.urisgraphic.default. — Path to HTML/FO image files Synopsis<xsl:param name="img.src.path"></xsl:param> Description Add a path prefix to the value of the fileref attribute of graphic, inlinegraphic, and imagedata elements. The resulting compound path is used in the output as the value of the src attribute of img (HTML) or external-graphic (FO). The path given by img.src.path could be relative to the directory where the HTML/FO files are created, or it could be an absolute URI. The default value is empty. Be sure to include a trailing slash if needed. This prefix is not applied to any filerefs that start with "/" or contain "//:". Namekeep.relative.image.uris — Should image URIs be resolved against xml:base? Synopsis<xsl:param name="keep.relative.image.uris" select="1"></xsl:param> Description If non-zero, relative URIs (in, for example fileref attributes) will be used in the generated output. Otherwise, the URIs will be made absolute with respect to the base URI. Note that the stylesheets calculate (and use) the absolute form for some purposes, this only applies to the resulting output. Namegraphic.default.extension — Default extension for graphic filenames Synopsis<xsl:param name="graphic.default.extension"></xsl:param> 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. Namedefault.image.width — The default width of images Synopsis<xsl:param name="default.image.width"></xsl:param> Description If specified, this value will be used for the width attribute on images that do not specify any viewport dimensions11 http://docbook.org/tdg5/en/html/imagedata.html#viewport.area. Namenominal.image.width — The nominal image width Synopsis<xsl:param name="nominal.image.width" select="6 * $pixels.per.inch"></xsl:param> 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. Namenominal.image.depth — Nominal image depth Synopsis<xsl:param name="nominal.image.depth" - select="4 * $pixels.per.inch"></xsl:param> Description See nominal.image.width. Nameuse.embed.for.svg — Use HTML embed for SVG? Synopsis<xsl:param name="use.embed.for.svg" select="0"></xsl:param> 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. Namemake.graphic.viewport — Use tables in HTML to make viewports for graphics Synopsis<xsl:param name="make.graphic.viewport" select="1"></xsl:param> 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. Namepreferred.mediaobject.role — Select which mediaobject to use based on this value of an object’s role attribute. Synopsis<xsl:param name="preferred.mediaobject.role"></xsl:param> Description A mediaobject may contain several objects such as imageobjects. If the parameter use.role.for.mediaobject is non-zero, then the role attribute on imageobjects and other objects within a mediaobject container will be used to select which object will be used. If one of the

PDF BY DBLATEX

objects has a role value that matches the preferred.mediaobject.role parameter, then it has first priority for selection. If more than one has such a role value, the first one is used. See the use.role.for.mediaobject parameter for the sequence of selection. Nameuse.role.for.mediaobject — Use role attribute value for selecting which of several objects within a mediaobject to use. Synopsis<xsl:param name="use.role.for.mediaobject" select="1"></xsl:param> Description If non-zero, the role attribute on imageobjects or other objects within a mediaobject container will be used to select which object will be used. The order of selection when then parameter is non-zero is: 1.If the stylesheet parameter preferred.mediaobject.role has a value, then the object whose role equals that value is selected. 2.Else if an object's role attribute has a value of html for HTML processing or fo for FO output, then the first of such objects is selected. 3.Else the first suitable object is selected. If the value of use.role.for.mediaobject is zero, then role attributes are not considered and the first suitable object with or without a role value is used. Nameignore.image.scaling — Tell the stylesheets to ignore the author's image scaling attributes Synopsis<xsl:param name="ignore.image.scaling" select="0"></xsl:param> Description If non-zero, the scaling attributes on graphics and media objects are ignored. ChunkingChunkingChunkingTable of Contentschunker.output.cdata-section-elementschunker.output.doctype-publicchunker.output.doctype-systemchunker.output.encodingchunker.output.indentchunker.output.media-typechunker.output.methodch xml-declarationchunker.output.standalonesaxon.character.representationhtml.extuse.id.as.filenamehtml.extra.head.linksroot.filenameba section-elements — List of elements to escape with CDATA sections Synopsis<xsl:param name="chunker.output.cdata-section-elements"></xsl:param> 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. NoteThis parameter is documented here, but the declaration is actually in the chunker.xsl stylesheet module. Namechunker.output.doctype-public — Public identifer to use in the document type of generated pages Synopsis<xsl:param name="chunker.output.doctype-public"></xsl:param> 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. NoteThis parameter is documented here, but the declaration is actually in the chunker.xsl stylesheet module. Namechunker.output.doctype-system — System identifier to use for the document type in generated pages Synopsis<xsl:param name="chunker.output.doctype-system"></xsl:param> 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. NoteThis parameter is documented here, but the declaration is actually in the chunker.xsl stylesheet module. Namechunker.output.encoding — Encoding used in generated pages Synopsis<xsl:param name="chunker.output.encoding">ISO-8859-1</xsl:param> 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. NoteThis parameter is documented here, but the declaration is actually in the chunker.xsl stylesheet module. Namechunker.output.indent — Specification of indentation on generated pages Synopsis<xsl:param name="chunker.output.indent">no</xsl:param> Description This parameter specifies the value of the indent specification for generated pages. Not all processors support specification of this parameter. NoteThis parameter is documented here, but the declaration is actually in the chunker.xsl stylesheet module. Namechunker.output.media-type — Media type to use in generated pages Synopsis<xsl:param name="chunker.output.media-type"></xsl:param> Description This parameter specifies the media type that should be used by the chunking stylesheet. Not all processors support specification of this parameter. This parameter specifies the media type that should be used by the chunking stylesheet. This should be one from those defined in [RFC2045]11 http://www.ietf.org/rfc/rfc2045.txt and [RFC2046]22 http://www.ietf.org/rfc/rfc2046.txt NoteThis parameter is documented here, but the declaration is actually in the chunker.xsl stylesheet module. It must be one from html, xml or text Namechunker.output.method — Method used in generated pages Synopsis<xsl:param name="chunker.output.method">html</xsl:param 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. NoteThis parameter is documented here, but the declaration is actually in the chunker.xsl stylesheet module. Namechunker.output.omit-xml-declaration — Omit-xml-declaration for generated pages Synopsis<xsl:param name="chunker.output.omit-xml-declaration">no</xsl:param> Description This parameter specifies the value of the omit-xml-declaration specification for generated pages. Not all processors support specification of this parameter. NoteThis parameter is documented here, but the declaration is actually in the chunker.xsl stylesheet module. Namechunker.output.standalone — Standalone declaration for generated pages Synopsis<xsl:param name="chunker.output.standalone">no</xsl:param> Description This parameter specifies the value of the standalone specification for generated pages. It must be either yes or no. Not all processors support specification of this parameter. NoteThis parameter is documented here, but the declaration is actually in the chunker.xsl stylesheet module. Namesaxon.character.representation — Saxon character representation used in generated HTML pages Synopsis<xsl:param name="saxon.character.representation" select="'entity;decimal'"></xsl:param> Description This parameter has effect only when Saxon 6 is used (version 6.4.2 or later). It sets the character representation in files generated by the chunking stylesheets. If you want to suppress entity references for characters with direct representations in chunker.output.encoding, set the parameter value to native. For more information, see DocBook XSL: TCG, Saxon output character representation11 http://www.sagehill.net/docbookxsl/OutputEncoding.html#SaxonCharacter. NoteThis parameter is documented here, but the declaration is actually in the chunker.xsl stylesheet module. Namehtml.ext — Identifies the extension of generated HTML files Synopsis<xsl:param name="html.ext">.html</xsl:param> Description The extension identified by html.ext will be used as the filename extension for chunks created by this stylesheet. Nameuse.id.as.filename — Use

PDF BY DBLATEX

ID value of chunk elements as the filename? Synopsis<xsl:param name="use.id.as.filename" select="0"></xsl:param> Description If use.id.as.filename is non-zero, the filename of chunk elements that have IDs will be derived from the ID value. Namehtml.extra.head.links — Toggle extra HTML head link information Synopsis<xsl:param name="html.extra.head.links" -select="0"></xsl:param> 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). Nameroot.filename — Identifies the name of the root HTML file when chunking Synopsis<xsl:param -name="root.filename">index</xsl:param> Description The root.filename is the base filename for the chunk created for the root of each document processed. Namebase.dir — The base directory of chunks Synopsis<xsl:param name="base.dir"></xsl:param> Description If specified, the base.dir parameter identifies the output directory for chunks. (If not specified, the output directory is system dependent.) Starting with version 1.77 of the stylesheets, the param's value will have a trailing slash added if it does not already have one. CautionDo not use base.dir to add a filename prefix string to chunked files. Instead, use the chunked.filename.prefix parameter. Namechunked.filename.prefix — Filename prefix for chunked files Synopsis<xsl:param -name="chunked.filename.prefix"></xsl:param> Description If specified, the chunked.filename.prefix parameter specifies a prefix string to add to each generated chunk filename. For example: <xsl:param name="chunked.filename.prefix">admin-<xsl:param> will produce chunked filenames like: admin-index.html admin-ch01.html admin-ch01s01.html ... CautionTrying to use the base.dir parameter to add a string prefix (by omitting the trailing slash) no longer works (it never worked completely anyway). That parameter value should contain only a directory path, and now gets a trailing slash appended if it was omitted from the param. Namegenerate.manifest — Generate a manifest file? Synopsis<xsl:param name="generate.manifest" select="0"></xsl:param> Description If non-zero, a list of HTML files generated by the stylesheet transformation is written to the file named by the manifest parameter. Namemanifest — Name of manifest file Synopsis <xsl:param name="manifest">HTML.manifest</xsl:param> Description The name of the file to which a manifest is written (if the value of the generate.manifest parameter is non-zero). Namemanifest.in.base.dir — Should the manifest file be written into base.dir? Synopsis<xsl:param name="manifest.in.base.dir" -select="0"></xsl:param> Description If non-zero, the manifest file as well as project files for HTML Help and Eclipse Help are written into base.dir instead of the current directory. Namechunk.toc — An explicit TOC to be used for chunking Synopsis<xsl:param name="chunk.toc"></xsl:param> 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). Namechunk.tocs.and.lots — Should ToC and LoTs be in separate chunks? Synopsis<xsl:param name="chunk.tocs.and.lots" select="0"></xsl:param> 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. Namechunk.separate.lots — Should each LoT be in its own separate chunk? Synopsis<xsl:param name="chunk.separate.lots" select="0"></xsl:param> Description If non-zero, each of the ToC and LoTs (List of Examples, List of Figures, etc.) will be put in its own separate chunk. The title page includes generated links to each of the separate files. This feature depends on the chunk.tocs.and.lots parameter also being non-zero. Namechunk.tocs.and.lots.has.title — Should ToC and LoTs in a separate chunks have title? Synopsis<xsl:param -name="chunk.tocs.and.lots.has.title" select="1"></xsl:param> Description If non-zero title of document is shown before ToC/LoT in separate chunk. Namechunk.section.depth — Depth to which sections should be chunked Synopsis<xsl:param name="chunk.section.d select="1"></xsl:param> Description This parameter sets the depth of section chunking. Namechunk.first.sections — Chunk the first top-level section? Synopsis<xsl:param name="chunk.first.sections" select="0"></xsl:param> 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. Namechunk.quietly — Omit the chunked filename messages. Synopsis<xsl:param name="chunk.quietly" -select="0"></xsl:param> 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. Namechunk.append — Specifies content to append to chunked HTML output Synopsis<xsl:param name="chunk.append"></xsl:param> Description Specifies content to append to the end of HTML files output by the html/chunk.xsl stylesheet, after the closing <html> tag. You probably don't want to set any value for this parameter; but if you do, the only value it should ever be set to is a newline character:
 or
 Namenavig.graphics — Use graphics in navigational headers and footers? Synopsis<xsl:param name="navig.graphics" -select="0"></xsl:param> Description If 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. If zero, text is used instead of graphics. Namenavig.graphics.extension — Extension for navigational graphics Synopsis<xsl:param name="navig.graphics.extension">.gif</xsl:param> Description Sets the filename extension to use on navigational graphics used in the headers and footers of chunked HTML. Namenavig.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. Namenavig.showtitles — Display titles in HTML headers and footers? Synopsis<xsl:param name="navig.showtitles">1</xsl:param> Description If 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. ProfilingProfilingProfiling The following parameters can be used for attribute-

based profiling of your document. For more information about profiling, see DocBook XSL: TCG, Profiling (conditional text)11 http://www.sagehill.net/docbookxsl/Profiling.html. Table of Contentsprofile.archprofile.audienceprofile.conditionprofile.conformancepr — Target profile for arch attribute Synopsis<xsl:param name="profile.arch"></xsl:param> Description The 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, ...). Nameprofile.audience — Target profile for audience attribute Synopsis<xsl:param name="profile.audience"></xsl:param> 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, ...). Nameprofile.condition — Target profile for condition attribute Synopsis<xsl:param name="profile.condition"></xsl:param> Description The 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, ...). Nameprofile.conformance — Target profile for conformance attribute Synopsis<xsl:param name="profile.conformance"></xsl:param> Description The 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, ...). Nameprofile.lang — Target profile for lang attribute Synopsis<xsl:param name="profile.lang"></xsl:param> Description The 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, ...). Nameprofile.os — Target profile for os attribute Synopsis<xsl:param name="profile.os"></xsl:param> Description The 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, ...). Nameprofile.revision — Target profile for revision attribute Synopsis<xsl:param name="profile.revision"></xsl:param> Description The 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, ...). Nameprofile.revisionflag — Target profile for revisionflag attribute Synopsis<xsl:param name="profile.revisionflag"></xsl:param> Description The 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, ...). Nameprofile.role — Target profile for role attribute Synopsis<xsl:param name="profile.role"></xsl:param> Description The 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, ...). WarningNote that role is often used for other purposes than profiling. For example it is commonly used to get emphasize in bold font: <emphasis role="bold">very important</emphasis> If you are using role for these purposes do not forget to add values like bold to value of this parameter. If you forgot you will get document with small pieces missing which are very hard to track. For this reason it is not recommended to use role attribute for profiling. You should rather use profiling specific attributes like userlevel, os, arch, condition, etc. Nameprofile.security — Target profile for security attribute Synopsis<xsl:param name="profile.security"></xsl:param> Description The 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, ...). Nameprofile.status — Target profile for status attribute Synopsis<xsl:param name="profile.status"></xsl:param> Description The 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, ...). Nameprofile.userlevel — Target profile for userlevel attribute Synopsis<xsl:param name="profile.userlevel"></xsl:param> Description The 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, ...). Nameprofile.vendor — Target profile for vendor attribute Synopsis<xsl:param -

PDF BY DBLATEX

name="profile.vendor"></xsl:param> Description The 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, ...). Nameprofile.wordsize — Target profile for wordsize attribute Synopsis<xsl:param name="profile.wordsize"></xsl:param> Description The 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, ...). Nameprofile.attribute — Name of user-specified profiling attribute Synopsis<xsl:param name="profile.attribute"></xsl:param> Description This parameter is used in conjuction 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, ...). Nameprofile.value — Target profile for user-specified attribute Synopsis<xsl:param name="profile.value"></xsl:param> Description When you are using this parameter you must also specify name of profiling attribute with parameter profile.attribute. The 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, ...). Nameprofile.separator — Separator character for compound profile values Synopsis<xsl:param name="profile.separator">;</xsl:param> Description Separator character used for compound profile values. See profile.arch HTML HelpHTML HelpHTML HelpTable of Contentshtmlhelp.encodinghtmlhelp.autolabelhtmlhelp.chmhtmlhelp.default.topichtmlhelp.display.progresshtmlhelp.hhphtmlhelp.hhchtmlhelp.hhkh — Character encoding to use in files for HTML Help compiler. Synopsis<xsl:param name="htmlhelp.encoding">iso-8859-1</xsl:param> Description The HTML Help Compiler is not UTF-8 aware, so you should always use an appropriate single-byte encoding here. See also DocBook XSL: TCG, Processing options11 http://www.sagehill.net/docbookxsl/HtmlHelp.html#HelpProcOptio Namehtmlhelp.autolabel — Should tree-like ToC use autonumbering feature? Synopsis<xsl:param name="htmlhelp.autolabel" select="0"></xsl:param> Description Set this to non-zero to include chapter and section numbers into ToC in the left panel. Namehtmlhelp.chm — Filename of output HTML Help file. Synopsis<xsl:param name="htmlhelp.chm">htmlhelp.chm</xsl:param> Description Set the name of resulting CHM file Namehtmlhelp.default.topic — Name of file with default topic Synopsis<xsl:param name="htmlhelp.default.topic"></xsl:param> 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">pr01.html</xsl:paran Namehtmlhelp.display.progress — Display compile progress? Synopsis<xsl:param name="htmlhelp.display.progress" select="1"></xsl: Description Set to non-zero to to display compile progress Namehtmlhelp.hhp — Filename of project file. Synopsis<xsl:param name="htmlhelp.hhp">htmlhelp.hhp</xsl:param> Description Change this parameter if you want different name of project file than htmlhelp.hhp. Namehtmlhelp.hhc — Filename of TOC file. Synopsis<xsl:param name="htmlhelp.hhc">toc.hhc</xsl:param> Description Set the name of the TOC file. The default is toc.hhc. Namehtmlhelp.hhk — Filename of index file. Synopsis<xsl:param name="htmlhelp.hhk">index.hhk</xsl:param> Description set the name of the index file. The default is index.hhk. Namehtmlhelp.hhp.tail — Additional content for project file. Synopsis<xsl:param name="htmlhelp.hhp.tail"></xsl:param> Description If you want to include some additional parameters into project file, store appropriate part of project file into this parameter. Namehtmlhelp.hhp.window — Name of default window. Synopsis<xsl:param name="htmlhelp.hhp.window">Main</xsl:param> Description Name of default window. If empty no [WINDOWS] section will be added to project file. Namehtmlhelp.hhp.windows — Definition of additional windows Synopsis<xsl:param name="htmlhelp.hhp.windows"></xsl:param> Description Content of this parameter is placed at the end of [WINDOWS] section of project file. You can use it for defining your own addtional windows. Namehtmlhelp.enhanced.decompilation — Allow enhanced decompilation of CHM? Synopsis<xsl:param name="htmlhelp.enhanced.decompilation" select="0"></xsl:param> Description When non-zero this parameter enables enhanced decompilation of CHM. Namehtmlhelp.enumerate.images — Should the paths to all used images be added to the project file? Synopsis<xsl:param name="htmlhelp.enumerate.images" select="0"></xsl:param> Description Set to non-zero if you insert images into your documents as external binary entities or if you are using absolute image paths. Namehtmlhelp.force.map.and.alias — Should [MAP] and [ALIAS] sections be added to the project file unconditionally? Synopsis<xsl:param name="htmlhelp.force.map.an select="0"></xsl:param> Description Set to non-zero if you have your own alias.h and context.h files and you want to include references to them in the project file. Namehtmlhelp.map.file — Filename of map file. Synopsis<xsl:param name="htmlhelp.map.file">cont Description Set the name of map file. The default is context.h. (used for context-sensitive help). Namehtmlhelp.alias.file — Filename of alias file. Synopsis<xsl:param name="htmlhelp.alias.file">alias.h</xsl:param> Description Specifies the filename of the alias file (used for context-sensitive help). Namehtmlhelp.hhc.section.depth — Depth of TOC for sections in a left pane. Synopsis<xsl:param name="htmlhelp.hhc.section.depth">5</xsl:param> Description Set the section depth in the left pane of HTML Help viewer. Namehtmlhelp.hhc.show.root — Should there be an entry for the root element in the ToC? Synopsis<xsl:param name="htmlhelp.hhc.show.root" select="1"></xsl:param> Description If set to zero, there will be no entry for the root element in the ToC. This is useful when you want to provide the user with an expanded ToC as a default. Namehtml-

help.hhc.folders.instead.books — Use folder icons in ToC (instead of book icons)? Synopsis<xsl:param name="htmlhelp.hhc.folders.inst select="1"></xsl:param> Description Set to non-zero for folder-like icons or zero for book-like icons in the ToC. If you want to use folder-like icons, you must switch off the binary ToC using htmlhelp.hhc.binary. Namehtmlhelp.hhc.binary — Generate binary ToC? Synopsis<xsl:param name="htmlhelp.hhc.binary" select="1"></xsl:param> Description Set to non-zero to generate a binary TOC. You must create a binary TOC if you want to add Prev/Next buttons to toolbar (which is default behaviour). Files with binary TOC can't be merged. Namehtmlhelp.hhc.width — Width of navigation pane Synopsis<xsl:param name="htmlhelp.hhc.width"></xsl:param> Description This parameter specifies the width of the navigation pane (containing TOC and other navigation tabs) in pixels. Namehtmlhelp.title — Title of HTML Help Synopsis<xsl:param name="htmlhelp.title"></xsl:p Description Content of this parameter will be used as a title for generated HTML Help. If empty, title will be automatically taken from document. Namehtmlhelp.show.menu — Should the menu bar be shown? Synopsis<xsl:param name="htmlhelp.show.menu" select="0"></xsl:param> Description Set to non-zero to have an application menu bar in your HTML Help window. Namehtmlhelp.show.toolbar.text — Show text under toolbar buttons? Synopsis<xsl:param name="htmlhelp.show.toolbar.text" select="1"></xsl:pa Description Set to non-zero to display texts under toolbar buttons, zero to switch off displays. Namehtmlhelp.show.advanced.search — Should advanced search features be available? Synopsis<xsl:param name="htmlhelp.show.advanced.search" select="0"></xsl:param> Description If you want advanced search features in your help, turn this parameter to 1. Namehtmlhelp.show.favorities — Should the Favorites tab be shown? Synopsis<xsl:param name="htmlhelp.show.favorities" select="0"></xsl:param> Description Set to non-zero to include a Favorites tab in the navigation pane of the help window. Namehtmlhelp.button.hideshow — Should the Hide/Show button be shown? Synopsis<xsl:param name="htmlhelp.button.hideshow" select="1"></xsl:param> Description Set to non-zero to include the Hide/Show button shown on toolbar Namehtmlhelp.button.back — Should the Back button be shown? Synopsis<xsl:param name="htmlhelp.button.back" select="1"></xsl:param> Description Set to non-zero to include the Hide/Show button shown on toolbar Namehtmlhelp.button.forward — Should the Forward button be shown? Synopsis<xsl:param name="htmlhelp.button.forward" select="0"></xsl:param> Description Set to non-zero to include the Forward button on the toolbar. Namehtmlhelp.button.stop — Should the Stop button be shown? Synopsis<xsl:param name="htmlhelp.button.stop" select="0"></xsl:param> Description If you want Stop button shown on toolbar, turn this parameter to 1. Namehtmlhelp.button.refresh — Should the Refresh button be shown? Synopsis<xsl:param name="htmlhelp.button.refresh" select="0"></xsl:param> Description Set to non-zero to include the Stop button on the toolbar. Namehtmlhelp.button.home — Should the Home button be shown? Synopsis<xsl:param name="htmlhelp.button.home" select="0"></xsl:param> Description Set to non-zero to include the Home button on the toolbar. Namehtmlhelp.button.home.url — URL address of page accessible by Home button Synopsis<xsl:param name="htmlhelp.button.home.url"></xsl:param> Description URL address of page accessible by Home button. Namehtmlhelp.button.options — Should the Options button be shown? Synopsis<xsl:param name="htmlhelp.button.options" select="1"></xsl:param> Description If you want Options button shown on toolbar, turn this parameter to 1. Namehtmlhelp.button.print — Should the Print button be shown? Synopsis<xsl:param name="htmlhelp.button.print" select="1"></xsl:param> Description Set to non-zero to include the Print button on the toolbar. Namehtmlhelp.button.locate — Should the Locate button be shown? Synopsis<xsl:param name="htmlhelp.button.locate" select="0"></xsl:param> Description If you want Locate button shown on toolbar, turn this parameter to 1. Namehtmlhelp.button.jump1 — Should the Jump1 button be shown? Synopsis<xsl:param name="htmlhelp.button.jump1" select="0"></xsl:param> Description Set to non-zero to include the Jump1 button on the toolbar. Namehtmlhelp.button.jump1.url — URL address of page accessible by Jump1 button Synopsis<xsl:param name="htmlhelp.button.jump1.url"></xsl:param> Description URL address of page accessible by Jump1 button. Namehtmlhelp.button.jump1.title — Title of Jump1 button Synopsis<xsl:param name="htmlhelp.button.jump1.title">User1</xsl:param> Description Title of Jump1 button. Namehtmlhelp.button.jump2 — Should the Jump2 button be shown? Synopsis<xsl:param name="htmlhelp.button.jump2" select="0"></xsl:param> Description Set to non-zero to include the Jump2 button on the toolbar. Namehtmlhelp.button.jump2.url — URL address of page accessible by Jump2 button Synopsis<xsl:param name="htmlhelp.button.jump2 Description URL address of page accessible by Jump2 button. Namehtmlhelp.button.jump2.title — Title of Jump2 button Synopsis<xsl:param name="htmlhelp.button.jump2.title">User2</xsl:param> Description Title of Jump2 button. Namehtmlhelp.button.next — Should the Next button be shown? Synopsis<xsl:param name="htmlhelp.button.next" select="1"></xsl:param> Description Set to non-zero to include the Next button on the toolbar. Namehtmlhelp.button.prev — Should the Prev button be shown? Synopsis<xsl:param name="htmlhelp.button.prev" select="1"></xsl:param> Description Set to non-zero to include the Prev button on the toolbar. Namehtmlhelp.button.zoom — Should the Zoom button be shown? Synopsis<xsl:param name="htmlhelp.button.zoom" select="0"></xsl:param> Description Set to non-zero to include the Zoom button on the toolbar. Namehtmlhelp.remember.window.position — Remember help window position? Synopsis<xsl:param name="htmlhelp.remember.windo select="0"></xsl:param> Description Set to non-zero to remember help window position between starts. Namehtmlhelp.window.geometr — Set initial geometry of help window Synopsis<xsl:param name="htmlhelp.window.geometry"></xsl:param> Description This parameter specifies initial position of help window. E.g. <xsl:param name="htmlhelp.window.geometry">[160,64,992,704]</xsl:param> Namehtmlhelp.use.hhk — Should the index be built using the HHK file? Synopsis<xsl:param name="htmlhelp.use.hhk" select="0"></xsl:param> Description If non-zero, the index is created using the HHK file (instead of using object elements in the HTML files). For more information, see DocBook XSL: TCG, Generating an index11 http://www.sagehill.net/docbookxsl/HtmlHelp.htm Namehtmlhelp.only — Should only project files be generated? Synopsis<xsl:param name="htmlhelp.only" select="0"></xsl:param>

Description Set to non-zero if you want to play with various HTML Help parameters and you don't need to regenerate all HTML files. This setting will not process whole document, only project files (hhp, hhc, hhk,...) will be generated. Eclipse Help PlatformEclipse Help PlatformEclipse Help PlatformTable of Contentseclipse.autolabeleclipse.plugin.nameeclipse.plugin.ideclipse.plugin.p — Should tree-like ToC use autonumbering feature? Synopsis<xsl:param name="eclipse.autolabel" select="0"></xsl:param> Description If you want to include chapter and section numbers into ToC in the left panel, set this parameter to 1. Nameeclipse.plugin.nam — Eclipse Help plugin name Synopsis<xsl:param name="eclipse.plugin.name">DocBook Online Help Sample</xsl:param> Description Eclipse Help plugin name. Nameeclipse.plugin.id — Eclipse Help plugin id Synopsis<xsl:param name="eclipse.plugin.id">cor Description Eclipse Help plugin id. You should change this id to something unique for each help. Nameeclipse.plugin.provider — Eclipse Help plugin provider name Synopsis<xsl:param name="eclipse.plugin.provider">Example provider</xsl:param> Description Eclipse Help plugin provider name. WebHelpWebHelpWebHelpTable of Contentswebhelp.autolabelwebhelp.base.dirwebhelp.c — Should tree-like ToC use autonumbering feature? Synopsis<xsl:param name="webhelp.autolabel">0</xsl:param> Description To include chapter and section numbers the table of contents pane, set this parameter to 1. Namewebhelp.base.dir — The base directory for webhelp output. Synopsis<xsl:param name="webhelp.base.dir">docs</xsl:param> Description If specified, the webhelp.base.dir parameter identifies the output directory for webhelp. (If not specified, the output directory is system dependent.) By default, this parameter is set to docs. Namewebhelp.common.dir — Path to the directory for the common webhelp resources (JavaScript, css, common images, etc). Synopsis<xsl:param name="webhelp.common.dir">../common/</xsl:param> Description By default, webhelp creates a common directory containing resources such as JavaScript files, css, common images, etc. In some cases you may prefer to store these files in a standard location on your site and point all webhelp documents to that location. You can use this parameter to control the urls written to these common resources. For example, you might set this parameter to /common and create a single common directory at the root of your web server. Namewebhelp.default.topic — The name of the file to which the start file in the webhelp base directory redirects Synopsis<xsl:param name="webhelp.default.topic">index.html</xsl:param> Description Currently webhelp creates a base directory and puts the output files in a content subdirectory. It creates a file in the base directory that redirects to a configured file in the content directory. The webhelp.default.topic parameter lets you configure the name of the file that is redirected to. NoteThis parameter will be removed from a future version of webhelp along with the content directory. Namewebhelp.include.search.tab — Should the webhelp output include a Search tab? Synopsis<xsl:param name="webhelp.include.search.tab">1</xsl:param> Description Set this parameter to 0 to suppress the search tab from webhelp output. Namewebhelp.indexer.language — The language to use for creating the webhelp search index. Synopsis<xsl:param name="webhelp.indexer.language">en</xsl:param> Description To support stemming in the client-side webhelp stemmer, you must provide the language code. By default, the following languages are supported: •en: English •de: German •fr: French •zh: Chinese •ja: Japanese •ko: Korean See the webhelp documentation for information on adding support for additional languages. Namewebhelp.start.filename — The name of the start file in the webhelp base directory. Synopsis<xsl:param name="webhelp.start.filename">index.html</xsl:param> Description Currently webhelp creates a base directory and puts the output files in a content subdirectory. It creates a file in the base directory that redirects to a configured file in the content directory. The webhelp.start.filename parameter lets you configure the name of the redirect file. NoteThis parameter will be removed from a future version of webhelp along with the content directory. Namewebhelp.tree.cookie.id — Controls how the cookie that stores the webhelp toc state is named. Synopsis<xsl:param name="webhelp.tree.cookie.id" select="concat('treeview-', count(//node()))"></xsl:param> Description The webhelp output does not use a frameset. Instead, the table of contents is a div on each page. To preserve the state of the table of contents as the user navigates from page to page, webhelp stores the state in a cookie and reads that cookie when you get to the next page. If you've published several webhelp documents on the same domain, it is important that each cookie have a unique id. In lieu of calling on a GUID generator, by default this parameter is just set to the number of nodes in the document on the assumption that it is unlikely that you will have more than one document with the exact number of nodes. A more optimal solution would be for the user to pass in some unique, stable identifier from the build system to use as the webhelp cookie id. For example, if you have safeguards in place to ensure that the xml:id of the root element of each document will be unique on your site, then you could set webhelp-tree.cookie.id as follows: <xsl:param name="webhelp.tree.cookie.id"> <xsl:choose> <xsl:when test="/*/@xml:id"> <xsl:value-of select="concat('treeview-',/*/@xml:id)"/> </xsl:when> <xsl:otherwise> <xsl:value-of select="concat('treeview-', count(//node()))"/> </xsl:otherwise> </xsl:choose> </xsl:param> JavaHelpJavaHelpJavaHelpTable of Contentsjavahelp.encodingNamejavahelp.encoding — Character encoding to use in control files for JavaHelp. Synopsis<xsl:param name="javahelp.encoding">iso-8859-1</xsl:param> Description JavaHelp crashes on some characters when written as character references. In that case you can use this parameter to select an appropriate encoding. LocalizationLocalizationLocalizationTable of Contentsl10n.gentext.languagel10n.gentext.default.languagel10n.gentext.use.xref.languagel10n.lang.value.rfc.compliantwriting.mod — Sets the gentext language Synopsis<xsl:param name="l10n.gentext.language">></xsl:param> Description If this parameter is set to any value other than the empty string, its value will be used as the value for the language when generating text. Setting l10n.gentext.language overrides any settings within the document being formatted. It's much more likely that you might want to set the l10n.gentext.default.language parameter. Namel10n.gentext.default.language — Sets the default language for generated text Synopsis<xsl:param name="l10n.gentext.default.language">en</xsl:param> Description The value of the l10n.gentext.default.language parameter is used as the language for generated text if no setting is provided in the source

document. Namel10n.gentext.use.xref.language — Use the language of target when generating cross-reference text? Synopsis<xsl:param name="l10n.gentext.use.xref.language" select="0"></xsl:param> Description If non-zero, the language of the target will be used when generating cross reference text. Usually, the “current” language is used when generating text (that is, the language of the element that contains the cross-reference element). But setting this parameter allows the language of the element pointed to to control the generated text. Consider the following example: <para lang="en">See also <xref -linkend="chap3"/>.</para> Suppose that Chapter 3 happens to be written in German. If l10n.gentext.use.xref.language is non-zero, the resulting text will be something like this: See also Kapital 3. Where the more traditional rendering would be: See also Chapter 3. Namel10n.lang.value.rfc.compliant — Make value of lang attribute RFC compliant? Synopsis<xsl:param -name="l10n.lang.value.rfc.compliant" select="1"></xsl:param> Description If non-zero, ensure that the values for all lang attributes in HTML output are RFC compliant11Section 8.1.1, Language Codes[http://www.w3.org/TR/REC-html40/struct/dirlang.html#h 8.1.1], in the HTML 4.0 Recommendation states that: [RFC1766] defines and explains the language codes that must be used in HTML documents.Briefly, language codes consist of a primary code and a possibly empty series of subcodes: language-code = primary-code ("-" subcode)* And in RFC 1766, Tags for the Identification of Languages[http://www.ietf.org/rfc/rfc1766.txt], the EBNF for "language tag" is given as: Language-Tag = Primary-tag *("-" Subtag) Primary-tag = 1*8ALPHA Subtag = 1*8ALPHA . by taking any underscore characters in any lang values found in source documents, and replacing them with hyphen characters in output HTML files. For example, zh_CN in a source document becomes zh-CN in the HTML output form that source. NoteThis parameter does not cause any case change in lang values, because RFC 1766 explicitly states that all "language tags" (as it calls them) "are to be treated as case insensitive". Namewriting.mode — Direction of text flow based on locale Synopsis<xsl:param name="writing.mode"> <xsl:call-template name="gentext"> <xsl:with-param name="key">writing-mode</xsl:with-param> <xsl:with-param name="lang"> <xsl:call-template name="l10n.language"> <xsl:with-param name="target" select="/*[1]"></xsl:with-param> </xsl:call-template> </xsl:with-param> </xsl:call-template> </xsl:param> Description Sets direction of text flow and text alignment based on locale. The value is normally taken from the gentext file for the lang attribute of the document’s root element, using the key name ’writing-mode’ to look it up in the gentext file. But this param can also be set on the command line to override that gentext value. Accepted values are: lr-tbLeft-to-right text flow in each line, lines stack top to bottom. rl-tbRight-to-left text flow in each line, lines stack top to bottom. tb-rlTop-to-bottom text flow in each vertical line, lines stack right to left. Supported by only a few XSL-FO processors. Not supported in HTML output. lrShorthand for lr-tb. rlShorthand for rl-tb. tbShorthand for tb-rl. Part II. FO Parameter Reference This is reference documentation for all user-configurable parameters in the DocBook XSL FO stylesheets (for generating XSL-FO output destined for final print/PDF output). FO Parameter ReferenceFO Parameter ReferenceTable of ContentsXXVIII. Admonitionsadmon.graphicsadmon.graphics.extensionadmon.graphics.pathadmon.textlabeladmonition.title.propertiesadmonition.propertiesgraphical.a
Calloutscalloutlist.propertiescallout.propertiescallout.defaultcolumncallout.graphicscallout.graphics.extensioncallout.graphics.number.l
ToC/LoT/Index Generationautotoc.label.separatorprocess.empty.source.tocprocess.source.tocgenerate.tocgenerate.indexmake.index.ma
Processor Extensionsarbortext.extensionsaxf.extensionsfop.extensionsfop1.extensionspassivetex.extensionstex.math.in.alttex.math.delin
Stylesheet Extensionslinenumbering.everyNthlinenumbering.extensionlinenumbering.separatorlinenumbering.widthtablecolumns.exten
Automatic labellingappendix.autolabelchapter.autolabelpart.autolabelreference.autolabelpreface.autolabelsection.autolabelsection.autol
XSLT ProcessingrootidXXXV. Meta/*Infomake.single.year.rangesmake.year.rangesauthor.othername.in.middleXXXVI. Reference Pagesfuncsynopsis.decorationfuncsynopsis.stylefunction.parensrefentry.generate.namerefentry.generate.titlerefentry.pagebreakrefe
Tablesdefault.table.widthnominal.table.widthdefault.table.framedefault.table.rulestable.cell.paddingtable.frame.border.thicknesstable.fr
Linkingcurrent.docidactivate.external.olinkscollect.xref.targetsinsert.olink.page.numberinsert.olink.pdf.fragolink.base.uriolink.debugol
Cross Referencesinsert.xref.page.numberxref.propertiesxref.label-title.separatorxref.label-page.separatorxref.title-page.separatorinsert.l
Listscompact.list.item.spacingitemizedlist.propertiesitemizedlist.label.propertiesitemizedlist.label.widthlist.block.propertieslist.block.sp
QAndASetqandadiv.autolabelqanda.inherit.numerationqanda.defaultlabelqanda.in.tocqanda.nested.in.tocXLII. Bibliographybibliography.stylebiblioentry.item.separatorbibliography.collectionbibliography.numberedbiblioentry.propertiesXLIII. Glossaryglossterm.a
Miscellaneousformal.proceduresformal.title.placementruninhead.default.title.end.punctruninhead.title.end.punctshow.commentspunct.h
Graphicsgraphic.default.extensiondefault.image.widthpreferred.mediaobject.roleuse.role.for.mediaobjectignore.image.scalingimg.src.pa
Pagination and General Stylespage.heightpage.height.portraitpage.margin.bottompage.margin.innerpage.margin.outerpage.margin.toppa
heightcolumn.count.backcolumn.count.bodycolumn.count.frontcolumn.count.indexcolumn.count.lotcolumn.count.titlepagecolumn.gap
Font Familiesbody.font.familydingbat.font.familymonospace.font.familysans.font.familytitle.font.familysymbol.font.familyXLVIII.
Property Setsformal.object.propertiesformal.title.propertiesinformal.object.propertiesmonospace.propertiesverbatim.propertiesmonospa
Profilingprofile.archprofile.audienceprofile.conditionprofile.conformanceprofile.langprofile.osprofile.revisionprofile.revisionflagprofile.
Localizationl10n.gentext.languagel10n.gentext.default.languagel10n.gentext.use.xref.languagel10n.lang.value.rfc.compliantwriting.mo
EBNFebnf.assignmentebnf.statement.terminatorLII. Prepresscrop.markscrop.mark.widthcrop.mark.offsetcrop.mark.bleedAdmonitionsA
of Contentsadmon.graphicsadmon.graphics.extensionadmon.graphics.pathadmon.textlabeladmonition.title.propertiesadmonition.proper
— Use graphics in admonitions? Synopsis<xsl:param name="admon.graphics" select="0"></xsl:param> Description If true (non-zero), admonitions are presented in an alternate style that uses a graphic. Default graphics are provided in the distribution. Nameadmon.graphics.extension — Filename extension for admonition graphics Synopsis<xsl:param name="admon.graphics.extension" Description Sets the filename extension to use on admonition graphics. The DocBook XSL distribution provides admonition

PDF BY DBLATEX

graphics in the following formats:•GIF (extension: .gif)•PNG (extension: .png)•SVG (extension: .svg)•TIFF (extension: .tif) Nameadmon.graphics.path — Path to admonition graphics Synopsis<xsl:param name="admon.graphics.path">images/</xsl:param> Description Sets the path to the directory containing the admonition graphics (caution.png, important.png etc). This location is normally relative to the output html directory. See base.dir Nameadmon.textlabel — Use text label in admonitions? Synopsis<xsl:param name="admon.textlabel" select="1"></xsl:param> Description If true (non-zero), admonitions are presented with a generated text label such as Note or Warning in the appropriate language. If zero, such labels are turned off, but any title child of the admonition element are still output. The default value is 1. Nameadmonition.title.properties — To set the style for admonitions titles. Synopsis<xsl:attribute-set name="admonition.title.properties"> <xsl:attribute name="font-size">14pt</xsl:attribute> <xsl:attribute name="font-weight">bold</xsl:attribute> <xsl:attribute name="hyphenate">false</xsl:attribu <xsl:attribute name="keep-with-next.within-column">always</xsl:attribute> </xsl:attribute-set> Description How do you want admonitions titles styled? Set the font-size, weight etc to the style required. Nameadmonition.properties — To set the style for admonitions. Synopsis<xsl:attribute-set name="admonition.properties"></xsl:attribute-set> Description How do you want admonitions styled? Set the font-size, weight, etc. to the style required Namegraphical.admonition.properties — To add properties to the outer block of a graphical admonition. Synopsis<xsl:attribute-set name="graphical.admonition.properties"> <xsl:attribute - name="space-before.optimum">1em</xsl:attribute> <xsl:attribute name="space-before.minimum">0.8em</xsl:attribute> - <xsl:attribute name="space-before.maximum">1.2em</xsl:attribute> <xsl:attribute name="space-after.optimum">1em</xsl:attribute> <xsl:attribute name="space-after.minimum">0.8em</xsl:attribute> <xsl:attribute name="space-after.maximum">1.2em</xsl:attribute> </xsl:attribute-set> Description These properties are added to the outer block containing the entire graphical admonition, including its title. It is used when the parameter admon.graphics is set to nonzero. Use this attribute-set to set the space above and below, and any indent for the whole admonition. In addition to these properties, a graphical admonition also applies the admonition.title.properties attribute-set to the title, and applies the admonition.properties attribute-set to the rest of the content. Namenongraphical.admonition.properties — To add properties to the outer block of a nongraphical admonition. Synopsis<xsl:attribute-set name="nongraphical.admonition.properties">> <xsl:attribute name="space-before.minimum">0.8em</xsl:att <xsl:attribute name="space-before.optimum">1em</xsl:attribute> <xsl:attribute name="space-before.maximum">1.2em</xsl:attribute <xsl:attribute name="margin-{$direction.align.start}">0.25in</xsl:attribute> <xsl:attribute name="margin-{$direction.align.end}">0.2 </xsl:attribute-set> Description These properties are added to the outer block containing the entire nongraphical admonition, including its title. It is used when the parameter admon.graphics is set to zero. Use this attribute-set to set the space above and below, and any indent for the whole admonition. In addition to these properties, a nongraphical admonition also applies the admonition.title.properties attribute-set to the title, and the admonition.properties attribute-set to the rest of the content. CalloutsCalloutsCalloutsTable of Contentscalloutlist.propertiescallout.propertiescallout.defaultcolumncallout.graphicscallout.graphics.extensioncall — Properties that apply to each list-block generated by calloutlist. Synopsis<xsl:attribute-set name="calloutlist.properties">> - <xsl:attribute name="space-before.optimum">1em</xsl:attribute> <xsl:attribute name="space-before.minimum">0.8em</xsl:attribute <xsl:attribute name="space-before.maximum">1.2em</xsl:attribute> <xsl:attribute name="provisional-distance-between-starts">2.2e <xsl:attribute name="provisional-label-separation">0.2em</xsl:attribute> </xsl:attribute-set> Description Properties that apply to the fo:list-block generated by calloutlist. Typically used to adjust spacing or margins of the entire list. Change the provisional-distance-between-starts attribute to change the indent of the list paragraphs relative to the callout numbers. Namecallout.properties — Properties that apply to the list-item generated by each callout within a calloutlist. Synopsis<xsl:attribute-set name="callout.properties"> </xsl:attribute-set> Description Properties that apply to the fo:list-item generated by each callout within a calloutlist. Typically used to add spacing properties. Namecallout.defaultcolumn — Indicates what column callouts appear in by default Synopsis<xsl:param name="callout.defaultcolumn">60</xsl:param> Description If a callout does not identify a column (for example, if it uses the linerange unit), it will appear in the default column. Namecallout.graphics — Use graphics for callouts? Synopsis<xsl:param name="callout.graphics" select="1"></xsl:param> 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. Namecallout.graphics.extension — Filename extension for callout graphics Synopsis<xsl:param - name="callout.graphics.extension">.svg</xsl:param> Description Sets the filename extension to use on callout graphics. The Docbook XSL distribution provides callout graphics in the following formats:•SVG (extension: .svg)•PNG (extension: .png)•GIF (extension: .gif) Namecallout.graphics.number.limit — Number of the largest callout graphic Synopsis<xsl:param name="callout.graphic Description If callout.graphics is non-zero, graphics are used to represent callout numbers instead of plain text. 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 "(plain text instead of a graphic)" will be used. Namecallout.graphics.path — Path to callout graphics Synopsis<xsl:param name="callout.graphics.path">images/callouts/</xsl:param> Description Sets the path to the directory holding the callout graphics. his location is normally relative to the output html directory. see base.dir. Always terminate the directory with / since the graphic file is appended to this string, hence needs the separator. Namecallout.icon.size — Specifies the size of callout marker icons Synopsis<xsl:param name="callout.icon.size">7pt</xsl:param> Description Specifies the size of the callout marker icons. The default size is 7 points. Namecallout.unicode — Use Unicode characters rather than images for callouts. Synopsis<xsl:param name="callout.unicode" select="0"></xsl:param> 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 back-

PDF BY DBLATEX

ground. Use this to select the Unicode character option. Namecallout.unicode.font — Specify a font for Unicode glyphs Synopsis<xsl:param name="callout.unicode.font">ZapfDingbats</xsl:param> Description The name of the font to specify around Unicode callout glyphs. If set to the empty string, no font change will occur. Namecallout.unicode.number.limit — Number of the largest unicode callout character Synopsis<xsl:param name="callout.unicode.number.limit">10</xsl:param> 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. Namecallout.unicode.start.character — First Unicode character to use, decimal value. Synopsis<xsl:param name="callout.unicode.start.character">10102</xsl:param> 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 values 9312 and 10102 are supported in the stylesheets for this parameter. Namecallouts.extension — Enable the callout extension Synopsis<xsl:param name="callouts.extension" select="1"></xsl:param> Description The callouts extension processes areaset elements in programlistingco and other text-based callout elements. ToC/LoT/Index GenerationToC/LoT/Index GenerationToC/LoT/Index GenerationTable of Contentsautotoc.label.separatorprocess.empty.source.tocprocess.source.tocgenerate.tocgenerate.indexmake.index.markupindex.methodindex.on.type — Separator between labels and titles in the ToC Synopsis<xsl:param name="autotoc.label.separator">. </xsl:param> Description String used to separate labels and titles in a table of contents. Nameprocess.empty.source.toc — Generate automated TOC if toc element occurs in a source document? Synopsis<xsl:param name="process.empty.source.toc" select="0"></xsl:param> Description Specifies that if an empty toc element is found in a source document, an automated TOC is generated at this point in the document. NoteDepending on what the value of the generate.toc parameter is, setting this parameter to 1 could result in generation of duplicate automated TOCs. So the process.empty.source.toc is primarily useful as an "override": by placing an empty toc in your document and setting this parameter to 1, you can force a TOC to be generated even if generate.toc says not to. Nameprocess.source.toc — Process a non-empty toc element if it occurs in a source document? Synopsis<xsl:param name="process.source.toc" select="0"></xsl:param> Description Specifies that the contents of a non-empty "hard-coded" toc element in a source document are processed to generate a TOC in output. NoteThis parameter has no effect on automated generation of TOCs. An automated TOC may still be generated along with the "hard-coded" TOC. To suppress automated TOC generation, adjust the value of the generate.toc paramameter. The process.source.toc parameter also has no effect if the toc element is empty; handling for empty toc is controlled by the process.empty.source.toc parameter. Namegenerate.toc — Control generation of ToCs and LoTs Synopsis<xsl:param name="generate.toc"> /appendix toc,title article/appendix nop /article toc,title book toc,title,figure,table,example,equation /chapter toc,title part toc,title /preface toc,title reference toc,title /sect1 toc /sect2 toc /sect3 toc /sect4 toc /sect5 toc /section toc set toc,title </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. Namegenerate.index — Do you want an index? Synopsis<xsl:param name="generate.index" select="1"></xsl:param> Description Specify if an index should be generated. Namemake.index.markup — Generate XML index markup in the index? Synopsis<xsl:param name="make.index.markup" select="0"></xsl:param> Description This parameter enables a very neat trick for getting properly merged, collated back-of-the-book indexes. G. Ken Holman suggested this trick at Extreme Markup Languages 2002 and I'm indebted to him for it. Jeni Tennison's excellent code in autoidx.xsl does a great job of merging and sorting indexterms in the document and building a back-of-the-book index. However, there's one thing that it cannot reasonably be expected to do: merge page numbers into ranges. (I would not have thought that it could collate and suppress duplicate page numbers, but in fact it appears to manage that task somehow.) Ken's trick is to produce a document in which the index at the back of the book is "displayed" in XML. Because the index is generated by the FO processor, all of the page numbers have been resolved. It's a bit hard to explain, but what it boils down to is that instead of having an index at the back of the book that looks like this: A. ap1, 1, 2, 3 you get one that looks like this: <indexdiv>A</indexdiv> <indexentry> <primaryie>ap1</primaryie>, <phrase role="pageno">1</phrase>, <phrase role="pageno">2</phrase>, <phrase role="pageno">3</phrase> </indexentry> After building a PDF file with this sort of odd-looking index, you can extract the text from the PDF file and the result is a proper index expressed in XML. Now you have data that's amenable to processing and a

PDF BY DBLATEX

simple Perl script (such as fo/pdf2index) can merge page ranges and generate a proper index. Finally, reformat your original document using this literal index instead of an automatically generated one and “bingo”! Nameindex.method — Select method used to group index entries in an index Synopsis<xsl:param name="index.method">basic</xsl:param> Description This parameter lets you select which method to use for sorting and grouping index entries in an index. Indexes in Latin-based languages that have accented characters typically sort together accented words and unaccented words. Thus “Á” (U+00C1 LATIN CAPITAL LETTER A WITH ACUTE) would sort together with “A” (U+0041 LATIN CAPITAL LETTER A), so both would appear in the “A” section of the index. Languages using other alphabets (such as Russian, which is written in the Cyrillic alphabet) and languages using ideographic chararacters (such as Japanese) require grouping specific to the languages and alphabets. The default indexing method is limited. It can group accented characters in Latin-based languages only. It cannot handle non-Latin alphabets or ideographic languages. The other indexing methods require extensions of one type or another, and do not work with all XSLT processors, which is why they are not used by default. The three choices for indexing method are: basic(default) Sort and groups words based only on the Latin alphabet. Words with accented Latin letters will group and sort with their respective primary letter, but words in non-Latin alphabets will be put in the “Symbols” section of the index. kosekThis method sorts and groups words based on letter groups configured in the DocBook locale file for the given language. See, for example, the French locale file common/fr.xml. This method requires that the XSLT processor supports the EXSLT extensions (most do). It also requires support for using user-defined functions in xsl:key (xsltproc does not). This method is suitable for any language for which you can list all the individual characters that should appear in each letter group in an index. It is probably not practical to use it for ideographic languages such as Chinese that have hundreds or thousands of characters. To use the kosek method, you must: 1.Use a processor that supports its extensions, such as Saxon 6 or Xalan (xsltproc and Saxon 8 do not). 2.Set the index.method parameter’s value to “kosek”. 3.Import the appropriate index extensions stylesheet module fo/autoidx-kosek.xsl or html/autoidx-kosek.xsl into your customization. kimberThis method uses extensions to the Saxon processor to implement sophisticated indexing processes. It uses its own configuration file, which can include information for any number of languages. Each language’s configuration can group words using one of two processes. In the enumerated process similar to that used in the kosek method, you indicate the groupings character-by-character. In the between-key process, you specify the break-points in the sort order that should start a new group. The latter configuration is useful for ideographic languages such as Chinese, Japanese, and Korean. You can also define your own collation algorithms and how you want mixed Latin-alphabet words sorted. •For a whitepaper describing the extensions, see: http://www.innodata-isogen.com/knowledge_center/white_papers/back_of_book_for_xsl_fo.pdf. •To download the extension library, see http://www.innodata-isogen.com/knowledge_center/tools_downloads/i18nsupport. To use the kimber method, you must: 1.Use Saxon (version 6 or 8) as your XSLT processor. 2.Install and configure the Innodata Isogen library, using the documentation that comes with it. 3.Set the index.method parameter’s value to “kimber”. 4.Import the appropriate index extensions stylesheet module fo/autoidx-kimber.xsl or html/autoidx-kimber.xsl into your customization. Nameindex.on.type — Select indexterms based on type attribute value Synopsis<xsl:param name="index.on.type" select="0"></xsl:param> Description If non-zero, then an index element that has a type attribute value will contain only those indexterm elements with a matching type attribute value. If an index has no type attribute or it is blank, then the index will contain all indexterms in the current scope. If index.on.type is zero, then the type attribute has no effect on selecting indexterms for an index. For those using DocBook version 4.2 or earlier, the type attribute is not available for index terms. However, you can achieve the same effect by using the role attribute in the same manner on indexterm and index, and setting the stylesheet parameter index.on.role to a nonzero value. Nameindex.on.role — Select indexterms based on role value Synopsis<xsl:param name="index.on.role" -select="0"></xsl:param> Description If non-zero, then an index element that has a role attribute value will contain only those indexterm elements with a matching role value. If an index has no role attribute or it is blank, then the index will contain all indexterms in the current scope. If index.on.role is zero, then the role attribute has no effect on selecting indexterms for an index. If you are using DocBook version 4.3 or later, you should use the type attribute instead of role on indexterm and index, and set the index.on.type to a nonzero value. Nameindex.preferred.page.properties — Properties used to emphasize page number references for significant index terms Synopsis<xsl:attribute-set name="index.preferred.page.properties"> <xsl:attribute -name="font-weight">bold</xsl:attribute> </xsl:attribute-set> Description Properties used to emphasize page number references for significant index terms (significance=preferred). Currently works only with XEP. Nameindex.entry.properties — Properties applied to the formatted entries in an index Synopsis<xsl:attribute-set name="index.entry.properties"> <xsl:attribute -name="start-indent">0pt</xsl:attribute> </xsl:attribute-set> Description This attribute set is applied to the block containing the entries in a letter division in an index. It can be used to set the font-size, font-family, and other inheritable properties that will be applied to all index entries. Nameindex.div.title.properties — Properties associated with the letter headings in an index Synopsis<xsl:attribute-set name="index.div.title.properties"> <xsl:attribute name="margin-{$direction.align.start}">0pt</xsl:attribute>
 <xsl:attribute name="font-size">14.4pt</xsl:attribute> <xsl:attribute name="font-family"><xsl:value-of select="$title.fontset"></xsl:v
of></xsl:attribute> <xsl:attribute name="font-weight">bold</xsl:attribute> <xsl:attribute name="keep-with-next.within-column">alw
 <xsl:attribute name="space-before.optimum"><xsl:value-of select="concat($body.font.master,’pt’)"></xsl:value-of></xsl:attribute>
 <xsl:attribute name="space-before.minimum"><xsl:value-of select="concat($body.font.master,’pt * 0.8’)"></xsl:value-of></xsl:attribu
 <xsl:attribute name="space-before.maximum"><xsl:value-of select="concat($body.font.master,’pt * 1.2’)"></xsl:value-of></xsl:attribu
 <xsl:attribute name="start-indent">0pt</xsl:attribute> </xsl:attribute-set> Description This attribute set is used on the letter

PDF BY DBLATEX

headings that separate the divisions in an index. Nameindex.number.separator — Override for punctuation separating page numbers in index Synopsis<xsl:param name="index.number.separator"></xsl:param> Description This parameter permits you to override the text to insert between page references in a formatted index entry. Typically that would be a comma and a space. Because this text may be locale dependent, this parameter's value is normally taken from a gentext template named 'number-separator' in the context 'index' in the stylesheet locale file for the language of the current document. This parameter can be used to override the gentext string, and would typically be used on the command line. This parameter would apply to all languages. So this text string can be customized in two ways. You can reset the default gentext string using the local.l10n.xml parameter, or you can override the gentext with the content of this parameter. The content can be a simple string, or it can be something more complex such as a call-template. In HTML index output, section title references are used instead of page number references. This punctuation appears between such section titles in an HTML index. Nameindex.range.separator — Override for punctuation separating the two numbers in a page range in index Synopsis<xsl:param name="index.range.separator"></xsl:param> Description This parameter permits you to override the text to insert between the two numbers of a page range in an index. This parameter is only used by those XSL-FO processors that support an extension for generating such page ranges (such as XEP). Because this text may be locale dependent, this parameter's value is normally taken from a gentext template named 'range-separator' in the context 'index' in the stylesheet locale file for the language of the current document. This parameter can be used to override the gentext string, and would typically be used on the command line. This parameter would apply to all languages. So this text string can be customized in two ways. You can reset the default gentext string using the local.l10n.xml parameter, or you can override the gentext with the content of this parameter. The content can be a simple string, or it can be something more complex such as a call-template. In HTML index output, section title references are used instead of page number references. So there are no page ranges and this parameter has no effect. Nameindex.term.separator — Override for punctuation separating an index term from its list of page references in an index Synopsis<xsl:param name="index.term.separator"></xsl:param> Description This parameter permits you to override the text to insert between the end of an index term and its list of page references. Typically that might be a comma and a space. Because this text may be locale dependent, this parameter's value is normally taken from a gentext template named 'term-separator' in the context 'index' in the stylesheet locale file for the language of the current document. This parameter can be used to override the gentext string, and would typically be used on the command line. This parameter would apply to all languages. So this text string can be customized in two ways. You can reset the default gentext string using the local.l10n.xml parameter, or you can fill in the content for this normally empty override parameter. The content can be a simple string, or it can be something more complex such as a call-template. For fo output, it could be an fo:leader element to provide space of a specific length, or a dot leader. Namexep.index.item.properties — Properties associated with XEP index-items Synopsis<xsl:attribute-set name="xep.index.item.properties" use-attribute-sets="index.page.number.properties"> - <xsl:attribute name="merge-subsequent-page-numbers">true</xsl:attribute> <xsl:attribute name="link-back">true</xsl:attribute> </xsl:attribute-set> Description Properties associated with XEP index-items, which generate page numbers in an index processed by XEP. For more info see the XEP documentation section "Indexes" in http://www.renderx.com/reference.html#Indexes. This attribute-set also adds by default any properties from the index.page.number.properties attribute-set. Nametoc.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. Nametoc.max.depth — How many levels should be created for each TOC? Synopsis<xsl:param name="toc.max.depth">8</xsl:param> Description Specifies the maximal depth of TOC on all levels. Nametoc.indent.width — Amount of indentation for TOC entries Synopsis<xsl:param - name="toc.indent.width">24</xsl:param> <!-- inconsistant point specification? --> Description Specifies, in points, the distance by which each level of the TOC is indented from its parent. This value is expressed in points, without a unit (in other words, it is a bare number). Using a bare number allows the stylesheet to perform calculations that would otherwise have to be performed by the FO processor because not all processors support expressions. Nametoc.line.properties — Properties for lines in ToCs and LoTs Synopsis<xsl:attribute-set name="toc.line.properties"> <xsl:attribute name="text-align-last">justify</xsl:attribute> - <xsl:attribute name="text-align">start</xsl:attribute> <xsl:attribute name="end-indent">><xsl:value-of select="concat($toc.indent.width 'pt')"></xsl:value-of></xsl:attribute> <xsl:attribute name="last-line-end-indent">><xsl:value-of select="concat('-', $toc.indent.width, - 'pt')"></xsl:value-of></xsl:attribute> </xsl:attribute-set> Description Properties which are applied to every line in ToC (or LoT). You can modify them in order to change appearance of all, or some lines. For example, in order to make lines for chapters bold, specify the following in your customization layer: <xsl:attribute-set name="toc.line.properties"> <xsl:attribute name="font-weight"> <xsl:choose> <xsl:when test="self::chapter">bold</xsl:when> <xsl:otherwise>normal</xsl:otherwise> - </xsl:choose> </xsl:attribute> </xsl:attribute-set> Nametoc.margin.properties — Margin properties used on Tables of Contents Synopsis<xsl:attribute-set name="toc.margin.properties"> <xsl:attribute name="space-before.minimum">0.5em</xsl:attribute> <xsl:attribute name="space-before.optimum">1em</xsl:attribute> <xsl:attribute name="space-before.maximum">2em</xsl:attribute> <xsl:attribute name="space-after.minimum">0.5em</xsl:attribute> <xsl:attribute name="space-after.optimum">1em</xsl:attribute> <xsl:attribute name="space-after.maximum">2em</xsl:attribute> </xsl:attribute-set> Description This attribute set is used on Tables of Contents. These attributes are set on the wrapper that surrounds the ToC block, not on each individual lines. Namebridgehead.in.toc — Should bridgehead elements appear in the TOC? Synopsis<xsl:param name="bridgehead.in.toc" select="0"></xsl:p Description If non-zero, bridgeheads appear in the TOC. Note that this option is not fully supported and may be removed in a

PDF BY DBLATEX

future version of the stylesheets. Namesimplesect.in.toc — Should simplesect elements appear in the TOC? Synopsis<xsl:param name="simplesect.in.toc" select="0"></xsl:param> Description If non-zero, simplesects will be included in the TOC. Namegenerate.section.toc.level — Control depth of TOC generation in sections Synopsis<xsl:param name="generate.section.toc.level" select="0"></xsl:param> 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. Processor ExtensionsProcessor ExtensionsProcessor ExtensionsTable of Contentsarbortext.extensionsaxf.extensionsfop.extensionsfop1.extensionspassivetex.extensionstex.math.in.alttex.math.delimsxep.extensions — Enable Arbortext extensions? Synopsis<xsl:param name="arbortext.extensions" select="0"></xsl:param> Description If non-zero, Arbortext11 http://www.arbortext.com/ extensions will be used. This parameter can also affect which graphics file formats are supported Nameaxf.extensions — Enable XSL Formatter extensions? Synopsis<xsl:param name="axf.extensions" select="0"></xsl:param> Description If non-zero, XSL Formatter11 http://www.antennahouse.com/ extensions will be used. XSL Formatter extensions consists of PDF bookmarks, document information and better index processing. This parameter can also affect which graphics file formats are supported Namefop.extensions — Enable extensions for FOP version 0.20.5 and earlier Synopsis<xsl:param name="fop.extensions" select="0"></xsl:param> Description If non-zero, extensions intended for FOP11 http://xml.apache.org/fop/ version 0.20.5 and earlier will be used. At present, this consists of PDF bookmarks. This parameter can also affect which graphics file formats are supported. If you are using a version of FOP beyond version 0.20.5, then use the fop1.extensions parameter instead. Namefop1.extensions — Enable extensions for FOP version 0.90 and later Synopsis<xsl:param name="fop1.extensions" select="0"></xsl:param> Description If non-zero, extensions for FOP11 http://xml.apache.org/fop/ version 0.90 and later will be used. This parameter can also affect which graphics file formats are supported. The original fop.extensions parameter should still be used for FOP version 0.20.5 and earlier. Namepassivetex.extensions — Enable PassiveTeX extensions? Synopsis<xsl:param name="passivetex.extensions" select="0"></xsl:param> Description ImportantThe PassiveTeX XSL-FO processor is no longer supported by DocBook XSL, beginning with version 1.78. PassiveTeX was never a complete implementation of XSL-FO, and development has ceased. Setting this parameter will have no effect on the output. Nametex.math.in.alt — TeX notation used for equations Synopsis<xsl:param name="tex.math.in.alt"></xsl:param> 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. WarningThis feature is useful for print/PDF output only if you use the obsolete and now unsupported PassiveTeX XSL-FO engine. Related Parameters tex.math.delims, passivetex.extensions, tex.math.file Nametex.math.delims — Should equations output for processing by TeX be surrounded by math mode delimiters? Synopsis<xsl:param name="tex.math.delims" select="1"></xsl:param> Description For compatibility with DSSSL based DBTeXMath from Allin Cottrell you should set this parameter to 0. WarningThis feature is useful for print/PDF output only if you use the obsolete and now unsupported PassiveTeX XSL-FO engine. Related Parameters tex.math.in.alt, passivetex.extensions See Also You can also use the <?dbtex delims?> processing instruction to control whether delimiters are output. Namexep.extensions — Enable XEP extensions? Synopsis<xsl:param name="xep.extensions" select="0"></xsl:param> Description If non-zero, XEP11 http://www.renderx.com/ extensions will be used. XEP extensions consists of PDF bookmarks, document information and better index processing. This parameter can also affect which graphics file formats are supported Stylesheet ExtensionsStylesheet ExtensionsStylesheet ExtensionsTable of Contentslinenumbering.everyNthlinenumbering.extensionlinenumbering.separatorlinenumbering — Indicate which lines should be numbered Synopsis<xsl:param name="linenumbering.everyNth">5</xsl:param> Description If line numbering is enabled, everyNth line will be numbered. Note that numbering is one based, not zero based. See also linenumbering.extension, linenumbering.separator, linenumbering.width and use.extensions Namelinenumbering.extension — Enable the line numbering extension Synopsis<xsl:param name="linenumbering.extension" select="1"></xsl:param> Description If non-zero, verbatim environments (address, literallayout, programlisting, screen, synopsis) that specify line numbering will have line numbers. Namelinenumbering.separator — Specify a separator between line numbers and lines Synopsis<xsl:param name="linenumbering.separator"><xsl:text> </xsl:text></xsl:param> Description The separator is inserted between line numbers and lines in the verbatim environment. The default value is a single white space. Note the interaction with linenumbering.width Namelinenumbering.width — Indicates the width of line numbers Synopsis<xsl:param name="linenumbering.width">3</xsl:p Description If line numbering is enabled, line numbers will appear right justified in a field "width" characters wide. Nametablecolumns.ex — Enable the table columns extension function Synopsis<xsl:param name="tablecolumns.extension" select="1"></xsl:param> Description The table columns extension function adjusts the widths of table columns in the HTML result to more accu-

PDF BY DBLATEX

rately reflect the specifications in the CALS table. Nametextinsert.extension — Enables the textinsert extension element Synopsis <xsl:param name="textinsert.extension" select="1"></xsl:param> Description The textinsert extension element inserts the contents of a file into the result tree (as text). NoteTo use the textinsert extension element, you must use either Saxon or Xalan as your XSLT processor (it doesn't work with xsltproc), along with either the DocBook Saxon extensions or DocBook Xalan extensions (for more information about those extensions, see DocBook XSL: TCG, DocBook Saxon Extensions11 http://www.sagehill.net/docbookxsl/InstallingAProcessor.html#SaxonExtensions and DocBook XSL: TCG, DocBook Xalan Extensions22 http://www.sagehill.net/docbookxsl/InstallingAProcessor.html#XalanExtensions), and you must set both the use.extensions and textinsert.extension parameters to 1. As an alternative to using the textinsert element, consider using an Xinclude element with the parse="text" attribute and value specified, as detailed in DocBook XSL: TCG, Using XInclude for text inclusions33 http://www.sagehill.net/docbookxsl/ExternalCode.html#XIncludeCode. See Also You can also use the <?dbhtm-include href?> processing instruction to insert external files — both files containing plain text and files with markup content (including HTML content). More information For how-to documentation on inserting contents of external code files and other text files into output, see DocBook XSL: TCG, External code files44 http://www.sagehill.net/docbookxsl/ExternalCode.html. For guidelines on inserting contents of HTML files into output, see DocBook XSL: TCG, Inserting external HTML code55 http://www.sagehill.net/docbookxsl/InsertExtHtml.html. Nametextdata.default.encoding — Default encoding of external text files which are included using textdata element Synopsis<xsl:param name="textdata.default.encoding"></xsl:param> Description Specifies the encoding of any external text files included using textdata element. This value is used only when you do not specify encoding by the appropriate attribute directly on textdata. An empty string is interpreted as the system default encoding. Nameuse.extensions — Enable extensions Synopsis<xsl:param name="use.extensions" select="0"></xsl:param> 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. Automatic labellingAutomatic labellingAutomatic labellingTable of Contentsappendix.autolabelchapter.autolabelp — Specifies the labeling format for Appendix titles Synopsis<xsl:param name="appendix.autolabel">A</xsl:param> Description If non-zero, then appendices will be numbered using the parameter value as the number format if the value matches one of the following: 1 or arabicArabic numeration (1, 2, 3 ...). A or upperalphaUppercase letter numeration (A, B, C ...). a or loweralphaLowercase letter numeration (a, b, c ...). I or upperromanUppercase roman numeration (I, II, III ...). i or lowerromanLowercase roman letter numeration (i, ii, iii ...). Any nonzero value other than the above will generate the default number format (upperalpha). Namechapter.autolabel — Specifies the labeling format for Chapter titles Synopsis<xsl:param name="chapter.autolabel" -select="1"></xsl:param> Description If non-zero, then chapters will be numbered using the parameter value as the number format if the value matches one of the following: 1 or arabicArabic numeration (1, 2, 3 ...). A or upperalphaUppercase letter numeration (A, B, C ...). a or loweralphaLowercase letter numeration (a, b, c ...). I or upperromanUppercase roman numeration (I, II, III ...). i or lowerromanLowercase roman letter numeration (i, ii, iii ...). Any nonzero value other than the above will generate the default number format (arabic). Namepart.autolabel — Specifies the labeling format for Part titles Synopsis<xsl:param -name="part.autolabel">I</xsl:param> Description If non-zero, then parts will be numbered using the parameter value as the number format if the value matches one of the following: 1 or arabicArabic numeration (1, 2, 3 ...). A or upperalphaUppercase letter numeration (A, B, C ...). a or loweralphaLowercase letter numeration (a, b, c ...). I or upperromanUppercase roman numeration (I, II, III ...). i or lowerromanLowercase roman letter numeration (i, ii, iii ...). Any nonzero value other than the above will generate the default number format (upperroman). Namereference.autolabel — Specifies the labeling format for Reference titles Synopsis <xsl:param name="reference.autolabel">I</xsl:param> Description If non-zero, references will be numbered using the parameter value as the number format if the value matches one of the following: 1 or arabicArabic numeration (1, 2, 3 ...). A or upperalphaUppercase letter numeration (A, B, C ...). a or loweralphaLowercase letter numeration (a, b, c ...). I or upperromanUppercase roman numeration (I, II, III ...). i or lowerromanLowercase roman letter numeration (i, ii, iii ...). Any non-zero value other than the above will generate the default number format (upperroman). Namepreface.autolabel — Specifices the labeling format for Preface titles Synopsis<xsl:param name="preface.autolabel" select="0"></xsl:param> Description If non-zero then prefaces will be numbered using the parameter value as the number format if the value matches one of the following: 1 or arabicArabic numeration (1, 2, 3 ...). A or upperalphaUppercase letter numeration (A, B, C ...). a or loweralphaLowercase letter numeration (a, b, c ...). I or upperromanUppercase roman numeration (I, II, III ...). i or lowerromanLowercase roman letter numeration (i, ii, iii ...). Any nonzero value other than the above will generate the default number format (arabic). Namesection.autolabel — Are sections enumerated? Synopsis<xsl:param name="section.autolabel" select="0"></xsl:param> Description If true (non-zero), unlabeled sections will be enumerated. Namesection.autolabel.max.depth — The deepest level of sections that are numbered. Synopsis<xsl:param name="section.autolabel.max.depth">8</xsl:param> Description When section numbering is turned on by the section.autolabel parameter, then this parameter controls the depth of section nesting that is numbered. Sections nested to a level deeper than this value will not be numbered. Namesection.label.includes.component.label — Do section labels include the component label? Synopsis<xsl:param name="section.label.includes.component.label" select="0"></xsl:param> Description If non-zero, section labels are prefixed with the label of the component that contains them. Namelabel.from.part — Renumber components in each part? Synopsis<xsl:param name="label.from.part" select="0"></xsl:param> Description If label.from.part is non-zero, then numbering of components — preface, chapter, appendix, and reference (when reference occurs at the component level) — is re-started within each part. If label.from.part is zero (the default), numbering of com-

PDF BY DBLATEX

ponents is not re-started within each part; instead, components are numbered sequentially throughout each book, regardless of whether or not they occur within part instances. Namecomponent.label.includes.part.label — Do component labels include the part label? Synopsis<xsl:param name="component.label.includes.part.label" select="0"></xsl:param> Description If non-zero, number labels for chapter, appendix, and other component elements are prefixed with the label of the part element that contains them. So you might see Chapter II.3 instead of Chapter 3. Also, the labels for formal elements such as table and figure will include the part label. If there is no part element container, then no prefix is generated. This feature is most useful when the label.from.part parameter is turned on. In that case, there would be more than one chapter "1", and the extra part label prefix will identify each chapter unambiguously. XSLT ProcessingXSLT ProcessingXSLT ProcessingTable of ContentsrootidNamerootid — Specify the root element to format Synopsis<xsl:param name="rootid"></xsl:param> Description If rootid is not empty, 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. Meta/*InfoMeta/*InfoMeta/*InfoTable of Contentsmake.single.year.rangesmake.year.rangesauthor.othername.in.middleNamemake.single.year.ranges — Print single-year ranges (e.g., 1998-1999) Synopsis<xsl:param name="make.single.year.ranges" select="0"></xsl:param> 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). Namemake.year.ranges — Collate copyright years into ranges? Synopsis<xsl:param name="make.year.ranges" select="0"></xsl:param> Description If non-zero, multiple copyright year elements will be collated into ranges. This works only if each year number is put into a separate year element. The copyright element permits multiple year elements. If a year element contains a dash or a comma, then that year element will not be merged into any range. Nameauthor.othername.in.middle — Is othername in author a middle name? Synopsis<xsl:param name="author.othername.in.middle" select="1"></xsl:param> Description If non-zero, the othername of an author appears between the firstname and surname. Otherwise, othername is suppressed. Reference PagesReference PagesReference PagesTable of Contentsfuncsynopsis.decorationfuncsynopsis.stylefunction.parensrefentry.generate.namerefentry.genera — Decorate elements of a funcsynopsis? Synopsis<xsl:param name="funcsynopsis.decoration" select="1"></xsl:param> Description If non-zero, elements of the funcsynopsis will be decorated (e.g. rendered as bold or italic text). The decoration is controlled by templates that can be redefined in a customization layer. Namefuncsynopsis.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. Namefunction.parens — Generate parens after a function? Synopsis<xsl:param name="function.parens" select="0"></xsl:param> Description If non-zero, the formatting of a function element will include generated parentheses. Namerefentry.generate.name — Output NAME header before refnames? Synopsis<xsl:param name="refentry.generate.name" select="1"></xsl:param> Description If non-zero, a "NAME" section title is output before the list of refnames. 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 non-zero unless you want get quite strange output. Namerefentry.generate.title — Output title before refnames? Synopsis<xsl:param name="refentry.generate.title" select="0"></xsl:param> Description If non-zero, the reference page title or first name is output before the list of refnames. This parameter and refentry.generate.name are mutually exclusive. This means that if you change this parameter to non-zero, you should set refentry.generate.name to zero unless you want get quite strange output. Namerefentry.pagebreak — Start each refentry on a new page Synopsis<xsl:param name="refentry.pagebreak" select="1"></xsl:param> Description If non-zero (the default), each refentry element will start on a new page. If zero, a page break will not be generated between refentry elements. The exception is when the refentry elements are children of a part element, in which case the page breaks are always retained. That is because a part element does not generate a page-sequence for its children, so each refentry must start its own page-sequence. Namerefentry.title.properties — Title properties for a refentry title Synopsis<xsl:attribute-set name="refentry.title.properties"> <xsl:attribute name="font-family"> <xsl:value-of select="$title.fontset"></xsl:value-of> </xsl:attribute> <xsl:attribute name="font-size">18pt</xsl:attribute> <xsl:attribute name="font-weight">bold</xsl:attribute> <xsl:attribute name="space-after">1em</xsl:attribute> <xsl:attribute name="hyphenate">false</xsl:attribute> <xsl:attribute name="keep-with-next.within-column">always</xsl:attribute> <xsl:attribute name="space-before.minimum">0.8em</xsl:attribute> <xsl:attribute name="space-before.optimum">1.0em</xsl:attribute> <xsl:attribute name="space-before.maximum">1.2em</xsl:attribu <xsl:attribute name="space-after.optimum">0.5em</xsl:attribute> <xsl:attribute name="space-after.minimum">0.4em</xsl:attribute> <xsl:attribute name="space-after.maximum">0.6em</xsl:attribute> <xsl:attribute name="start-indent"><xsl:value-of select="$title.ma of></xsl:attribute> </xsl:attribute-set> Description Formatting properties applied to the title generated for the refnamediv part of output for refentry when the value of the refentry.generate.title parameter is non-zero. The font size is supplied by the appropriate section.levelX.title.properties attribute-set, computed from the location of the refentry in the section hierarchy. NoteThis parameter has no effect on the the title generated for the refnamediv part of output for refentry when the value of the refentry.generate.name parameter is non-zero. By default, that title is formatted with the same properties as the titles for all other first-level children of refentry. Namerefentry.xref.manvolnum — Output manvolnum as part of refentry cross-reference? Synopsis<xsl:param name="refentry.xref.manvolnum" select="1"></xsl:param> Description if non-zero, the manvolnum is used when cross-referencing refentrys, either with xref or citerefentry. Namerefclass.suppress — Suppress display

PDF BY DBLATEX

of refclass contents? Synopsis<xsl:param name="refclass.suppress" select="0"></xsl:param> Description If the value of refclass.suppress is non-zero, then display of refclass contents is suppressed in output. TablesTablesTablesTable of Contentsdefault.table.widthnominal.table.widthdefault.table.framedefault.table.rulestable.cell.paddingtable.frame.border.thicknesstable.frame.borc
— The default width of tables Synopsis<xsl:param name="default.table.width"></xsl:param> Description If non-zero, this value will be used for the width attribute on tables that do not specify an alternate width (with the <?dbhtml table-width?> or <?dbfo table-width?> processing instruction). Namenominal.table.width — The (absolute) nominal width of tables Synopsis<xsl:param name="nominal.table.width">6in</xsl:param> 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 percentage). Namedefault.table.frame — The default framing of tables Synopsis<xsl:param name="default.table.frame">all</xsl:param> Description This value will be used when there is no frame attribute on the table. Namedefault.table.rules — The default column and row rules for tables using HTML markup Synopsis<xsl:param name="default.table.rules">none</xsl:param> Description Tables using HTML markup elements can use an attribute named rules on the table or informaltable element to specify whether column and row border rules should be displayed. This parameter lets you specify a global default style for all HTML tables that don't otherwise have that attribute. These are the supported values: allRules will appear between all rows and columns. rowsRules will appear between rows only. colsRules will appear between columns only. groupsRules will appear between row groups (thead, tfoot, tbody). No support for rules between column groups yet. noneNo rules. This is the default value. The border after the last row and the border after the last column are not affected by this setting. Those borders are controlled by the frame attribute on the table element. Nametable.cell.padding — Specifies the padding of table cells Synopsis<xsl:attribute-set name="table.cell.padding"> <xsl:attribute name="padding-start">2pt</xsl:attribute> <xsl:attribute name="padding-end">2pt</xsl:attribute> <xsl:attribute name="padding-top">2pt</xsl:attribut <xsl:attribute name="padding-bottom">2pt</xsl:attribute> </xsl:attribute-set> Description Specifies the padding of table cells. Nametable.frame.border.thickness — Specifies the thickness of the frame border Synopsis<xsl:param name="table.frame.border.thicknes Description Specifies the thickness of the border on the table's frame. Nametable.frame.border.style — Specifies the border style of table frames Synopsis<xsl:param name="table.frame.border.style">solid</xsl:param> Description Specifies the border style of table frames. Nametable.frame.border.color — Specifies the border color of table frames Synopsis<xsl:param name="table.frame.border.color">black</xsl:param> Description Specifies the border color of table frames. Nametable.cell.border.thick — Specifies the thickness of table cell borders Synopsis<xsl:param name="table.cell.border.thickness">0.5pt</xsl:param> Description If non-zero, specifies the thickness of borders on table cells. The units are points. See CSS11 http://www.w3.org/TR/CSS21/box width-properties NoteTo control properties of cell borders in HTML output, you must also turn on the table.borders.with.css parameter. Nametable.cell.border.style — Specifies the border style of table cells Synopsis<xsl:param name="table.cell.border.style">solid Description Specifies the border style of table cells. NoteTo control properties of cell borders in HTML output, you must also turn on the table.borders.with.css parameter. Nametable.cell.border.color — Specifies the border color of table cells Synopsis<xsl:param name="table.cell.border.color">black</xsl:param> Description Set the color of table cell borders. If non-zero, the value is used for the border coloration. See CSS11 http://www.w3.org/TR/CSS21/syndata.html#value-def-color. A color is either a keyword or a numerical RGB specification. Keywords are aqua, black, blue, fuchsia, gray, green, lime, maroon, navy, olive, orange, purple, red, silver, teal, white, and yellow. NoteTo control properties of cell borders in HTML output, you must also turn on the table.borders.with.css parameter. Nametable.table.properties — Properties associated with a table Synopsis<xsl:attribute-set name="table.table.properties">> <xsl:attribute name="border-before-width.conditionality">retain</xsl:attribute> <xsl:attribute name="border-collapse">collapse</xsl:attribute> </xsl:attribute-set> Description The styling for tables. This parameter should really have been called table.properties, but that parameter name was inadvertently established for the block-level properties of the table as a whole. See also table.properties. Nametable.caption.properties — Properties associated with a table caption Synopsis<xsl:attribute-set name="table.caption.properties">> <xsl:attribute name="keep-together.within-column">always</xsl:attribute </xsl:attribute-set> Description The styling for table caption element (not the table title). See also table.properties. LinkingLinkingLinkingTable of Contentscurrent.docidactivate.external.olinkscollect.xref.targetsinsert.olink.page.numberinsert.olink.pdf.fragolink.b
— targetdoc identifier for the document being processed Synopsis<xsl:param name="current.docid"></xsl:param> 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.doctitle to enable that feature. Nameactivate.external.olinks — Make external olinks into active links Synopsis<xsl:param name="activate.external.olinks" select="1"></xsl:param> Description If activate.external.olinks is nonzero (the default), then any olinks that reference another document become active links that can be clicked on to follow the link. If the parameter is set to zero, then external olinks will have the appropriate link text generated, but the link is not made active. Olinks to destinations in the current document remain active. To make an external olink active for HTML outputs, the link text is wrapped in an a element with an href attribute. To make an external olink active for FO outputs, the link text is wrapped in an fo:basic-link element with

an external-destination attribute. This parameter is useful when you need external olinks to resolve but not be clickable. For example, if documents in a collection are available independently of each other, then having active links between them could lead to unresolved links when a given target document is missing. The epub stylesheets set this parameter to zero by default because there is no standard linking mechanism between Epub documents. If external links are made inactive, you should consider setting the stylesheet parameter olink.doctitle to yes. That will append the external document's title to the link text, making it easier for the user to locate the other document. An olink is considered external when the current.docid stylesheet parameter is set to some value, and the olink's targetdoc attribute has a different value. If the two values match, then the link is considered internal. If the current.docid parameter is blank, or the olink element does not have a targetdoc attribute, then the link is considered to be internal and will become an active link. See also olink.doctitle, prefer.internal.olink. Namecollect.xref.targets — Controls whether cross reference data is collected Synopsis<xsl:param name="collect.xref.targets">no</xsl:param> 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. Nameinsert.olink.page.number — Turns page numbers in olinks on and off Synopsis<xsl:param name="insert.olink.page.number">no</xsl:param> Description The value of this parameter determines if cross references made between documents with olink will include page number citations. In most cases this is only applicable to references in printed output. The parameter has three possible values. noNo page number references will be generated for olinks. yesPage number references will be generated for all olink references. The style of page reference may be changed if an xrefstyle attribute is used. maybePage number references will not be generated for an olink element unless it has an xrefstyle attribute whose value specifies a page reference. Olinks that point to targets within the same document are treated as xrefs, and controlled by the insert.xref.page.number parameter. Page number references for olinks to external documents can only be inserted if the information exists in the olink database. This means each olink target element (div or obj) must have a page attribute whose value is its page number in the target document. The XSL stylesheets are not able to extract that information during processing because pages have not yet been created in XSLT transformation. Only the XSL-FO processor knows what page each element is placed on. Therefore some postprocessing must take place to populate page numbers in the olink database. Nameinsert.olink.pdf.frag — Add fragment identifiers for links into PDF files Synopsis<xsl:param name="insert.olink.pdf.frag" select="0"></xsl:param> Description The value of this parameter determines whether the cross reference URIs to PDF documents made with olink will include fragment identifiers. When forming a URI to link to a PDF document, a fragment identifier (typically a '#' followed by an id value) appended to the PDF filename can be used by the PDF viewer to open the PDF file to a location within the document instead of the first page. However, not all PDF files have id values embedded in them, and not all PDF viewers can handle fragment identifiers. If insert.olink.pdf.frag is set to a non-zero value, then any olink targeting a PDF file will have the fragment identifier appended to the URI. The URI is formed by concatenating the value of the olink.base.uri parameter, the value of the baseuri attribute from the document element in the olink database with the matching targetdoc value, and the value of the href attribute for the targeted element in the olink database. The href attribute contains the fragment identifier. If insert.olink.pdf.frag is set to zero (the default value), then the href attribute from the olink database is not appended to PDF olinks, so the fragment identifier is left off. A PDF olink is any olink for which the baseuri attribute from the matching document element in the olink database ends with '.pdf'. Any other olinks will still have the fragment identifier added. Nameolink.base.uri — Base URI used in olink hrefs Synopsis<xsl:param name="olink.base.uri"></xsl:param> 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. Nameolink.debug — Turn on debugging messages for olinks Synopsis<xsl:param name="olink.debug" select="0"></xsl:param> Description If non-zero, then each olink will generate several messages about how it is being resolved during processing. This is useful when an olink does not resolve properly and the standard error messages are not sufficient to find the problem. You may need to read through the olink XSL templates to understand the context for some of the debug messages. Nameolink.doctitle — show the document title for external olinks? Synopsis<xsl:param name="olink.doctitle">no</xsl:param> Description When olinks between documents are resolved, 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. •This olink.doctitle parameter should be set to either yes or maybe to enable this feature. •And you should also set the current.docid parameter to the document id for the document currently being processed for output. Then if an olink's targetdoc id differs from the current.docid value, the stylesheet knows that it is a reference to another document and can append the target document's title to the generated olink text. The text for the target document's title is copied from the olink database from the ttl element of the top-level div for that document. If that ttl element is missing or empty, no title is output. The supported values

PDF BY DBLATEX

for olink.doctitle are: yesAlways insert the title to the target document if it is not the current document. noNever insert the title to the target document, even if requested in an xrefstyle attribute. maybeOnly insert the title to the target document, if requested in an xrefstyle attribute. An xrefstyle attribute may override the global setting for individual olinks. The following values are supported in an xrefstyle attribute using the select: syntax: docnameInsert the target document name for this olink using the docname gentext template, but only if the value of olink.doctitle is not no. docnamelongInsert the target document name for this olink using the docnamelong gentext template, but only if the value of olink.doctitle is not no. nodocnameOmit the target document name even if the value of olink.doctitle is yes. Another way of inserting the target document name for a single olink is to employ an xrefstyle attribute using the template: syntax. The %o placeholder (the letter o, not zero) in such a template will be filled in with the target document's title when it is processed. This will occur regardless of the value of olink.doctitle. Note that prior to version 1.66 of the XSL stylesheets, the allowed values for this parameter were 0 and 1. Those values are still supported and mapped to 'no' and 'yes', respectively. Nameolink.lang.fallback.sequence — look up translated documents if olink not found? Synopsis<xsl:param name="olink.lang.fallback.sequence"></xsl:param> Description This parameter defines a list of lang values to search among to resolve olinks. Normally an olink tries to resolve to a document in the same language as the olink itself. The language of an olink is determined by its nearest ancestor element with a lang attribute, otherwise the value of the l10n.gentext.default.lang parameter. An olink database can contain target data for the same document in multiple languages. Each set of data has the same value for the targetdoc attribute in the document element in the database, but with a different lang attribute value. When an olink is being resolved, the target is first sought in the document with the same language as the olink. If no match is found there, then this parameter is consulted for additional languages to try. The olink.lang.fallback.sequence must be a whitespace separated list of lang values to try. The first one with a match in the olink database is used. The default value is empty. For example, a document might be written in German and contain an olink with targetdoc="adminguide". When the document is processed, the processor first looks for a target dataset in the olink database starting with: <document targetdoc="adminguide" lang="de">. If there is no such element, then the olink.lang.fallback.sequence parameter is consulted. If its value is, for example, "fr en", then the processor next looks for targetdoc="adminguide" lang="fr", and then for targetdoc="adminguide" lang="en". If there is still no match, it looks for targetdoc="adminguide" with no lang attribute. This parameter is useful when a set of documents is only partially translated, or is in the process of being translated. If a target of an olink has not yet been translated, then this parameter permits the processor to look for the document in other languages. This assumes the reader would rather have a link to a document in a different language than to have a broken link. Nameolink.properties — Properties associated with the cross-reference text of an olink. Synopsis<xsl:attribute-set name="olink.properties"> <xsl:attribute name="show-destination">replace</xsl:attribute> </xsl:attribute-set> Description This attribute set is applied to the fo:basic-link element of an olink. It is not applied to the optional page number or optional title of the external document. Nameprefer.internal.olink — Prefer a local olink reference to an external reference Synopsis<xsl:param name="prefer.internal.olink" select="0"></xsl:param> Description If you are re-using XML content modules in multiple documents, you may want to redirect some of your olinks. This parameter permits you to redirect an olink to the current document. For example: you are writing documentation for a product, which includes 3 manuals: a little installation booklet (booklet.xml), a user guide (user.xml), and a reference manual (reference.xml). All 3 documents begin with the same introduction section (intro.xml) that contains a reference to the customization section (custom.xml) which is included in both user.xml and reference.xml documents. How do you write the link to custom.xml in intro.xml so that it is interpreted correctly in all 3 documents? •If you use xref, it will fail in user.xml. •If you use olink (pointing to reference.xml), the reference in user.xml will point to the customization section of the reference manual, while it is actually available in user.xml. If you set the prefer.internal.olink parameter to a non-zero value, then the processor will first look in the olink database for the olink's targetptr attribute value in document matching the current.docid parameter value. If it isn't found there, then it tries the document in the database with the targetdoc value that matches the olink's targetdoc attribute. This feature permits an olink reference to resolve to the current document if there is an element with an id matching the olink's targetptr value. The current document's olink data must be included in the target database for this to work. CautionThere is a potential for incorrect links if the same id attribute value is used for different content in different documents. Some of your olinks may be redirected to the current document when they shouldn't be. It is not possible to control individual olink instances. Nametarget.database.document — Name of master database file for resolving olinks Synopsis <xsl:param name="target.database.document">olinkdb.xml</xsl:param> 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. Nametargets.filename — Name of cross reference targets data file Synopsis<xsl:param name="targets.filename">target.db</xsl:param> 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. Nameuse.local.olink.style — Process olinks using xref style of current document Synopsis<xsl:param -

name="use.local.olink.style" select="0"></xsl:param> 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. Cross ReferencesCross ReferencesCross ReferencesTable of Contentsinsert.xref.page.numberxref.propertiesxref.label-title.separatorxref.label-page.separatorxref.title-page.separatorinsert.link.page.numberNameinsert.xref.page.number — Turns page numbers in xrefs on and off Synopsis<xsl:param name="insert.xref.page.number">no</xsl:param> Description The value of this parameter determines if cross references (xrefs) in printed output will include page number citations. It has three possible values. noNo page number references will be generated. yesPage number references will be generated for all xref elements. The style of page reference may be changed if an xrefstyle attribute is used. maybePage number references will not be generated for an xref element unless it has an xrefstyle attribute whose value specifies a page reference. Namexref.properties — Properties associated with cross-reference text Synopsis<xsl:attribute-set name="xref.properties"> </xsl:attribute-set> Description This attribute set is used to set properties on cross reference text. Namexref.label-title.separator — Punctuation or space separating label from title in xref Synopsis<xsl:param name="xref.label-title.separator">: </xsl:param> Description This parameter allows you to control the punctuation of certain types of generated cross reference text. When cross reference text is generated for an xref or olink element using an xrefstyle attribute that makes use of the select: feature, and the selected components include both label and title, then the value of this parameter is inserted between label and title in the output. Namexref.label-page.separator — Punctuation or space separating label from page number in xref Synopsis<xsl:param name="xref.label-page.separator"><xsl:text> </xsl:text></xsl:param> Description This parameter allows you to control the punctuation of certain types of generated cross reference text. When cross reference text is generated for an xref or olink element using an xrefstyle attribute that makes use of the select: feature, and the selected components include both label and page but no title, then the value of this parameter is inserted between label and page number in the output. If a title is included, then other separators are used. Namexref.title-page.separator — Punctuation or space separating title from page number in xref Synopsis<xsl:param name="xref.title-page.separator"><xsl:text> </xsl:text></xsl:param> Description This parameter allows you to control the punctuation of certain types of generated cross reference text. When cross reference text is generated for an xref or olink element using an xrefstyle attribute that makes use of the select: feature, and the selected components include both title and page number, then the value of this parameter is inserted between title and page number in the output. Nameinsert.link.page.number — Turns page numbers in link elements on and off Synopsis<xsl:param name="insert.link.page.number">no</xsl:param> Description The value of this parameter determines if cross references using the link element in printed output will include standard page number citations. It has three possible values. noNo page number references will be generated. yesPage number references will be generated for all link elements. The style of page reference may be changed if an xrefstyle attribute is used. maybePage number references will not be generated for a link element unless it has an xrefstyle attribute whose value specifies a page reference. Although the xrefstyle attribute can be used to turn the page reference on or off, it cannot be used to control the formatting of the page number as it can in xref. In link it will always format with the style established by the gentext template with name="page.citation" in the l:context name="xref". ListsListsListsTable of Contentscompact.list.item.spacingitemizedlist.propertiesitemizedlist.label.propertiesitemizedlist.label.widthlist.block.propertieslist.block.sp — What space do you want between list items (when spacing="compact")? Synopsis<xsl:attribute-set name="compact.list.item.spacing" <xsl:attribute name="space-before.optimum">0em</xsl:attribute> <xsl:attribute name="space-before.minimum">0em</xsl:attribute> <xsl:attribute name="space-before.maximum">0.2em</xsl:attribute> </xsl:attribute-set> Description Specify what spacing you want between each list item when spacing is “compact”. Nameitemizedlist.properties — Properties that apply to each list-block generated by itemizedlist. Synopsis<xsl:attribute-set name="itemizedlist.properties" use-attribute-sets="list.block.properties">
</xsl:attribute-set> Description Properties that apply to each fo:list-block generated by itemizedlist. Nameitemizedlist.label.properties — Properties that apply to each label inside itemized list. Synopsis<xsl:attribute-set name="itemizedlist.label.properties">
</xsl:attribute-set> Description Properties that apply to each label inside itemized list. E.g.: <xsl:attribute-set name="itemizedlist.label.p <xsl:attribute name="text-align">right</xsl:attribute> </xsl:attribute-set> Nameitemizedlist.label.width — The default width of the label (bullet) in an itemized list. Synopsis <xsl:param name="itemizedlist.label.width">1.0em</xsl:param> Description Specifies the default width of the label (usually a bullet or other symbol) in an itemized list. You can override the default value on any particular list with the “dbfo” processing instruction using the “label-width” pseudoattribute. Namelist.block.properties — Properties that apply to each list-block generated by list. Synopsis<xsl:attribute-set name="list.block.properties"> <xsl:attribute name="provisional-label-separation">0.2em</xsl:attribute> <xsl:attribute name="provisional-distance-between-starts">1.5em</xsl:att
</xsl:attribute-set> Description Properties that apply to each fo:list-block generated by itemizedlist/orderedlist. Namelist.block.spacing — What spacing do you want before and after lists? Synopsis<xsl:attribute-set name="list.block.spacing">
<xsl:attribute name="space-before.optimum">1em</xsl:attribute> <xsl:attribute name="space-before.minimum">0.8em</xsl:attribute> <xsl:attribute name="space-before.maximum">1.2em</xsl:attribute> <xsl:attribute name="space-after.optimum">1em</xsl:attribute> <xsl:attribute name="space-after.minimum">0.8em</xsl:attribute> <xsl:attribute name="space-after.maximum">1.2em</xsl:attribute>

PDF BY DBLATEX

</xsl:attribute-set> Description Specify the spacing required before and after a list. It is necessary to specify the space after a list block because lists can come inside of paras. Namelist.item.spacing — What space do you want between list items? Synopsis<xsl:attribute-set name="list.item.spacing"> <xsl:attribute name="space-before.optimum">1em</xsl:attribute> <xsl:attribute name="space-before.minimum">0.8em</xsl:attribute> <xsl:attribute name="space-before.maximum">1.2em</xsl:attribute> </xsl:attribute-set> Description Specify what spacing you want between each list item. Nameorderedlist.properties — Properties that apply to each list-block generated by orderedlist. Synopsis<xsl:attribute-set name="orderedlist.properties" use-attribute-sets="list.block.properties"> <xsl:attribute name="provisional-distance-between-starts">2em</xsl:attribute> </xsl:attribute-set> Description Properties that apply to each fo:list-block generated by orderedlist. Nameorderedlist.label.properties — Properties that apply to each label inside ordered list. Synopsis<xsl:attribute-set name="orderedlist.label.properties"> </xsl:attribute-set> Description Properties that apply to each label inside ordered list. E.g.: <xsl:attribute-set name="orderedlist.label.properties"> <xsl:attribute name="text-align">right</xsl:attribute> </xsl:attribute-set> Nameorderedlist.label.width — The default width of the label (number) in an ordered list. Synopsis<xsl:param name="orderedlist.label.width">1.2em</xsl:param> Description Specifies the default width of the label (usually a number or sequence of numbers) in an ordered list. You can override the default value on any particular list with the “dbfo” processing instruction using the “label-width” pseudoattribute. Namevariablelist.max.termlength — Specifies the longest term in variablelists Synopsis<xsl:param name="variablelist.max.termlength">24</xsl:p Description In variablelists, the listitem is indented to leave room for the term elements. That indent may be computed if it is not specified with a termlength attribute on the variablelist element. The computation counts characters in the term elements in the list to find the longest term. However, some terms are very long and would produce extreme indents. This parameter lets you set a maximum character count. Any terms longer than the maximum would line wrap. The default value is 24. The character counts are converted to physical widths by multiplying by 0.50em. There will be some variability in how many actual characters fit in the space since some characters are wider than others. Namevariablelist.term.separator — Text to separate terms within a multi-term varlistentry Synopsis<xsl:param name="variablelist.term.separator">, </xsl:param> Description When a varlistentry contains multiple term elements, the string specified in the value of the variablelist.term.separator parameter is placed after each term except the last. NoteTo generate a line break between multiple terms in a varlistentry, set a non-zero value for the variablelist.term.break.after parameter. If you do so, you may also want to set the value of the variablelist.term.separator parameter to an empty string (to suppress rendering of the default comma and space after each term). Namevariablelist.term.properties — To add properties to the term elements in a variablelist. Synopsis<xsl:attribute-set name="variablelist.term.properties">
</xsl:attribute-set> Description These properties are added to the block containing a term in a variablelist. Use this attribute-set to set font properties or alignment, for example. Namevariablelist.term.break.after — Generate line break after each term within a multi-term varlistentry? Synopsis<xsl:param name="variablelist.term.break.after">0</xsl:param> Description Set a non-zero value for the variablelist.term.break.after parameter to generate a line break between terms in a multi-term varlistentry. NoteIf you set a non-zero value for variablelist.term.break.after, you may also want to set the value of the variablelist.term.separator parameter to an empty string (to suppress rendering of the default comma and space after each term). QAndASetQAndASetQAndASet-Table of Contentsqandadiv.autolabelqanda.inherit.numerationqanda.defaultlabelqanda.in.tocqanda.nested.in.tocNameqandadiv.autolabe — Are divisions in QAndASets enumerated? Synopsis<xsl:param name="qandadiv.autolabel" select="1"></xsl:param> Description If non-zero, unlabeled qandadivs will be enumerated. Nameqanda.inherit.numeration — Does enumeration of QandASet components inherit the numeration of parent elements? Synopsis<xsl:param name="qanda.inherit.numeration" select="1"></xsl:p Description If non-zero, numbered qandadiv elements and question and answer inherit the enumeration of the ancestors of the qandaset. Nameqanda.defaultlabel — Sets the default for defaultlabel on QandASet. Synopsis<xsl:param name="qanda.defaultlabel">nu Description If no defaultlabel attribute is specified on a qandaset, this value is used. It is generally one of the legal values for the defaultlabel attribute (none, number or qanda), or one of the additional stylesheet-specific values (qnumber or qnumberanda). The default value is ’number’. The values are rendered as follows: qandaquestions are labeled "Q:" and answers are labeled "A:". numberThe questions are enumerated and the answers are not labeled. qnumberThe questions are labeled "Q:" followed by a number, and answers are not labeled. When sections are numbered, adding a label to the number distinguishes the question numbers from the section numbers. This value is not allowed in the defaultlabel attribute of a qandaset element. qnumberandaThe questions are labeled "Q:" followed by a number, and the answers are labeled "A:". When sections are numbered, adding a label to the number distinguishes the question numbers from the section numbers. This value is not allowed in the defaultlabel attribute of a qandaset element. noneNo distinguishing label precedes Questions or Answers. Nameqanda.in.toc — Should qandaentry questions appear in the document table of contents? Synopsis<xsl:param name="qanda.in.toc" select="0"></xsl:param> Description If true (non-zero), then the generated table of contents for a document will include qandaset titles, qandadiv titles, and question elements. The default value (zero) excludes them from the TOC. This parameter does not affect any tables of contents that may be generated inside a qandaset or qandadiv. Nameqanda.nested.in.toc — Should nested answer/qandaentry instances appear in TOC? Synopsis<xsl:param name="qanda.nested.in.toc" select="0"></xsl:param> Description If non-zero, instances of qandaentry that are children of answer elements are shown in the TOC. BibliographyBibliographyBibliographyTable of Contentsbibliography.stylebiblioentry.item.separatorbibliography.collectionbibliography.numberedbiblioentry.propertiesNamebibliography.style — Style used for formatting of biblioentries. Synopsis<xsl:param name="bibliography.style">normal</xsl:param> Description Currently only normal and iso690 styles are supported. In order to use ISO690 style to the full extent you might need to

use additional markup described on the following WiKi page11 http://wiki.docbook.org/topic/ISO690Bibliography. Namebiblioentry.item.separator — Text to separate bibliography entries Synopsis<xsl:param name="biblioentry.item.separator">. - </xsl:param> Description Text to separate bibliography entries Namebibliography.collection — Name of the bibliography collection file Synopsis<xsl:param name="bibliography.collection">http://docbook.sourceforge.net/release/bibliography/bibliography.xml</x 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. A relative path in the parameter is interpreted in one of two ways: a.If your document contains no links to empty bibliographic elements, then the path is relative to the file containing the first bibliomixed element in the document. b.If your document does contain links to empty bibliographic elements, then the path is relative to the file containing the first such link element in the document. Once the collection file is opened by the first instance described above, it stays open for the current document and the relative path is not reinterpreted again. The stylesheets will format the bibliography in your document as if all of the entries referenced appeared there literally. Namebibliography.numbered — Should bibliography entries be numbered? Synopsis<xsl:param - name="bibliography.numbered" select="0"></xsl:param> Description If non-zero bibliography entries will be numbered Namebiblioentry.properties — To set the style for biblioentry. Synopsis<xsl:attribute-set name="biblioentry.properties" use-attribute-sets="normal.para.spacing"> <xsl:attribute name="start-indent">0.5in</xsl:attribute> <xsl:attribute name="text-indent">- 0.5in</xsl:attribute> </xsl:attribute-set> Description How do you want biblioentry styled? Set the font-size, weight, space-above and space-below, indents, etc. to the style required GlossaryGlossaryGlossaryTable of Contentsglossterm.auto.linkfirstterm.only.linkglos — Generate links from glossterm to glossentry automatically? Synopsis<xsl:param name="glossterm.auto.link" select="0"></xsl:param Description If non-zero, links from inline glossterms to the corresponding glossentry elements in a glossary or glosslist will be automatically generated. This is useful when your glossterms are consistent and you don't want to add links manually. The automatic link generation feature is not used on glossterm elements that have a linkend attribute. Namefirstterm.only.link — Does automatic glossterm linking only apply to firstterms? Synopsis<xsl:param name="firstterm.only.link" select="0"></xsl:param> Description If non-zero, only firstterms will be automatically linked to the glossary. If glossary linking is not enabled, this parameter has no effect. Nameglossary.collection — Name of the glossary collection file Synopsis<xsl:param name="glossary.collection"></xsl:par 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,

PDF BY DBLATEX

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="partsofspeach">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 automic 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. A relative path in the parameter is interpreted in one of two ways: a.If the parameter glossterm.auto.link is set to zero, then the path is relative to the file containing the empty glossary element in the document. b.If the parameter glossterm.auto.link is set to non-zero, then the path is relative to the file containing the first inline glossterm or firstterm in the document to be linked. Once the collection file is opened by the first instance described above, it stays open for the current document and the relative path is not reinterpreted again. The stylesheets will format the glossary in your document as if all of the entries implicilty 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> Nameglossary.as.blocks — Present glossarys using blocks instead of lists? Synopsis<xsl:param name="glossary.as.blocks" select="0"></xsl:param> Description If non-zero, glossarys will be formatted as blocks. If you have long glossterms, proper list markup in the FO case may produce unattractive lists. By setting this parameter, you can force the stylesheets to produce block markup instead of proper lists. You can override this setting with a processing instruction as the child of glossary: <?dbfo glossary-presentation="blocks"?> or <?dbfo glossary-presentation="list"?> Nameglosslist.as.blocks — Use blocks for glosslists? Synopsis<xsl:param name="glosslist.as.blocks" - select="0"></xsl:param> Description See glossary.as.blocks. Nameglossentry.list.item.properties — To add properties to each glossentry in a list. Synopsis<xsl:attribute-set name="glossentry.list.item.properties"> <xsl:attribute name="space-before.optimum">1e <xsl:attribute name="space-before.minimum">0.8em</xsl:attribute> <xsl:attribute name="space-before.maximum">1.2em</xsl:attrib </xsl:attribute-set> Description These properties are added to the fo:list-item containing a glossentry in a glossary when the glos-

PDF BY DBLATEX

sary.as.blocks parameter is zero. Use this attribute-set to set spacing between entries, for example. Nameglossterm.block.properties — To add properties to the block of a glossentry's glossterm. Synopsis<xsl:attribute-set name="glossterm.block.properties"> - <xsl:attribute name="space-before.optimum">1em</xsl:attribute> <xsl:attribute name="space-before.minimum">0.8em</xsl:attribute <xsl:attribute name="space-before.maximum">1.2em</xsl:attribute> <xsl:attribute name="keep-with-next.within-column">always</x <xsl:attribute name="keep-together.within-column">always</xsl:attribute> </xsl:attribute-set> Description These properties are added to the block containing a glossary term in a glossary when the glossary.as.blocks parameter is non-zero. Use this attribute-set to set the space above and below, font properties, and any indent for the glossary term. Nameglossdef.block.properties — To add properties to the block of a glossary definition. Synopsis<xsl:attribute-set name="glossdef.block.properties"> - <xsl:attribute name="margin-{$direction.align.start}">.25in</xsl:attribute> </xsl:attribute-set> Description These properties are added to the block containing a glossary definition in a glossary when the glossary.as.blocks parameter is non-zero. Use this attribute-set to set the space above and below, any font properties, and any indent for the glossary definition. Nameglossterm.list.properties — To add properties to the glossterm in a list. Synopsis<xsl:attribute-set name="glossterm.list.properties"> </xsl:attribute-set> Description These properties are added to the block containing a glossary term in a glossary when the glossary.as.blocks parameter is zero. Use this attribute-set to set font properties, for example. Nameglossdef.list.properties — To add properties to the glossary definition in a list. Synopsis<xsl:attribute-set name="glossdef.list.properties"> </xsl:attribute-set> Description These properties are added to the block containing a glossary definition in a glossary when the glossary.as.blocks parameter is zero. Use this attribute-set to set font properties, for example. Nameglossterm.width — Width of glossterm in list presentation mode Synopsis<xsl:param name="glossterm.width">2in</xsl:param> Description This parameter specifies the width reserved for glossary terms when a list presentation is used. Nameglossterm.separation — Separation between glossary terms and descriptions in list mode Synopsis<xsl:param name="glossterm.separation">0.25in</xsl:param> Description Specifies the miminum horizontal separation between glossary terms and descriptions when they are presented side-by-side using lists when the glossary.as.blocks is zero. Nameglossentry.show.acronym — Display glossentry acronyms? Synopsis<xsl:param name="glossentry.show.acronym">no</xsl:param> Description A setting of "yes" means they should be displayed; "no" means they shouldn't. If "primary" is used, then they are shown as the primary text for the entry. NoteThis setting controls both acronym and abbrev elements in the glossentry. Nameglossary.sort — Sort glossentry elements? Synopsis<xsl:param name="glossary.sort" select="0"></xsl:param> Description If non-zero, then the glossentry elements within a glossary, glossdiv, or glosslist are sorted on the glossterm, using the current lang setting. If zero (the default), then glossentry elements are not sorted and are presented in document order. MiscellaneousMiscellaneousMiscellaneousTable of Contentsformal.proceduresformal.title.placementruninhead.default.title.end.punctruninhead.title.end.punctshow.commentspunct.honorificsegmente — Selects formal or informal procedures Synopsis<xsl:param name="formal.procedures" select="1"></xsl:param> Description Formal procedures are numbered and always have a title. Nameformal.title.placement — Specifies where formal object titles should occur Synopsis<xsl:param name="formal.title.placement"> figure before example before equation before table before procedure before task 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". Nameruninhead.default.title.end.punct — Default punctuation character on a run-in-head Synopsis<xsl:param name="runinhead.default.title.end.pu Description If non-zero, For a formalpara, use the specified string as the separator between the title and following text. The period is the default value. Nameruninhead.title.end.punct — Characters that count as punctuation on a run-in-head Synopsis<xsl:param name="runinhead.title.end.punct">.!?:</xsl:param> Description Specify which characters are to be counted as punctuation. These characters are checked for a match with the last character of the title. If no match is found, the runinhead.default.title.end.punct contents are inserted. This is to avoid duplicated punctuation in the output. Nameshow.comments — Display remark elements? Synopsis<xsl:param name="show.comments" select="1"></xsl:param> Description If non-zero, comments will be displayed, otherwise they are suppressed. Comments here refers to the remark element (which was called comment prior to DocBook 4.0), not XML comments (<-- like this -->) which are unavailable. Namepunct.honorific — Punctuation after an honorific in a personal name. Synopsis<xsl:param name="punct.honorific">.</xsl:param> Description This parameter specifies the punctuation that should be added after an honorific in a personal name. Namesegmentedlist.as.table — Format segmented lists as tables? Synopsis<xsl:param name="segmentedlist.as.table" select="0"></xsl:param> Description If non-zero, segmentedlists will be formatted as tables. Namevariablelist.as.blocks — Format variablelists lists as blocks? Synopsis<xsl:param - name="variablelist.as.blocks" select="0"></xsl:param> Description If non-zero, variablelists will be formatted as blocks. If you have long terms, proper list markup in the FO case may produce unattractive lists. By setting this parameter, you can force the stylesheets to produce block markup instead of proper lists. You can override this setting with a processing instruction as the child of variablelist: <?dbfo list-presentation="blocks"?> or <?dbfo list-presentation="list"?>. When using list-presentation="list", you can also control the amount of space used for the terms with the <?dbfo term-width=".25in"?> processing instruction, the termlength attribute on variablelist, or allow the stylesheets to attempt to calculate the amount of space to leave based on the number of letters in the longest term. <variablelist> <?dbfo list-presentation="list"?> <?dbfo term-width="1.5in"?> <?dbhtml list-presentation="table"?> <?dbhtml term-width="1.5in"?> <varlistentry> <term>list</term> - <listitem> <para> Formatted as a list even if variablelist.as.blocks is set to 1. </para> </listitem> </varlistentry> </variablelist> Nameblockquote.properties — To set the style for block quotations. Synopsis<xsl:attribute-

set name="blockquote.properties"> <xsl:attribute name="margin-{$direction.align.start}">0.5in</xsl:attribute> <xsl:attribute - name="margin-{$direction.align.end}">0.5in</xsl:attribute> <xsl:attribute name="space-after.minimum">0.5em</xsl:attribute> <xsl:attribute name="space-after.optimum">1em</xsl:attribute> <xsl:attribute name="space-after.maximum">2em</xsl:attribute> </xsl:attribute-set> Description The blockquote.properties attribute set specifies the formating properties of block quotations. Nameulink.show — Display URLs after ulinks? Synopsis<xsl:param name="ulink.show" select="1"></xsl:param> Description If non-zero, the URL of each ulink will appear after the text of the link. If the text of the link and the URL are identical, the URL is suppressed. See also ulink.footnotes. NoteDocBook 5 does not have an ulink element. When processing DocBoook 5 documents, ulink.show applies to all inline elements that are marked up with xlink:href attributes that point to external resources. Nameulink.footnotes — Generate footnotes for ulinks? Synopsis<xsl:param name="ulink.footnotes" - select="0"></xsl:param> Description If non-zero, and if ulink.show also is non-zero, the URL of each ulink will appear as a footnote. NoteDocBook 5 does not have an ulink element. When processing DocBoook 5 documents, ulink.footnotes applies to all inline elements that are marked up with xlink:href attributes that point to external resources. Nameulink.hyphenate — Allow URLs to be automatically hyphenated Synopsis<xsl:param name="ulink.hyphenate"></xsl:param> Description If not empty, the specified character (or more generally, content) is added to URLs after every character included in the string in the ulink.hyphenate.chars parameter (default is "/") to enable hyphenation of ulinks. If the character in this parameter is a Unicode soft hyphen (0x00AD) or Unicode zero-width space (0x200B), some FO processors will be able to reasonably hyphenate long URLs. Note that this hyphenation process is only applied when the ulink element is empty and the url attribute is reused as the link text. It is not applied if the ulink has literal text content. The same applies in in DocBook 5, where ulink was replaced with link with an xlink:href attribute. Nameulink.hyphenate.chars — List of characters to allow ulink URLs to be automatically hyphenated on Synopsis<xsl:param name="ulink.hyphenate.chars">/</xsl:param> Description If the ulink.hyphenate parameter is not empty, then hyphenation of ulinks is turned on, and any character contained in this parameter is treated as an allowable hyphenation point. This and ulink.hyphenate work together, one is pointless without the other being set to a non-empty value The default value is "/", but the parameter could be customized to contain other URL characters, as for example: <xsl:param name="ulink.hyphenate.chars">:/@&?.#</xsl:param> Nameshade.verbatim — Should verbatim environments be shaded? Synopsis<xsl:param name="shade.verbatim" select="0"></xsl:param> Description In the FO stylesheet, if this parameter is non-zero then the shade.verbatim.style properties will be applied to verbatim environments. In the HTML stylesheet, this parameter is now deprecated. Use CSS instead. Nameshade.verbatim.style — Properties that specify the style of shaded verbatim listings Synopsis<xsl:attribute-set name="shade.verbatim.style"> <xsl:attribute - name="background-color">#E0E0E0</xsl:attribute> </xsl:attribute-set> Description Properties that specify the style of shaded verbatim listings. The parameters specified (the border and background color) are added to the styling of the xsl-fo output. A border might be specified as "thin black solid" for example. See xsl-fo11 http://www.w3.org/TR/2004/WD-xsl11-20041216/#border Namehyphenate.verbatim — Should verbatim environments be hyphenated on space characters? Synopsis<xsl:param name="hyphenate.verbatim" select="0"></xsl:param> Description If the lines of program listing are too long to fit into one line it is quite common to split them at space and indicite by hook arrow that code continues on the next line. You can turn on this behaviour for programlisting, screen and synopsis elements by using this parameter. Note that you must also enable line wrapping for verbatim environments and select appropriate hyphenation character (e.g. hook arrow). This can be done using monospace.verbatim.properties attribute set: <xsl:attribute-set name="monospace.verbatim.properties" - use-attribute-sets="verbatim.properties monospace.properties"> <xsl:attribute name="wrap-option">wrap</xsl:attribute> - <xsl:attribute name="hyphenation-character">►</xsl:attribute> </xsl:attribute-set> For a list of arrows available in Unicode see http://www.unicode.org/charts/PDF/U2190.pdf and http://www.unicode.org/charts/PDF/U2900.pdf and make sure that selected character is available in the font you are using for verbatim environments. Namehyphenate.verbatim.characters — List of characters after which a line break can occur in listings Synopsis<xsl:param name="hyphenate.verbatim.characters"></xsl:param> Description If you enable hyphenate.verbatim line breaks are allowed only on space characters. If this is not enough for your document, you can specify list of additional characters after which line break is allowed in this parameter. Nameuse.svg — Allow SVG in the result tree? Synopsis<xsl:param name="use.svg" select="1"></xsl:param> 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. Nameuse.role.as.xrefstyle — Use role attribute for xrefstyle on xref? Synopsis<xsl:param name="use.role.as.xrefstyle" select="1"></xsl Description In DocBook documents that conform to a schema older than V4.3, this parameter allows role to serve the purpose of specifying the cross reference style. If non-zero, the role attribute on xref will be used to select the cross reference style. In DocBook V4.3, the xrefstyle attribute was added for this purpose. If the xrefstyle attribute is present, role will be ignored, regardless of the setting of this parameter. Example The following small stylesheet shows how to configure the stylesheets to make use of the cross reference style: <?xml version="1.0"?> <xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" - version="1.0"> <xsl:import href="../xsl/html/docbook.xsl"/> <xsl:output method="html"/> <xsl:param name="local.l10n.xml" - select="document(")"/> <l:i18n xmlns:l="http://docbook.sourceforge.net/xmlns/l10n/1.0"> <l:l10n xmlns:l="http://docbook.sourcefor language="en"> <l:context name="xref"> <l:template name="chapter" style="title" text="Chapter %n, %t"/> - <l:template name="chapter" text="Chapter %n"/> </l:context> </l:l10n> </l:i18n> </xsl:stylesheet> With this stylesheet,

the cross references in the following document: <?xml version="1.0" encoding="utf-8"?> <!DOCTYPE book PUBLIC "-//OASIS//DTD DocBook XML V4.2//EN" "http://www.oasis-open.org/docbook/xml/4.2/docbookx.dtd"> <book id="book"><title>Book</title> <preface> <title>Preface</title> <para>Normal: <xref linkend="ch1"/>.</para> <para>Title: <xref xrefstyle="title" linkend="ch1"/>.</para> </preface> <chapter id="ch1"> <title>First Chapter</title> <para>Irrelevant.</para> </chapter> </book> will appear as: Normal: Chapter 1. Title: Chapter 1, First Chapter. Namemenuchoice.separator — Separator between items of a menuchoice other than guimenuitem and guisubmenu Synopsis<xsl:param name="menuchoice.separator">+</xsl:par Description Separator used to connect items of a menuchoice other than guimenuitem and guisubmenu. The latter elements are linked with menuchoice.menu.separator. Namemenuchoice.menu.separator — Separator between items of a menuchoice with guimenuitem or guisubmenu Synopsis<xsl:param name="menuchoice.menu.separator">→ </xsl:param> Description Separator used to connect items of a menuchoice with guimenuitem or guisubmenu. Other elements are linked with menuchoice.separator. The default value is →, which is the → (right arrow) character entity. The current FOP (0.20.5) requires setting the font-family explicitly. The default value also includes spaces around the arrow, which will allow a line to break. Replace the spaces with   (nonbreaking space) if you don't want those spaces to break. Namedefault.float.class — Specifies the default float class Synopsis<xsl:param name="default.float.class"> <xsl:choose> <xsl:when test="contains($stylesheet.result.type,'html')">le <xsl:otherwise>before</xsl:otherwise> </xsl:choose> </xsl:param> Description Selects the direction in which a float should be placed. for xsl-fo this is before, for html it is left. For Western texts, the before direction is the top of the page. Namefootnote.number.format — Identifies the format used for footnote numbers Synopsis<xsl:param name="footnote.number.format">1</xsl:para Description The footnote.number.format specifies the format to use for footnote numeration (1, i, I, a, or A). Nametable.footnote.number.f — Identifies the format used for footnote numbers in tables Synopsis<xsl:param name="table.footnote.number.format">a</xsl:param> Description The table.footnote.number.format specifies the format to use for footnote numeration (1, i, I, a, or A) in tables. Namefootnote.number.symbols — Special characters to use as footnote markers Synopsis<xsl:param name="footnote.number.symbols"></xsl: Description If footnote.number.symbols is not the empty string, footnotes will use the characters it contains as footnote symbols. For example, "*†‡◊✠" will identify footnotes with "*", "†", "‡", "◊", and "✠". If there are more footnotes than symbols, the stylesheets will fall back to numbered footnotes using footnote.number.format. The use of symbols for footnotes depends on the ability of your processor (or browser) to render the symbols you select. Not all systems are capable of displaying the full range of Unicode characters. If the quoted characters in the preceding paragraph are not displayed properly, that's a good indicator that you may have trouble using those symbols for footnotes. Nametable.footnote.number.symbols — Special characters to use a footnote markers in tables Synopsis<xsl:param name="table.footnote.number.symbols"></xsl:param> Description If table.footnote.number.symbols is not the empty string, table footnotes will use the characters it contains as footnote symbols. For example, "*†‡◊✠" will identify footnotes with "*", "†", "‡", "◊", and "✠". If there are more footnotes than symbols, the stylesheets will fall back to numbered footnotes using table.footnote.number.format. The use of symbols for footnotes depends on the ability of your processor (or browser) to render the symbols you select. Not all systems are capable of displaying the full range of Unicode characters. If the quoted characters in the preceding paragraph are not displayed properly, that's a good indicator that you may have trouble using those symbols for footnotes. Namefootnote.properties — Properties applied to each footnote body Synopsis<xsl:attribute-set name="footnote.properties"> <xsl:attribute name="font-family"><xsl:value-of select="$body.fontset"></xsl:value-of></xsl:attribute> <xsl:attribute name="font-size"><xsl:value-of select="$footnote.font.size"></xsl:value-of></xsl:attribute> <xsl:attribute name="font-weight">normal</xsl:attribute> <xsl:attribute name="font-style">normal</xsl:attribute> <xsl:attribute name="text-align"><xsl:value-of select="$alignment"></xsl:value-of></xsl:a <xsl:attribute name="start-indent">0pt</xsl:attribute> <xsl:attribute name="end-indent">0pt</xsl:attribute> <xsl:attribute name="text-indent">0pt</xsl:attribute> <xsl:attribute name="hyphenate"><xsl:value-of select="$hyphenate"></xsl:value-of></xsl:attr <xsl:attribute name="wrap-option">wrap</xsl:attribute> <xsl:attribute name="linefeed-treatment">treat-as-space</xsl:attribute> </xsl:attribute-set> Description This attribute set is applied to the footnote-block for each footnote. It can be used to set the font-size, font-family, and other inheritable properties that will be applied to all footnotes. Nametable.footnote.properties — Properties applied to each table footnote body Synopsis<xsl:attribute-set name="table.footnote.properties"> <xsl:attribute name="font-family"><xsl:value-of select="$body.fontset"></xsl:value-of></xsl:attribute> <xsl:attribute name="font-size"><xsl:value-of select="$footnote.font.size"></xsl:value-of></xsl:attribute> <xsl:attribute name="font-weight">normal</xsl:attribute> <xsl:attribute name="font-style">normal</xsl:attribute> <xsl:attribute name="space-before">2pt</xsl:attribute> <xsl:attribute name="text-align"><xsl:value-of select="$alignment"></xsl:value-of></xsl:attribute> </xsl:attribute-set> Description This attribute set is applied to the footnote-block for each table footnote. It can be used to set the font-size, font-family, and other inheritable properties that will be applied to all table footnotes. Namefootnote.mark.properties — Properties applied to each footnote mark Synopsis<xsl:attribute-set name="footnote.mark.properties"> <xsl:attribute name="font-family"><xsl:value-of select="$body.fontset"></xsl:value-of></xsl:attribute> <xsl:attribute name="font-size">75%</xsl:attribute> <xsl:attribute name="font-weight">normal</xsl:attribute> <xsl:attribute name="font-style">normal</xsl:attribute> </xsl:attribute-set> Description This attribute set is applied to the footnote mark used for each footnote. It should contain only inline properties. The property to make the mark a superscript is contained in the footnote template itself, because the current version of FOP reports an error if baseline-shift is used. Namefootnote.sep.leader.properties — Properties associated with footnote separators Synopsis<xsl:attribute-set name="footnote.sep.leader.properties"> <xsl:attribute name="color">black</xsl:attribute> -

PDF BY DBLATEX

<xsl:attribute name="leader-pattern">rule</xsl:attribute> <xsl:attribute name="leader-length">1in</xsl:attribute> </xsl:attribute-set> Description The styling for the rule line that separates the footnotes from the body text. These are properties applied to the fo:leader used as the separator. If you want to do more than just set properties on the leader element, then you can customize the template named footnote.separator in fo/pagesetup.xsl. Namexref.with.number.and.title — Use number and title in cross references Synopsis<xsl:param name="xref.with.number.and.title" select="1"></xsl:param> Description A cross reference may include the number (for example, the number of an example or figure) and the title which is a required child of some targets. This parameter inserts both the relevant number as well as the title into the link. Namesuperscript.properties — Properties associated with superscripts Synopsis<xsl:attribute-set name="superscript.properties"> <xsl:attribute name="font-size">75%</xsl:attribute> </xsl:attribute-set> Description Specifies styling properties for superscripts. Namesubscript.properties — Properties associated with subscripts Synopsis<xsl:attribute-set name="subscript.properties"> <xsl:attribute name="font-size">75%</xsl:attribute> </xsl:attribute-set> Description Specifies styling properties for subscripts. Namepgwide.properties — Properties to make a figure or table page wide. Synopsis<xsl:attribute-set name="pgwide.properties"> <xsl:attribute -name="start-indent">0pt</xsl:attribute> </xsl:attribute-set> Description This attribute set is used to set the properties that make a figure or table "page wide" in fo output. It comes into effect when an attribute pgwide="1" is used. By default, it sets start-indent to 0pt. In a stylesheet that sets the parameter body.start.indent to a non-zero value in order to indent body text, this attribute set can be used to outdent pgwide figures to the start margin. If a document uses a multi-column page layout, then this attribute set could try setting span to a value of all. However, this may not work with some processors because a span property must be on an fo:block that is a direct child of fo:flow. It may work in some processors anyway. Namehighlight.source — Should the content of programlisting be syntactically highlighted? Synopsis<xsl:param name="highlight.source" select="0"></xsl:param> Description When this parameter is non-zero, the stylesheets will try to do syntax highlighting of the content of programlisting elements. You specify the language for each programlisting by using the language attribute. The highlight.default.language parameter can be used to specify the language for programlistings without a language attribute. Syntax highlighting also works for screen and synopsis elements. The actual highlighting work is done by the XSLTHL extension module. This is an external Java library that has to be downloaded separately (see below). In order to use this extension, you must•add xslthl-2.x.x.jar to your Java classpath. The latest version is available from the XSLT syntax highlighting project11 http://sourceforge.net/projects/xslthl at SourceForge. •use a customization layer in which you import one of the following stylesheet modules: •html/highlight.xsl •xhtml/highlight.xsl •xhtml-1_1/highlight.xsl •fo/highlight.xsl •let either the xslthl.config Java system property or the highlight.xslthl.config parameter point to the configuration file for syntax highlighting (using URL syntax). DocBook XSL comes with a ready-to-use configuration file, highlighting/xslthl-config.xml. The extension works with Saxon 6.5.x and Xalan-J. (Saxon 8.5 or later is also supported, but since it is an XSLT 2.0 processor it is not guaranteed to work with DocBook XSL in all circumstances.) The following is an example of a Saxon 6 command adapted for syntax highlighting, to be used on Windows: java -cp c:/Java/saxon.jar;c:/Java/xslthl-2.0.1.jar -Dxslthl.config=file:///c:/docbook-xsl/highlighting/xslthl-config.xml com.icl.saxon.StyleSheet -o test.html test.xml myhtml.xsl Namehighlight.xslthl.config — Location of XSLTHL configuration file Synopsis<xsl:param name="highlight.xslthl.config"></xsl:param> Description This location has precedence over the corresponding Java property. Please note that usually you have to specify location as URL not just as a simple path on the local filesystem. E.g. file:///home/user/xslthl/my-xslthl-config.xml. Namehighlight.default.language — Default language of programlisting Synopsis<xsl:param name="highlight.default.language"></xsl:param> Description This language is used when there is no language attribute on programlisting. Nameemail.delimiters.enabled — Generate delimiters around email addresses? Synopsis<xsl:param name="email.delimiters.enabled" select="1"></xsl:param> Description If non-zero, delimiters 11For delimiters, the stylesheets are currently hard-coded to output angle brackets. are generated around e-mail addresses (the output of the email element). Nameemail.mailto.enabled — Generate mailto: links for email addresses? Synopsis<xsl:param -name="email.mailto.enabled" select="0"></xsl:param> Description If non-zero the generated output for the email element will be a clickable mailto: link that brings up the default mail client on the system. Namesection.container.element — Select XSL-FO element name to contain sections Synopsis<xsl:param name="section.container.element">block</xsl:param> Description Selects the element name for outer container of each section. The choices are block (default) or wrapper. The fo: namespace prefix is added by the stylesheet to form the full element name. This element receives the section id attribute and the appropriate section level attribute-set. Changing this parameter to wrapper is only necessary when producing multi-column output that contains page-wide spans. Using fo:wrapper avoids the nesting of fo:block elements that prevents spans from working (the standard says a span must be on a block that is a direct child of fo:flow). If set to wrapper, the section attribute-sets only support properties that are inheritable. That's because there is no block to apply them to. Properties such as font-family are inheritable, but properties such as border are not. Only some XSL-FO processors need to use this parameter. The Antenna House processor, for example, will handle spans in nested blocks without changing the element name. The RenderX XEP product and FOP follow the XSL-FO standard and need to use wrapper. Namemonospace.verbatim.font.width — Width of a single monospace font character Synopsis<xsl:param name="monospace.verbatim.font.width">0.60em</xsl:param> Description Specifies with em units the width of a single character of the monospace font. The default value is 0.6em. This parameter is only used when a screen or programlisting element has a width attribute, which is expressed as a plain integer to indicate the maximum character count of each line. To convert this character count to an actual maximum width measurement, the width of the font

PDF BY DBLATEX

characters must be provided. Different monospace fonts have different character width, so this parameter should be adjusted to fit the monospace font being used. Nameexsl.node.set.available — Is the test function-available('exsl:node-set') true? Synopsis<xsl:param name="exsl.node.set.available"> <xsl:choose> <xsl:when exsl:foo="" test="function-available('exsl:node-set') or contains(system-property('xsl:vendor'), 'Apache Software Foundation')">1</xsl:when> <xsl:otherwise>0</xsl:otherwise> </xsl:choose> </xsl:param> Description If non-zero, then the exsl:node-set() function is available to be used in the stylesheet. If zero, then the function is not available. This param automatically detects the presence of the function and does not normally need to be set manually. This param was created to handle a long-standing bug in the Xalan processor that fails to detect the function even though it is available. Namebookmarks.collapse — Specifies the initial state of bookmarks Synopsis<xsl:param name="bookmarks.collapse" select="1"></xsl:param> Description If non-zero, the bookmark tree is collapsed so that only the top-level bookmarks are displayed initially. Otherwise, the whole tree of bookmarks is displayed. This parameter currently works with FOP 0.93 or later. Namegenerate.consistent.ids — Generate consistent id values if document is unchanged Synopsis<xsl:param name="generate.consistent.ids" select="0"></xsl:param> Description When the stylesheet assigns an id value to an output element, the generate-id() function may be used. That function may not produce consistent values between runs. Version control systems may misidentify the changing id values as changes to the document. If you set this parameter's value to 1, then the template named object.id will replace the use of the function generate-id() with <xsl:number level="multiple" count="*"/>. This counts preceding elements to generate a unique number for the id value. CautionThis param does not associate permanent unique id values with particular elements. The id values are consistent only as long as the document structure does not change. If the document structure changes, then the counting of elements changes, and all id values after the first such change may be different, even when there is no change to the element itself or its output. The default value of this parameter is zero, so generate-id() is used by default. Namebase.dir — The base directory of chunks Synopsis<xsl:param name="base.dir"></xsl:param> Description If specified, the base.dir parameter identifies the output directory for chunks. (If not specified, the output directory is system dependent.) Starting with version 1.77 of the stylesheets, the param's value will have a trailing slash added if it does not already have one. CautionDo not use base.dir to add a filename prefix string to chunked files. Instead, use the chunked.filename.prefix parameter. Namechunk.quietly — Omit the chunked filename messages. Synopsis<xsl:param name="chunk.quietly" select="0"></xsl:param> 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. GraphicsGraphicsGraphicsTable of Contentsgraphic.default.extensiondefault.image.widthpreferred.mediaobject.roleuse.role.for.mediaobjectignore.image.scalingimg.src — Default extension for graphic filenames Synopsis<xsl:param name="graphic.default.extension"></xsl:param> 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. Namedefault.image.width — The default width of images Synopsis<xsl:param name="default.image.width"></xsl:param> Description If specified, this value will be used for the width attribute on images that do not specify any viewport dimensions11 http://docbook.org/tdg5/en/html/imagedata.html#viewport.area. Namepreferred.mediaobject.r — Select which mediaobject to use based on this value of an object's role attribute. Synopsis<xsl:param name="preferred.mediaobject.ro Description A mediaobject may contain several objects such as imageobjects. If the parameter use.role.for.mediaobject is non-zero, then the role attribute on imageobjects and other objects within a mediaobject container will be used to select which object will be used. If one of the objects has a role value that matches the preferred.mediaobject.role parameter, then it has first priority for selection. If more than one has such a role value, the first one is used. See the use.role.for.mediaobject parameter for the sequence of selection. Nameuse.role.for.mediaobject — Use role attribute value for selecting which of several objects within a mediaobject to use. Synopsis<xsl:param name="use.role.for.mediaobject" select="1"></xsl:param> Description If non-zero, the role attribute on imageobjects or other objects within a mediaobject container will be used to select which object will be used. The order of selection when then parameter is non-zero is: 1.If the stylesheet parameter preferred.mediaobject.role has a value, then the object whose role equals that value is selected. 2.Else if an object's role attribute has a value of html for HTML processing or fo for FO output, then the first of such objects is selected. 3.Else the first suitable object is selected. If the value of use.role.for.mediaobject is zero, then role attributes are not considered and the first suitable object with or without a role value is used. Nameignore.image.scaling — Tell the stylesheets to ignore the author's image scaling attributes Synopsis<xsl:param name="ignore.image.scaling" select="0"></xsl:param> Description If non-zero, the scaling attributes on graphics and media objects are ignored. Nameimg.src.path — Path to HTML/FO image files Synopsis<xsl:param name="img.src.path"></xsl:param> Description Add a path prefix to the value of the fileref attribute of graphic, inlinegraphic, and imagedata elements. The resulting compound path is used in the output as the value of the src attribute of img (HTML) or external-graphic (FO). The path given by img.src.path could be relative to the directory where the HTML/FO files are created, or it could be an absolute URI. The default value is empty. Be sure to include a trailing slash if needed. This prefix is not applied to any filerefs that start with "/" or contain "//:". Namekeep.relative.image.uris — Should image URIs be resolved against xml:base? Synopsis<xsl:param name="keep.relative.image.uris" select="0"></xsl:param> Description If non-zero, relative URIs (in, for example fileref attributes) will be used in the generated output. Otherwise, the URIs will be made absolute with respect to the base URI. Note that the stylesheets calculate (and use) the absolute form for some purposes, this only applies to the resulting output. Pagination and General StylesPagination and General StylesPagination and General Styles Understand-

PDF BY DBLATEX

ing XSL FO MarginsUnderstanding XSL FO Margins To make sense of the parameters in this section, it's useful to consider Figure 1, "Page Model". Figure 1. Page Model First, let's consider the regions on the page. The white region is the physical page. Its dimensions are determined by the page.height and page.width parameters. The yellow region is the region-body. The size and placement of the region body is constrained by the dimensions labelled in the figure. The pink region at the top of the page is the region-before. The darker area inside the region-before is the header text. In XSL, the default display alignment for a region is before, but the DocBook stylesheets still explicitly make it before. That's why the darker area is at the top. The pink region at the bottom of the page is the region-after. The darker area is the footer text. In XSL, the default display alignment for a region is before, but the DocBook stylesheets explicitly make it after. That's why the darker area is at the bottom. The dimensions in the figure are: A.The page-master margin-top. B.The region-before extent. C.The region-body margin-top. D.The region-after extent. E.The page-master margin-bottom. F.The region-body margin-bottom. G.The sum of the page-master margin-left and the region-body margin-left. In DocBook, the region-body margin-left is zero by default, so this is simply the page-master margin-left. H.The sum of the page-master margin-right and the region-body margin-right. In DocBook, the region-body margin-right is zero by default, so this is simply the page-master margin-right. Table of Contentspage.heightpage.height.portraitpage.margin.bottompage.margin.innerpage.margin.outerpage.margin.toppage.orientationpage.v heightcolumn.count.backcolumn.count.bodycolumn.count.frontcolumn.count.indexcolumn.count.lotcolumn.count.titlepagecolumn.gap — The height of the physical page Synopsis<xsl:param name="page.height"> <xsl:choose> <xsl:when test="$page.orientation - = 'portrait'"> <xsl:value-of select="$page.height.portrait"></xsl:value-of> </xsl:when> <xsl:otherwise> - <xsl:value-of select="$page.width.portrait"></xsl:value-of> </xsl:otherwise> </xsl:choose> </xsl:param> Description The page height is generally calculated from the paper.type and page.orientation parameters. Namepage.height.portrait — Specify the physical size of the long edge of the page Synopsis<xsl:param name="page.height.portrait"> <xsl:choose> <xsl:when - test="$paper.type = 'A4landscape'">210mm</xsl:when> <xsl:when test="$paper.type = 'USletter'">11in</xsl:when> - <xsl:when test="$paper.type = 'USlandscape'">8.5in</xsl:when> <xsl:when test="$paper.type = 'USlegal'">14in</xsl:when> <xsl:when test="$paper.type = 'USlegallandscape'">8.5in</xsl:when> <xsl:when test="$paper.type = '4A0'">2378mm</xsl:when> <xsl:when test="$paper.type = '2A0'">1682mm</xsl:when> <xsl:when test="$paper.type = 'A0'">1189mm</xsl:when> <xsl:when test="$paper.type = 'A1'">841mm</xsl:when> <xsl:when test="$paper.type = 'A2'">594mm</xsl:when> - <xsl:when test="$paper.type = 'A3'">420mm</xsl:when> <xsl:when test="$paper.type = 'A4'">297mm</xsl:when> - <xsl:when test="$paper.type = 'A5'">210mm</xsl:when> <xsl:when test="$paper.type = 'A6'">148mm</xsl:when> - <xsl:when test="$paper.type = 'A7'">105mm</xsl:when> <xsl:when test="$paper.type = 'A8'">74mm</xsl:when> - <xsl:when test="$paper.type = 'A9'">52mm</xsl:when> <xsl:when test="$paper.type = 'A10'">37mm</xsl:when> - <xsl:when test="$paper.type = 'B0'">1414mm</xsl:when> <xsl:when test="$paper.type = 'B1'">1000mm</xsl:when> - <xsl:when test="$paper.type = 'B2'">707mm</xsl:when> <xsl:when test="$paper.type = 'B3'">500mm</xsl:when> - <xsl:when test="$paper.type = 'B4'">353mm</xsl:when> <xsl:when test="$paper.type = 'B5'">250mm</xsl:when> - <xsl:when test="$paper.type = 'B6'">176mm</xsl:when> <xsl:when test="$paper.type = 'B7'">125mm</xsl:when> - <xsl:when test="$paper.type = 'B8'">88mm</xsl:when> <xsl:when test="$paper.type = 'B9'">62mm</xsl:when> - <xsl:when test="$paper.type = 'B10'">44mm</xsl:when> <xsl:when test="$paper.type = 'C0'">1297mm</xsl:when> - <xsl:when test="$paper.type = 'C1'">917mm</xsl:when> <xsl:when test="$paper.type = 'C2'">648mm</xsl:when> - <xsl:when test="$paper.type = 'C3'">458mm</xsl:when> <xsl:when test="$paper.type = 'C4'">324mm</xsl:when> - <xsl:when test="$paper.type = 'C5'">229mm</xsl:when> <xsl:when test="$paper.type = 'C6'">162mm</xsl:when> - <xsl:when test="$paper.type = 'C7'">114mm</xsl:when> <xsl:when test="$paper.type = 'C8'">81mm</xsl:when> - <xsl:when test="$paper.type = 'C9'">57mm</xsl:when> <xsl:when test="$paper.type = 'C10'">40mm</xsl:when> - <xsl:otherwise>11in</xsl:otherwise> </xsl:choose> </xsl:param> Description The portrait page height is the length of the long edge of the physical page. Namepage.margin.bottom — The bottom margin of the page Synopsis<xsl:param name="page.margin.bottom Description The bottom page margin is the distance from the bottom of the region-after to the physical bottom of the page. Namepage.margin.inner — The inner page margin Synopsis<xsl:param name="page.margin.inner"> <xsl:choose> <xsl:when - test="$double.sided != 0">1.25in</xsl:when> <xsl:otherwise>1in</xsl:otherwise> </xsl:choose> </xsl:param> Description The inner page margin is the distance from bound edge of the page to the first column of text. The inner page margin is the distance from bound edge of the page to the outer edge of the first column of text. In left-to-right text direction, this is the left margin of recto (front side) pages. For single-sided output, it is the left margin of all pages. In right-to-left text direction, this is the right margin of recto pages. For single-sided output, this is the right margin of all pages. NoteCurrent versions (at least as of version 4.13) of the XEP XSL-FO processor do not correctly handle these margin settings for documents with right-to-left text direction. The workaround in that situation is to reverse the values for page.margin.inner and page.margin.outer, until this bug is fixed by RenderX. It does not affect documents with left-to-right text direction. See also writing.mode. Namepage.margin.outer — The outer page margin Synopsis<xsl:param name="page.margin.outer"> <xsl:choose> <xsl:when test="$double.sided != - 0">0.75in</xsl:when> <xsl:otherwise>1in</xsl:otherwise> </xsl:choose> </xsl:param> Description The outer page margin is the distance from non-bound edge of the page to the outer edge of the last column of text. In left-to-right text direction, this is the right margin of recto (front side) pages. For single-sided output, it is the right margin of all pages. In right-to-left text direction, this is the left margin of recto pages. For single-sided output, this is the left margin of all pages. NoteCurrent versions (at least as

PDF BY DBLATEX

of version 4.13) of the XEP XSL-FO processor do not correctly handle these margin settings for documents with right-to-left text direction. The workaround in that situation is to reverse the values for page.margin.inner and page.margin.outer, until this bug is fixed by RenderX. It does not affect documents with left-to-right text direction. See also writing.mode. Namepage.margin.top — The top margin of the page Synopsis<xsl:param name="page.margin.top">0.5in</xsl:param> Description The top page margin is the distance from the physical top of the page to the top of the region-before. Namepage.orientation — Select the page orientation Synopsis<xsl:param name="page.orientation">portrait</xsl:param> Description Select one from portrait or landscape. In portrait orientation, the short edge is horizontal; in landscape orientation, it is vertical. Namepage.width — The width of the physical page Synopsis<xsl:param name="page.width"> <xsl:choose> <xsl:when test="$page.orientation = 'portrait'"> <xsl:value-of select="$page.width.portrait"></xsl:value-of> </xsl:when> <xsl:otherwise> <xsl:value-of select="$page.height.portrait"></xsl:value-of> </xsl:otherwise> </xsl:choose> </xsl:param> Description The page width is generally calculated from the paper.type and page.orientation parameters. Namepage.width.portrait — Specify the physical size of the short edge of the page Synopsis<xsl:param name="page.width.portrait"> <xsl:choose> <xsl:when test="$paper.type = 'USletter'">8.5in</xsl:when> <xsl:when test="$paper.type = 'USlandscape'">11in</xsl:when> <xsl:when test="$paper.type = 'USlegal'">8.5in</xsl:when> <xsl:when test="$paper.type = 'USlegallandscape'">14in</xsl:when> <xsl:when test="$paper.type = '4A0'">1682mm</xsl:when> <xsl:when test="$paper.type = '2A0'">1189mm</xsl:when> <xsl:when test="$paper.type = 'A0'">841mm</xsl:when> <xsl:when test="$paper.type = 'A1'">594mm</xsl:when> <xsl:when test="$paper.type = 'A2'">420mm</xsl:when> <xsl:when test="$paper.type = 'A3'">297mm</xsl:when> <xsl:when test="$paper.type = 'A4'">210mm</xsl:when> <xsl:when test="$paper.type = 'A5'">148mm</xsl:when> <xsl:when test="$paper.type = 'A6'">105mm</xsl:when> <xsl:when test="$paper.type = 'A7'">74mm</xsl:when> <xsl:when test="$paper.type = 'A8'">52mm</xsl:when> <xsl:when test="$paper.type = 'A9'">37mm</xsl:when> <xsl:when test="$paper.type = 'A10'">26mm</xsl:when> <xsl:when test="$paper.type = 'B0'">1000mm</xsl:when> <xsl:when test="$paper.type = 'B1'">707mm</xsl:when> <xsl:when test="$paper.type = 'B2'">500mm</xsl:when> <xsl:when test="$paper.type = 'B3'">353mm</xsl:when> <xsl:when test="$paper.type = 'B4'">250mm</xsl:when> <xsl:when test="$paper.type = 'B5'">176mm</xsl:when> <xsl:when test="$paper.type = 'B6'">125mm</xsl:when> <xsl:when test="$paper.type = 'B7'">88mm</xsl:when> <xsl:when test="$paper.type = 'B8'">62mm</xsl:when> <xsl:when test="$paper.type = 'B9'">44mm</xsl:when> <xsl:when test="$paper.type = 'B10'">31mm</xsl:when> <xsl:when test="$paper.type = 'C0'">917mm</xsl:when> <xsl:when test="$paper.type = 'C1'">648mm</xsl:when> <xsl:when test="$paper.type = 'C2'">458mm</xsl:when> <xsl:when test="$paper.type = 'C3'">324mm</xsl:when> <xsl:when test="$paper.type = 'C4'">229mm</xsl:when> <xsl:when test="$paper.type = 'C5'">162mm</xsl:when> <xsl:when test="$paper.type = 'C6'">114mm</xsl:when> <xsl:when test="$paper.type = 'C7'">81mm</xsl:when> <xsl:when test="$paper.type = 'C8'">57mm</xsl:when> <xsl:when test="$paper.type = 'C9'">40mm</xsl:when> <xsl:when test="$paper.type = 'C10'">28mm</xsl:when> <xsl:otherwise>8.5in</xsl:otherwise> </xsl:choose> </xsl:param> Description The portrait page width is the length of the short edge of the physical page. Namepaper.type — Select the paper type Synopsis<xsl:param name="paper.type">USletter</xsl:param> Description The paper type is a convenient way to specify the paper size. The list of known paper sizes includes USletter and most of the A, B, and C sizes. See page.width.portrait, for example. Namedouble.sided — Is the document to be printed double sided? Synopsis<xsl:param name="double.sided" select="0"></xsl:param> Description This parameter is useful when printing a document on both sides of the paper. if set to non-zero, documents are formatted using different page-masters for odd and even pages. These can differ by using a slightly wider margin on the binding edge of the page, and alternating left-right positions of header or footer elements. If set to zero (the default), then only the 'odd' page masters are used for both even and odd numbered pages. See also force.blank.pages, page.margin.inner and page.margin.outer. Nameforce.blank.pages — Generate blank page to end on even page number Synopsis<xsl:param name="force.blank.pages" select="1"></xsl:param> Description If non-zero (the default), then each page sequence will be forced to end on an even-numbered page, by inserting a blank page if necessary. This will force the next page sequence to start on an odd-numbered page, which is a standard convention for printed and bound books. If zero, then such blank pages will not be inserted. Chapters will start on the next available page, regardless of whether it is an even or odd number. This is useful when publishing online where blank pages are not needed. This param is independent of the double.sided parameter, which just triggers the use of even and odd page sequence masters that differ in their header and footer placement. So you can combine the two params for alternating headers/footers and no blank pages. Namebody.margin.bottom — The bottom margin of the body text Synopsis<xsl:param name="body.margin.bottom">0.5in</xsl:param> Description The body bottom margin is the distance from the last line of text in the page body to the bottom of the region-after. Namebody.margin.top — To specify the size of the top margin of a page Synopsis<xsl:param name="body.margin.top">0.5in</xsl:param> Description The body top margin is the distance from the top of the region-before to the first line of text in the page body. Namebody.margin.inner — Specify the size of the inner margin of the body region Synopsis<xsl:param name="body.margin.inner">0in</xsl:param> Description The inner body margin is the extra inner side (binding side) margin taken from the body region in addition to the inner page margin. It makes room for a side region for text content whose width is specified by the region.inner.extent parameter. For double-sided output, this side region is fo:region-start on a odd-numbered page, and fo:region-end on an even-numbered page. For single-sided output, this side region is fo:region-start for all pages. This correspondence applies to all languages,

PDF BY DBLATEX

both left-to-right and right-to-left writing modes. The default value is zero. See also region.inner.extent, region.outer.extent, body.margin.outer, side.region.precedence. Namebody.margin.outer — Specify the size of the outer margin of the body region Synopsis<xsl:param name="body.margin.outer">0in</xsl:param> Description The outer body margin is the extra outer side (opposite the binding side) margin taken from the body region in addition to the outer page margin. It makes room for a side region for text content whose width is specified by the region.outer.extent parameter. For double-sided output, this side region is fo:region-end on a odd-numbered page, and fo:region-start on an even-numbered page. For single-sided output, this side region is fo:region-end for all pages. This correspondence applies to all languages, both left-to-right and right-to-left writing modes. The default value is zero. See also region.inner.extent, region.outer.extent, body.margin.inner, side.region.precedence. Namebody.start.indent — The start-indent for the body text Synopsis<xsl:param name="body.start.indent"> <xsl:choose> - <xsl:when test="$fop.extensions != 0">0pt</xsl:when> <xsl:when test="$passivetex.extensions != 0">0pt</xsl:when> <xsl:otherwise>4pc</xsl:otherwise> </xsl:choose> </xsl:param> Description This parameter provides the means of indenting the body text relative to section titles. For left-to-right text direction, it indents the left side. For right-to-left text direction, it indents the right side. It is used in place of the title.margin.left for all XSL-FO processors except FOP 0.25. It enables support for side floats to appear in the indented margin area. This start-indent property is added to the fo:flow for certain page sequences. Which page-sequences it is applied to is determined by the template named set.flow.properties. By default, that template adds it to the flow for page-sequences using the “body” master-reference, as well as appendixes and prefaces. If this parameter is used, section titles should have a start-indent value of 0pt if they are to be outdented relative to the body text. If you are using FOP, then set this parameter to a zero width value and set the title.margin.left parameter to the negative value of the desired indent. See also body.end.indent and title.margin.left. Namebody.end.indent — The end-indent for the body text Synopsis<xsl:param name="body.end.indent">0pt</xsl:param> Description This end-indent property is added to the fo:flow for certain page sequences. Which page-sequences it is applied to is determined by the template named set.flow.properties. By default, that template adds it to the flow for page-sequences using the “body” master-reference, as well as appendixes and prefaces. See also body.start.indent. Namealignment — Specify the default text alignment Synopsis<xsl:param name="alignment">justify</xsl:param> Description The default text alignment is used for most body text. Allowed values are left, right, start, end, center, justify. The default value is justify. Namehyphenate — Specify hyphenation behavior Synopsis<xsl:param name="hyphenate">true</xsl:param> Description If true, words may be hyphenated. Otherwise, they may not. See also ulink.hyphenate.chars Nameline-height — Specify the line-height property Synopsis<xsl:param name="line-height">normal</xsl:param> Description Sets the line-height property. Namecolumn.count.back — Number of columns on back matter pages Synopsis<xsl:param name="column.count.back" select="1"></xsl:param> Description Number of columns on back matter (appendix, glossary, etc.) pages. Namecolumn.count.body — Number of columns on body pages Synopsis<xsl:param - name="column.count.body" select="1"></xsl:param> Description Number of columns on body pages. Namecolumn.count.front — Number of columns on front matter pages Synopsis<xsl:param name="column.count.front" select="1"></xsl:param> Description Number of columns on front matter (dedication, preface, etc.) pages. Namecolumn.count.index — Number of columns on index pages Synopsis<xsl:param name="column.count.index">2</xsl:param> Description Number of columns on index pages. Namecolumn.count.lot — Number of columns on a ’List-of-Titles’ page Synopsis<xsl:param name="column.count.lot" - select="1"></xsl:param> Description Number of columns on a page sequence containing the Table of Contents, List of Figures, etc. Namecolumn.count.titlepage — Number of columns on a title page Synopsis<xsl:param name="column.count.titlepage" - select="1"></xsl:param> Description Number of columns on a title page Namecolumn.gap.back — Gap between columns in back matter Synopsis<xsl:param name="column.gap.back">12pt</xsl:param> Description Specifies the gap between columns in back matter (if column.count.back is greater than one). Namecolumn.gap.body — Gap between columns in the body Synopsis<xsl:param name="column.gap.body">12pt</xsl:param> Description Specifies the gap between columns in body matter (if column.count.body is greater than one). Namecolumn.gap.front — Gap between columns in the front matter Synopsis<xsl:param name="column.gap.front">12pt</xsl:param> Description Specifies the gap between columns in front matter (if column.count.front is greater than one). Namecolumn.gap.index — Gap between columns in the index Synopsis<xsl:param - name="column.gap.index">12pt</xsl:param> Description Specifies the gap between columns in indexes (if column.count.index is greater than one). Namecolumn.gap.lot — Gap between columns on a ’List-of-Titles’ page Synopsis<xsl:param name="column.gap.lo Description Specifies the gap between columns on ’List-of-Titles’ pages (if column.count.lot is greater than one). Namecolumn.gap.titlepage — Gap between columns on title pages Synopsis<xsl:param name="column.gap.titlepage">12pt</xsl:param> Description Specifies the gap between columns on title pages (if column.count.titlepage is greater than one). Nameregion.after.extent — Specifies the height of the footer. Synopsis<xsl:param name="region.after.extent">0.4in</xsl:param> Description The region after extent is the height of the area where footers are printed. Nameregion.before.extent — Specifies the height of the header Synopsis<xsl:param name="region.before.extent">0.4in</xsl:param> Description The region before extent is the height of the area where headers are printed. Nameregion.inner.extent — Specifies the width of the inner side region Synopsis<xsl:param - name="region.inner.extent">0in</xsl:param> Description The region inner extent is the width of the optional text area next to the inner side (binding side) of the body region. For double-sided output, this side region is fo:region-start on a odd-numbered page, and fo:region-end on an even-numbered page. For single-sided output, this side region is fo:region-start for all pages. This correspondence applies to all languages, both left-to-right and right-to-left writing modes. The default value of this pa-

PDF BY DBLATEX

rameter is zero. If you enlarge this extent, be sure to also enlarge the body.margin.inner parameter to make room for its content, otherwise any text in the side region may overlap with the body text. See also region.outer.extent, body.margin.inner, body.margin.outer, side.region.precedence. Nameregion.outer.extent — Specifies the width of the outer side region Synopsis<xsl:param name="region.outer.extent">0in</xsl:param> Description The region outer extent is the width of the optional text area next to the outer side (opposite the binding side) of the body region. For double-sided output, this side region is fo:region-end on a odd-numbered page, and fo:region-start on an even-numbered page. For single-sided output, this side region is fo:region-end for all pages. This correspondence applies to all languages, both left-to-right and right-to-left writing modes. The default value of this parameter is zero. If you enlarge this extent, be sure to also enlarge the body.margin.outer parameter to make room for its content, otherwise any text in the side region may overlap with the body text. See also region.inner.extent, body.margin.inner, body.margin.outer, side.region.precedence. Namedefault.units — Default units for an unqualified dimension Synopsis<xsl:param name="default.units">pt</xsl:param> Description If an unqualified dimension is encountered (for example, in a graphic width), the default.units will be used for the units. Unqualified dimensions are not allowed in XSL Formatting Objects. Namenormal.para.spacing — What space do you want between normal paragraphs Synopsis<xsl:attribute-set name="normal.para.spacing"> <xsl:attribute name="space-before.optimum">1em</xsl:attribute> <xsl:attribute name="space-before.minimum">0.8em</xsl:attribute> <xsl:attribute name="space-before.maximum">1.2em</xsl:attribute> </xsl:attribute-set> Description Specify the spacing required between normal paragraphs as well as the following block-level elements: ackno acknowledgements cmdsynopsis glosslist sidebar simpara simplelist To customize the spacing, you need to reset all three attributes. To specify properties on just para elements without affecting these other elements, use the para.properties attribute-set. Namepara.properties — Properties to apply to para elements Synopsis<xsl:attribute-set name="para.properties" use-attribute-sets="normal.para.spacing"> </xsl:attribute-set> Description Specify properties to apply to the fo:block of a para element, such as text-indent. Although the default attribute-set is empty, it uses the attribute-set named normal.para.spacing to add vertical space before each para. The para.properties attribute-set can override those spacing properties for para only. See also normal.para.spacing. Namebody.font.master — Specifies the default point size for body text Synopsis<xsl:param name="body.font.master"> Description The body font size is specified in two parameters (body.font.master and body.font.size) so that math can be performed on the font size by XSLT. Namebody.font.size — Specifies the default font size for body text Synopsis<xsl:param name="body.font.size"> <xsl:value-of select="$body.font.master"></xsl:value-of><xsl:text>pt</xsl:text> </xsl:param> Description The body font size is specified in two parameters (body.font.master and body.font.size) so that math can be performed on the font size by XSLT. Namefootnote.font.size — The font size for footnotes Synopsis<xsl:param name="footnote.font.size">
<xsl:value-of select="$body.font.master * 0.8"></xsl:value-of><xsl:text>pt</xsl:text> </xsl:param> Description The footnote font size is used for...footnotes! Nametitle.margin.left — Adjust the left margin for titles Synopsis<xsl:param name="title.margin.left">
<xsl:choose> <xsl:when test="$fop.extensions != 0">-4pc</xsl:when> <xsl:when test="$passivetex.extensions != 0">0pt</xsl:when> <xsl:otherwise>0pt</xsl:otherwise> </xsl:choose> </xsl:param> Description This parameter provides the means of adjusting the left margin for titles when the XSL-FO processor being used is an old version of FOP (0.25 and earlier). It is only useful when the fop.extensions is nonzero. The left margin of the body region is calculated to include this space, and titles are outdented to the left outside the body region by this amount, effectively leaving titles at the intended left margin and the body text indented. Currently this method is only used for old FOP because it cannot properly use the body.start.indent parameter. The default value when the fop.extensions parameter is nonzero is -4pc, which means the body text is indented 4 picas relative to the titles. The default value when the fop.extensions parameter equals zero is 0pt, and the body indent should instead be specified using the body.start.indent parameter. If you set the value to zero, be sure to still include a unit indicator such as 0pt, or the FO processor will report errors. Namedraft.mode — Select draft mode Synopsis<xsl:param name="draft.mode">no</xsl:param> Description Selects draft mode. If draft.mode is “yes”, the entire document will be treated as a draft. If it is “no”, the entire document will be treated as a final copy. If it is “maybe”, individual sections will be treated as draft or final independently, depending on how their status attribute is set. Namedraft.watermark.image — The URI of the image to be used for draft watermarks Synopsis<xsl:param name="draft.watermark.image">images/draft.png</xsl:param> Description The image to be used for draft watermarks. Nameheaders.on.blank.pages — Put headers on blank pages? Synopsis<xsl:param name="headers.on.blank.pages" select="1"></xsl:param> Description If non-zero, headers will be placed on blank pages. Namefooters.on.blank.pages — Put footers on blank pages? Synopsis<xsl:param name="footers.on.blank.pages" select="1"></xsl:param> Description If non-zero, footers will be placed on blank pages. Nameheader.rule — Rule under headers? Synopsis<xsl:param name="header.rule" select="1"></xsl:param> Description If non-zero, a rule will be drawn below the page headers. Namefooter.rule — Rule over footers? Synopsis<xsl:param name="footer.rule" select="1"></xsl:param> Description If non-zero, a rule will be drawn above the page footers. Nameheader.column.widths — Specify relative widths of header areas Synopsis<xsl:param name="header.column.widths">1 1 1</xsl:param> Description Page headers in print output use a three column table to position text at the left, center, and right side of the header on the page. This parameter lets you specify the relative sizes of the three columns. The default value is "1 1 1". The parameter value must be three numbers, separated by white space. The first number represents the relative width of the inside header for double-sided output. The second number is the relative width of the center header. The third number is the relative width of the outside header for double-sided output. For single-sided output, the first number is the relative width of left header for left-to-right text direction, or the right

PDF BY DBLATEX

header for right-to-left text direction. The third number is the relative width of right header for left-to-right text direction, or the left header for right-to-left text direction. The numbers are used to specify the column widths for the table that makes up the header area. In the FO output, this looks like: <fo:table-column column-number="1" column-width="proportional-column-width(1)"/> The proportional-column-width() function computes a column width by dividing its argument by the total of the arguments for all the columns, and then multiplying the result by the width of the whole table (assuming all the column specs use the function). Its argument can be any positive integer or floating point number. Zero is an acceptable value, although some FO processors may warn about it, in which case using a very small number might be more satisfactory. For example, the value "1 2 1" means the center header should have twice the width of the other areas. A value of "0 0 1" means the entire header area is reserved for the right (or outside) header text. Note that to keep the center area centered on the page, the left and right values must be the same. A specification like "1 2 3" means the center area is no longer centered on the page since the right area is three times the width of the left area. Namefooter.column.widths — Specify relative widths of footer areas Synopsis<xsl:param name="footer.column.widths">1 1 1</xsl:param> Description Page footers in print output use a three column table to position text at the left, center, and right side of the footer on the page. This parameter lets you specify the relative sizes of the three columns. The default value is "1 1 1". The parameter value must be three numbers, separated by white space. The first number represents the relative width of the inside footer for double-sided output. The second number is the relative width of the center footer. The third number is the relative width of the outside footer for double-sided output. For single-sided output, the first number is the relative width of left footer for left-to-right text direction, or the right footer for right-to-left text direction. The third number is the relative width of right footer for left-to-right text direction, or the left footer for right-to-left text direction. The numbers are used to specify the column widths for the table that makes up the footer area. In the FO output, this looks like: <fo:table-column column-number="1" column-width="proportional-column-width(1)"/> The proportional-column-width() function computes a column width by dividing its argument by the total of the arguments for all the columns, and then multiplying the result by the width of the whole table (assuming all the column specs use the function). Its argument can be any positive integer or floating point number. Zero is an acceptable value, although some FO processors may warn about it, in which case using a very small number might be more satisfactory. For example, the value "1 2 1" means the center footer should have twice the width of the other areas. A value of "0 0 1" means the entire footer area is reserved for the right (or outside) footer text. Note that to keep the center area centered on the page, the left and right values must be the same. A specification like "1 2 3" means the center area is no longer centered on the page since the right area is three times the width of the left area. Namheader.table.properties — Apply properties to the header layout table Synopsis<xsl:attribute-set name="header.table.properties"> <xsl:attribute name="table-layout">fixed</xsl:attribute> <xsl:attribute name="width">100%</xsl:attribute> </xsl:attribute-set> Description Properties applied to the table that lays out the page header. Nameheader.table.height — Specify the minimum height of the table containing the running page headers Synopsis<xsl:param name="header.table.height">14pt</xsl:param> Description Page headers in print output use a three column table to position text at the left, center, and right side of the header on the page. This parameter lets you specify the minimum height of the single row in the table. Since this specifies only the minimum height, the table should automatically grow to fit taller content. The default value is "14pt". Namefooter.table.properties — Apply properties to the footer layout table Synopsis<xsl:attribute-set name="footer.table.properties"> <xsl:attribute name="table-layout">fixed</xsl:attribute> <xsl:attribute name="width">100%</xsl:attribute> </xsl:attribute-set> Description Properties applied to the table that lays out the page footer. Namefooter.table.height — Specify the minimum height of the table containing the running page footers Synopsis<xsl:param name="footer.table.height">14pt</xsl:param> Description Page footers in print output use a three column table to position text at the left, center, and right side of the footer on the page. This parameter lets you specify the minimum height of the single row in the table. Since this specifies only the minimum height, the table should automatically grow to fit taller content. The default value is "14pt". Nameheader.content.properties — Properties of page header content Synopsis<xsl:attribute-set name="header.content.properties"> <xsl:attribute name="font-family"> <xsl:value-of select="$body.fontset"></xsl:value-of> </xsl:attribute> <xsl:attribute name="margin-left"> <xsl:value-of select="$title.margin.left"></xsl:value-of> </xsl:attribute> </xsl:attribute-set> Description Properties of page header content. Namefooter.content.properties — Properties of page footer content Synopsis<xsl:attribute-set name="footer.content.properties"> <xsl:attribute name="font-family"> <xsl:value-of select="$body.fontset"></xsl:value-of> </xsl:attribute> <xsl:attribute name="margin-left"> <xsl:value-of select="$title.margin.left"></xsl:value-of> </xsl:attribute> </xsl:attribute-set> Description Properties of page footer content. Namemarker.section.level — Control depth of sections shown in running headers or footers Synopsis<xsl:param name="marker.section.level">2</xsl:param> Description The marker.section.level parameter controls the depth of section levels that may be displayed in running headers and footers. For example, if the value is 2 (the default), then titles from sect1 and sect2 or equivalent section elements are candidates for use in running headers and footers. Each candidate title is marked in the FO output with a <fo:marker marker-class-name="section.head.marker"> element. In order for such titles to appear in headers or footers, the header.content or footer.content template must be customized to retrieve the marker using an output element such as: <fo:retrieve-marker retrieve-class-name="section.head.marker" retrieve-position="first-including-carryover" retrieve-boundary="page-sequence"/> Nameside.region.precedence — Determines side region page layout precedence Synopsis<xsl:param name="side.region.precedence">false</xsl:param> Description If optional side regions on a page are established using parameters such as body.margin.inner, region.inner.extent,

PDF BY DBLATEX

etc., then this parameter determines what happens at the corners where the side regions meet the header and footer regions. If the value of this parameter is true, then the side regions have precedence and extend higher and lower, while the header and footer regions are narrower and fit inside the side regions. If the value of this parameter is false (the default value), then the header and footer regions have precedence and extend over and below the side regions. Any value other than true or false is taken to be false. If you need to set precedence separately for individual regions, then you can set four parameters that are normally internal to the stylesheet. These four parameters are normally set based on the value from side.region.precedence: region.before.precedence region.after.precedence region.start.precedence region.end.precedence See also region.inner.extent, region.outer.extent, body.margin.inner, body.margin.outer. Nameregion.inner.properties — Properties of running inner side region Synopsis<xsl:attribute-set name="region.inner.properties"> <xsl:attribute name="border-width">0</xsl:attribute> <xsl:attribute name="padding">0</xsl:attribute> <xsl:attribute name="reference-orientation">90</xsl:attribute> </xsl:attribute-set> Description The FO stylesheet supports optional side regions similar to the header and footer regions. Any attributes declared in this attribute-set are applied to the region element in the page master on the inner side (binding side) of the page. This corresponds to <fo:regin-start> on odd-numbered pages and <fo:region-end> on even-numbered pages. For single-sided output, it always corresponds to <fo:regin-start>. You can customize the template named inner.region.content to specify the content of the inner side region. See also inner.region.content.properties, page.margin.inner, body.margin.inner, and the corresponding outer parameters. Nameregion.outer.properties — Properties of running outer side region Synopsis<xsl:attribute-set name="region.outer.properties"> <xsl:attribute name="border-width">0</xsl:attribute> <xsl:attribute name="padding">0</xsl:attribu <xsl:attribute name="reference-orientation">90</xsl:attribute> </xsl:attribute-set> Description The FO stylesheet supports optional side regions similar to the header and footer regions. Any attributes declared in this attribute-set are applied to the region element in the page master on the outer side (opposite the binding side) of the page. This corresponds to <fo:regin-start> on odd-numbered pages and <fo:region-end> on even-numbered pages. For single-sided output, it always corresponds to <fo:regin-start>. You can customize the template named outer.region.content to specify the content of the outer side region. See also outer.region.content.properties, page.margin.outer, body.margin.outer, and the corresponding inner parameters. Nameinner.region.content.properties — Properties of running inner side content Synopsis<xsl:attribute-set name="inner.region.content.propertie </xsl:attribute-set> Description The FO stylesheet supports optional side regions similar to the header and footer regions. Any attributes declared in this attribute-set are applied to the fo:block in the side region on the inner side (binding side) of the page. This corresponds to the start region on odd-numbered pages and the end region on even-numbered pages. For single-sided output, it always corresponds to the start region. You can customize the template named inner.region.content to specify the content of the inner side region. See also region.inner.properties, page.margin.inner, body.margin.inner, and the corresponding outer parameters. Nameouter.region.content.properties — Properties of running outer side content Synopsis<xsl:attribute-set name="outer.region.content.properties"> </xsl:attribute-set> Description The FO stylesheet supports optional side regions similar to the header and footer regions. Any attributes declared in this attribute-set are applied to the fo:block in the side region on the outer side (opposite the binding side) of the page. This corresponds to the start region on odd-numbered pages and the end region on even-numbered pages. For single-sided output, it always corresponds to the start region. You can customize the template named outer.region.content to specify the content of the outer side region. See also region.outer.properties, page.margin.outer, body.margin.outer, and the corresponding inner parameters. Font FamiliesFont FamiliesFont FamiliesTable of Contentsbody.font.familydingbat.font.familymonospace.font.familysans.font.familytitle.font.familysymbol.font.familyNamebody.font.f — The default font family for body text Synopsis<xsl:param name="body.font.family">serif</xsl:param> Description The body font family is the default font used for text in the page body. If more than one font is required, enter the font names, separated by a comma, e.g. <xsl:param name="body.font.family">Arial, SimSun, serif</xsl:param> Namedingbat.font.family — The font family for copyright, quotes, and other symbols Synopsis<xsl:param name="dingbat.font.family">serif</xsl:param> Description The dingbat font family is used for dingbats. If it is defined as the empty string, no font change is effected around dingbats. Namemonospace.font.family — The default font family for monospace environments Synopsis<xsl:param name="monospace.font.family">monospace</xsl:param> Description The monospace font family is used for verbatim environments (program listings, screens, etc.). If more than one font is required, enter the font names, separated by a comma, e.g. <xsl:param name="body.font.family">Arial, SimSun, serif</xsl:param> Namesans.font.family — The default sans-serif font family Synopsis<xsl:param name="sans.font.family">sans-serif</xsl:param> Description The default sans-serif font family. At the present, this isn't actually used by the stylesheets. Nametitle.font.family — The default font family for titles Synopsis<xsl:param name="title.font.family">sans-serif</xsl:param> Description The title font family is used for titles (chapter, section, figure, etc.) If more than one font is required, enter the font names, separated by a comma, e.g. <xsl:param name="body.font.family">Arial, SimSun, serif</xsl:param> Namesymbol.font.family — The font families to be searched for symbols outside of the body font Synopsis<xsl:param name="symbol.font.family">Symbol,ZapfDingbats</xsl:param> Description A typical body or title font does not contain all the character glyphs that DocBook supports. This parameter specifies additional fonts that should be searched for special characters not in the normal font. These symbol font names are automatically appended to the body or title font family name when fonts are specified in a font-family property in the FO output. The symbol font names should be entered as a comma-separated list. The default value is Symbol,ZapfDingbats. Property SetsProperty SetsProperty SetsTable of Contentsformal.object.propertiesformal.title.propertiesinformal.object.propertiesmonospace.propertiesverbatim.proper

— Properties associated with a formal object such as a figure, or other component that has a title Synopsis<xsl:attribute-set name="formal.object.properties"> <xsl:attribute name="space-before.minimum">0.5em</xsl:attribute> <xsl:attribute name="space-before.optimum">1em</xsl:attribute> <xsl:attribute name="space-before.maximum">2em</xsl:attribute> <xsl:attribute name="space-after.minimum">0.5em</xsl:attribute> <xsl:attribute name="space-after.optimum">1em</xsl:attribute> <xsl:attribute name="space-after.maximum">2em</xsl:attribute> <xsl:attribute name="keep-together.within-column">always</xsl:attribute> </xsl:attribute-set> Description The styling for formal objects in docbook. Specify the spacing before and after the object. Nameformal.title.properties — Style the title element of formal object such as a figure. Synopsis<xsl:attribute-set name="formal.title.properties" use-attribute-sets="normal.para.spacing"> <xsl:attribute name="font-weight">bold</xsl:attribute> <xsl:attribute name="font-size"> <xsl:value-of select="$body.font.master * 1.2"></xsl:value-of> <xsl:text>pt</xsl:text> </xsl:attribute> <xsl:attribute name="hyphenate">false</xsl:attribute> <xsl:attribute name="space-after.minimum">0.4em</xsl:attribute> <xsl:attribute name="space-after.optimum">0.6em</xsl:attribute> <xsl:attribute name="space-after.maximum">0.8em</xsl:attribute> </xsl:attribute-set> Description Specify how the title should be styled. Specify the font size and weight of the title of the formal object. Namein-formal.object.properties — Properties associated with an informal (untitled) object, such as an informalfigure Synopsis<xsl:attribute-set name="informal.object.properties"> <xsl:attribute name="space-before.minimum">0.5em</xsl:attribute> <xsl:attribute name="space-before.optimum">1em</xsl:attribute> <xsl:attribute name="space-before.maximum">2em</xsl:attribute> <xsl:attribute name="space-after.minimum">0.5em</xsl:attribute> <xsl:attribute name="space-after.optimum">1em</xsl:attribute> <xsl:attribute name="space-after.maximum">2em</xsl:attribute> </xsl:attribute-set> Description The styling for informal objects in docbook. Specify the spacing before and after the object. Namemonospace.properties — Properties of monospaced content Synopsis<xsl:attribute-set name="monospace.properties"> <xsl:attribute name="font-family"> <xsl:value-of select="$monospace.font.family"></xsl:value-of> </xsl:attribute> </xsl:attribute-set> Description Specifies the font name for monospaced output. This property set used to set the font-size as well, but that doesn't work very well when different fonts are used (as they are in titles and paragraphs, for example). If you want to set the font-size in a customization layer, it's probably going to be more appropriate to set font-size-adjust, if your formatter supports it. Nameverbatim.properties — Properties associated with verbatim text Synopsis<xsl:attribute-set name="verbatim.properties"> <xsl:attribute name="space-before.minimum">0.8em</xsl:attribute> <xsl:attribute name="space-before.optimum">1em</xsl:attribute> <xsl:attribute name="space-before.maximum">1.2em</xsl:attribute> <xsl:attribute name="space-after.minimum">0.8em</xsl:attribute> <xsl:attribute name="space-after.optimum">1em</xsl:attribute> <xsl:attribute name="space-after.maximum">1.2em</xsl:attribute> <xsl:attribute name="hyphenate">false</xsl:attribute> - <xsl:attribute name="wrap-option">no-wrap</xsl:attribute> <xsl:attribute name="white-space-collapse">false</xsl:attribute> <xsl:attribute name="white-space-treatment">preserve</xsl:attribute> <xsl:attribute name="linefeed-treatment">preserve</xsl:attribut <xsl:attribute name="text-align">start</xsl:attribute> </xsl:attribute-set> Description This attribute set is used on all verbatim environments. Namemonospace.verbatim.properties — What font and size do you want for monospaced content? Synopsis<xsl:attribute-set name="monospace.verbatim.properties" use-attribute-sets="verbatim.properties monospace.properties"> <xsl:attribute name="text-align">start</xsl:attribute> <xsl:attribute name="wrap-option">no-wrap</xsl:attribute> </xsl:attribute-set> Description Specify the font name and size you want for monospaced output Namesidebar.properties — Attribute set for sidebar properties Synopsis<xsl:attribute-set name="sidebar.properties" use-attribute-sets="formal.object.properties"> <xsl:attribute name="border-style">solid</xsl:attribute> <xsl:attribute name="border-width">1pt</xsl:attribute> <xsl:attribute name="border-color">black</xsl:attribute> <xsl:attribute name="background-color">#DDDDDD</xsl:attribute> <xsl:attribute name="padding-start">12pt</xsl:attribute> <xsl:attribute name="padding-end">12pt</xsl:attribute> <xsl:attribute name="padding-top">6pt</xsl:attrib <xsl:attribute name="padding-bottom">6pt</xsl:attribute> <xsl:attribute name="margin-{$direction.align.start}">0pt</xsl:attribute> <xsl:attribute name="margin-{$direction.align.end}">0pt</xsl:attribute> <!-- <xsl:attribute name="margin-top">6pt</xsl:attribute> <xsl:attribute name="margin-bottom">6pt</xsl:attribute> --> </xsl:attribute-set> Description The styling for sidebars. Namesidebar.title.properties — Attribute set for sidebar titles Synopsis<xsl:attribute-set name="sidebar.title.properties"> <xsl:attribute name="font-weight">bold</xsl:attribute> <xsl:attribute name="hyphenate">false</xsl:attribute> <xsl:attribute name="text-align">start</xsl:attribute> <xsl:attribute name="keep-with-next.within-column">always</xsl:attribute> </xsl:attribute-set> Description The styling for sidebars titles. Namesidebar.float.type — Select type of float for sidebar elements Synopsis<xsl:param name="sidebar.float.type">none</xsl:param> Description Selects the type of float for sidebar elements. •If sidebar.float.type is "none", then no float is used. •If sidebar.float.type is "before", then the float appears at the top of the page. On some processors, that may be the next page rather than the current page. •If sidebar.float.type is "left", then a left side float is used. •If sidebar.float.type is "start", then when the text direction is left-to-right a left side float is used. When the text direction is right-to-left, a right side float is used. •If sidebar.float.type is "right", then a right side float is used. •If sidebar.float.type is "end", then when the text direction is left-to-right a right side float is used. When the text direction is right-to-left, a left side float is used. •If your XSL-FO processor supports floats positioned on the "inside" or "outside" of double-sided pages, then you have those two options for side floats as well. Namesidebar.float.width — Set the default width for sidebars Synopsis<xsl:param name="sidebar.float.width">1in</xsl:param> Description Sets the default width for sidebars when used as a side float. The width determines the degree to which the sidebar block intrudes into the text area. If sidebar.float.type is "before" or "none", then this parameter is ignored. Namemargin.note.properties — Attribute set for margin.note properties Synopsis<xsl:attribute-set name="margin.note.properties"> <xsl:attribute name="font-size">90%</xsl:attribute> <xsl:attribute name="text-align">start</xsl:attr

PDF BY DBLATEX

</xsl:attribute-set> Description The styling for margin notes. By default, margin notes are not implemented for any element. A stylesheet customization is needed to make use of this attribute-set. You can use a template named “floater” to create the customization. That template can create side floats by specifying the content and characteristics as template parameters. For example: <xsl:template match="para[@role=’marginnote’]"> <xsl:call-template name="floater"> <xsl:with-param name="position"> <xsl:value-of select="$margin.note.float.type"/> </xsl:with-param> <xsl:with-param name="width"> <xsl:value-of select="$margin.note.width"/> </xsl:with-param> <xsl:with-param name="content"> <xsl:apply-imports/> </xsl:with-param> </xsl:call-template> </xsl:template> Namemargin.note.title.properties — Attribute set for margin note titles Synopsis<xsl:attribute-set name="margin.note.title.properties"> <xsl:attribute name="font-weight">bold</xsl:attribute> <xsl:attribute name="hyphenate">false</xsl:attribute> <xsl:attribute name="text-align">start</xsl:attrib <xsl:attribute name="keep-with-next.within-column">always</xsl:attribute> </xsl:attribute-set> Description The styling for margin note titles. Namemargin.note.float.type — Select type of float for margin note customizations Synopsis<xsl:param name="margin.note.float.type">none</xsl:param> Description Selects the type of float for margin notes. DocBook does not define a margin note element, so this feature must be implemented as a customization of the stylesheet. See margin.note.properties for an example. •If margin.note.float.type is “none”, then no float is used. •If margin.note.float.type is “before”, then the float appears at the top of the page. On some processors, that may be the next page rather than the current page. •If margin.note.float.type is “left” or “start”, then a left side float is used. •If margin.note.float.type is “right” or “end”, then a right side float is used. •If your XSL-FO processor supports floats positioned on the “inside” or “outside” of double-sided pages, then you have those two options for side floats as well. Namemargin.note.width — Set the default width for margin notes Synopsis<xsl:param name="margin.note.width">1in</xsl:param> Description Sets the default width for margin notes when used as a side float. The width determines the degree to which the margin note block intrudes into the text area. If margin.note.float.type is “before” or “none”, then this parameter is ignored. Namecomponent.title.properties — Properties for component titles Synopsis<xsl:attribute-set name="component.title.properties"> <xsl:attribute name="keep-with-next.within-column">always</xsl:attribute> <xsl:attribute name="space-before.optimum"><xsl:value-of select="concat($body.font.master, ’pt’)"></xsl:value-of></xsl:attribute> <xsl:attribute name="space-before.minimum"><xsl:value-of select="concat($body.font.master, ’pt * 0.8’)"></xsl:value-of></xsl:attribute> <xsl:attribute name="space-before.maximum"><xsl:value-of select="concat($body.font.ma ’pt * 1.2’)"></xsl:value-of></xsl:attribute> <xsl:attribute name="hyphenate">false</xsl:attribute> <xsl:attribute name="text-align"> <xsl:choose> <xsl:when test="((parent::article | parent::articleinfo | parent::info/parent::article) and not(ancestor::book) and not(self::bibliography)) or (parent::slides | parent::slidesinfo)">center</xsl:when> <xsl:otherwise>start</xsl:otherwise> </xsl:choose> </xsl:attribute> <xsl:attribute name="start-indent"><xsl:value-of select="$title.margin.left"></xsl:value-of></xsl:attribute> </xsl:attribute-set> Description The properties common to all component titles. Namecomponent.titlepage.properties — Properties for component titlepages Synopsis<xsl:attribute-set name="component.titlepage.properties"> </xsl:attribute-set> Description The properties that are applied to the outer block containing all the component title page information. Its main use is to set a span="all" property on the block that is a direct child of the flow. This attribute-set also applies to index titlepages. It is empty by default. Namesection.title.properties — Properties for section titles Synopsis<xsl:attribute-set name="section.title.properties"> <xsl:attribute name="font-family"> <xsl:value-of select="$title.fontset"></xsl:value-of> </xsl:attribute> <xsl:attribute name="font-weight">bold</xsl:attribute> <!-- font size is calculated dynamically by section.heading template --> <xsl:attribute name="keep-with-next.within-column">always</xsl:attribute> <xsl:attribute name="space-before.minimur <xsl:attribute name="space-before.optimum">1.0em</xsl:attribute> <xsl:attribute name="space-before.maximum">1.2em</xsl:attribu <xsl:attribute name="text-align">start</xsl:attribute> <xsl:attribute name="start-indent"><xsl:value-of select="$title.margin.left"></x of></xsl:attribute> </xsl:attribute-set> Description The properties common to all section titles. Namesection.title.level1.properties — Properties for level-1 section titles Synopsis<xsl:attribute-set name="section.title.level1.properties"> <xsl:attribute name="font-size"> <xsl:value-of select="$body.font.master * 2.0736"></xsl:value-of> <xsl:text>pt</xsl:text> </xsl:attribute> </xsl:attribute-set> Description The properties of level-1 section titles. Namesection.title.level2.properties — Properties for level-2 section titles Synopsis<xsl:attribute-set name="section.title.level2.properties"> <xsl:attribute name="font-size"> <xsl:value-of select="$body.font.master * 1.728"></xsl:value-of> <xsl:text>pt</xsl:text> </xsl:attribute> </xsl:attribute-set> Description The properties of level-2 section titles. Namesection.title.level3.properties — Properties for level-3 section titles Synopsis<xsl:attribute-set name="section.title.level3.properties"> <xsl:attribute name="font-size"> <xsl:value-of select="$body.font.master * 1.44"></xsl:value-of> <xsl:text>pt</xsl:text> </xsl:attribute> </xsl:attribute-set> Description The properties of level-3 section titles. Namesection.title.level4.properties — Properties for level-4 section titles Synopsis<xsl:attribute-set name="section.title.level4 <xsl:attribute name="font-size"> <xsl:value-of select="$body.font.master * 1.2"></xsl:value-of> <xsl:text>pt</xsl:text> </xsl:attribute> </xsl:attribute-set> Description The properties of level-4 section titles. Namesection.title.level5.properties — Properties for level-5 section titles Synopsis<xsl:attribute-set name="section.title.level5.properties"> <xsl:attribute name="font-size"> <xsl:value-of select="$body.font.master"></xsl:value-of> <xsl:text>pt</xsl:text> </xsl:attribute> </xsl:attribute-set> Description The properties of level-5 section titles. Namesection.title.level6.properties — Properties for level-6 section titles Synopsis<xsl:attribute-set name="section.title.level6.properties"> <xsl:attribute name="font-size"> <xsl:value-of select="$body.font.master"></xsl:value-of> <xsl:text>pt</xsl:text> </xsl:attribute> </xsl:attribute-set> Description The properties of level-6 section titles. This property set is actually used for all titles below level 5. Namesection.properties —

PDF BY DBLATEX

Properties for all section levels Synopsis<xsl:attribute-set name="section.properties"> </xsl:attribute-set> Description The properties that apply to the containing block of all section levels, and therefore apply to the whole section. This attribute set is inherited by the more specific attribute sets such as section.level1.properties. The default is empty. Namesection.level1.properties — Properties for level-1 sections Synopsis<xsl:attribute-set name="section.level1.properties" use-attribute-sets="section.properties">
</xsl:attribute-set> Description The properties that apply to the containing block of a level-1 section, and therefore apply to the whole section. This includes sect1 elements and section elements at level 1. For example, you could start each level-1 section on a new page by using: <xsl:attribute-set name="section.level1.properties"> <xsl:attribute name="break-before">page</xsl:attribute>
</xsl:attribute-set> This attribute set inherits attributes from the general section.properties attribute set. Namesection.level2.properties — Properties for level-2 sections Synopsis<xsl:attribute-set name="section.level2.properties" use-attribute-sets="section.properties">
</xsl:attribute-set> Description The properties that apply to the containing block of a level-2 section, and therefore apply to the whole section. This includes sect2 elements and section elements at level 2. For example, you could start each level-2 section on a new page by using: <xsl:attribute-set name="section.level2.properties"> <xsl:attribute name="break-before">page</xsl:attribute>
</xsl:attribute-set> This attribute set inherits attributes from the general section.properties attribute set. Namesection.level3.properties — Properties for level-3 sections Synopsis<xsl:attribute-set name="section.level3.properties" use-attribute-sets="section.properties">
</xsl:attribute-set> Description The properties that apply to the containing block of a level-3 section, and therefore apply to the whole section. This includes sect3 elements and section elements at level 3. For example, you could start each level-3 section on a new page by using: <xsl:attribute-set name="section.level3.properties"> <xsl:attribute name="break-before">page</xsl:attribute>
</xsl:attribute-set> This attribute set inherits attributes from the general section.properties attribute set. Namesection.level4.properties — Properties for level-4 sections Synopsis<xsl:attribute-set name="section.level4.properties" use-attribute-sets="section.properties">
</xsl:attribute-set> Description The properties that apply to the containing block of a level-4 section, and therefore apply to the whole section. This includes sect4 elements and section elements at level 4. For example, you could start each level-4 section on a new page by using: <xsl:attribute-set name="section.level4.properties"> <xsl:attribute name="break-before">page</xsl:attribute>
</xsl:attribute-set> This attribute set inherits attributes from the general section.properties attribute set. Namesection.level5.properties — Properties for level-5 sections Synopsis<xsl:attribute-set name="section.level5.properties" use-attribute-sets="section.properties">
</xsl:attribute-set> Description The properties that apply to the containing block of a level-5 section, and therefore apply to the whole section. This includes sect5 elements and section elements at level 5. For example, you could start each level-5 section on a new page by using: <xsl:attribute-set name="section.level5.properties"> <xsl:attribute name="break-before">page</xsl:attribute>
</xsl:attribute-set> This attribute set inherits attributes from the general section.properties attribute set. Namesection.level6.properties — Properties for level-6 sections Synopsis<xsl:attribute-set name="section.level6.properties" use-attribute-sets="section.properties">
</xsl:attribute-set> Description The properties that apply to the containing block of a level 6 or lower section, and therefore apply to the whole section. This includes section elements at level 6 and lower. For example, you could start each level-6 section on a new page by using: <xsl:attribute-set name="section.level6.properties"> <xsl:attribute name="break-before">page</xsl:attribute> </xsl:attribute-set> This attribute set inherits attributes from the general section.properties attribute set. Namefigure.properties — Properties associated with a figure Synopsis<xsl:attribute-set name="figure.properties" use-attribute-sets="formal.object.properties"></xsl:attribute-set> Description The styling for figures. Nameexample.properties — Properties associated with a example Synopsis<xsl:attribute-set name="example.properties" use-attribute-sets="formal.object.properties">
<xsl:attribute name="keep-together.within-column">auto</xsl:attribute> </xsl:attribute-set> Description The styling for examples. Nameequation.properties — Properties associated with a equation Synopsis<xsl:attribute-set name="equation.properties" -
use-attribute-sets="formal.object.properties"></xsl:attribute-set> Description The styling for equations. Nameequation.number.properti
— Properties that apply to the fo:table-cell containing the number of an equation that does not have a title. Synopsis<xsl:attribute-set name="equation.number.properties"> <xsl:attribute name="text-align">end</xsl:attribute> <xsl:attribute name="display-align">center</xsl:attribute> </xsl:attribute-set> Description Properties that apply to the fo:table-cell containing the number of an equation when it has no title. The number in an equation with a title is formatted along with the title, and this attribute-set does not apply. Nametable.properties — Properties associated with the block surrounding a table Synopsis<xsl:attribute-set -
name="table.properties" use-attribute-sets="formal.object.properties"> <xsl:attribute name="keep-together.within-column">auto</xsl:
</xsl:attribute-set> Description Block styling properties for tables. This parameter should really have been called table.block.properties or something like that, but we're leaving it to avoid backwards-compatibility problems. See also table.table.properties. Nametask.properti
— Properties associated with a task Synopsis<xsl:attribute-set name="task.properties" use-attribute-sets="formal.object.properties">
<xsl:attribute name="keep-together.within-column">auto</xsl:attribute> </xsl:attribute-set> Description Properties to style the entire block containing a task element. Nameinformalfigure.properties — Properties associated with an informalfigure Synopsis<xsl:attribute-set name="informalfigure.properties" use-attribute-sets="informal.object.properties"></xsl:attribute-set> Description The styling for informalfigures. Nameinformalexample.properties — Properties associated with an informalexample Synopsis<xsl:attribute-set name="informalexample.properties" use-attribute-sets="informal.object.properties"></xsl:attribute-set> Description The styling for informalexamples. Nameinformalequation.properties — Properties associated with an informalequation Synopsis<xsl:attribute-set name="informalequation.properties" use-attribute-sets="informal.object.properties"></xsl:attribute-set> Description The styling for informalequations. Nameinformaltable.properties — Properties associated with the block surrounding an informaltable Synopsis<xsl:attribute-set name="informaltable.properties" use-attribute-sets="informal.object.properties"><

PDF BY DBLATEX

set> Description Block styling properties for informaltables. This parameter should really have been called informaltable.block.propertie
or something like that, but we're leaving it to avoid backwards-compatibility problems. See also table.table.properties. Name-
procedure.properties — Properties associated with a procedure Synopsis<xsl:attribute-set name="procedure.properties" use-
attribute-sets="formal.object.properties"> <xsl:attribute name="keep-together.within-column">auto</xsl:attribute> </xsl:attribute-
set> Description The styling for procedures. Nameroot.properties — The properties of the fo:root element Synopsis<xsl:attribute-
set name="root.properties"> <xsl:attribute name="font-family"> <xsl:value-of select="$body.fontset"></xsl:value-of> -
</xsl:attribute> <xsl:attribute name="font-size"> <xsl:value-of select="$body.font.size"></xsl:value-of> </xsl:attribute>
<xsl:attribute name="text-align"> <xsl:value-of select="$alignment"></xsl:value-of> </xsl:attribute> <xsl:attribute -
name="line-height"> <xsl:value-of select="$line-height"></xsl:value-of> </xsl:attribute> <xsl:attribute name="font-
selection-strategy">character-by-character</xsl:attribute> <xsl:attribute name="line-height-shift-adjustment">disregard-shifts</xsl:att
<xsl:attribute name="writing-mode"> <xsl:value-of select="$direction.mode"></xsl:value-of> </xsl:attribute> </xsl:attribute-
set> Description This property set is used on the fo:root element of an FO file. It defines a set of default, global parameters. Name-
qanda.title.properties — Properties for qanda set titles Synopsis<xsl:attribute-set name="qanda.title.properties"> <xsl:attribute -
name="font-family"> <xsl:value-of select="$title.fontset"></xsl:value-of> </xsl:attribute> <xsl:attribute name="font-
weight">bold</xsl:attribute> <!-- font size is calculated dynamically by qanda.heading template --> <xsl:attribute name="keep-
with-next.within-column">always</xsl:attribute> <xsl:attribute name="space-before.minimum">0.8em</xsl:attribute> <xsl:attribute
name="space-before.optimum">1.0em</xsl:attribute> <xsl:attribute name="space-before.maximum">1.2em</xsl:attribute> </xsl:attri
set> Description The properties common to all qanda set titles. Nameqanda.title.level1.properties — Properties for level-1 qanda
set titles Synopsis<xsl:attribute-set name="qanda.title.level1.properties"> <xsl:attribute name="font-size"> <xsl:value-of -
select="$body.font.master * 2.0736"></xsl:value-of> <xsl:text>pt</xsl:text> </xsl:attribute> </xsl:attribute-set> Descrip-
tion The properties of level-1 qanda set titles. Nameqanda.title.level2.properties — Properties for level-2 qanda set titles
Synopsis<xsl:attribute-set name="qanda.title.level2.properties"> <xsl:attribute name="font-size"> <xsl:value-of select="$body.font.m
* 1.728"></xsl:value-of> <xsl:text>pt</xsl:text> </xsl:attribute> </xsl:attribute-set> Description The properties of level-2
qanda set titles. Nameqanda.title.level3.properties — Properties for level-3 qanda set titles Synopsis<xsl:attribute-set name="qanda.title.l
<xsl:attribute name="font-size"> <xsl:value-of select="$body.font.master * 1.44"></xsl:value-of> <xsl:text>pt</xsl:text>
</xsl:attribute> </xsl:attribute-set> Description The properties of level-3 qanda set titles. Nameqanda.title.level4.properties —
Properties for level-4 qanda set titles Synopsis<xsl:attribute-set name="qanda.title.level4.properties"> <xsl:attribute name="font-
size"> <xsl:value-of select="$body.font.master * 1.2"></xsl:value-of> <xsl:text>pt</xsl:text> </xsl:attribute> </xsl:attribute-
set> Description The properties of level-4 qanda set titles. Nameqanda.title.level5.properties — Properties for level-5 qanda
set titles Synopsis<xsl:attribute-set name="qanda.title.level5.properties"> <xsl:attribute name="font-size"> <xsl:value-of -
select="$body.font.master"></xsl:value-of> <xsl:text>pt</xsl:text> </xsl:attribute> </xsl:attribute-set> Description The prop-
erties of level-5 qanda set titles. Nameqanda.title.level6.properties — Properties for level-6 qanda set titles Synopsis<xsl:attribute-
set name="qanda.title.level6.properties"> <xsl:attribute name="font-size"> <xsl:value-of select="$body.font.master"></xsl:value-
of> <xsl:text>pt</xsl:text> </xsl:attribute> </xsl:attribute-set> Description The properties of level-6 qanda set titles. This
property set is actually used for all titles below level 5. Namearticle.appendix.title.properties — Properties for appendix titles that
appear in an article Synopsis<xsl:attribute-set name="article.appendix.title.properties" use-attribute-sets="section.title.properties -
section.title.level1.properties"> </xsl:attribute-set> Description The properties for the title of an ap-
pendix that appears inside an article. The default is to use the properties of sect1 titles. Nameabstract.properties — Proper-
ties associated with the block surrounding an abstract Synopsis<xsl:attribute-set name="abstract.properties"> <xsl:attribute -
name="start-indent">0.0in</xsl:attribute> <xsl:attribute name="end-indent">0.0in</xsl:attribute> </xsl:attribute-set> Descrip-
tion Block styling properties for abstract. See also abstract.title.properties. Nameabstract.title.properties — Properties for
abstract titles Synopsis<xsl:attribute-set name="abstract.title.properties"> <xsl:attribute name="font-family"><xsl:value-of -
select="$title.fontset"></xsl:value-of></xsl:attribute> <xsl:attribute name="font-weight">bold</xsl:attribute> <xsl:attribute -
name="keep-with-next.within-column">always</xsl:attribute> <xsl:attribute name="keep-with-next.within-column">always</xsl:attri
<xsl:attribute name="space-before.optimum"><xsl:value-of select="concat($body.font.master, 'pt')"></xsl:value-of></xsl:attribute>
<xsl:attribute name="space-before.minimum"><xsl:value-of select="concat($body.font.master, 'pt * 0.8')"></xsl:value-of></xsl:attribu
<xsl:attribute name="space-before.maximum"><xsl:value-of select="concat($body.font.master, 'pt * 1.2')"></xsl:value-of></xsl:attribu
<xsl:attribute name="hyphenate">false</xsl:attribute> <xsl:attribute name="text-align">center</xsl:attribute> </xsl:attribute-
set> Description The properties for abstract titles. See also abstract.properties. Nameindex.page.number.properties — Properties
associated with index page numbers Synopsis<xsl:attribute-set name="index.page.number.properties"> </xsl:attribute-set> De-
scription Properties associated with page numbers in indexes. Changing color to indicate the page number is a link is one
possibility. Namerevhistory.table.properties — The properties of table used for formatting revhistory Synopsis<xsl:attribute-
set name="revhistory.table.properties"> </xsl:attribute-set> Description This property set defines appearance of revhistory ta-
ble. Namerevhistory.table.cell.properties — The properties of table cells used for formatting revhistory Synopsis<xsl:attribute-
set name="revhistory.table.cell.properties"> </xsl:attribute-set> Description This property set defines appearance of individ-
ual cells in revhistory table. Namerevhistory.title.properties — The properties of revhistory title Synopsis<xsl:attribute-set -
name="revhistory.title.properties"> </xsl:attribute-set> Description This property set defines appearance of revhistory title. Pro-

PDF BY DBLATEX

filingProfilingProfiling The following parameters can be used for attribute-based profiling of your document. For more information about profiling, see DocBook XSL: TCG, Profiling (conditional text)11 http://www.sagehill.net/docbookxsl/Profiling.html. Table of Contentsprofile.archprofile.audienceprofile.conditionprofile.conformanceprofile.langprofile.osprofile.revisionprofile.revisionfla — Target profile for arch attribute Synopsis<xsl:param name="profile.arch"></xsl:param> Description The 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, ...). Nameprofile.audience — Target profile for audience attribute Synopsis<xsl:param name="profile.audience"></xsl:param> 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, ...). Nameprofile.condition — Target profile for condition attribute Synopsis<xsl:param name="profile.condition"></xsl:param> Description The 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, ...). Nameprofile.conformance — Target profile for conformance attribute Synopsis<xsl:param name="profile.conformance"></xsl:param> Description The 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, ...). Nameprofile.lang — Target profile for lang attribute Synopsis<xsl:param name="profile.lang"></xsl:param> Description The 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, ...). Nameprofile.os — Target profile for os attribute Synopsis<xsl:param name="profile.os"></xsl:param> Description The 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, ...). Nameprofile.revision — Target profile for revision attribute Synopsis<xsl:param name="profile.revision"></xsl:param> Description The 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, ...). Nameprofile.revisionflag — Target profile for revisionflag attribute Synopsis<xsl:param name="profile.revisionflag"></xsl:param> Description The 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, ...). Nameprofile.role — Target profile for role attribute Synopsis<xsl:param name="profile.role"></xsl:param> Description The 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, ...). WarningNote that role is often used for other purposes than profiling. For example it is commonly used to get emphasize in bold font: <emphasis role="bold">very important</emphasis> If you are using role for these purposes do not forget to add values like bold to value of this parameter. If you forgot you will get document with small pieces missing which are very hard to track. For this reason it is not recommended to use role attribute for profiling. You should rather use profiling specific attributes like userlevel, os, arch, condition, etc. Nameprofile.security — Target profile for security attribute Synopsis<xsl:param name="profile.security"></xsl:param> Description The 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, ...). Nameprofile.status — Target profile for status attribute Synopsis<xsl:param name="profile.status"></xsl:param> Description The 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, ...). Nameprofile.userlevel — Target profile for userlevel attribute Synopsis<xsl:param name="profile.userlevel"></xsl:param> Description The 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, ...). Nameprofile.vendor — Target profile

PDF BY DBLATEX

for vendor attribute Synopsis<xsl:param name="profile.vendor"></xsl:param> Description The 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, ...). Nameprofile.wordsize — Target profile for wordsize attribute Synopsis<xsl:param name="profile.wordsize"></xsl:param> Description The 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, ...). Nameprofile.attribute — Name of user-specified profiling attribute Synopsis<xsl:param name="profile.attribute"></xsl:param> Description This parameter is used in conjuction 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, ...). Nameprofile.value — Target profile for user-specified attribute Synopsis<xsl:param name="profile.value"></xsl:param> Description When you are using this parameter you must also specify name of profiling attribute with parameter profile.attribute. The 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, ...). Nameprofile.separator — Separator character for compound profile values Synopsis<xsl:param name="profile.separator">;</xsl:param> Description Separator character used for compound profile values. See profile.arch LocalizationLocalizationLocalizationTable of Contentsl10n.gentext.languagel10n.gentext.default.languagel10n.gentext.use.xref.languagel10n.lang.value.rfc.compliantwriting.modeN — Sets the gentext language Synopsis<xsl:param name="l10n.gentext.language"></xsl:param> Description If this parameter is set to any value other than the empty string, its value will be used as the value for the language when generating text. Setting l10n.gentext.language overrides any settings within the document being formatted. It's much more likely that you might want to set the l10n.gentext.default.language parameter. Namel10n.gentext.default.language — Sets the default language for generated text Synopsis<xsl:param name="l10n.gentext.default.language">en</xsl:param> Description The value of the l10n.gentext.default.language parameter is used as the language for generated text if no setting is provided in the source document. Namel10n.gentext.use.xref.language — Use the language of target when generating cross-reference text? Synopsis<xsl:param name="l10n.gentext.use.xref.language" select="0"></xsl:param> Description If non-zero, the language of the target will be used when generating cross reference text. Usually, the "current" language is used when generating text (that is, the language of the element that contains the cross-reference element). But setting this parameter allows the language of the element pointed to to control the generated text. Consider the following example: <para lang="en">See also <xref linkend="chap3"/>.</para> Suppose that Chapter 3 happens to be written in German. If l10n.gentext.use.xref.language is non-zero, the resulting text will be something like this: See also Kapital 3. Where the more traditional rendering would be: See also Chapter 3. Namel10n.lang.value.rfc.compliant — Make value of lang attribute RFC compliant? Synopsis<xsl:param name="l10n.lang.value.rfc.compliant" select="1"></xsl:param> Description If non-zero, ensure that the values for all lang attributes in HTML output are RFC compliant11Section 8.1.1, Language Codes[http://www.w3.org/TR/REC-html40/struct/dirlang.html#h 8.1.1], in the HTML 4.0 Recommendation states that: [RFC1766] defines and explains the language codes that must be used in HTML documents.Briefly, language codes consist of a primary code and a possibly empty series of subcodes: language-code = primary-code ("-" subcode)* And in RFC 1766, Tags for the Identification of Languages[http://www.ietf.org/rfc/rfc1766.txt], the EBNF for "language tag" is given as: Language-Tag = Primary-tag *("-" Subtag) Primary-tag = 1*8ALPHA Subtag = 1*8ALPHA . by taking any underscore characters in any lang values found in source documents, and replacing them with hyphen characters in output HTML files. For example, zh_CN in a source document becomes zh-CN in the HTML output form that source. NoteThis parameter does not cause any case change in lang values, because RFC 1766 explicitly states that all "language tags" (as it calls them) "are to be treated as case insensitive". Namewriting.mode — Direction of text flow based on locale Synopsis<xsl:param name="writing.mode"> <xsl:call-template name="gentext"> <xsl:with-param name="key">writing-mode</xsl:with-param> <xsl:with-param name="lang"> <xsl:call-template name="l10n.language"> <xsl:with-param name="target" select="/*[1]"></xsl:with-param> </xsl:call-template> </xsl:with-param> </xsl:call-template> </xsl:param> Description Sets direction of text flow and text alignment based on locale. The value is normally taken from the gentext file for the lang attribute of the document's root element, using the key name 'writing-mode' to look it up in the gentext file. But this param can also be set on the command line to override that gentext value. Accepted values are: lr-tbLeft-to-right text flow in each line, lines stack top to bottom. rl-tbRight-to-left text flow in each line, lines stack top to bottom. tb-rlTop-to-bottom text flow in each vertical line, lines stack right to left. Supported by only a few XSL-FO processors. Not supported in HTML output. lrShorthand for lr-tb. rlShorthand for rl-tb. tbShorthand for tb-rl. EBNFEBNFEBNFTable of Contentsebnf.assignmentebnf.statement.terminatorNameebnf.assignment — The EBNF production assignment operator Synopsis<xsl:param name="ebnf.assignment"> <fo:inline font-family="{$monospace.font.family}"> <xsl:text>::=</xsl:text> </fo:inline> </xsl:param> Description The ebnf.assignment parameter determines what text is used to show "assignment" in productions in productionsets. While "::=" is common, so are several other operators. Nameebnf.statement.terminator — Punctuation that ends an EBNF statement. Synopsis<xsl:param name="ebnf.statement.terminator"></xsl:param> Description The

PDF BY DBLATEX

ebnf.statement.terminator parameter determines what text is used to terminate each production in productionset. Some notations end each statement with a period. PrepressPrepressPrepressTable of Contentscrop.markscrop.mark.widthcrop.mark.offsetcrop.mark.blee — Output crop marks? Synopsis<xsl:param name="crop.marks" select="0"></xsl:param> Description If non-zero, crop marks will be added to each page. Currently this works only with XEP if you have xep.extensions set. Namecrop.mark.width — Width of crop marks. Synopsis<xsl:param name="crop.mark.width">0.5pt</xsl:param> Description Width of crop marks. Crop marks are controlled by crop.marks parameter. Namecrop.mark.offset — Length of crop marks. Synopsis<xsl:param name="crop.mark.offset">24pt</xsl:param> Description Length of crop marks. Crop marks are controlled by crop.marks parameter. Namecrop.mark.bleed — Length of invisible part of crop marks. Synopsis<xsl:param name="crop.mark.bleed">6pt</xsl:param> Description Length of invisible part of crop marks. Crop marks are controlled by crop.marks parameter. Part III. Manpages Parameter Reference This is reference documentation for all user-configurable parameters in the DocBook XSL "manpages" stylesheet (for generating groff/nroff output). Note that the manpages stylesheet is a customization layer of the DocBook XSL HTML stylesheet. Therefore, you can also use a number of HTML stylesheet parameters11 ../html/ to control manpages output (in addition to the manpages-specific parameters listed in this section). Manpages Parameter ReferenceManpages Parameter ReferenceTable of ContentsLIII. Hyphenation, justification, and breakingman.hyphenateman.hyphenate.urlsman.hyphenate.filenamesman.h Indentationman.indent.widthman.indent.refsectman.indent.blurbsman.indent.listsman.indent.verbatimsLV. Fontsman.font.funcprototype SYNOPSIS sectionman.funcsynopsis.styleLVII. AUTHORS and COPYRIGHT sectionsman.authors.section.enabledman.copyright.sect Endnotes and link handlingman.endnotes.list.enabledman.endnotes.list.headingman.endnotes.are.numberedman.base.url.for.relative.link Listsman.segtitle.suppressLX. Character/string substitutionman.charmap.enabledman.charmap.uriman.charmap.use.subsetman.charmap Refentry metadata gatheringrefentry.meta.get.quietlyrefentry.date.profilerefentry.date.profile.enabledrefentry.manual.profilerefentry.man Page header/footerman.th.extra1.suppressman.th.extra2.suppressman.th.extra3.suppressman.th.title.max.lengthman.th.extra2.max.lengt Outputman.output.manifest.enabledman.output.manifest.filenameman.output.in.separate.dirman.output.lang.in.name.enabledman.outpu Otherman.table.footnotes.dividerman.subheading.divider.enabledman.subheading.dividerHyphenation, justification, and breakingHyphenation, justification, and breakingHyphenation, justification, and breakingTable of Contentsman.hyphenateman.hyphenate.urls — Enable hyphenation? Synopsis<xsl:param name="man.hyphenate">0</xsl:param> Description If non-zero, hyphenation is enabled. NoteThe default value for this parameter is zero because groff is not particularly smart about how it does hyphenation; it can end up hyphenating a lot of things that you don't want hyphenated. To mitigate that, the default behavior of the stylesheets is to suppress hyphenation of computer inlines, filenames, and URLs. (You can override the default behavior by setting non-zero values for the man.hyphenate.urls, man.hyphenate.filenames, and man.hyphenate.computer.inlines parameters.) But the best way is still to just globally disable hyphenation, as the stylesheets do by default. The only good reason to enabled hyphenation is if you have also enabled justification (which is disabled by default). The reason is that justified text can look very bad unless you also hyphenate it; to quote the "Hypenation" node from the groff info page: Since the odds are not great for finding a set of words, for every output line, which fit nicely on a line without inserting excessive amounts of space between words, 'gtroff' hyphenates words so that it can justify lines without inserting too much space between words. So, if you set a non-zero value for the man.justify parameter (to enable justification), then you should probably also set a non-zero value for man.hyphenate (to enable hyphenation). Nameman.hyphenate.urls — Hyphenate URLs? Synopsis<xsl:param name="man.hyphenate.urls">0</xsl:param> Description If zero (the default), hyphenation is suppressed for output of the ulink url attribute. NoteIf hyphenation is already turned off globally (that is, if man.hyphenate is zero, setting man.hyphenate.urls is not necessary. If man.hyphenate.urls is non-zero, URLs will not be treated specially and are subject to hyphenation just like other words. NoteIf you are thinking about setting a non-zero value for man.hyphenate.urls in order to make long URLs break across lines, you'd probably be better off experimenting with setting the man.break.after.slash parameter first. That will cause long URLs to be broken after slashes. Nameman.hyphenate.filenames — Hyphenate filenames? Synopsis<xsl:param name="man.hyphenate.filenames">0</xsl:param> Description If zero (the default), hyphenation is suppressed for filename output. NoteIf hyphenation is already turned off globally (that is, if man.hyphenate is zero, setting man.hyphenate.filenames is not necessary. If man.hyphenate.filenames is non-zero, filenames will not be treated specially and are subject to hyphenation just like other words. NoteIf you are thinking about setting a non-zero value for man.hyphenate.filenames in order to make long filenames/pathnames break across lines, you'd probably be better off experimenting with setting the man.break.after.slash parameter first. That will cause long pathnames to be broken after slashes. Nameman.hyphenate.computer.inlines — Hyphenate computer inlines? Synopsis<xsl:param name="man.hyphenate.computer.inlines">0</xsl:param> Description If zero (the default), hyphenation is suppressed for "computer inlines" such as environment variables, constants, etc. This parameter current affects output of the following elements: classname, constant, envar, errorcode, option, replaceable, userinput, type, varname NoteIf hyphenation is already turned off globally (that is, if man.hyphenate is zero, setting the man.hyphenate.computer.inlines is not necessary. If man.hyphenate.computer.inlines is non-zero, computer inlines will not be treated specially and will be hyphenated like other words when needed. Nameman.justify — Justify text to both right and left margins? Synopsis<xsl:param name="man.justify">0</xsl:param> Description If non-zero, text is justified to both the right and left margins (or, in roff terminology, "adjusted and filled" to both the right and left margins). If zero (the default), text is adjusted to the left margin only -- producing what is traditionally called "ragged-right" text. NoteThe default value for this parameter is zero because justified text looks good only when it is also hyphenated. Without hyphenation, excessive amounts of space often end up get-

PDF BY DBLATEX

ting between words, in order to "pad" lines out to align on the right margin. The problem is that groff is not particularly smart about how it does hyphenation; it can end up hyphenating a lot of things that you don't want hyphenated. So, disabling both justification and hyphenation ensures that hyphens won't get inserted where you don't want to them, and you don't end up with lines containing excessive amounts of space between words. However, if do you decide to set a non-zero value for the man.justify parameter (to enable justification), then you should probably also set a non-zero value for man.hyphenate (to enable hyphenation). Yes, these default settings run counter to how most existing man pages are formatted. But there are some notable exceptions, such as the perl man pages. Nameman.break.after.slash — Enable line-breaking after slashes? Synopsis<xsl:param name="man.break.after.slash">0</xsl:param> Description If non-zero, line-breaking after slashes is enabled. This is mainly useful for causing long URLs or pathnames/filenames to be broken up or "wrapped" across lines (though it also has the side effect of sometimes causing relatively short URLs and pathnames to be broken up across lines too). If zero (the default), line-breaking after slashes is disabled. In that case, strings containing slashes (for example, URLs or filenames) are not broken across lines, even if they exceed the maximum column widith. WarningIf you set a non-zero value for this parameter, check your man-page output carefuly afterwards, in order to make sure that the setting has not introduced an excessive amount of breaking-up of URLs or pathnames. If your content contains mostly short URLs or pathnames, setting a non-zero value for man.break.after.slash will probably result in in a significant number of relatively short URLs and pathnames being broken across lines, which is probably not what you want. IndentationIndentationIndentationTable of Contentsman.indent.widthman.indent.refsectman.indent.blurbsman.indent.listsman.indent.verbatimsNameman.indent.width — Specifies width used for adjusted indents Synopsis<xsl:param name="man.indent.width">4</xsl:param> Description The man.indent.width parameter specifies the width used for adjusted indents. The value of man.indent.width is used for indenting of lists, verbatims, headings, and elsewhere, depending on whether the values of certain man.indent.* boolean parameters are non-zero. The value of man.indent.width should include a valid roff measurement unit (for example, n or u). The default value of 4n specifies a 4-en width; when viewed on a console, that amounts to the width of four characters. For details about roff measurment units, see the Measurements node in the groff info page. Nameman.indent.refsect — Adjust indentation of refsect* and refsection? Synopsis<xsl:param name="man.indent.refsect" select="0"></xsl:param> Description If the value of man.indent.refsect is non-zero, the width of the left margin for refsect1, refsect2 and refsect3 contents and titles (and first-level, second-level, and third-level nested refsectioninstances) is adjusted by the value of the man.indent.width parameter. With man.indent.width set to its default value of 3n, the main results are that: •contents of refsect1 are output with a left margin of three characters instead the roff default of seven or eight characters •contents of refsect2 are displayed in console output with a left margin of six characters instead the of the roff default of seven characters •the contents of refsect3 and nested refsection instances are adjusted accordingly. If instead the value of man.indent.refsect is zero, no margin adjustment is done for refsect* output. TipIf your content is primarly comprised of refsect1 and refsect2 content (or the refsection equivalent) – with few or no refsect3 or lower nested sections , you may be able to "conserve" space in your output by setting man.indent.refsect to a non-zero value. Doing so will "squeeze" the left margin in such as way as to provide an additional four characters of "room" per line in refsect1 output. That extra room may be useful if, for example, you have many verbatim sections with long lines in them. Nameman.indent.blurbs — Adjust indentation of blurbs? Synopsis<xsl:param name="man.indent.blurbs" select="1"></xsl:param> Description If the value of man.indent.blurbs is non-zero, the width of the left margin for authorblurb, personblurb, and contrib output is set to the value of the man.indent.width parameter (3n by default). If instead the value of man.indent.blurbs is zero, the built-in roff default width (7.2n) is used. Nameman.indent.lists — Adjust indentation of lists? Synopsis<xsl:param name="man.indent.lists" select="1"></xsl:param> Description If the value of man.indent.lists is non-zero, the width of the left margin for list items in itemizedlist, orderedlist, variablelist output (and output of some other lists) is set to the value of the man.indent.width parameter (4n by default). If instead the value of man.indent.lists is zero, the built-in roff default width (7.2n) is used. Nameman.indent.verbatims — Adjust indentation of verbatims? Synopsis<xsl:param name="man.indent.verbatims" select="1"></xsl:param> Description If the value of man.indent.verbatims is non-zero, the width of the left margin for output of verbatim environments (programlisting, screen, and so on) is set to the value of the man.indent.width parameter (3n by default). If instead the value of man.indent.verbatims is zero, the built-in roff default width (7.2n) is used. FontsFontsFontsTable of Contentsman.font.funcprototypeman.font.funcsynopsisinfoman.font.linksman.font.table.headingsman.font.table.titleNameman.font.funcprototyp — Specifies font for funcprototype output Synopsis <xsl:param name="man.font.funcprototype">BI</xsl:param> Description The man.font.funcprototype parameter specifies the font for funcprototype output. It should be a valid roff font name, such as BI or B. Nameman.font.funcsynopsisinfo — Specifies font for funcsynopsisinfo output Synopsis <xsl:param name="man.font.funcsynopsis Description The man.font.funcsynopsisinfo parameter specifies the font for funcsynopsisinfo output. It should be a valid roff font name, such as B or I. Nameman.font.links — Specifies font for links Synopsis<xsl:param name="man.font.links">B</xsl:param> Description The man.font.links parameter specifies the font for output of links (ulink instances and any instances of any element with an xlink:href attribute). The value of man.font.links must be either B or I, or empty. If the value is empty, no font formatting is applied to links. If you set man.endnotes.are.numbered and/or man.endnotes.list.enabled to zero (disabled), then you should probably also set an empty value for man.font.links. But if man.endnotes.are.numbered is non-zero (enabled), you should probably keep man.font.links set to B or I11The main purpose of applying a font format to links in most output formats it to indicate that the formatted text is "clickable"; given that links rendered in man pages are not "real" hyperlinks that

PDF BY DBLATEX

users can click on, it might seem like there is never a good reason to have font formatting for link contents in man output. In fact, if you suppress the display of inline link references (by setting man.endnotes.are.numbered to zero), there is no good reason to apply font formatting to links. However, if man.endnotes.are.numbered is non-zero, having font formatting for links (arguably) serves a purpose: It provides "context" information about exactly what part of the text is being "annotated" by the link. Depending on how you mark up your content, that context information may or may not have value.. Related Parameters man.endnotes.list.enabled, man.endnotes.are.numbered Nameman.font.table.headings — Specifies font for table headings Synopsis <xsl:param name="man.font.table.headings">B</xsl:param> Description The man.font.table.headings parameter specifies the font for table headings. It should be a valid roff font, such as B or I. Nameman.font.table.title — Specifies font for table headings Synopsis <xsl:param name="man.font.table.title">B</xsl:param> Description The man.font.table.title parameter specifies the font for table titles. It should be a valid roff font, such as B or I. SYNOPSIS sectionSYNOPSIS sectionSYNOPSIS sectionTable of Contentsman.funcsynopsis.styleNameman.funcsynopsis.style — What style of funcsynopsis should be generated? Synopsis<xsl:param name="man.funcsynopsis.style">ansi</xsl:param> Description If man.funcsynopsis.style is ansi, ANSI-style function synopses are generated for a funcsynopsis, otherwise K&R-style function synopses are generated. AUTHORS and COPYRIGHT sectionsAUTHORS and COPYRIGHT sectionsAUTHORS and COPYRIGHT sectionsTable of Contentsman.authors.section.enabledman.copyright.section.enabledNameman.authors.section.enabled — Display auto-generated AUTHORS section? Synopsis<xsl:param name="man.authors.section.enabled">1</xsl:param> Description If the value of man.authors.section.enabled is non-zero (the default), then an AUTHORS section is generated near the end of each man page. The output of the AUTHORS section is assembled from any author, editor, and othercredit metadata found in the contents of the child info or refentryinfo (if any) of the refentry itself, or from any author, editor, and othercredit metadata that may appear in info contents of any ancestors of the refentry. If the value of man.authors.section.enabled is zero, the the auto-generated AUTHORS section is suppressed. Set the value of man.authors.section.enabled to zero if you want to have a manually created AUTHORS section in your source, and you want it to appear in output instead of the auto-generated AUTHORS section. Nameman.copyright.section.enabled — Display auto-generated COPYRIGHT section? Synopsis<xsl:param name="man.copyright.section.enabled">1</xsl:param> Description If the value of man.copyright.section.enabled is non-zero (the default), then a COPYRIGHT section is generated near the end of each man page. The output of the COPYRIGHT section is assembled from any copyright and legalnotice metadata found in the contents of the child info or refentryinfo (if any) of the refentry itself, or from any copyright and legalnotice metadata that may appear in info contents of any ancestors of the refentry. If the value of man.copyright.section.enabled is zero, the the auto-generated COPYRIGHT section is suppressed. Set the value of man.copyright.section.enabled to zero if you want to have a manually created COPYRIGHT section in your source, and you want it to appear in output instead of the auto-generated COPYRIGHT section. Endnotes and link handlingEndnotes and link handlingEndnotes and link handlingTable of Contentsman.endnotes.list.enabledman.endnotes.list.headingman.endnotes.are.numberedman.base — Display endnotes list at end of man page? Synopsis<xsl:param name="man.endnotes.list.enabled">1</xsl:param> Description If the value of man.endnotes.list.enabled is non-zero (the default), then an endnotes list is added to the end of the output man page. If the value of man.endnotes.list.enabled is zero, the list is suppressed — unless link numbering is enabled (that is, if man.endnotes.are.numbered is non-zero), in which case, that setting overrides the man.endnotes.list.enabled setting, and the endnotes list is still displayed. The reason is that inline numbering of notesources associated with endnotes only makes sense if a (numbered) list of endnotes is also generated. NoteLeaving man.endnotes.list.enabled at its default (non-zero) value ensures that no "out of line" information (such as the URLs for hyperlinks and images) gets lost in your man-page output. It just gets "rearranged". So if you're thinking about disabling endnotes listing by setting the value of man.endnotes.list.enabled to zero: Before you do so, first take some time to carefully consider the information needs and experiences of your users. The "out of line" information has value even if the presentation of it in text output is not as interactive as it may be in other output formats. As far as the specific case of URLs: Even though the URLs displayed in text output may not be "real" (clickable) hyperlinks, many X terminals have convenience features for recognizing URLs and can, for example, present users with an options to open a URL in a browser with the user clicks on the URL is a terminal window. And short of those, users with X terminals can always manually cut and paste the URLs into a web browser. Also, note that various "man to html" tools, such as the widely used man2html11 http://users.actrix.gen.nz/michael/vhman2html.html (VH-Man2html) application, automatically mark up URLs with a@href markup during conversion — resulting in "real" hyperlinks in HTML output from those tools. To "turn off" numbering of endnotes in the endnotes list, set man.endnotes.are.numbered to zero. The endnotes list will still be displayed; it will just be displayed without the numbers22It can still make sense to have the list of endnotes displayed even if you have endnotes numbering turned off. In that case, your endnotes list basically becomes a "list of references" without any association with specific text in your document. This is probably the best option if you find the inline endnotes numbering obtrusive. Your users will still have access to all the "out of line" such as URLs for hyperlinks. The default heading for the endnotes list is NOTES. To change that, set a non-empty value for the man.endnotes.list.heading parameter. In the case of notesources that are links: Along with the URL for each link, the endnotes list includes the contents of the link. The list thus includes only non-empty33A "non-empty" link is one that looks like this:<ulink url="http://docbook.sf.net/snapshot/xsl/doc/manpages/">manpages</ulink> an "empty link" is on that looks like this:<ulink url="http://docbook.sf.net/snapshot/xsl/doc/manpages/"/> links. Empty links are never included, and never numbered. They are simply displayed inline, without any numbering. In addition, if there are

PDF BY DBLATEX

multiple instances of links in a refentry that have the same URL, the URL is listed only once. The contents listed for that link in the endnotes list are the contents of the first link which has that URL. If you disable endnotes listing, you should probably also set man.links.are.underlined to zero (to disable link underlining). Nameman.endnotes.list.heading — Specifies an alternate name for endnotes list Synopsis<xsl:param name="man.endnotes.list.heading"></xsl:param> Description If the value of the man.endnotes.are.numbered parameter and/or the man.endnotes.list.enabled parameter is non-zero (the defaults for both are non-zero), a numbered list of endnotes is generated near the end of each man page. The default heading for the list of endnotes is the equivalent of the English word NOTES in the current locale. To cause an alternate heading to be displayed, set a non-empty value for the man.endnotes.list.heading parameter — for example, REFERENCES. Nameman.endnotes.are.numbered — Number endnotes? Synopsis<xsl:param name="man.endnotes.are.numbered">1</xsl:param> Description If the value of man.endnotes.are.numbered is non-zero (the default), then for each non-empty11A “non-empty” notesource is one that looks like this:<ulink url="http://docbook.sf.net/snapshot/xsl/doc/manpages/">manpages</ulink> an “empty” notesource is on that looks like this:<ulink url="http://docbook.sf.net/snapshot/xsl/doc/manpages/"/> “notesource”: •a number (in square brackets) is displayed inline after the rendered inline contents (if any) of the notesource •the contents of the notesource are included in a numbered list of endnotes that is generated at the end of each man page; the number for each endnote corresponds to the inline number for the notesource with which it is associated The default heading for the list of endnotes is NOTES. To output a different heading, set a value for the man.endnotes.section.heading parameter. NoteThe endnotes list is also displayed (but without numbers) if the value of man.endnotes.list.enabled is non-zero. If the value of man.endnotes.are.numbered is zero, numbering of endnotess is suppressed; only inline contents (if any) of the notesource are displayed inline. ImportantIf you are thinking about disabling endnote numbering by setting the value of man.endnotes.are.numbered to zero, before you do so, first take some time to carefully consider the information needs and experiences of your users. The square-bracketed numbers displayed inline after notesources may seem obstrusive and aesthetically unpleasing22As far as notesources that are links, ytou might think it would be better to just display URLs for non-empty links inline, after their content, rather than displaying square-bracketed numbers all over the place. But it’s not better. In fact, it’s not even practical, because many (most) URLs for links are too long to be displayed inline. They end up overflowing the right margin. You can set a non-zero value for man.break.after.slash parameter to deal with that, but it could be argued that what you end up with is at least as ugly, and definitely more obstrusive, then having short square-bracketed numbers displayed inline., but in a text-only output format, the numbered-notesources/endnotes-listing mechanism is the only practical way to handle this kind of content. Also, users of “text based” browsers such as lynx will already be accustomed to seeing inline numbers for links. And various "man to html" applications, such as the widely used man2html33 http://users.actrix.gen.nz/michael/vhman2html.html (VH-Man2html) application, can automatically turn URLs into "real" HTML hyperlinks in output. So leaving man.endnotes.are.numbered at its default (non-zero) value ensures that no information is lost in your man-page output. It just gets “rearranged”. The handling of empty links is not affected by this parameter. Empty links are handled simply by displaying their URLs inline. Empty links are never auto-numbered. If you disable endnotes numbering, you should probably also set man.font.links to an empty value (to disable font formatting for links. Related Parameters man.endnotes.list.enabled, man.font.links Nameman.base.url.for.relative.links — Specifies a base URL for relative links Synopsis<xsl:param name="man.base.url.for.relative.links">[set $man.base.url.for.relative.links]/</xsl:param> Description For any “notesource” listed in the auto-generated “NOTES” section of output man pages (which is generated when the value of the man.endnotes.list.enabled parameter is non-zero), if the notesource is a link source with a relative URI, the URI is displayed in output with the value of the man.base.url.for.relative.links parameter prepended to the value of the link URI. NoteA link source is an notesource that references an external resource: •a ulink element with a url attribute •any element with an xlink:href attribute •an imagedata, audiodata, or videodata element If you use relative URIs in link sources in your DocBook refentry source, and you leave man.base.url.for.relative.links unset, the relative links will appear “as is” in the “Notes” section of any man-page output generated from your source. That’s probably not what you want, because such relative links are only usable in the context of HTML output. So, to make the links meaningful and usable in the context of man-page output, set a value for man.base.url.for.relative.links that points to the online version of HTML output generated from your DocBook refentry source. For example: <xsl:param name="man.base.url.for.relative.links" - >http://www.kernel.org/pub/software/scm/git/docs/</xsl:param> Related Parameters man.endnotes.list.enabled ListsListsListsTable of Contentsman.segtitle.suppressNameman.segtitle.suppress — Suppress display of segtitle contents? Synopsis<xsl:param - name="man.segtitle.suppress" select="0"></xsl:param> Description If the value of man.segtitle.suppress is non-zero, then display of segtitle contents is suppressed in output. Character/string substitutionCharacter/string substitutionCharacter/string substitutionTable of Contentsman.charmap.enabledman.charmap.uriman.charmap.use.subsetman.charmap.subset.profileman.charmap.subset. — Apply character map before final output? Synopsis<xsl:param name="man.charmap.enabled" select="1"></xsl:param> Description If the value of the man.charmap.enabled parameter is non-zero, a "character map" is used to substitute certain Unicode symbols and special characters with appropriate roff/groff equivalents, just before writing each man-page file to the filesystem. If instead the value of man.charmap.enabled is zero, Unicode characters are passed through "as is". Details For converting certain Unicode symbols and special characters in UTF-8 or UTF-16 encoded XML source to appropriate groff/roff equivalents in man-page output, the DocBook XSL Stylesheets distribution includes a roff character map11 http://docbook.sourceforge.net/snapshot/xsl/man that is compliant with the XSLT character map22 http://www.w3.org/TR/xslt20/#character-maps format as detailed in the XSLT

PDF BY DBLATEX

2.0 specification. The map contains more than 800 character mappings and can be considered the standard roff character map for the distribution. You can use the man.charmap.uri parameter to specify a URI for the location for an alternate roff character map to use in place of the standard roff character map provided in the distribution. You can also use a subset of a character map. For details, see the man.charmap.use.subset, man.charmap.subset.profile, and man.charmap.subset.profile.english parameters. Nameman.charmap.uri — URI for custom roff character map Synopsis<xsl:param name="man.charmap.uri"></xsl:param> Description For converting certain Unicode symbols and special characters in UTF-8 or UTF-16 encoded XML source to appropriate groff/roff equivalents in man-page output, the DocBook XSL Stylesheets distribution includes an XSLT character map11 http://www.w3.org/TR/xslt20/#character-maps. That character map can be considered the standard roff character map for the distribution. If the value of the man.charmap.uri parameter is non-empty, that value is used as the URI for the location for an alternate roff character map to use in place of the standard roff character map provided in the distribution. WarningDo not set a value for man.charmap.uri unless you have a custom roff character map that differs from the standard one provided in the distribution. Nameman.charmap.use.subset — Use subset of character map instead of full map? Synopsis<xsl:param name="man.charmap.use.subset" select="1"></xsl:param> Description If the value of the man.charmap.use.subset parameter is non-zero, a subset of the roff character map is used instead of the full roff character map. The profile of the subset used is determined either by the value of the man.charmap.subset.profile parameter (if the source is not in English) or the man.charmap.subset.profile.english parameter (if the source is in English). NoteYou may want to experiment with setting a non-zero value of man.charmap.use.subset, so that the full character map is used. Depending on which XSLT engine you run, setting a non-zero value for man.charmap.use.subset may significantly increase the time needed to process your documents. Or it may not. For example, if you set it and run it with xsltproc, it seems to dramatically increase processing time; on the other hand, if you set it and run it with Saxon, it does not seem to increase processing time nearly as much. If processing time is not a important concern and/or you can tolerate the increase in processing time imposed by using the full character map, set man.charmap.use.subset to zero. Details For converting certain Unicode symbols and special characters in UTF-8 or UTF-16 encoded XML source to appropriate groff/roff equivalents in man-page output, the DocBook XSL Stylesheets distribution includes a roff character map11 http://docbook.sourceforge.net/snapshot/xsl/manpages/charmap.groff.xsl that is compliant with the XSLT character map22 http://www.w3.org/TR/xslt20/#character-maps format as detailed in the XSLT 2.0 specification. The map contains more than 800 character mappings and can be considered the standard roff character map for the distribution. NoteYou can use the man.charmap.uri parameter to specify a URI for the location for an alternate roff character map to use in place of the standard roff character map provided in the distribution. Because it is not terrifically efficient to use the standard 800-character character map in full -- and for most (or all) users, never necessary to use it in full -- the DocBook XSL Stylesheets support a mechanism for using, within any given character map, a subset of character mappings instead of the full set. You can use the man.charmap.subset.profile or man.charmap.subset.profile.english parameter to tune the profile of that subset to use. Nameman.charmap.subset.profile — Profile of character map subset Synopsis<xsl:param name="man.charmap.subset.profile"> @*[local-name() = 'block'] = 'Miscellaneous Technical' or (@*[local-name() = 'block'] = 'C1 Controls And Latin-1 Supplement (Latin-1 Supplement)' and (@*[local-name() = 'class'] = 'symbols' or @*[local-name() = 'class'] = 'letters')) or @*[local-name() = 'block'] = 'Latin Extended-A' or (@*[local-name() = 'block'] = 'General Punctuation' and (@*[local-name() = 'class'] = 'spaces' or @*[local-name() = 'class'] = 'dashes' or @*[local-name() = 'class'] = 'quotes' or @*[local-name() = 'class'] = 'bullets')) or @*[local-name() = 'name'] = 'HORIZONTAL ELLIPSIS' or @*[local-name() = 'name'] = 'WORD JOINER' or @*[local-name() = 'name'] = 'SERVICE MARK' or @*[local-name() = 'name'] = 'TRADE MARK SIGN' or @*[local-name() = 'name'] = 'ZERO WIDTH NO-BREAK SPACE' </xsl:param> Description If the value of the man.charmap.use.subset parameter is non-zero, and your DocBook source is not written in English (that is, if the lang or xml:lang attribute on the root element in your DocBook source or on the first refentry element in your source has a value other than en), then the character-map subset specified by the man.charmap.subset.profile parameter is used instead of the full roff character map. Otherwise, if the lang or xml:lang attribute on the root element in your DocBook source or on the first refentry element in your source has the value en or if it has no lang or xml:lang attribute, then the character-map subset specified by the man.charmap.subset.profile.english parameter is used instead of man.charmap.subset.profile. The difference between the two subsets is that man.charmap.subset.profile provides mappings for characters in Western European languages that are not part of the Roman (English) alphabet (ASCII character set). The value of man.charmap.subset.profile is a string representing an XPath expression that matches attribute names and values for output-character elements in the character map. The attributes supported in the standard roff character map included in the distribution11 http://docbook.sourceforge.net/snapshot/xsl/manpages/charmap.groff.xsl are: charactera raw Unicode character or numeric Unicode character-entity value (either in decimal or hex); all characters have this attribute namea standard full/long ISO/Unicode character name (e.g., "OHM SIGN"); all characters have this attribute blocka standard Unicode "block" name (e.g., "General Punctuation"); all characters have this attribute. For the full list of Unicode block names supported in the standard roff character map, see the section called “Supported Unicode block names and "class" values”. classa class of characters (e.g., "spaces"). Not all characters have this attribute; currently, it is used only with certain characters within the "C1 Controls And Latin-1 Supplement" and "General Punctuation" blocks. For details, see the section called “Supported Unicode block names and "class" values”. entityan ISO entity name (e.g., "ohm"); not all characters have this attribute, because not all characters have ISO entity names; for example, of the 800 or so characters in the standard roff character map

PDF BY DBLATEX

included in the distribution, only around 300 have ISO entity names. stringa string representing an roff/groff escape-code (with "@esc@" used in place of the backslash), or a simple ASCII string; all characters in the roff character map have this attribute The value of man.charmap.subset.profile is evaluated as an XPath expression at run-time to select a portion of the roff character map to use. You can tune the subset used by adding or removing parts. For example, if you need to use a wide range of mathematical operators in a document, and you want to have them converted into roff markup properly, you might add the following: @*[local-name() = 'block'] ='MathematicalOperators' That will cause a additional set of around 67 additional "math" characters to be converted into roff markup. NoteDepending on which XSLT engine you use, either the EXSLT dyn:evaluate extension function (for xsltproc or Xalan) or saxon:evaluate extension function (for Saxon) are used to dynamically evaluate the value of man.charmap.subset.profile at run-time. If you don't use xsltproc, Saxon, Xalan -- or some other XSLT engine that supports dyn:evaluate -- you must either set the value of the man.charmap.use.subset parameter to zero and process your documents using the full character map instead, or set the value of the man.charmap.enabled parameter to zero instead (so that character-map processing is disabled completely. An alternative to using man.charmap.subset.profile is to create your own custom character map, and set the value of man.charmap.uri to the URI/filename for that. If you use a custom character map, you will probably want to include in it just the characters you want to use, and so you will most likely also want to set the value of man.charmap.use.subset to zero. You can create a custom character map by making a copy of the standard roff character map22 http://docbook.sourceforge.net/snapshot/xsl/manpages/charmap.groff.xsl provided in the distribution, and then adding to, changing, and/or deleting from that. CautionIf you author your DocBook XML source in UTF-8 or UTF-16 encoding and aren't sure what OSes or environments your man-page output might end up being viewed on, and not sure what version of nroff/groff those environments might have, you should be careful about what Unicode symbols and special characters you use in your source and what parts you add to the value of man.charmap.subset.profile. Many of the escape codes used are specific to groff and using them may not provide the expected output on an OS or environment that uses nroff instead of groff. On the other hand, if you intend for your man-page output to be viewed only on modern systems (for example, GNU/Linux systems, FreeBSD systems, or Cygwin environments) that have a good, up-to-date groff, then you can safely include a wide range of Unicode symbols and special characters in your UTF-8 or UTF-16 encoded DocBook XML source and add any of the supported Unicode block names to the value of man.charmap.subset.profile. For other details, see the documentation for the man.charmap.use.subset parameter. Supported Unicode block names and "class" values Below is the full list of Unicode block names and "class" values supported in the standard roff stylesheet provided in the distribution, along with a description of which codepoints from the Unicode range corresponding to that block name or block/class combination are supported. •C1 Controls And Latin-1 Supplement (Latin-1 Supplement)33 http://zvon.org/other/charSearch/PHP/search.php?searchType=103&id=C1%20Controls%20and%20Latin-1%20Supplement%20(Latin 1%20Supplement) (x00a0 to x00ff) class values•symbols •letters •Latin Extended-A44 http://zvon.org/other/charSearch/PHP/search.php A (x0100 to x017f, partial) •Spacing Modifier Letters55 http://zvon.org/other/charSearch/PHP/search.php?searchType=103&id=Spacing (x02b0 to x02ee, partial) •Greek and Coptic66 http://zvon.org/other/charSearch/PHP/search.php?searchType=103&id=Greek%20and%2 (x0370 to x03ff, partial) •General Punctuation77 http://zvon.org/other/charSearch/PHP/search.php?searchType=103&id=General%20Pu (x2000 to x206f, partial) class values•spaces88 http://zvon.org/other/charSearch/PHP/search.php?searchType=103&start=8192&end=82 •dashes99 http://zvon.org/other/charSearch/PHP/search.php?searchType=103&start=8208&end=8213 •quotes •daggers •bullets •leaders •primes •Superscripts and Subscripts1010 http://zvon.org/other/charSearch/PHP/search.php?searchType=103&id=Superscripts (x2070 to x209f) •Currency Symbols1111 http://zvon.org/other/charSearch/PHP/search.php?searchType=103&id=Currency%20Symbol (x20a0 to x20b1) •Letterlike Symbols1212 http://zvon.org/other/charSearch/PHP/search.php?searchType=103&id=Letterlike%20Symbo (x2100 to x214b) •Number Forms1313 http://zvon.org/other/charSearch/PHP/search.php?searchType=103&id=Number%20Forms (x2150 to x218f) •Arrows1414 http://zvon.org/other/charSearch/PHP/search.php?searchType=103&id=Arrows (x2190 to x21ff, partial) •Mathematical Operators1515 http://zvon.org/other/charSearch/PHP/search.php?searchType=103&id=Mathematical%20Operat (x2200 to x22ff, partial) •Control Pictures1616 http://zvon.org/other/charSearch/PHP/search.php?searchType=103&id=Control%20Pictu (x2400 to x243f) •Enclosed Alphanumerics1717 http://zvon.org/other/charSearch/PHP/search.php?searchType=103&id=Enclosed%20A (x2460 to x24ff) •Geometric Shapes1818 http://zvon.org/other/charSearch/PHP/search.php?searchType=103&id=Geometric%20Shapes (x25a0 to x25f7, partial) •Miscellaneous Symbols1919 http://zvon.org/other/charSearch/PHP/search.php?searchType=103&id=Miscella (x2600 to x26ff, partial) •Dingbats2020 http://zvon.org/other/charSearch/PHP/search.php?searchType=103&id=Dingbats (x2700 to x27be, partial) •Alphabetic Presentation Forms2121 http://zvon.org/other/charSearch/PHP/search.php?searchType=103&id=Alphabet (xfb00 to xfb04 only) Nameman.charmap.subset.profile.english — Profile of character map subset Synopsis<xsl:param name="man.char @*[local-name() = 'block'] = 'Miscellaneous Technical' or (@*[local-name() = 'block'] = 'C1 Controls And Latin-1 Supplement - (Latin-1 Supplement)' and @*[local-name() = 'class'] = 'symbols') or (@*[local-name() = 'block'] = 'General Punctuation' and (@*[local-name() = 'class'] = 'spaces' or @*[local-name() = 'class'] = 'dashes' or @*[local-name() = 'class'] = 'quotes' or @*[local-name() = 'class'] = 'bullets')) or @*[local-name() = 'name'] = 'HORIZONTAL ELLIPSIS' or @*[local-name() = - 'name'] = 'WORD JOINER' or @*[local-name() = 'name'] = 'SERVICE MARK' or @*[local-name() = 'name'] = 'TRADE - MARK SIGN' or @*[local-name() = 'name'] = 'ZERO WIDTH NO-BREAK SPACE' </xsl:param> Description If the value of the man.charmap.use.subset parameter is non-zero, and your DocBook source is written in English (that is, if its lang or xml:lang attribute on the root element in your DocBook source or on the first refentry element in your source has the value en or if it has no lang or xml:lang attribute), then the character-map subset specified by the man.charmap.subset.profile.english

PDF BY DBLATEX

parameter is used instead of the full roff character map. Otherwise, if the lang or xml:lang attribute on the root element in your DocBook source or on the first refentry element in your source has a value other than en, then the character-map subset specified by the man.charmap.subset.profile parameter is used instead of man.charmap.subset.profile.english. The difference between the two subsets is that man.charmap.subset.profile provides mappings for characters in Western European languages that are not part of the Roman (English) alphabet (ASCII character set). The value of man.charmap.subset.profile.english is a string representing an XPath expression that matches attribute names and values for output-character elements in the character map. For other details, see the documentation for the man.charmap.subset.profile.english and man.charmap.use.subset parameters. Name- man.string.subst.map.local.pre — Specifies “local” string substitutions Synopsis <xsl:param name="man.string.subst.map.local.pre"></ Description Use the man.string.subst.map.local.pre parameter to specify any “local” string substitutions to perform over the entire roff source for each man page before performing the string substitutions specified by the man.string.subst.map parameter. For details about the format of this parameter, see the documentation for the man.string.subst.map parameter. Name- man.string.subst.map — Specifies a set of string substitutions Synopsis<xsl:param name="man.string.subst.map"> <!-- - * remove no-break marker at beginning of line (stylesheet artifact) --> <ss:substitution oldstring="▒▀" - newstring="▒"></ss:substitution> <!-- * replace U+2580 no-break marker (stylesheet-added) w/ no-break space --> - <ss:substitution oldstring="▀" newstring="\ "></ss:substitution> <!-- ========== --> <!-- * squeeze multiple newlines before a roff request --> <ss:substitution oldstring=" ." newstring=" ."></ss:substitution> - <!-- * remove any .sp instances that directly precede a .PP --> <ss:substitution oldstring=".sp .PP" newstring=".PP"></ss:substitution> <!-- * remove any .sp instances that directly follow a .PP --> <ss:substitution oldstring=".sp .sp" newstring=".sp"></ss:substitution> <!-- * squeeze multiple .sp instances into a single .sp--> <ss:substitution oldstring=".PP .sp" newstring=".PP"></ss:substitution> <!-- * squeeze multiple newlines after start of no-fill (verbatim) env. --> <ss:substitution oldstring=".nf " newstring=".nf "></ss:substitution> <!-- * squeeze multiple newlines after REstoring margin --> <ss:substitution oldstring=".RE " newstring=".RE "></ss:substitution> <!-- * U+2591 is a marker we add before and after every Parameter in --> <!-- * Funcprototype - output --> <ss:substitution oldstring="░" newstring=" "></ss:substitution> <!-- * U+2592 is a marker we add for - the newline before output of <sbr>; --> <ss:substitution oldstring="▒" newstring=" "></ss:substitution> <!-- * --> <!-- * Now deal with some other characters that are added by the --> <!-- * stylesheets during processing. --> <!-- * - --> <!-- * bullet --> <ss:substitution oldstring="•" newstring="\(bu"></ss:substitution> <!-- * left double quote --> - <ss:substitution oldstring="“" newstring="\(lq"></ss:substitution> <!-- * right double quote --> <ss:substitution oldstring="”" - newstring="\(rq"></ss:substitution> <!-- * left single quote --> <ss:substitution oldstring="‘" newstring="\(oq"></ss:substitution> <!-- * right single quote --> <ss:substitution oldstring="’" newstring="\(cq"></ss:substitution> <!-- * copyright sign --> - <ss:substitution oldstring="©" newstring="\(co"></ss:substitution> <!-- * registered sign --> <ss:substitution oldstring="®" - newstring="\(rg"></ss:substitution> <!-- * ...servicemark... --> <!-- * There is no groff equivalent for it. --> <ss:substitution - oldstring="℠" newstring="(SM)"></ss:substitution> <!-- * ...trademark... --> <!-- * We don’t do "\(tm" because for - console output, --> <!-- * groff just renders that as "tm"; that is: --> <!-- * --> <!-- * Product™ -> Producttm --> <!-- * --> <!-- * So we just make it to "(TM)" instead; thus: --> <!-- * --> <!-- * Product™ -> Product(TM) - --> <ss:substitution oldstring="™" newstring="(TM)"></ss:substitution> </xsl:param> Description The man.string.subst.map parameter contains a map that specifies a set of string substitutions to perform over the entire roff source for each man page, either just before generating final man-page output (that is, before writing man-page files to disk) or, if the value of the man.charmap.enabled parameter is non-zero, before applying the roff character map. You can use man.string.subst.map as a “lightweight” character map to perform “essential” substitutions -- that is, substitutions that are always performed, even if the value of the man.charmap.enabled parameter is zero. For example, you can use it to replace quotation marks or other special characters that are generated by the DocBook XSL stylesheets for a particular locale setting (as opposed to those characters that are actually in source XML documents), or to replace any special characters that may be automatically generated by a particular customization of the DocBook XSL stylesheets. WarningDo you not change value of the man.string.subst.map parameter unless you are sure what you are doing. First consider adding your string-substitution mappings to either or both of the following parameters: man.string.subst.map.local.preapplied before man.string.subst.mapman.string.subst.map.local.postapplied after man.string.subst.map By default, both of those parameters contain no string substitutions. They are intended as a means for you to specify your own local string-substitution mappings. If you remove any of default mappings from the value of the man.string.subst.map parameter, you are likely to end up with broken output. And be very careful about adding anything to it; it’s used for doing string substitution over the entire roff source of each man page – it causes target strings to be replaced in roff requests and escapes, not just in the visible contents of the page. Contents of the substitution map The string-substitution map contains one or more ss:substitution elements, each of which has two attributes: oldstringstring to replace newstringstring with which to replace oldstring It may also include XML comments (that is, delimited with "<!--" and "-->"). Name- man.string.subst.map.local.post — Specifies “local” string substitutions Synopsis<xsl:param name="man.string.subst.map.local.post"><
Description Use the man.string.subst.map.local.post parameter to specify any “local” string substitutions to perform over the entire roff source for each man page after performing the string substitutions specified by the man.string.subst.map parameter. For details about the format of this parameter, see the documentation for the man.string.subst.map parameter. Refentry metadata gatheringRefentry metadata gatheringRefentry metadata gatheringTable of Contentsrefentry.meta.get.quietlyrefentry.date.profilerefentry.date

PDF BY DBLATEX

— Suppress notes and warnings when gathering refentry metadata? Synopsis<xsl:param name="refentry.meta.get.quietly" -select="0"></xsl:param> Description If zero (the default), notes and warnings about “missing” markup are generated during gathering of refentry metadata. If non-zero, the metadata is gathered “quietly” -- that is, the notes and warnings are suppressed. TipIf you are processing a large amount of refentry content, you may be able to speed up processing significantly by setting a non-zero value for refentry.meta.get.quietly. Namerefentry.date.profile — Specifies profile for refentry "date" data Synopsis<xsl:param -name="refentry.date.profile"> (($info[//date])[last()]/date)[1]| (($info[//pubdate])[last()]/pubdate)[1] </xsl:param> Description The value of refentry.date.profile is a string representing an XPath expression. It is evaluated at run-time and used only if refentry.date.profile.enabled is non-zero. Otherwise, the refentry metadata-gathering logic "hard coded" into the stylesheets is used. The man(7) man page describes this content as "the date of the last revision". In man pages, it is the content that is usually displayed in the center footer. Namerefentry.date.profile.enabled — Enable refentry "date" profiling? Synopsis<xsl:param -name="refentry.date.profile.enabled">0</xsl:param> Description If the value of refentry.date.profile.enabled is non-zero, then during refentry metadata gathering, the info profile specified by the customizable refentry.date.profile parameter is used. If instead the value of refentry.date.profile.enabled is zero (the default), then "hard coded" logic within the DocBook XSL stylesheets is used for gathering refentry "date" data. If you find that the default refentry metadata-gathering behavior is causing incorrect "date" data to show up in your output, then consider setting a non-zero value for refentry.date.profile.enabled and adjusting the value of refentry.date.profile to cause correct data to be gathered. Note that the terms "source" and "date" have special meanings in this context. For details, see the documentation for the refentry.date.profile parameter. Namerefentry.manual.profile — Specifies profile for refentry "manual" data Synopsis<xsl:param name="refentry.manual.profile"> (($info[//title])[last()]/title)[1]| ../title/node() </xsl:param> Description The value of refentry.manual.profile is a string representing an XPath expression. It is evaluated at run-time and used only if refentry.manual.profile.enabled is non-zero. Otherwise, the refentry metadata-gathering logic "hard coded" into the stylesheets is used. In man pages, this content is usually displayed in the middle of the header of the page. The man(7) man page describes this as "the title of the manual (e.g., Linux Programmer’s Manual)". Here are some examples from existing man pages: •dpkg utilities (dpkg-name) •User Contributed Perl Documentation (GET) •GNU Development Tools (ld) •Emperor Norton Utilities (ddate) •Debian GNU/Linux manual (faked) •GIMP Manual Pages (gimp) •KDOC Documentation System (qt2kdoc) Namerefentry.manual.profile.enabled — Enable refentry "manual" profiling? Synopsis<xsl:param -name="refentry.manual.profile.enabled">0</xsl:param> Description If the value of refentry.manual.profile.enabled is non-zero, then during refentry metadata gathering, the info profile specified by the customizable refentry.manual.profile parameter is used. If instead the value of refentry.manual.profile.enabled is zero (the default), then "hard coded" logic within the DocBook XSL stylesheets is used for gathering refentry "manual" data. If you find that the default refentry metadata-gathering behavior is causing incorrect "manual" data to show up in your output, then consider setting a non-zero value for refentry.manual.profile.enabled and adjusting the value of refentry.manual.profile to cause correct data to be gathered. Note that the term "manual" has a special meanings in this context. For details, see the documentation for the refentry.manual.profile parameter. Namerefentry.source.name.suppress — Suppress "name" part of refentry "source" contents? Synopsis<xsl:param -name="refentry.source.name.suppress">0</xsl:param> Description If the value of refentry.source.name.suppress is non-zero, then during refentry metadata gathering, no "source name" data is added to the refentry "source" contents. Instead (unless refentry.version.suppress is also non-zero), only "version" data is added to the "source" contents. If you find that the refentry metadata gathering mechanism is causing unwanted "source name" data to show up in your output -- for example, in the footer (or possibly header) of a man page -- then you might consider setting a non-zero value for refentry.source.name.suppress. Note that the terms "source", "source name", and "version" have special meanings in this context. For details, see the documentation for the refentry.source.name.profile parameter. Namerefentry.source.name.profile — Specifies profile for refentry "source name" data Synopsis<xsl:param name="refentry.source.name.profile"> (($info[//productname])[last()]/productname)[1]| -(($info[//corpname])[last()]/corpname)[1]| (($info[//corpcredit])[last()]/corpcredit)[1]| (($info[//corpauthor])[last()]/corpauthor)[1]| (($info[//orgname])[last()]/orgname)[1]| (($info[//publishername])[last()]/publishername)[1] </xsl:param> Description The value of refentry.source.name.profile is a string representing an XPath expression. It is evaluated at run-time and used only if refentry.source.name.profile.enabled is non-zero. Otherwise, the refentry metadata-gathering logic "hard coded" into the stylesheets is used. A "source name" is one part of a (potentially) two-part Name Version "source" field. In man pages, it is usually displayed in the left footer of the page. It typically indicates the software system or product that the item documented in the man page belongs to. The man(7) man page describes it as "the source of the command", and provides the following examples: •For binaries, use something like: GNU, NET-2, SLS Distribution, MCC Distribution. •For system calls, use the version of the kernel that you are currently looking at: Linux 0.99.11. •For library calls, use the source of the function: GNU, BSD 4.3, Linux DLL 4.4.1. In practice, there are many pages that simply have a Version number in the "source" field. So, it looks like what we have is a two-part field, Name Version, where: Nameproduct name (e.g., BSD) or org. name (e.g., GNU) Versionversion number Each part is optional. If the Name is a product name, then the Version is probably the version of the product. Or there may be no Name, in which case, if there is a Version, it is probably the version of the item itself, not the product it is part of. Or, if the Name is an organization name, then there probably will be no Version. Namerefentry.source.name.profile.enabled — Enable refentry "source name" profiling? Synopsis<xsl:param -name="refentry.source.name.profile.enabled">0</xsl:param> Description If the value of refentry.source.name.profile.enabled is

non-zero, then during refentry metadata gathering, the info profile specified by the customizable refentry.source.name.profile parameter is used. If instead the value of refentry.source.name.profile.enabled is zero (the default), then "hard coded" logic within the DocBook XSL stylesheets is used for gathering refentry "source name" data. If you find that the default refentry metadata-gathering behavior is causing incorrect "source name" data to show up in your output, then consider setting a non-zero value for refentry.source.name.profile.enabled and adjusting the value of refentry.source.name.profile to cause correct data to be gathered. Note that the terms "source" and "source name" have special meanings in this context. For details, see the documentation for the refentry.source.name.profile parameter. Namerefentry.version.suppress — Suppress "version" part of refentry "source" contents? Synopsis<xsl:param name="refentry.version.suppress">0</xsl:param> Description If the value of refentry.version.suppress is non-zero, then during refentry metadata gathering, no "version" data is added to the refentry "source" contents. Instead (unless refentry.source.name.suppress is also non-zero), only "source name" data is added to the "source" contents. If you find that the refentry metadata gathering mechanism is causing unwanted "version" data to show up in your output -- for example, in the footer (or possibly header) of a man page -- then you might consider setting a non-zero value for refentry.version.suppress. Note that the terms "source", "source name", and "version" have special meanings in this context. For details, see the documentation for the refentry.source.name.profile parameter. Namerefentry.version.profile — Specifies profile for refentry "version" data Synopsis<xsl:param name="refentry.version.profile"> (($info[//productnumber])[last()]/productnumber)[1]| (($info[//edition])[last()]/editior (($info[//releaseinfo])[last()]/releaseinfo)[1] </xsl:param> Description The value of refentry.version.profile is a string representing an XPath expression. It is evaluated at run-time and used only if refentry.version.profile.enabled is non-zero. Otherwise, the refentry metadata-gathering logic "hard coded" into the stylesheets is used. A "source.name" is one part of a (potentially) two-part Name Version "source" field. For more details, see the documentation for the refentry.source.name.profile parameter. Namerefentry.version.profile.enabled — Enable refentry "version" profiling? Synopsis<xsl:param name="refentry.version.profile.enabled">0</xsl:p Description If the value of refentry.version.profile.enabled is non-zero, then during refentry metadata gathering, the info profile specified by the customizable refentry.version.profile parameter is used. If instead the value of refentry.version.profile.enabled is zero (the default), then "hard coded" logic within the DocBook XSL stylesheets is used for gathering refentry "version" data. If you find that the default refentry metadata-gathering behavior is causing incorrect "version" data to show up in your output, then consider setting a non-zero value for refentry.version.profile.enabled and adjusting the value of refentry.version.profile to cause correct data to be gathered. Note that the terms "source" and "version" have special meanings in this context. For details, see the documentation for the refentry.version.profile parameter. Namerefentry.manual.fallback.profile — Specifies profile of "fallback" for refentry "manual" data Synopsis<xsl:param name="refentry.manual.fallback.profile"> refmeta/refmiscinfo[not(@class = -'date')][1]/node()</xsl:param> Description The value of refentry.manual.fallback.profile is a string representing an XPath expression. It is evaluated at run-time and used only if no "manual" data can be found by other means (that is, either using the refentry metadata-gathering logic "hard coded" in the stylesheets, or the value of refentry.manual.profile, if it is enabled). ImportantDepending on which XSLT engine you run, either the EXSLT dyn:evaluate extension function (for xsltproc or Xalan) or saxon:evaluate extension function (for Saxon) are used to dynamically evaluate the value of refentry.manual.fallback.profile at run-time. If you don't use xsltproc, Saxon, Xalan -- or some other XSLT engine that supports dyn:evaluate -- you must manually disable fallback processing by setting an empty value for the refentry.manual.fallback.profile parameter. Namerefentry.source.fallback.profile — Specifies profile of "fallback" for refentry "source" data Synopsis<xsl:param name="refentry.source.fallbac refmeta/refmiscinfo[not(@class = 'date')][1]/node()</xsl:param> Description The value of refentry.source.fallback.profile is a string representing an XPath expression. It is evaluated at run-time and used only if no "source" data can be found by other means (that is, either using the refentry metadata-gathering logic "hard coded" in the stylesheets, or the value of the refentry.source.name.profile and refentry.version.profile parameters, if those are enabled). ImportantDepending on which XSLT engine you run, either the EXSLT dyn:evaluate extension function (for xsltproc or Xalan) or saxon:evaluate extension function (for Saxon) are used to dynamically evaluate the value of refentry.source.fallback.profile at run-time. If you don't use xsltproc, Saxon, Xalan -- or some other XSLT engine that supports dyn:evaluate -- you must manually disable fallback processing by setting an empty value for the refentry.source.fallback.profile parameter. Page header/footerPage header/footerPage header/footerTable of Contentsman.th.extra1.suppressman.th.extra2.suppressman.th.extra3.suppressman.th.title.max.lengthman.th.extra2.max.lengthman.th.e — Suppress extra1 part of header/footer? Synopsis<xsl:param name="man.th.extra1.suppress">0</xsl:param> Description If the value of man.th.extra1.suppress is non-zero, then the extra1 part of the .TH title line header/footer is suppressed. The content of the extra1 field is almost always displayed in the center footer of the page and is, universally, a date. Nameman.th.extra2.suppress — Suppress extra2 part of header/footer? Synopsis<xsl:param name="man.th.extra2.suppress">0</xsl:param> Description If the value of man.th.extra2.suppress is non-zero, then the extra2 part of the .TH title line header/footer is suppressed. The content of the extra2 field is usually displayed in the left footer of the page and is typically "source" data, often in the form Name Version; for example, "GTK+ 1.2" (from the gtk-options(7) man page). NoteYou can use the refentry.source.name.suppress and refentry.version.suppress parameters to independently suppress the Name and Version parts of the extra2 field. Nameman.th.extra3.suppress — Suppress extra3 part of header/footer? Synopsis<xsl:param name="man.th.extra3.suppress">0</xsl:param> Description If the value of man.th.extra3.suppress is non-zero, then the extra3 part of the .TH title line header/footer is suppressed. The content of the extra3 field is usually displayed in the middle header of the page and is typically a "manual name"; for example, "GTK+ User's Manual" (from the gtk-options(7) man page). Nameman.th.title.max.length — Maximum

PDF BY DBLATEX

length of title in header/footer Synopsis<xsl:param name="man.th.title.max.length">20</xsl:param> Description Specifies the maximum permitted length of the title part of the man-page .TH title line header/footer. If the title exceeds the maxiumum specified, it is truncated down to the maximum permitted length. Details Every man page generated using the DocBook stylesheets has a title line, specified using the TH roff macro. Within that title line, there is always, at a minimum, a title, followed by a section value (representing a man "section" -- usually just a number). The title and section are displayed, together, in the visible header of each page. Where in the header they are displayed depends on OS the man page is viewed on, and on what version of nroff/groff/man is used for viewing the page. But, at a minimum and across all systems, the title and section are displayed on the right-hand column of the header. On many systems -- those with a modern groff, including Linux systems -- they are displayed twice: both in the left and right columns of the header. So if the length of the title exceeds a certain percentage of the column width in which the page is viewed, the left and right titles can end up overlapping, making them unreadable, or breaking to another line, which doesn't look particularly good. So the stylesheets provide the man.th.title.max.length parameter as a means for truncating titles that exceed the maximum length that can be viewing properly in a page header. The default value is reasonable but somewhat arbitrary. If you have pages with long titles, you may want to experiment with changing the value in order to achieve the correct aesthetic results. Nameman.th.extra2.max.length — Maximum length of extra2 in header/footer Synopsis<xsl:param name="man.th.extra2.max.length">30</xsl:param> Description Specifies the maximum permitted length of the extra2 part of the man-page part of the .TH title line header/footer. If the extra2 content exceeds the maxiumum specified, it is truncated down to the maximum permitted length. The content of the extra2 field is usually displayed in the left footer of the page and is typically "source" data indicating the software system or product that the item documented in the man page belongs to, often in the form Name Version; for example, "GTK+ 1.2" (from the gtk-options(7) man page). The default value for this parameter is reasonable but somewhat arbitrary. If you are processing pages with long "source" information, you may want to experiment with changing the value in order to achieve the correct aesthetic results. Nameman.th.extra3.max.length — Maximum length of extra3 in header/footer Synopsis<xsl:param name="man.th.extra3.max.length">30</xsl:param> Description Specifies the maximum permitted length of the extra3 part of the man-page .TH title line header/footer. If the extra3 content exceeds the maxiumum specified, it is truncated down to the maximum permitted length. The content of the extra3 field is usually displayed in the middle header of the page and is typically a "manual name"; for example, "GTK+ User's Manual" (from the gtk-options(7) man page). The default value for this parameter is reasonable but somewhat arbitrary. If you are processing pages with long "manual names" -- or especially if you are processing pages that have both long "title" parts (command/function, etc. names) and long manual names -- you may want to experiment with changing the value in order to achieve the correct aesthetic results. OutputOutputOutputTable of Contentsman.output.manifest.enabledman.output.manifest.filenameman.output.in.separate.dirman.output.lang.in.name.enabledman.output.base. — Generate a manifest file? Synopsis<xsl:param name="man.output.manifest.enabled" select="0"></xsl:param> Description If non-zero, a list of filenames for man pages generated by the stylesheet transformation is written to the file named by the man.output.manifest.filename parameter. Nameman.output.manifest.filename — Name of manifest file Synopsis<xsl:param name="man.output.manifest.filename">MAN.MANIFEST</xsl:param> Description The man.output.manifest.filename parameter specifies the name of the file to which the manpages manifest file is written (if the value of the man.output.manifest.enabled parameter is non-zero). Nameman.output.in.separate.dir — Output man-page files in separate output directory? Synopsis<xsl:param name="man.output.in.separate.dir" select="0"></xsl:param> Description If the value of man.output.in.separate.dir parameter is non-zero, man-page files are output in a separate directory, specified by the man.output.base.dir parameter; otherwise, if the value of man.output.in.separate.dir is zero, man-page files are not output in a separate directory. Nameman.output.lang.in.name.enabled — Include $LANG value in man-page filename/pathname? Synopsis<xsl:param name="man.output.lang.in.name.enabled" select="0"></xsl:param> Description The man.output.lang.in.name.enabled parameter specifies whether a $lang value is included in man-page filenames and pathnames. If the value of man.output.lang.in.name.enabled is non-zero, man-page files are output with the $lang value included in their filenames or pathnames as follows; •if man.output.subdirs.enabled is non-zero, each file is output to, e.g., a man/$lang/man8/foo.8 pathname •if man.output.subdirs.enabled is zero, each file is output with a foo.$lang.8 filename Nameman.output.base.dir — Specifies separate output directory Synopsis<xsl:param name="man.output.base.dir">> Description The man.output.base.dir parameter specifies the base directory into which man-page files are output. The man.output.subdirs parameter controls whether the files are output in subdirectories within the base directory. NoteThe values of the man.output.base.dir and man.output.subdirs.enabled parameters are used only if the value of man.output.in.separate.dir parameter is non-zero. If the value of the man.output.in.separate.dir is zero, man-page files are not output in a separate directory. Nameman.output.subdirs.enabled — Output man-page files in subdirectories within base output directory? Synopsis<xsl:param name="man.output.subdirs.enabled" select="1"></xsl:param> Description The man.output.subdirs.enabled parameter controls whether man-pages files are output in subdirectories within the base directory specified by the directory specified by the man.output.base.dir parameter. NoteThe values of the man.output.base.dir and man.output.subdirs.enabled parameters are used only if the value of man.output.in.separate.dir parameter is non-zero. If the value of the man.output.in.separate.dir is zero, man-page files are not output in a separate directory. Nameman.output.quietly — Suppress filename messages emitted when generating output? Synopsis<xsl:param name="man.output.quietly" select="0"></xsl:param> Description If zero (the default), for each man-page file created, a message with the name of the file is emitted. If non-zero, the files are output "quietly" -- that is, the filename messages are sup-

pressed. TipIf you are processing a large amount of refentry content, you may be able to speed up processing significantly by setting a non-zero value for man.output.quietly. Nameman.output.encoding — Encoding used for man-page output Synopsis<xsl:param name="man.output.encoding">UTF-8</xsl:param> Description This parameter specifies the encoding to use for files generated by the manpages stylesheet. Not all processors support specification of this parameter. ImportantIf the value of the man.charmap.enabled parameter is non-zero (the default), keeping the man.output.encoding parameter at its default value (UTF-8) or setting it to UTF-16 does not cause your man pages to be output in raw UTF-8 or UTF-16 -- because any Unicode characters for which matches are found in the enabled character map will be replaced with roff escape sequences before the final man-page files are generated. So if you want to generate "real" UTF-8 man pages, without any character substitution being performed on your content, you need to set man.charmap.enabled to zero (which will completely disable character-map processing). You may also need to set man.charmap.enabled to zero if you want to output man pages in an encoding other than UTF-8 or UTF-16. Character-map processing is based on Unicode character values and may not work with other output encodings. Nameman.output.better.ps.enabled — Enable enhanced print/PostScript output? Synopsis<xsl:param name="man.output.better.ps.enabled">0</xsl:param> Description If the value of the man.output.better.ps.enabled parameter is non-zero, certain markup is embedded in each generated man page such that PostScript output from the man -Tps command for that page will include a number of enhancements designed to improve the quality of that output. If man.output.better.ps.enabled is zero (the default), no such markup is embedded in generated man pages, and no enhancements are included in the PostScript output generated from those man pages by the man -Tps command. WarningThe enhancements provided by this parameter rely on features that are specific to groff (GNU troff) and that are not part of “classic” AT&T troff or any of its derivatives. Therefore, any man pages you generate with this parameter enabled will be readable only on systems on which the groff (GNU troff) program is installed, such as GNU/Linux systems. The pages will not not be readable on systems on with the classic troff (AT&T troff) command is installed. The value of this parameter only affects PostScript output generated from the man command. It has no effect on output generated using the FO backend. TipYou can generate PostScript output for any man page by running the following command: man FOO -Tps > FOO.ps You can then generate PDF output by running the following command: ps2pdf FOO.ps OtherOtherOtherTable of Contentsman.table.footnotes.dividerman.subheading.divider.enabledman.subheading.dividerNameman.table.footn — Specifies divider string that appears before table footnotes Synopsis<xsl:param name="man.table.footnotes.divider">----</xsl:param> Description In each table that contains footenotes, the string specified by the man.table.footnotes.divider parameter is output before the list of footnotes for the table. Nameman.subheading.divider.enabled — Add divider comment to roff source before/after subheadings? Synopsis<xsl:param name="man.subheading.divider.enabled">0</xsl:param> Description If the value of the man.subheading.divider.enabled parameter is non-zero, the contents of the man.subheading.divider parameter are used to add a "divider" before and after subheadings in the roff output. The divider is not visisble in the rendered man page; it is added as a comment, in the source, simply for the purpose of increasing reability of the source. If man.subheading.divider.enabled is zero (the default), the subheading divider is suppressed. Nameman.subheading.divider — Specifies string to use as divider comment before/after subheadings Synopsis<xsl:param name="man.subheading.divider">========== Description If the value of the man.subheading.divider.enabled parameter is non-zero, the contents of the man.subheading.divider parameter are used to add a "divider" before and after subheadings in the roff output. The divider is not visisble in the rendered man page; it is added as a comment, in the source, simply for the purpose of increasing reability of the source. If man.subheading.divider.enabled is zero (the default), the subheading divider is suppressed. Part IV. Roundtrip Parameter Reference This is reference documentation for all user-configurable parameters in the DocBook “Roundtrip” Stylesheets (for transforming DocBook to WordML, OpenDocument, and Apple Pages, and for converting from those formats back to DocBook). wordml.templatewordml.templateNamewordml.template — Specify the template WordML document Synopsis<xsl:param name="wordn Description The wordml.template parameter specifies a WordML document to use as a template for the generated document. The template document is used to define the (extensive) headers for the generated document, in particular the paragraph and character styles that are used to format the various elements. Any content in the template document is ignored. A template document is used in order to allow maintenance of the paragraph and character styles to be done using Word itself, rather than these XSL stylesheets. pages.templatepages.templateNamepages.template — Specify the template Pages document Synopsis<xsl:param name="pages.template">></xsl:param> Description The pages.template parameter specifies a Pages (the Apple word processing application) document to use as a template for the generated document. The template document is used to define the (extensive) headers for the generated document, in particular the paragraph and character styles that are used to format the various elements. Any content in the template document is ignored. A template document is used in order to allow maintenance of the paragraph and character styles to be done using Pages itself, rather than these XSL stylesheets. Part V. Slides Parameter Reference This is reference documentation for all user-configurable parameters in the DocBook XSL Slides stylesheets (for generating HTML and PDF slide presentations). NoteThe Slides stylesheet for HTML output is a customization layer of the DocBook XSL HTML stylesheet; the Slides stylesheet for FO output is a customization layer of the DocBook XSL FO stylesheet. Therefore, in addition to the slides-specific parameters listed in this section, you can also use a number of HTML stylesheet parameters11 ../html and FO stylesheet parameters22 ../fo to control Slides output. Slides Parameter ReferenceSlides Parameter ReferenceTable of ContentsLXV. XHTML: General Paramsdisable.collapsibledisable.incrementalgenerate.copyrightgenerate.foilgroup.numbered.tocgenerate.foilgroup.tocgenerate.handoutnotesgene

PDF BY DBLATEX

XHTML: S5 Paramss5.controlss5.defaultviews5.opera.csss5.outline.csss5.path.prefixs5.print.csss5.slides.csss5.slides.jsLXVII. XHTML: Slidy Paramsslidy.durationslidy.path.prefixslidy.slidy.cssslidy.slidy.jsslidy.user.cssLXVIII. FO: General Paramsfoil.title.maste FO: Property Setsfoil.header.propertiesfoil.master.propertiesfoil.page-sequence.propertiesfoil.propertiesfoil.region-after.propertiesfoil.r before.propertiesfoil.region-body.propertiesfoil.subtitle.propertiesfoil.title.propertiesfoil.footer.propertieshandoutnotes.propertiesslides. body.propertiesslides.titlepage.corpauthor.propertiesslides.titlepage.title.propertiesslides.titlepage.subtitle.propertiesslides.titlepage.auth General ParamsXHTML: General ParamsXHTML: General ParamsTable of Contentsdisable.collapsibledisable.incrementalgenerate.cop — Specifies whether collapsible rendering is enabled Synopsis <xsl:param name="disable.collapsible">0</xsl:param> - Description This parameter specifies whether elements marked as collapsible are generated as such in the output document. Namedisable.incremental — Specifies whether incremental rendering is enabled Synopsis <xsl:param name="disable.incremental">0</ Description This parameter specifies whether elements marked as incremental are generated as such in the output document. Namegenerate.copyright — Specifies whether copyright is generated Synopsis <xsl:param name="generate.copyright">1</xsl:param> Description This parameter specifies whether the copyright info is generated in the footer area. Namegenerate.foilgroup.numbered.toc — Specifies whether foilgroups have a numbered TOC Synopsis <xsl:param name="generate.foilgroup.numbered.toc">1</xsl:param> Description If TOC generation is turned on, this parameter specifies whether foilgroups have a numbered TOC. If disabled, TOC items will be bulleted, not numbered. Namegenerate.foilgroup.toc — Specifies whether foilgroups have a TOC Synopsis <xsl:param name="generate.foilgroup.toc">1</xsl:param> Description This parameter specifies whether foilgroups will contain a table of contents of the included foils. Namegenerate.handoutnotes — Specifies whether handoutnotes are generated Synopsis <xsl:param name="generate.handoutnotes">0</xsl:param> Description This parameter specifies whether handoutnotes shall be generated to the output. Namegenerate.pubdate — Specifies whether the pubdate is generated Synopsis <xsl:param name="generate.pubdate">1</xsl:param> Description This parameter specifies whether the publication date is generated in the footer area. Namegenerate.speakernotes — Specifies whether speakernotes are generated Synopsis <xsl:param name="generate.speakernotes">0</xsl:param> Description This parameter specifies whether speakernotes shall be generated to the output. Namegenerate.titlepage — Specifies whether titlepage is generated Synopsis - <xsl:param name="generate.titlepage">1</xsl:param> Description This parameter specifies whether titlepage is generated for the presentation. Namemml.embedding.mode — Specifies how inline MathML is processed Synopsis <xsl:param - name="mml.embedding.mode">inline</xsl:param> Description This parameter specifies how inline MathML formulas are embedded into the output document. inlineContent is copied over inline with its namespace. objectContent is extracted into an externel file and referenced by an object element. imageContent is extracted into an externel file and referenced by an img element. linkContent is extracted into an externel file and referenced by an a element. iframeContent is extracted into an externel file and referenced by an iframe element. embedContent is extracted into an externel file and referenced by an embed element. Namesvg.embedding.mode — Specifies how inline SVG is processed Synopsis <xsl:param name="svg.embedding.mode">object</xsl: Description This parameter specifies how inline SVG graphics are embedded into the output document. inlineContent is copied over inline with its namespace. objectContent is extracted into an externel file and referenced by an object element. imageContent is extracted into an externel file and referenced by an img element. linkContent is extracted into an externel file and referenced by an a element. iframeContent is extracted into an externel file and referenced by an iframe element. embedContent is extracted into an externel file and referenced by an embed element. Nameuser.css — Specifies the path to user-supplied CSS Synopsis - <xsl:param name="user.css">user.css</xsl:param> Description This parameter specifies the path from where the CSS styling is read. This file can be used to add additional styling to the slides. Namewrap.slidecontent — Specifies whether the foil content is wrapped into a div Synopsis <xsl:param name="wrap.slidecontent">0</xsl:param> Description This parameter specifies whether the foil content is wrapped into a div so that additional styling can be applied. XHTML: S5 ParamsXHTML: S5 ParamsXHTML: S5 ParamsTable of Contentss5.controlss5.defaultviews5.opera.csss5.outline.csss5.path.prefixs5.print.csss5.slides.csss5.slides — Specifies whether S5 controls are visible Synopsis <xsl:param name="s5.controls">0</xsl:param> Description This parameter specifies whether S5 navigation controls are visible by default. Names5.defaultview — Specifies the default S5 view Synopsis <xsl:param name="s5.defaultview">slideshow</xsl:param> Description This parameter specifies, which is the default view in the generated S5 presentation. Names5.opera.css — Specifies the name of the S5 Opera-specific CSS file Synopsis <xsl:param name="s5.opera.css">opera.css</xsl:param> Description This parameter specifies the name of the S5 Opera-specific CSS file. Names5.outline.css — Specifies the name of the S5 outline CSS file Synopsis <xsl:param - name="s5.outline.css">outline.css</xsl:param> Description This parameter specifies the name of the S5 outline CSS file. Names5.path.prefix — Specifies the path to S5 files Synopsis <xsl:param name="s5.path.prefix">files/s5/ui/default/</xsl:param> Description This parameter specifies the path where S5 CSS and JavaScript files reside. All the CSS and JavaScript paths will be generated relative to this directory. Names5.print.css — Specifies the name of the S5 print CSS file Synopsis - <xsl:param name="s5.print.css">print.css</xsl:param> Description This parameter specifies the name of the S5 print CSS file. Names5.slides.css — Specifies the name of the S5 slides CSS file Synopsis <xsl:param name="s5.slides.css">slides.css</xsl:param> Description This parameter specifies the name of the S5 slides CSS file. Names5.slides.js — Specifies the name of the S5 slides JavaScript file Synopsis <xsl:param name="s5.slides.js">slides.js</xsl:param> Description This parameter specifies the name of the S5 slides JavaScript file. XHTML: Slidy ParamsXHTML: Slidy ParamsXHTML: Slidy ParamsTable of Contentsslidy.durationslidy.path.prefixslidy.slidy.cssslidy.slidy.jsslidy.user.cssNameslidy.duration — Specifies the duration of

PDF BY DBLATEX

the presentation Synopsis <xsl:param name="slidy.duration">0</xsl:param> Description This parameter specifies the duration of the presentation in minutes. A JavaScript clock will count down to help the speaker in not running out of time. Can be disabled if set to 0. Nameslidy.path.prefix — Specifies the path to Slidy files Synopsis <xsl:param name="slidy.path.prefix">files/slidy/</xsl:para Description This parameter specifies the path where Slidy CSS and JavaScript files reside. All the CSS and JavaScript paths will be generated relative to this directory. Nameslidy.slidy.css — Specifies the name of the main Slidy CSS file Synopsis - <xsl:param name="slidy.slidy.css">styles/slidy.css</xsl:param> Description This parameter specifies the name of the main Slidy CSS file. Nameslidy.slidy.js — Specifies the name of the Slidy JavaScript file Synopsis <xsl:param name="slidy.slidy.js">scripts/s Description This parameter specifies the name of the Slidy JavaScript file. Nameslidy.user.css — Specifies the name of the Slidy user CSS file Synopsis <xsl:param name="slidy.user.css">styles/w3c-blue.css</xsl:param> Description This parameter specifies the name of the Slidy user CSS file. FO: General ParamsFO: General ParamsFO: General ParamsTable of Contentsfoil.title.masterfoil.title.sizegenerate.copyrightgenerate.foilgroup.numbered.tocgenerate.foilgroup.tocgenerate.handoutnotesgenerate.pa — Specifies unitless font size to use for foil titles Synopsis<xsl:param name="foil.title.master">36</xsl:param> <!-- Inconsistant - use of point size? --> Description Specifies a unitless font size to use for foil titles; used in combination with the foil.title.size parameter. Namefoil.title.size — Specifies font size to use for foil titles, including units Synopsis <xsl:param name="foil.title.size">

<xsl:value-of select="$foil.title.master"></xsl:value-of><xsl:text>pt</xsl:text> </xsl:param> Description This parameter combines the value of the foil.title.master parameter with a unit specification. The default unit is pt (points). Namegenerate.copyright — Specifies whether copyright is generated Synopsis <xsl:param name="generate.copyright">1</xsl:param> Description This parameter specifies whether the copyright info is generated in the footer area. Namegenerate.foilgroup.numbered.toc — Specifies whether foilgroups have a numbered TOC Synopsis <xsl:param name="generate.foilgroup.numbered.toc">1</xsl:param> Description If TOC generation is turned on, this parameter specifies whether foilgroups have a numbered TOC. If disabled, TOC items will be bulleted, not numbered. Namegenerate.foilgroup.toc — Specifies whether foilgroups have a TOC Synopsis <xsl:param name="generate.foilgroup.toc">1</xsl:param> Description This parameter specifies whether foilgroups will contain a table of contents of the included foils. Namegenerate.handoutnotes — Specifies whether handoutnotes are generated Synopsis <xsl:param name="generate.handoutnotes">0</xsl:param> Description This parameter specifies whether handoutnotes shall be generated to the output. Namegenerate.page.number — Specifies whether page numbers are generated Synopsis <xsl:param name="generate.page.number">compact</xsl:param> Description This parameter specifies how page numbers are generated in the footer area. noNo page numbers generated at all. fullCurrent page number, a slash and the total number of pages compactCurrent page number only noNo page numbers generated at all. Namegenerate.pubdate — Specifies whether the pubdate is generated Synopsis <xsl:param name="generate.pubdate">1</xsl:param> Description This parameter specifies whether the publication date is generated in the footer area. Namegenerate.speakernotes — Specifies whether speakernotes are generated Synopsis <xsl:param name="generate.speakernotes">0</xsl:param> Description This parameter specifies whether speakernotes shall be generated to the output. Namegenerate.titlepage — Specifies whether titlepage is generated Synopsis <xsl:param name="generate.titlepage">1</xsl:param> Description This parameter specifies whether titlepage is generated for the presentation. Namemml.embedding.mode — Specifies how inline MathML is processed Synopsis <xsl:param name="mml.embedding.mode">external-graphic</xsl:param> Description This parameter specifies how inline MathML formulas are embedded into the output document. inlineContent is copied over inline with its namespace. external-graphicContent is extracted into an externel file and referenced by an external-graphic element. instream-foreign-objectContent is copied over with its namespace inside an instream-foreign-object element. Nameslide.font.family — Specifies font family to use for slide bodies Synopsis<xsl:param name="slide.font.family">Helvetica</xsl:param> Description Specifies the font family to use for slides bodies. Nameslide.title.font.family — Specifies font family to use for slide titles Synopsis<xsl:param name="slide.title.font.family">Helvetica</xsl:param> Description Specifies the font family to use for slides titles. Namesvg.embedding.mode — Specifies how inline SVG is processed Synopsis <xsl:param name="svg.embedding.mode">instream-foreign-object</xsl:param> Description This parameter specifies how inline SVG graphics are embedded into the output document. inlineContent is copied over inline with its namespace. external-graphicContent is extracted into an externel file and referenced by an external-graphic element. instream-foreign-objectContent is copied over with its namespace inside an instream-foreign-object element. FO: Property SetsFO: Property SetsFO: Property SetsTable of Contentsfoil.header.propertiesfoil.master.propertie sequence.propertiesfoil.propertiesfoil.region-after.propertiesfoil.region-before.propertiesfoil.region-body.propertiesfoil.subtitle.properti body.propertiesslides.titlepage.corpauthor.propertiesslides.titlepage.title.propertiesslides.titlepage.subtitle.propertiesslides.titlepage.auth — Specifies properties for foil header area Synopsis <xsl:attribute-set name="foil.header.properties">

<xsl:attribute name="background-color">white</xsl:attribute>

<xsl:attribute name="color">black</xsl:attribute>

<xsl:attribute name="font-weight">bold</xsl:attribute>

<xsl:attribute name="text-align">center</xsl:attribute>

<xsl:attribute name="font-family">

<xsl:value-of select="$slide.title.font.family"></xsl:value-of> </xsl:attribute>

<xsl:attribute name="space-after">12pt</xsl:attribute> </xsl:attribute-set> Description This parameter specifies properties for the foil header area. Namefoil.master.properties — Specifies properties for foil master Synopsis <xsl:attribute-set name="foil.master.properties

<xsl:attribute name="page-width">

<xsl:value-of select="$page.width"></xsl:value-of> </xsl:attribute>

<xsl:attribute name="page-height">

<xsl:value-of select="$page.height"></xsl:value-of> </xsl:attribute>

<xsl:attribute name="margin-top">

<xsl:value-of select="$page.margin.top"></xsl:value-of> </xsl:attribute>

PDF BY DBLATEX

<xsl:attribute name="margin-bottom"> <xsl:value-of select="$page.margin.bottom"></xsl:value-of> </xsl:attribute> <xsl:attribute name="margin-left"> <xsl:value-of select="$page.margin.inner"></xsl:value-of> </xsl:attribute> <xsl:attribute name="margin-right"> <xsl:value-of select="$page.margin.outer"></xsl:value-of> </xsl:attribute> - </xsl:attribute-set> Description This parameter specifies properties for the foil master. Namefoil.page-sequence.properties — Specifies properties for foil page-sequence Synopsis <xsl:attribute-set name="foil.page-sequence.properties"> - <xsl:attribute name="hyphenate"> <xsl:value-of select="$hyphenate"></xsl:value-of> </xsl:attribute> </xsl:attribute-set> Description This parameter specifies properties for foil page-sequence. Namefoil.properties — Specifies properties for all foils Synopsis <xsl:attribute-set name="foil.properties"> <xsl:attribute name="font-family"> <xsl:value-of - select="$slide.font.family"></xsl:value-of> </xsl:attribute> <xsl:attribute name="margin-{$direction.align.start}">1in</xsl:attrib <xsl:attribute name="margin-{$direction.align.end}">1in</xsl:attribute> <xsl:attribute name="font-size"> - <xsl:value-of select="$body.font.size"></xsl:value-of> </xsl:attribute> </xsl:attribute-set> Description This parameter specifies properties that are applied to all foils. Namefoil.region-after.properties — Specifies properties for foil region-after Synopsis <xsl:attribute-set name="foil.region-after.properties"> <xsl:attribute name="extent"> <xsl:value-of - select="$region.after.extent"></xsl:value-of> </xsl:attribute> <xsl:attribute name="display-align">after</xsl:attribute> </xsl:attribute-set> Description This parameter specifies properties for the foil region-after. Namefoil.region-before.properties — Specifies properties for foil region-before Synopsis <xsl:attribute-set name="foil.region-before.properties"> <xsl:attribute - name="extent"> <xsl:value-of select="$region.before.extent"></xsl:value-of> </xsl:attribute> <xsl:attribute - name="display-align"> <xsl:value-of select="'before'"></xsl:value-of> </xsl:attribute> </xsl:attribute-set> Description This parameter specifies properties for the foil region-before. Namefoil.region-body.properties — Specifies properties for foil region-body Synopsis <xsl:attribute-set name="foil.region-body.properties"> <xsl:attribute name="margin-bottom"> <xsl:value-of select="$body.margin.bottom"></xsl:value-of> </xsl:attribute> <xsl:attribute name="margin-top"> <xsl:value-of select="$body.margin.top"></xsl:value-of> </xsl:attribute> <xsl:attribute name="column-count"> <xsl:value-of select="$column.count.body"></xsl:value-of> </xsl:attribute> </xsl:attribute-set> Description This parameter specifies properties for the foil region-body. Namefoil.subtitle.properties — Specifies properties for all foil subtitles Synopsis <xsl:attribute-set name="foil.subtitle.properties"> <xsl:attribute name="font-family"> <xsl:value-of - select="$slide.title.font.family"></xsl:value-of> </xsl:attribute> <xsl:attribute name="text-align">center</xsl:attribute> <xsl:attribute name="font-size"> <xsl:value-of select="$foil.title.master * 0.8"></xsl:value-of><xsl:text>pt</xsl:text> </xsl:attribute> <xsl:attribute name="space-after">12pt</xsl:attribute> </xsl:attribute-set> Description This parameter specifies properties that are applied to all foil subtitles. Namefoil.title.properties — Specifies properties for foil title Synopsis <xsl:attribute-set name="foil.title.properties"> <xsl:attribute name="font-size"> <xsl:value-of - select="$foil.title.size"></xsl:value-of> </xsl:attribute> </xsl:attribute-set> Description This parameter specifies properties for the foil title. Namefoil.footer.properties — Specifies properties for slides footer Synopsis <xsl:attribute-set - name="foil.footer.properties"></xsl:attribute-set> Description This parameter specifies properties for the foil footer. Namefootnote.properties — Properties applied to handoutnotes Synopsis<xsl:attribute-set name="handoutnotes.properties"></xsl:attribute-set> Description This attribute set is applied to handoutnotes. Nameslides.properties — Specifies properties for all slides Synopsis <xsl:attribute-set name="slides.properties"> <xsl:attribute name="font-family"> <xsl:value-of select="$slide.font.family">< of> </xsl:attribute> </xsl:attribute-set> Description This parameter specifies properties that are applied to all slides. Nameslides.titlepage.master.properties — Specifies properties for slides titlepage master Synopsis <xsl:attribute-set - name="slides.titlepage.master.properties"> <xsl:attribute name="page-width"> <xsl:value-of select="$page.width"></xsl:value-of> </xsl:attribute> <xsl:attribute name="page-height"> <xsl:value-of select="$page.height"></xsl:value-of> </xsl:attribute> <xsl:attribute name="margin-top"> <xsl:value-of select="$page.margin.top"></xsl:value-of> - </xsl:attribute> <xsl:attribute name="margin-bottom"> <xsl:value-of select="$page.margin.bottom"></xsl:value-of> </xsl:attribute> <xsl:attribute name="margin-left"> <xsl:value-of select="$page.margin.inner"></xsl:value-of> </xsl:attribute> <xsl:attribute name="margin-right"> <xsl:value-of select="$page.margin.outer"></xsl:value-of> </xsl:attribute> </xsl:attribute-set> Description This parameter specifies properties for the slides titlepage master. Nameslides.titlepage.region-body.properties — Specifies properties for slides titlepage region-body Synopsis <xsl:attribute-set name="slides.titlepage.region-body.properties"> <xsl:attribute name="margin-bottom">0pt</xsl:attribute> <xsl:attribute - name="margin-top">0pt</xsl:attribute> <xsl:attribute name="column-count"> <xsl:value-of select="$column.count.body"></xsl of> </xsl:attribute> </xsl:attribute-set> Description This parameter specifies properties for the slides titlepage region-body. Nameslides.titlepage.corpauthor.properties — Specifies properties for slides titlepage title Synopsis <xsl:attribute-set - name="slides.titlepage.corpauthor.properties"> <xsl:attribute name="text-align">center</xsl:attribute> <xsl:attribute - name="space-after">1em</xsl:attribute> <xsl:attribute name="font-size">20.736pt</xsl:attribute> </xsl:attribute-set> Description This parameter specifies properties for the corpauthor on the default titlepage. Nameslides.titlepage.title.properties — Specifies properties for slides titlepage title Synopsis <xsl:attribute-set name="slides.titlepage.title.properties"> - <xsl:attribute name="text-align">center</xsl:attribute> <xsl:attribute name="space-after">1em</xsl:attribute> - <xsl:attribute name="padding-top">1.5in</xsl:attribute> <xsl:attribute name="keep-with-next">always</xsl:attribute> - <xsl:attribute name="font-size"> <xsl:value-of select="$foil.title.size"></xsl:value-of> </xsl:attribute> <xsl:attribute -

PDF BY DBLATEX

name="font-weight">bold</xsl:attribute> <xsl:attribute name="font-family"> <xsl:value-of select="$slide.title.font.family"></xsl:va of> </xsl:attribute> </xsl:attribute-set> Description This parameter specifies properties for the title on the default titlepage. Nameslides.titlepage.subtitle.properties — Specifies properties for slides titlepage title Synopsis <xsl:attribute-set name="slides.titlepage.subtitle.properties">> <xsl:attribute name="text-align">center</xsl:attribute> <xsl:attribute - name="space-after">1em</xsl:attribute> <xsl:attribute name="font-family"> <xsl:value-of select="$slide.title.font.family"></xsl:val of> </xsl:attribute> </xsl:attribute-set> Description This parameter specifies properties for the subtitle on the default titlepage. Nameslides.titlepage.author.properties — Specifies properties for slides titlepage title Synopsis <xsl:attribute-set - name="slides.titlepage.author.properties">> <xsl:attribute name="text-align">center</xsl:attribute> <xsl:attribute - name="space-after">1em</xsl:attribute> <xsl:attribute name="font-size">20.736pt</xsl:attribute> </xsl:attribute-set> Description This parameter specifies properties for the author on the default titlepage. Nameslides.titlepage.pubdate.properties — Specifies properties for slides titlepage title Synopsis <xsl:attribute-set name="slides.titlepage.pubdate.properties">> - <xsl:attribute name="text-align">center</xsl:attribute> <xsl:attribute name="space-after">1em</xsl:attribute> - <xsl:attribute name="font-size">17.28pt</xsl:attribute> </xsl:attribute-set> Description This parameter specifies properties for the pubdate on the default titlepage. Nameslides.titlepage.authorgroup.properties — Specifies properties for slides titlepage title Synopsis <xsl:attribute-set name="slides.titlepage.authorgroup.properties"></xsl:attribute-set> Description This parameter specifies properties for the authorgroup on the default titlepage. Namefootnote.properties — Properties applied to speakernotes Synopsis<xsl:attribute-set name="speakernotes.properties"></xsl:attribute-set> Description This attribute set is applied to speakernotes. Part VI. Website Parameter Reference This is reference documentation for all user-configurable parameters in the DocBook XSL Website stylesheet (for generating websites from DocBook XML sources). Note that the Website stylesheet is a customization layer of the DocBook XSL HTML stylesheet. Therefore, in addition to the Website-specific parameters listed in this section, you can also use a number of HTML stylesheet parameters11 ../html/ to control Website output. Website Parameter ReferenceWebsite Parameter ReferenceTable of ContentsLXX. General Parametersautolayout-filebody.attributescurrentpage.markerdry-runfeedback.hreffeedback.link.textfeedback.with.idsfilename-prefixfooter.hrheader.hroutput-rootrebuild-allsequential.linkssuppress.homepage.titletable.spacer.imageLXXI. Navigation Parametersbanner.before.navigationnavbgco ToC Parameterstoc.blank.graphictoc.blank.imagetoc.blank.texttoc.pointer.graphictoc.pointer.imagetoc.pointer.texttoc.spacer.graphictoc. ParametersGeneral ParametersGeneral ParametersTable of Contentsautolayout-filebody.attributescurrentpage.markerdry-runfeedback.hr prefixfooter.hrheader.hroutput-rootrebuild-allsequential.linkssuppress.homepage.titletable.spacer.imageNameautolayout-file — Identifies the autolayout.xml file Synopsis<xsl:param name="autolayout-file">autolayout.xml</xsl:param> Description When the source pages are spread over several directories, this parameter can be set (for example, from the command line of a batch-mode XSLT processor) to indicate the location of the autolayout.xml file. FIXME: for browser-based use, there needs to be a PI for this... Namebody.attributes — DEPRECATED Synopsis<xsl:attribute-set name="body.attributes">> <xsl:attribute - name="bgcolor">white</xsl:attribute> <xsl:attribute name="text">black</xsl:attribute> <xsl:attribute name="link">#0000FF</xsl:att <xsl:attribute name="vlink">#840084</xsl:attribute> <xsl:attribute name="alink">#0000FF</xsl:attribute> </xsl:attribute-set> Description DEPRECATED Namecurrentpage.marker — The text symbol used to mark the current page Synopsis<xsl:param - name="currentpage.marker">@</xsl:param> Description Character to use as identifying the current page in Namedry-run — Indicates that no files should be produced Synopsis <xsl:param name="dry-run" select="0"></xsl:param> Description When using the XSLT processor to manage dependencies and construct the website, this parameter can be used to suppress the generation of new and updated files. Effectively, this allows you to see what the stylesheet would do, without actually making any changes. Only applies when XSLT-based chunking is being used. Namefeedback.href — HREF (URI) for feedback link Synopsis<xsl:param name="feedback.href"></xsl:param> Description The feedback.href value is used as the value for the href attribute on the feedback link. If feedback.href is empty, no feedback link is generated. Namefeedback.link.text — The text of the feedback link Synopsis<xsl:param name="feedback.link.text">Feedback</xsl:param> Description The contents of this variable is used as the text of the feedback link if feedback.href is not empty. If feedback.href is empty, no feedback link is generated. Namefeedback.with.ids — Toggle use of IDs in feedback Synopsis<xsl:param name="feedback.with.ids" - select="0"></xsl:param> Description If feedback.with.ids is non-zero, the ID of the current page will be added to the feedback link. This can be used, for example, if the feedback.href is a CGI script. Namefilename-prefix — Prefix added to all filenames Synopsis<xsl:param name="filename-prefix"></xsl:param> Description To produce the “text-only” (that is, non-tabular) layout of a website simultaneously with the tabular layout, the filenames have to be distinguished. That’s accomplished by adding the filename-prefix to the front of each filename. Namefooter.hr — Toggle <HR> before footer Synopsis<xsl:param - name="footer.hr" select="1"></xsl:param> Description If non-zero, an <HR> is generated at the bottom of each web page, before the footer. Nameheader.hr — Toggle <HR> after header Synopsis<xsl:param name="header.hr" select="1"></xsl:param> Description If non-zero, an <HR> is generated at the bottom of each web page, before the footer. Nameoutput-root — Specifies the root directory of the website Synopsis<xsl:param name="output-root">.</xsl:param> Description When using the XSLT processor to manage dependencies and construct the website, this parameter can be used to indicate the root directory where the resulting pages are placed. Only applies when XSLT-based chunking is being used. Namerebuild-all — Indicates that all files should be produced Synopsis<xsl:param name="rebuild-all" select="0"></xsl:param> Description When using the XSLT processor to manage dependencies and construct the website, this parameter can be used to regenerate the whole website, updating even

PDF BY DBLATEX

pages that don't appear to need to be updated. The dependency extension only looks at the source documents. So if you change something in the stylesheet, for example, that has a global effect, you can use this parameter to force the stylesheet to rebuild the whole website. Only applies when XSLT-based chunking is being used. Namesequential.links — Make sequentional links? Synopsis<xsl:param name="sequential.links" select="0"></xsl:param> Description FIXME Namesuppress.homepage.title — Suppress title on homepage? Synopsis<xsl:param name="suppress.homepage.title" select="1"></xsl:param> Description FIXME:If non-zero, the title on the homepage is suppressed? Nametable.spacer.image — Invisible pixel for tabular accessibility Synopsis<xsl:param name="table.spacer.image">graphics/spacer.gif</xsl:param> Description This is the 1x1 pixel, transparent pixel used for the table trick11 http://diveintoaccessibility.org/day_10_presenting_your_main_content_first.html to increase the accessibility of the tabular website presentation. Navigation ParametersNavigation ParametersNavigation ParametersTable of Contentsbanner.before.navigationnavbgcolornavbodywidthnav.table.summarynavtocwidthtextbgcolorNamebanner.before.navigation — Put banner before navigation? Synopsis<xsl:param name="banner.before.navigation" select="1"></xsl:param> Description FIXME Namenavbgcolor — The background color of the navigation TOC Synopsis<xsl:param name="navbgcolor">#4080FF</xsl:para Description The background color of the navigation TOC. Only applies with the tabular presentation is being used. Namenavbodywidth — Specifies the width of the navigation table body Synopsis<xsl:param name="navbodywidth"></xsl:param> Description The width of the body column. Only applies with the tabular presentation is being used. Namenav.table.summary — HTML Table summary attribute value for navigation tables Synopsis<xsl:param name="nav.table.summary">Navigation</xsl:param> Description The value of this parameter is used as the value of the table summary attribute for the navigation table. Only applies with the tabular presentation is being used. Namenavtocwidth — Specifies the width of the navigation table TOC Synopsis<xsl:param name="navtocwidth">220</xsl:param> Description The width, in pixels, of the navigation column. Only applies with the tabular presentation is being used. Nametextbgcolor — The background color of the table body Synopsis<xsl:param name="textbgcolor">white</xsl:param> Description The background color of the table body. Only applies with the tabular presentation is being used. ToC ParametersToC ParametersToC ParametersTable of Contentstoc.blank.graphictoc.blank.imagetoc.blank.text — Use graphic for "blanks" in TOC? Synopsis<xsl:param name="toc.blank.graphic" select="1"></xsl:param> Description If non-zero, "blanks" in the the TOC will be accomplished with the graphic identified by toc.spacer.image. Only applies with the tabular presentation is being used. Nametoc.blank.image — The image for "blanks" in the TOC Synopsis<xsl:param name="toc.blank.image">graphics/blank.gif</xsl:param> Description If toc.blank.graphic is non-zero, this image will be used to for "blanks" in the TOC. Only applies with the tabular presentation is being used. Nametoc.blank.text — The text for "blanks" in the TOC Synopsis<xsl:param name="toc.blank.text"> </xsl:param> Description If toc.blank.graphic is zero, this text string will be used for "blanks" in the TOC. Only applies with the tabular presentation is being used. Nametoc.pointer.graphic — Use graphic for TOC pointer? Synopsis<xsl:param name="toc.pointer.graphic" select="1"></xsl:param> Description If non-zero, the "pointer" in the TOC will be displayed with the graphic identified by toc.pointer.image. Only applies with the tabular presentation is being used. Nametoc.pointer.image — The image for the "pointer" in the TOC Synopsis<xsl:param name="toc.pointer.image">graphics/arrow.gif</xsl:param> Description If toc.pointer.graphic is non-zero, this image will be used for the "pointer" in the TOC. Only applies with the tabular presentation is being used. Nametoc.pointer.text — The text for the "pointer" in the TOC Synopsis<xsl:param name="toc.pointer.text"> > </xsl:param> Description If toc.pointer.graphic is zero, this text string will be used to display the "pointer" in the TOC. Only applies with the tabular presentation is being used. Nametoc.spacer.graphic — Use graphic for TOC spacer? Synopsis<xsl:param name="toc.spacer.graphic" select="1"></xsl:param> Description If non-zero, the indentation in the TOC will be accomplished with the graphic identified by toc.spacer.image. Only applies with the tabular presentation is being used. Nametoc.spacer.image — The image for spacing the TOC Synopsis<xsl:param name="toc.spacer.image">graphics/blank.gif</xsl:param> Description If toc.spacer.graphic is non-zero, this image will be used to indent the TOC. Only applies with the tabular presentation is being used. Nametoc.spacer.text — The text for spacing the TOC Synopsis<xsl:param name="toc.spacer.text"> </xsl:param> Description If toc.spacer.graphic is zero, this text string will be used to indent the TOC. Only applies with the tabular presentation is being used. DocBook XSL Stylesheets User Reference: PIsDocBook XSL Stylesheets User Reference: PIsAbstractThis is generated reference documentation for all user-specifiable processing instructions in the DocBook XSL stylesheets. NoteYou add these PIs at particular points in a document to cause specific "exceptions" to formatting/output behavior. To make global changes in formatting/output behavior across an entire document, it's better to do it by setting an appropriate stylesheet parameter (if there is one). DocBook XSL Stylesheets User Reference: PIsDocBook XSL Stylesheets User Reference: PIsTable of ContentsI. HTML Processing Instruction Referencedbhtml_background-colordbhtml_bgcolordbhtml_cellpaddingdbhtml_cellspacingdbhtml_classdbhtml_dirdbhtml_filenamedb styledbhtml_img.src.pathdbhtml_label-widthdbhtml_linenumbering.everyNthdbhtml_linenumbering.separatordbhtml_linenumbering.w presentationdbhtml_list-widthdbhtml_row-heightdbhtml_startdbhtml_stop-chunkingdbhtml_table-summarydbhtml_table-widthdbhtml_ presentationdbhtml_term-separatordbhtml_term-widthdbhtml_tocdbcmdlistdbfunclistdbhtml-include_hrefdbhhII. FO Processing Instruction Referencedbfo_background-colordbfo_bgcolordbfo_float-typedbfo_funcsynopsis-styledbfo_glossary-presentationdbfo_glos presentationdbfo_glossterm-widthdbfo_keep-togetherdbfo_label-widthdbfo_linenumbering.everyNthdbfo_linenumbering.separatordbfo presentationdbfo_list-widthdbfo_orientationdbfo_pgwidedbfo_rotated-widthdbfo_sidebar-widthdbfo_startdbfo_table-widthdbfo_term-widthdbfo_tocdbfo-needdbfo_row-heightIII. manpages Processing Instruction Referencedbman_funcsynopsis-styleIV. Common Processing Instruction Referencedbchoice_choicedbtimestampdbtex_delimsPart I. HTML Processing Instruction ReferenceIn-

PDF BY DBLATEX

troduction This is generated reference documentation for all user-specifiable processing instructions (PIs) in the DocBook XSL stylesheets for HTML output. NoteYou add these PIs at particular points in a document to cause specific “exceptions” to formatting/output behavior. To make global changes in formatting/output behavior across an entire document, it’s better to do it by setting an appropriate stylesheet parameter (if there is one). dbhtml_background-colordbhtml_background-colorNamedbhtml_background-color — Sets background color for an image Synopsis<?dbhtml background-color="color"?> Description Use the <?dbhtml background-color?> PI before or after an image (graphic, inlinegraphic, imagedata, or videodata element) as a sibling to the element, to set a background color for the image. Parameters background-color="color"An HTML color value Related Information in DocBook XSL: The Complete Guide11 http://www.sagehill.net/docbookxsl/ Background color22 http://www.sagehill.net/docbookxsl/BGcolor.html dbhtml_bgcolordbhtml_bgcolorNamedbhtml_bgcolor — Sets background color on a CALS table row or table cell Synopsis<?dbhtml bgcolor="color"?> Description Use the <?dbhtml bgcolor?> PI as child of a CALS table row or cell to set a background color for that table row or cell. Parameters bgcolor="color"An HTML color value Related Information in DocBook XSL: The Complete Guide11 http://www.sagehill.net/docbookxsl/ Cell background color22 http://www.sagehill.net/docbookxsl/BGtableColor.html#CellBGColor dbhtml_cellpaddingdbhtml_cellpaddingNamedbhtml_cel — Specifies cellpadding in CALS table or qandaset output Synopsis<?dbhtml cellpadding="number"?> Description Use the <?dbhtml cellpadding?> PI as a child of a CALS table or qandaset to specify the value for the HTML cellpadding attribute in the output HTML table. Parameters cellpadding="number"Specifies the cellpadding Related Global Parameters html.cellpadding Related Information in DocBook XSL: The Complete Guide11 http://www.sagehill.net/docbookxsl/ Cell spacing and cell padding22 http://www.sagehill.net/docbookxsl/CellSpacing.html, Q and A formatting33 http://www.sagehill.net/docbookxsl/QandAformat.html dbhtml_cellspacingdbhtml_cellspacingNamedbhtml_cellspacing — Specifies cellspacing in CALS table or qandaset output Synopsis<?dbhtml cellspacing="number"?> Description Use the <?dbhtml cellspacing?> PI as a child of a CALS table or qandaset to specify the value for the HTML cellspacing attribute in the output HTML table. Parameters cellspacing="number"Specifies the cellspacing Related Global Parameters html.cellspacing Related Information in DocBook XSL: The Complete Guide11 http://www.sagehill.net/docbookxsl/ Cell spacing and cell padding22 http://www.sagehill.net/docbookxsl/CellSpacing.html, Q and A formatting33 http://www.sagehill.net/docbookxsl/QandAformat.html dbhtml_classdbhtml_classNamedbhtml_class — Set value of the class attribute for a CALS table row Synopsis<?dbhtml class="name"?> Description Use the <?dbhtml class?> PI as a child of a row to specify a class attribute and value in the HTML output for that row. Parameters class="name"Specifies the class name Related Information in DocBook XSL: The Complete Guide11 http://www.sagehill.net/docbookxsl/ Table styles in HTML output22 http://www.sagehill.net/docbookxsl/CSSTableCells.html dbhtml_dirdbhtml_dirNamedbhtml_dir — Specifies a directory name in which to write files Synopsis<?dbhtml dir="path"?> Description When chunking output, use the <?dbhtml dir?> PI as a child of a chunk source to cause the output of that chunk to be written to the specified directory; also, use it as a child of a mediaobject to specify a directory into which any long-description files for that mediaobject will be written. The output directory specification is inherited by all chunks of the descendants of the element. If descendants need to go to a different directory, then add another <?dbhtml dir?> processing instruction as a child of the source element for that chunk, and specify the path relative to the ancestor path. For example, to put most chunk files into shared but one chapter into exception at the same level, use: <book> <?dbhtml dir="shared"?> ... <chapter> <?dbhtml dir="../exception"?> </chapter> </book> Parameters dir="path"Specifies the pathname for the directory Related Global Parameters base.dir Related Information in DocBook XSL: The Complete Guide11 http://www.sagehill.net/docbookxsl/ dbhtml dir processing instruction22 http://www.sagehill.net/docbookxsl/Chunking.html#dbhtmlDirPI dbhtml_filenamedbhtml_filenameNamedbhtml_filename — Specifies a filename for a chunk Synopsis<?dbhtml filename="filename"?> Description When chunking output, use the <?dbhtml filename?> PI as a child of a chunk source to specify a filename for the output file for that chunk. Include the filename suffix. You cannot include a directory path in the filename value, or your links may not work. Add a <?dbhtml dir?> processing instruction to specify the output directory. You can also combine the two specifications in one processing instruction: <?dbhtml dir="mydir" filename="myfile.html"?>. Parameters filename="path"Specifies the filename for the file Related Global Parameters use.id.as.filename Related Information in DocBook XSL: The Complete Guide11 http://www.sagehill.net/docbookxsl/ dbhtml filenames22 http://www.sagehill.net/docbookxsl/Chunking.html#DbhtmlFilenames dbhtml_funcsynopsis-styledbhtml_funcsynopsis-styleNamedbhtml_funcsynopsis-style — Specifies presentation style for a funcsynopsis Synopsis<?dbhtml funcsynopsis-style="kr"|"ans Description Use the <?dbhtml funcsynopsis-style?> PI as a child of a funcsynopsis or anywhere within a funcsynopsis to control the presentation style for output of all funcprototype instances within that funcsynopsis. Parameters funcsynopsis-style="kr"Displays funcprototype output in K&R style funcsynopsis-style="ansi"Displays funcprototype output in ANSI style Related Global Parameters funcsynopsis.style dbhtml_img.src.pathdbhtml_img.src.pathNamedbhtml_img.src.path — Specifies a path to the location of an image file Synopsis<?dbhtml img.src.path="path"?> Description Use the <?dbhtml img.src.path?> PI before or after an image (graphic, inlinegraphic, imagedata, or videodata element) as a sibling to the element, to specify a path to the location of the image; in HTML output, the value specified for the img.src.path attribute is prepended to the filename. Parameters img.src.path="path"Specifies the pathname to prepend to the name of the image file Related Global Parameters img.src.path Related Information in DocBook XSL: The Complete Guide11 http://www.sagehill.net/docbookxsl/ Using fileref22 http://www.sagehill.net/docbookxsl/GraphicsLocations.html#UsingFileref dbhtml_label-widthdbhtml_label-widthNamedbhtml_label-width — Specifies the label width for a qandaset Synopsis<?dbhtml label-width="width"?> Description Use the <?dbhtml label-

width?> PI as a child of a qandaset to specify the width of labels. Parameters label-width="width"Specifies the label width (including units) Related Information in DocBook XSL: The Complete Guide11 http://www.sagehill.net/docbookxsl/ Q and A formatting22 http://www.sagehill.net/docbookxsl/QandAformat.html dbhtml_linenumbering.everyNthdbhtml_linenumbering.everyNthNa
— Specifies interval for line numbers in verbatims Synopsis<?dbhtml linenumbering.everyNth="N"?> Description Use the <?dbhtml linenumbering.everyNth?> PI as a child of a "verbatim" element – programlisting, screen, synopsis — to specify the interval at which lines are numbered. Parameters linenumbering.everyNth="N"Specifies numbering interval; a number is output before every Nth line Related Global Parameters linenumbering.everyNth Related Information in DocBook XSL: The Complete Guide11 http://www.sagehill.net/docbookxsl/ Line numbering22 http://www.sagehill.net/docbookxsl/AnnotateListing.html#LineNumbe
dbhtml_linenumbering.separatordbhtml_linenumbering.separatorNamedbhtml_linenumbering.separator — Specifies separator text for line numbers in verbatims Synopsis<?dbhtml linenumbering.separator="text"?> Description Use the <?dbhtml linenumbering.separator?> PI as a child of a "verbatim" element – programlisting, screen, synopsis — to specify the separator text output between the line numbers and content. Parameters linenumbering.separator="text"Specifies the text (zero or more characters) Related Global Parameters linenumbering.separator Related Information in DocBook XSL: The Complete Guide11 http://www.sagehill.net/docbookxsl/ Line numbering22 http://www.sagehill.net/docbookxsl/AnnotateListing.html#LineNumbering
dbhtml_linenumbering.widthdbhtml_linenumbering.widthNamedbhtml_linenumbering.width — Specifies width for line numbers in verbatims Synopsis<?dbhtml linenumbering.width="width"?> Description Use the <?dbhtml linenumbering.width?> PI as a child of a "verbatim" element – programlisting, screen, synopsis — to specify the width set aside for line numbers. Parameters linenumbering.width="width"Specifies the width (inluding units) Related Global Parameters linenumbering.width Related Information in DocBook XSL: The Complete Guide11 http://www.sagehill.net/docbookxsl/ Line numbering22 http://www.sagehill.net/docbookxsl/AnnotateListing.html#LineNumbering dbhtml_list-presentationdbhtml_list-presentationNamedbht
presentation — Specifies presentation style for a variablelist or segmentedlist Synopsis<?dbhtml list-presentation="list"|"table"?> Description Use the <?dbhtml list-presentation?> PI as a child of a variablelist or segmentedlist to control the presentation style for the list (to cause it, for example, to be displayed as a table). Parameters list-presentation="list"Displays the list as a list list-presentation="table"Displays the list as a table Related Global Parameters •variablelist.as.table •segmentedlist.as.table Related Information in DocBook XSL: The Complete Guide11 http://www.sagehill.net/docbookxsl/ Variable list formatting in HTML22 http://www.sagehill.net/docbookxsl/Variablelists.html#VarListFormatting dbhtml_list-widthdbhtml_list-widthNamedbhtml_list-width — Specifies the width of a variablelist or simplelist Synopsis<?dbhtml list-width="width"?> Description Use the <?dbhtml list-width?> PI as a child of a variablelist or a simplelist presented as a table, to specify the output width. Parameters list-width="width"Specifies the output width (including units) Related Information in DocBook XSL: The Complete Guide11 http://www.sagehill.net/docbookxsl/ Variable list formatting in HTML22 http://www.sagehill.net/docbookxsl/Variablelists.html#VarList
dbhtml_row-heightdbhtml_row-heightNamedbhtml_row-height — Specifies the height for a CALS table row Synopsis<?dbhtml -row-height="height"?> Description Use the <?dbhtml row-height?> PI as a child of a row to specify the height of the row. Parameters row-height="height"Specifies the row height (including units) Related Information in DocBook XSL: The Complete Guide11 http://www.sagehill.net/docbookxsl/ Row height22 http://www.sagehill.net/docbookxsl/RowHeight.html dbhtml_startdbhtml_s
— (obsolete) Sets the starting number on an ordered list Synopsis<?dbhtml start="character"?> Description This PI is obsolete. The intent of this PI was to provide a means for setting a specific starting number for an ordered list. Instead of this PI, set a value for the override attribute on the first listitem in the list; that will have the same effect as what this PI was intended for. Parameters start="character"Specifies the character to use as the starting number; use 0-9, a-z, A-Z, or lowercase or uppercase Roman numerals Related Information in DocBook XSL: The Complete Guide11 http://www.sagehill.net/docbookxsl/ List starting number22 http://www.sagehill.net/docbookxsl/Orderedlists.html#ListStartNum dbhtml_stop-chunkingdbhtml_stop-chunkingNamedbhtml_stop-chunking — Do not chunk any descendants of this element. Synopsis<?dbhtml stop-chunking?> Description When generating chunked HTML output, adding this PI as the child of an element that contains elements that would normally be generated on separate pages if generating chunked output causes chunking to stop at this point. No descendants of the current element will be split into new HTML pages: <section> <title>Configuring pencil</title> <?dbhtml stop-chunking?> ... </section> Related Information in DocBook XSL: The Complete Guide11 http://www.sagehill.net/docbookxsl/ Chunking into multiple HTML files22 http://www.sagehill.net/docbookxsl/Chunking.html dbhtml_table-summarydbhtml_table-summaryNamedbhtml_table-summary — Specifies summary for CALS table, variablelist, segmentedlist, or qandaset output Synopsis<?dbhtml table-summary="text"?> Description Use the <?dbhtml table-summary?> PI as a child of a CALS table, variablelist, segmentedlist, or qandaset to specify the text for the HTML summary attribute in the output HTML table. Parameters table-summary="text"Specifies the summary text (zero or more characters) Related Information in DocBook XSL: The Complete Guide11 http://www.sagehill.net/docbookxsl/ Variable list formatting in HTML22 http://www.sagehill.net/docbookxsl/Variablelists.html Table summary text33 http://www.sagehill.net/docbookxsl/TableSummary.html dbhtml_table-widthdbhtml_table-widthNamedbhtml_ta
width — Specifies the width for a CALS table Synopsis<?dbhtml table-width="width"?> Description Use the <?dbhtml table-width?> PI as a child of a CALS table to specify the width of the table in output. Parameters table-width="width"Specifies the table width (including units or as a percentage) Related Global Parameters default.table.width Related Information in DocBook XSL: The Complete Guide11 http://www.sagehill.net/docbookxsl/ Table width22 http://www.sagehill.net/docbookxsl/Tables.html#Table
dbhtml_term-presentationdbhtml_term-presentationNamedbhtml_term-presentation — Sets character formatting for terms in a

PDF BY DBLATEX

variablelist Synopsis<?dbhtml term-presentation="bold"|"italic"|"bold-italic"?> Description Use the <?dbhtml term-presentation?> PI as a child of a variablelist to set character formatting for the term output of the list. Parameters term-presentation="bold"Specifies that terms are displayed in bold term-presentation="italic"Specifies that terms are displayed in italic term-presentation="bold-italic"Specifies that terms are displayed in bold-italic Related Information in DocBook XSL: The Complete Guide11 http://www.sagehill. Variable list formatting in HTML22 http://www.sagehill.net/docbookxsl/Variablelists.html#VarListFormatting dbhtml_term-separatordb separatorNamedbhtml_term-separator — Specifies separator text among terms in a varlistentry Synopsis<?dbhtml term-separator="text" Description Use the <?dbhtml term-separator?> PI as a child of a variablelist to specify the separator text among term instances. Parameters term-separator="text"Specifies the text (zero or more characters) Related Global Parameters variablelist.term.separator Related Information in DocBook XSL: The Complete Guide11 http://www.sagehill.net/docbookxsl/ Variable list formatting in HTML22 http://www.sagehill.net/docbookxsl/Variablelists.html#VarListFormatting dbhtml_term-widthdbhtml_term-widthNamedbhtm width — Specifies the term width for a variablelist Synopsis<?dbhtml term-width="width"?> Description Use the <?dbhtml term-width?> PI as a child of a variablelist to specify the width for term output. Parameters term-width="width"Specifies the term width (including units) Related Information in DocBook XSL: The Complete Guide11 http://www.sagehill.net/docbookxsl/ Variable list formatting in HTML22 http://www.sagehill.net/docbookxsl/Variablelists.html#VarListFormatting dbhtml_tocdbhtml_tocNamed — Specifies whether a TOC should be generated for a qandaset Synopsis<?dbhtml toc="0"|"1"?> Description Use the <?dbhtml toc?> PI as a child of a qandaset to specify whether a table of contents (TOC) is generated for the qandaset. Parameters toc="0"If zero, no TOC is generated toc="1"If 1 (or any non-zero value), a TOC is generated Related Information in DocBook XSL: The Complete Guide11 http://www.sagehill.net/docbookxsl/ Q and A list of questions22 http://www.sagehill.net/docbookxsl/QandAtoc.html, Q and A formatting33 http://www.sagehill.net/docbookxsl/QandAformat.html dbcmdlistdbcmdlistNamedbcmdlist — Generates a hyperlinked list of commands Synopsis<?dbcmdlist?> Description Use the <?dbcmdlist?> PI as the child of any element (for example, refsynopsisdiv) containing multiple cmdsynopsis instances; a hyperlinked navigational “command list” will be generated at the top of output for that element, enabling users to quickly jump to each command synopsis. Parameters [No parameters] dbfunclistdbfunclistNamedbfunclist — Generates a hyperlinked list of functions Synopsis<?dbfunclist?> Description Use the <?dbfunclist?> PI as the child of any element (for example, refsynopsisdiv) containing multiple funcsynopsis instances; a hyperlinked navigational “function list” will be generated at the top of output for that element, enabling users to quickly jump to to each function synopsis. Parameters [No parameters] dbhtml-include_hrefdbhtml-include_hrefNamedbhtml-include_href — Copies an external well-formed HTML/XML file into current doc Synopsis<?dbhtml-include href="URI"?> Description Use the <?dbhtml-include href?> PI anywhere in a document to cause the contents of the file referenced by the href pseudo-attribute to be copied/inserted “as is” into your HTML output at the point in document order where the PI occurs in the source. NoteThe referenced file may contain plain text (as long as it is “wrapped” in an html element — see the note below) or markup in any arbitrary vocabulary, including HTML — but it must conform to XML well-formedness constraints (because the feature in XSLT 1.0 for opening external files, the document() function, can only handle files that meet XML well-formedness constraints). Among other things, XML well-formedness constraints require a document to have a single root element. So if the content you want to include is plain text or is markup that does not have a single root element, wrap the content in an html element. The stylesheets will strip out that surrounding html “wrapper” when they find it, leaving just the content you want to insert. Parameters href="URI"Specifies the URI for the file to include; the URI can be, for example, a remote http: URI, or a local filesystem file: URI Related Global Parameters textinsert.extension Related Information in DocBook XSL: The Complete Guide11 http://www.sagehill.net/docbookxsl/ Inserting external HTML code22 http://www.sagehill.net/docbookxsl/InsertExtHtml.html, External code files33 http://www.sagehill.net/docbookxsl/ExternalCode.html dbhhdbhhNamedbhh — Sets topic name and topic id for context-sensitive HTML Help Synopsis<?dbhh topicname="name" topicid="id"?> Description Use the <?dbhh?> PI as a child of components that should be used as targets for context-sensitive help requests. Parameters topicname="name"Specifies a unique string constant that identifies a help topic topicid="id"Specifies a unique integer value for the topicname string Related Information in DocBook XSL: The Complete Guide11 http://www.sagehill.net/docbookxsl/ Context-sensitive help22 http://www.sagehill.n Part II. FO Processing Instruction ReferenceIntroduction This is generated reference documentation for all user-specifiable processing instructions (PIs) in the DocBook XSL stylesheets for FO output. NoteYou add these PIs at particular points in a document to cause specific “exceptions” to formatting/output behavior. To make global changes in formatting/output behavior across an entire document, it’s better to do it by setting an appropriate stylesheet parameter (if there is one). dbfo_background-colordbfo_background-colorNamedbfo_background-color — Sets background color for an image Synopsis<?dbfo background-color="color"?> Description Use the <?dbfo background-color?> PI before or after an image (graphic, inlinegraphic, imagedata, or videodata element) as a sibling to the element, to set a background color for the image. Parameters background-color="color"An HTML color value Related Information in DocBook XSL: The Complete Guide11 http://www.sagehill.net/docbookxsl/ Background color22 http://www.sagehill.net/docbookxsl/BGcolor.html dbfo_bgcolordbfo_bgcolorNamedbfo_bgcolor — Sets background color on a table row or table cell Synopsis<?dbfo bgcolor="color"?> Description Use the <?dbfo bgcolor?> PI as child of a table row or cell to set a background color for that table row or cell. This PI works for both CALS and HTML tables. Parameters bgcolor="color"An HTML color value Related Information in DocBook XSL: The Complete Guide11 http://www.sagehill.net/docbookxsl/ Cell background color22 http://www.sagehill.net/docbookxsl/BGtableColor.html#CellBGColor dbfo_float-typedbfo_float-typeNamedbfo_float-type — Specifies float behavior for a sidebar Synopsis<?dbfo float-type="margin.note"?>

PDF BY DBLATEX

Description Use the <?dbfo float-type?> PI to specify the float behavior for a sidebar (to cause the sidebar to be displayed as a marginal note). Parameters float-type="margin.note"Specifies that the sidebar should be displayed as a marginal note. Related Global Parameters sidebar.float.type (parameter), sidebar.float.width (parameter), sidebar.properties (attribute-set), sidebar.title.properties (attribute-set) Related Information in DocBook XSL: The Complete Guide11 http://www.sagehill.net/docbookxsl/ A sidebar as side float22 http://www.sagehill.net/docbookxsl/SideFloats.html#SidebarFloats dbfo_funcsynopsis-styledbfo_funcsynopsis-styleNamedbfo_funcsynopsis-style — Specifies presentation style for a funcsynopsis Synopsis<?dbfo funcsynopsis-style="kr"|"ansi"?> Description Use the <?dbfo funcsynopsis-style?> PI as a child of a funcsynopsis or anywhere within a funcsynopsis to control the presentation style for output of all funcprototype instances within that funcsynopsis. Parameters funcsynopsis-style="kr"Displays funcprototype output in K&R style funcsynopsis-style="ansi"Displays funcprototype output in ANSI style Related Global Parameters funcsynopsis.style dbfo_glossary-presentationdbfo_glossary-presentationNamedbfo_glossary-presentation — Specifies presentation style for a glossary Synopsis<?dbfo glossary-presentation="list"|"blocks"?> Description Use the <?dbfo glossary-presentation?> PI as a child of a glossary to control its presentation style. Parameters glossary-presentation="list"Displays the glossary as a list glossary-presentation="blocks"Displays the glossary as blocks Related Global Parameters glossary.as.blocks Related Information in DocBook XSL: The Complete Guide11 http://www.sagehill.net/docbookxsl/ Glossary formatting in print22 http://www.sagehill.net/docbookxsl/Glossaries.html#GlossaryFormatPrint dbfo_glosslist-presentationdbfo_glosslist-presentationNamed presentation — Specifies presentation style for a glosslist Synopsis<?dbfo glosslist-presentation="list"|"blocks"?> Description Use the <?dbfo glosslist-presentation?> PI as a child of a glosslist to control its presentation style. Parameters glosslist-presentation="list"Displays the glosslist as a list glosslist-presentation="blocks"Displays the glosslist as blocks Related Global Parameters glosslist.as.blocks Related Information in DocBook XSL: The Complete Guide11 http://www.sagehill.net/docbookxsl/ Glossary formatting in print22 http://www.sagehill.net/docbookxsl/Glossaries.html#GlossaryFormatPrint dbfo_glossterm-widthdbfo_gl widthNamedbfo_glossterm-width — Specifies the glossterm width for a glossary or glosslist Synopsis<?dbfo glossterm-width="width"?> Description Use the <?dbfo glossterm-width?> PI as a child of a glossary or glosslist to specify the width for output of glossterm instances in the output. Parameters glossterm-width="width"Specifies the glossterm width (including units) Related Global Parameters glossterm.width, glossterm.separation Related Information in DocBook XSL: The Complete Guide11 http://www.sagehill.net/d Glossary formatting in print22 http://www.sagehill.net/docbookxsl/Glossaries.html#GlossaryFormatPrint dbfo_keep-togetherdbfo_keep togetherNamedbfo_keep-together — Specifies “keep” behavior for a table, example, figure, equation, procedure, or task Synopsis<?dbfo keep-together="auto"|"always"?> Description Use the <?dbfo keep-together?> PI as a child of a formal object (table, example, figure, equation, procedure, or task) to specify “keep” behavior (to allow the object to “break” across a page). The PI also works with informaltable, informalexample, informalfigure and informalequation. Parameters keep-together="auto"Enables the object to break across a page keep-together="always"Prevents the object from breaking across a page (the default stylesheet behavior) Related Global Parameters formal.object.properties Related Information in DocBook XSL: The Complete Guide11 http://www.sagehill.net/docbookxsl/ Keep-together processing instruction22 http://www.sagehill.net/docbookxsl/PageBreaking.html#Ke dbfo_label-widthdbfo_label-widthNamedbfo_label-width — Specifies the label width for a qandaset, itemizedlist, orderedlist or calloutlist Synopsis<?dbfo label-width="width"?> Description Use the <?dbfo label-width?> PI as a child of a qandaset, itemizedlist, orderedlist, or calloutlist to specify the width of labels. Parameters label-width="width"Specifies the label width (including units) Related Information in DocBook XSL: The Complete Guide11 http://www.sagehill.net/docbookxsl/ Q and A formatting22 http://www.sagehill.net/docbookxsl/QandAformat.html dbfo_linenumbering.everyNthdbfo_linenumbering.everyNthNamedbfo_ — Specifies interval for line numbers in verbatims Synopsis<?dbfo linenumbering.everyNth="N"?> Description Use the <?dbfo linenumbering.everyNth?> PI as a child of a “verbatim” element – programlisting, screen, synopsis — to specify the interval at which lines are numbered. Parameters linenumbering.everyNth="N"Specifies numbering interval; a number is output before every Nth line Related Global Parameters linenumbering.everyNth Related Information in DocBook XSL: The Complete Guide11 http://www.sagehill.net/docbookxsl/ Line numbering22 http://www.sagehill.net/docbookxsl/AnnotateListing.html#LineNumbering dbfo_linenumbering.separatordbfo_linenumbering.separatorNamedbfo_linenumbering.separator — Specifies separator text for line numbers in verbatims Synopsis<?dbfo linenumbering.separator="text"?> Description Use the <?dbfo linenumbering.separator?> PI as a child of a “verbatim” element – programlisting, screen, synopsis — to specify the separator text output between the line numbers and content. Parameters linenumbering.separator="text"Specifies the text (zero or more characters) Related Global Parameters linenumbering.separator Related Information in DocBook XSL: The Complete Guide11 http://www.sagehill.net/docbookxsl/ Line numbering22 http://www.sagehill.net/docbookxsl/AnnotateListing.html#LineNumbering dbfo_linenumbering.widthdbfo_linenum — Specifies width for line numbers in verbatims Synopsis<?dbfo linenumbering.width="width"?> Description Use the <?dbfo linenumbering.width?> PI as a child of a “verbatim” element – programlisting, screen, synopsis — to specify the width set aside for line numbers. Parameters linenumbering.width="width"Specifies the width (inluding units) Related Global Parameters linenumbering.width Related Information in DocBook XSL: The Complete Guide11 http://www.sagehill.net/docbookxsl/ Line numbering22 http://www.sagehill.net/docbookxsl/AnnotateListing.html#LineNumbering dbfo_list-presentationdbfo_list-presentationN presentation — Specifies presentation style for a variablelist or segmentedlist Synopsis<?dbfo list-presentation="list"|"blocks"|"table"?> Description Use the <?dbfo list-presentation?> PI as a child of a variablelist or segmentedlist to control the presentation style for the list (to cause it, for example, to be displayed as a table). Parameters list-presentation="list"Displays the list as a list list-presentation="blocks"(variablelist only) Displays the list as blocks list-presentation="table"(segmentedlist only) Displays the

list as a table Related Global Parameters •variablelist.as.blocks •variablelist.as.table Related Information in DocBook XSL: The Complete Guide11 http://www.sagehill.net/docbookxsl/ Variable list formatting in print22 http://www.sagehill.net/docbookxsl/Variableli dbfo_list-widthdbfo_list-widthNamedbfo_list-width — Specifies the width of a horizontal simplelist Synopsis<?dbfo list-width="width" Description Use the <?dbfo list-width?> PI as a child of a simplelist whose class value is horizontal, to specify the width of the simplelist. Parameters list-width="width"Specifies the simplelist width (including units) dbfo_orientationdbfo_orientationNamedbfo_ori — Specifies the orientation for a CALS table row or cell Synopsis<?dbfo orientation="0"|"90"|"180"|"270"|"-90"|"-180"|"-270"?> Description Use the <?dbfo orientation?> PI as a child of a CALS table row or cell to specify the orientation (rotation) for the row or cell. Parameters orientation="0"|"90"|"180"|"270"|"-90"|"-180"|"-270"Specifies the number of degrees by which the cell or row is rotated dbfo_pgwidedbfo_pgwideNamedbfo_pgwide — Specifies if an equation or example goes across full page width Synopsis<?dbfo pgwide="0"|"1"?> Description Use the <?dbfo pgwide?> PI as a child of an equation or example to specify that the content should rendered across the full width of the page. Parameters pgwide="0"If zero, the content is rendered across the current text flow pgwide="1"If 1 (or any non-zero value), the content is rendered across the full width of the page Related Global Parameters pgwide.properties dbfo_rotated-widthdbfo_rotated-widthNamedbfo_rotated-width — Specifies the width for a CALS table entry or row Synopsis<?dbfo rotated-width="width"?> Description Use the <?dbfo rotated-width?> PI as a child of entry or row instance in a CALS table to specify the width of that the entry or row; or use it higher up in table to cause the width to be inherited recursively down. Parameters rotated-width="width"Specifies the width of a row or cell (including units) dbfo_sidebar-widthdbfo_sidebar-widthNamedbfo_sidebar-width — Specifies the width of a sidebar Synopsis<?dbfo sidebar-width="width"?> Description Use the <?dbfo sidebar-width?> PI as a child of a sidebar to specify the width of the sidebar. Parameters sidebar-width="width"Specifies the sidebar width (including units) Related Global Parameters sidebar.float.type parameter, sidebar.float.width parameter, sidebar.properties attribute-set, sidebar.title.properties Related Information in DocBook XSL: The Complete Guide11 http://www.sagehill.net/docbookxsl/ A sidebar as side float22 http://www.sagehill.net/docbookxsl/SideFloats.html#SidebarFloats dbfo_startdbfo_startNamedbfo_start — (obsolete) Sets the starting number on an ordered list Synopsis<?dbfo start="character"?> Description This PI is obsolete. The intent of it was to provide a means for setting a specific starting number for an ordered list. Instead of this PI, set a value for the override attribute on the first listitem in the list; that will have the same effect as what this PI was intended for. Parameters start="character"Specifies the character to use as the starting number; use 0-9, a-z, A-Z, or lowercase or uppercase Roman numerals Related Information in DocBook XSL: The Complete Guide11 http://www.sagehill.net/docbookxsl/ List starting number22 http://www.sagehill.net/docbookxsl/Orderedlists.html#ListStartNum dbfo_table-widthdbfo_table-widthNamedbfo_table-width — Specifies the width for a CALS table or for revhistory output Synopsis<?dbfo table-width="width"?> Description Use the <?dbfo table-width?> PI as a child or sibling of a CALS table, or as a child of an informaltable, entrytbl, or revhistory instance (which is rendered as a table in output) to specify the width of the table in output. Parameters table-width="width"Specifies the table width (including units or as a percentage) Related Information in DocBook XSL: The Complete Guide11 http://www.sagehill.net/docl Table width22 http://www.sagehill.net/docbookxsl/Tables.html#TableWidth dbfo_term-widthdbfo_term-widthNamedbfo_term-width — Specifies the term width for a variablelist Synopsis<?dbfo term-width="width"?> Description Use the <?dbfo term-width?> PI as a child of a variablelist to specify the width for term output. Parameters term-width="width"Specifies the term width (including units) Related Information in DocBook XSL: The Complete Guide11 http://www.sagehill.net/docbookxsl/ Variable list formatting in print22 http://www.sagehill.net/docbookxsl/Variablelists.html#ListIndents dbfo_tocdbfo_tocNamedbfo_toc — Specifies whether a TOC should be generated for a qandaset Synopsis<?dbfo toc="0"|"1"?> Description Use the <?dbfo toc?> PI as a child of a qandaset to specify whether a table of contents (TOC) is generated for the qandaset. Parameters toc="0"If zero, no TOC is generated toc="1"If 1 (or any non-zero value), a TOC is generated Related Information in DocBook XSL: The Complete Guide11 http://www.sagehill.net/docbookxsl/ Q and A list of questions22 http://www.sagehill.net/docbookxsl/QandAtoc.html, Q and A formatting33 http://www.sagehill.net/docbookxsl/QandAformat.html dbfo-needdbfo-needNamedbfo-need — Specify a need for space (a kind of soft page break) Synopsis<?dbfo-need height="n" [space-before="n"]?> Description A “need” is a request for space on a page. If the requested space is not available, the page breaks and the content that follows the need request appears on the next page. If the requested space is available, then no page break is inserted. Parameters height="n"The amount of height needed (including units) space-before="n"The amount of extra vertical space to add (including units) Related Information in DocBook XSL: The Complete Guide11 http://www.sagehill.net/docbookxsl/ Soft page breaks22 http://www.sagehill.net/docbookxsl/P dbfo_row-heightdbfo_row-heightNamedbfo_row-height — Specifies the height for a CALS table row Synopsis<?dbfo row-height="height"?> Description Use the <?dbfo row-height?> PI as a child of a row to specify the height of the row. Parameters row-height="height"Specifies the row height (including units) Related Information in DocBook XSL: The Complete Guide11 http://www.sagehill.net/docbookxsl/ Row height22 http://www.sagehill.net/docbookxsl/RowHeight.html Part III. manpages Processing Instruction ReferenceIntroduction This is generated reference documentation for all user-specifiable processing instructions (PIs) in the DocBook XSL stylesheets for manpages output. NoteYou add these PIs at particular points in a document to cause specific “exceptions” to formatting/output behavior. To make global changes in formatting/output behavior across an entire document, it’s better to do it by setting an appropriate stylesheet parameter (if there is one). dbman_funcsynopsis-styledbman_funcsynopsis-styleNamedbman_funcsynopsis-style — Specifies presentation style for a funcsynopsis. Synopsis<?dbman -funcsynopsis-style="kr"|"ansi"?> Description Use the <?dbman funcsynopsis-style?> PI as a child of a funcsynopsis or anywhere

PDF BY DBLATEX

within a funcsynopsis to control the presentation style for output of all funcprototype instances within that funcsynopsis. Parameters funcsynopsis-style="kr"Displays the funcprototype in K&R style funcsynopsis-style="ansi"Displays the funcprototype in ANSI style Related Global Parameters man.funcsynopsis.style Part IV. Common Processing Instruction ReferenceIntroduction This is generated reference documentation for all user-specifiable processing instructions (PIs) in the “common” part of the DocBook XSL stylesheets. NoteYou add these PIs at particular points in a document to cause specific “exceptions” to formatting/output behavior. To make global changes in formatting/output behavior across an entire document, it’s better to do it by setting an appropriate stylesheet parameter (if there is one). dbchoice_choicedbchoice_choiceNamedbchoice_choice — Generates a localized choice separator Synopsis<?dbchoice choice="and"|"or"|string"?> Description Use the <?dbchoice choice?> PI to generate an appropriate localized “choice” separator (for example, and or or) before the final item in an inline simplelist WarningThis PI is a less-than-ideal hack; support for it may disappear in the future (particularly if and when a more appropriate means for marking up "choice" lists becomes available in DocBook). Parameters choice="and"generates a localized and separator choice="or"generates a localized or separator choice="string"generates a literal string separator dbtimestampdbtimestampNamedbtimestamp — Inserts a date timestamp Synopsis<?dbtimestamp format="formatstring" [padding="0"|"1"]?> Description Use the <?dbtimestamp?> PI at any point in a source document to cause a date timestamp (a formatted string representing the current date and time) to be inserted in output of the document. Parameters format="formatstring"Specifies format in which the date and time are output NoteFor details of the content of the format string, see DocBook XSL: TCG, Date and time11 http://www.sagehill.net/docbookxsl/Datetime.html. padding="0"|"1"Specifies padding behavior; if non-zero, padding is is added dbtex_delimsdbtex_delimsNamedbtex_delims — Generates delimiters around embedded TeX equations in output Synopsis<?dbtex delims="no"|"yes"?> Description Use the <?dbtex delims?> PI as a child of a textobject containing embedded TeX markup, to cause that markup to be surrounded by $ delimiter characters in output. WarningThis feature is useful for print/PDF output only if you use the obsolete and now unsupported PassiveTeX XSL-FO engine. Parameters dbtex delims="no"|"yes"Specifies whether delimiters are output Related Global Parameters tex.math.delims DocBook XSL Stylesheets Developer ReferenceDocBook XSL Stylesheets Developer ReferenceAbstractThis is technical reference documentation for developers using the DocBook XSL Stylesheets. It is not intended to be user documentation, but is instead provided for developers writing customization layers for the stylesheets.DocBook XSL Stylesheets Developer ReferenceDocBook XSL Stylesheets Developer ReferenceTable of ContentsI. XSL Library Template ReferenceI. General Library Templatesdot.countcopy-stringstring.substxpo
magnitudelength-unitslength-speclength-in-pointspi-attributelookup.keyxpath.locationcomment-escape-stringcomment-escape-string.re
string-subst-mapII. Relative URI Functionscount.uri.path.depthtrim.common.uri.pathsresolve.pathdirnameII. Common Template ReferenceIII. Common » Base Template Referenceis.componentis.sectionsection.levelqanda.section.levelselect.mediaobjectselect.mediaobj
typeIV. Common » Refentry Metadata Template Referenceget.refentry.metadataget.refentry.titleget.refentry.sectionget.refentry.dateget.r
Common » Utility Template Referencelog.messageget.doc.titlepad-stringVI. Common » Character-Map Template Referenceapply-character-mapread-character-mapIII. Formatting Object Table Referencecalc.column.widthIV. Titlepage Template Stylesheet Referencet:templatesxsl:*t:titlepage@* (in copy.literal.atts mode)t:titlepage-contentt:titlepage-separatort:titlepage-before* (in copy mode)@* (in copy mode)* (in document.order mode)* (in document.order mode)* (in titlepage.specialrules mode)* (in titlepage.subrules mode)t:ort:or (in titlepage.subrules mode)element-or-listPart I. XSL Library Template ReferenceIntroduction This is technical reference documentation for the vocabulary-independent “library” templates in the DocBook XSL Stylesheets. This is not intended to be user documentation. It is provided for developers writing customization layers for the stylesheets. XSL Library Template ReferenceXSL Library Template ReferenceTable of ContentsI. General Library Templatesdot.countcopy-stringstring.substxpointer.idrefidreflength-magnitudelength-unitslength-speclength-in-pointspi-attributelookup.keyxpath.locationcomm
escape-stringcomment-escape-string.recursivetrim.textstr.tokenize.keep.delimitersapply-string-subst-mapII. Relative URI Functionscount.uri.path.depthtrim.common.uri.pathsresolve.pathdirnameGeneral Library TemplatesGeneral Library TemplatesGeneral Library TemplatesTable of Contentsdot.countcopy-stringstring.substxpointer.idrefidreflength-magnitudelength-unitslength-speclength-in-pointspi-attributelookup.keyxpath.locationcomment-escape-stringcomment-escape-string.recursivetrim.textstr.tokenize.k
string-subst-mapNamedot.count — Returns the number of “.” characters in a string Description Given a string, the dot.count template returns the number of dot/period characters in the string. This template is useful, for example, when testing the nesting level of nested inline markup (for nested emphasis, quotations, etc.). <xsl:template name="dot.count"> <!-- Returns the number of "." characters in a string --> <xsl:param name="string"></xsl:param> <xsl:param name="count" select="0"></xsl:param> <xsl:choose> <xsl:when test="contains($string, ’.’)"> <xsl:call-template name="dot.count">
<xsl:with-param name="string" select="substring-after($string, ’.’)"></xsl:with-param> <xsl:with-param name="count" select="$count+1"></xsl:wi
param> </xsl:call-template> </xsl:when> <xsl:otherwise> <xsl:value-of select="$count"></xsl:value-of> -
</xsl:otherwise> </xsl:choose> </xsl:template> Namecopy-string — Returns “count” copies of a string Description Given a string, the copy-string template creates n copies of the string, when the value of n is given by the count parameter. <xsl:template name="copy-string">
<!-- returns ’count’ copies of ’string’ --> <xsl:param name="string"></xsl:param> <xsl:param name="count" select="0"></xsl:param> <xsl:param name="result"></xsl:param> <xsl:choose> <xsl:when test="$count>0">
<xsl:call-template name="copy-string">
<xsl:with-param name="string" select="$string"></xsl:with-param>
<xsl:with-param name="count" select="$count - 1"></xsl:with-param>
<xsl:with-param name="result">
<xsl:value-of select="$result"></xsl:value-of>
<xsl:value-of select="$string"></xsl:value-of>
</xsl:with-param>
</xsl:call-

PDF BY DBLATEX

template> </xsl:when> <xsl:otherwise> <xsl:value-of select="$result"></xsl:value-of> </xsl:otherwise> </xsl:choose> </xsl:template> Namestring.subst — Substitute one text string for another in a string Description The string.subst template replaces all occurances of target in string with replacement and returns the result. <xsl:template name="string.subst"> <xsl:param name="string"></xsl:param> <xsl:param name="target"></xsl:param> <xsl:param name="replacement"></xsl:param> <xsl:choose> <xsl:when test="contains($string, $target)"> <xsl:variable name="rest"> <xsl:call-template name="string.subst"> <xsl:with-param name="string" select="substring-after($string, $target)"></xsl:with-param> <xsl:with-param name="target" select="$target"></xsl:with-param> <xsl:with-param name="replacement" select="$replacement"></xsl:with-param> </xsl:call-template> </xsl:variable> <xsl:value-of select="concat(substring-before($string, $target), $replacement, $rest)"></xsl:value-of> </xsl:when> <xsl:otherwise> <xsl:value-of select="$string"></xsl:value-of> </xsl:otherwise> </xsl:choose> </xsl:template> Namexpointer.idref — Extract IDREF from an XPointer Description The xpointer.idref template returns the ID portion of an XPointer which is a pointer to an ID within the current document, or the empty string if it is not. In other words, xpointer.idref returns “foo” when passed either #foo or #xpointer(id(’foo’)), otherwise it returns the empty string. <xsl:template name="xpointer.idref">
<xsl:param name="xpointer">http://...</xsl:param> <xsl:choose> <xsl:when test="starts-with($xpointer, ’#xpointer(id(’)">
<xsl:variable name="rest" select="substring-after($xpointer, ’#xpointer(id(’)"></xsl:variable> <xsl:variable name="quote" select="substring($rest, 1, 1)"></xsl:variable> <xsl:value-of select="substring-before(substring-after($xpointer, $quote), $quote)"></xsl:value-of> </xsl:when> <xsl:when test="starts-with($xpointer, ’#’)"> <xsl:value-of select="substring-after($xpointer, ’#’)"></xsl:value-of> </xsl:when> <!-- otherwise it’s a pointer to some other document --> </xsl:choose> </xsl:template> Nameidref — Extract IDREF from a linking element Description The idref template accepts any xref or universally linking element as a parameter and returns the local ID that it points to. If it has a linkend attribute, it returns that value. If it has a xlink:href, it returns that value, parsed as needed to extract just the id reference. Otherwise it returns an empty string. <xsl:template name="idref">
<xsl:param name="xref" select="."></xsl:param> <xsl:param name="linkend" select="$xref/@linkend"></xsl:param> <xsl:param name="xpointer">
<xsl:call-template name="xpointer.idref">
<xsl:with-param name="xpointer" select="$xref/@xlink:href"></xsl:with-param> </xsl:call-template> </xsl:param>
<xsl:choose> <xsl:when test="string-length($linkend) != 0">
<xsl:value-of select="$linkend"></xsl:value-of>
</xsl:when> <xsl:when test="string-length($xpointer) != 0">
<xsl:value-of select="$xpointer"></xsl:value-of>
</xsl:when> <!-- otherwise it’s a pointer to some other document --> </xsl:choose> </xsl:template> Namelength-magnitude — Return the unqualified dimension from a length specification Description The length-magnitude template returns the unqualified length ("20" for "20pt") from a dimension. <xsl:template name="length-magnitude">
<xsl:param name="length" select="’0pt’"></xsl:param> <xsl:choose> <xsl:when test="string-length($length) = 0"></xsl:when>
<xsl:when test="substring($length,1,1) = ’0’ or substring($length,1,1) = ’1’ or substring($length,1,1) = ’2’ or substring($length,1,1) = ’3’ or substring($length,1,1) = ’4’ or substring($length,1,1) = ’5’ or substring($length,1,1) = ’6’ or substring($length,1,1) = ’7’ or substring($length,1,1) = ’8’ or substring($length,1,1) = ’9’ or substring($length,1,1) = ’.’"> <xsl:value-of select="substring($length,1,1)"></xsl:value-of>
<xsl:call-template name="length-magnitude">
<xsl:with-param name="length" select="substring($length,2)"></xsl:with-param> </xsl:call-template> </xsl:when> </xsl:choose> </xsl:template> Namelength-units — Return the units from a length specification Description The length-units template returns the units ("pt" for "20pt") from a length. If no units are supplied on the length, the defauilt.units are returned. <xsl:template name="length-units">
<xsl:param name="length" select="’0pt’"></xsl:param> <xsl:param name="default.units" select="’px’"></xsl:param> <xsl:variable name="magnitude">
<xsl:call-template name="length-magnitude">
<xsl:with-param name="length" select="$length"></xsl:with-param>
</xsl:call-template> </xsl:variable> <xsl:variable name="units">
<xsl:value-of select="substring($length, string-length($magnitude)+1)"></xsl:value-of> </xsl:variable> <xsl:value-of select="$magnitude"></xsl:value-of> <xsl:choose> <xsl:when test="$units=’cm’ or $units=’mm’ or $units=’in’ or $units=’pt’ or $units=’pc’ or $units=’px’ or $units=’em’"> <xsl:value-of select="$units"></xsl:value-of> </xsl:when>
<xsl:when test="$units = ’’"> <xsl:value-of select="$default.units"></xsl:value-of> </xsl:when> <xsl:otherwise>
<xsl:message> <xsl:text>Unrecognized unit of measure: </xsl:text> <xsl:value-of select="$units"></xsl:value-of>
<xsl:text>.</xsl:text> </xsl:message> </xsl:otherwise> </xsl:choose> </xsl:template> Namelength-in-points — Returns the size, in points, of a specified length Description The length-in-points template converts a length specification to points

PDF BY DBLATEX

and returns that value as an unqualified number. CautionThere is no way for the template to infer the size of an em. It relies on the default em.size which is initially 10 (for 10pt). Similarly, converting pixels to points relies on the pixels.per.inch parameter which is initially 90. <xsl:template name="length-in-points"> <xsl:param name="length" select="'0pt'"></xsl:param> <xsl:param name="em.size" select="10"></xsl:param> <xsl:param name="pixels.per.inch" select="90"></xsl:param> <xsl:variable name="magnitude"> <xsl:call-template name="length-magnitude"> <xsl:with-param name="length" select="$length"></xsl:with-param> </xsl:call-template> </xsl:variable> <xsl:variable name="units"> <xsl:value-of select="substring($length, string-length($magnitude)+1)"></xsl:value-of> </xsl:variable> <xsl:choose> <xsl:when test="$units = 'pt'"> <xsl:value-of select="$magnitude"></xsl:value-of> </xsl:when> <xsl:when test="$units = 'cm'"> <xsl:value-of select="$magnitude div 2.54 * 72.0"></xsl:value-of> </xsl:when> <xsl:when test="$units = 'mm'"> <xsl:value-of select="$magnitude div 25.4 * 72.0"></xsl:value-of> </xsl:when> <xsl:when test="$units = 'in'"> <xsl:value-of select="$magnitude * 72.0"></xsl:value-of> </xsl:when> <xsl:when test="$units = 'pc'"> <xsl:value-of select="$magnitude * 12.0"></xsl:value-of> </xsl:when> <xsl:when test="$units = 'px'"> <xsl:value-of select="$magnitude div $pixels.per.inch * 72.0"></xsl:value-of> </xsl:when> <xsl:when test="$units = 'em'"> <xsl:value-of select="$magnitude * $em.size"></xsl:value-of> </xsl:when> <xsl:otherwise> <xsl:message> <xsl:text>Unrecognized unit of measure: </xsl:text> <xsl:value-of select="$units"></xsl:value-of> <xsl:text>.</xsl:text> </xsl:message> </xsl:otherwise> </xsl:choose> </xsl:template> Namepi-attribute — Extract a pseudo-attribute from a PI Description The pi-attribute template extracts a pseudo-attribute from a processing instruction. For example, given the PI “<?foo bar="1" baz='red'?>”, <xsl:call-template name="pi-attribute"> <xsl:with-param name="pis" select="processing-instruction('foo')"/> <xsl:with-param name="attribute" select="'baz'"/> </xsl:call-template> will return “red”. This template returns the first matching attribute that it finds. Presented with processing instructions that contain badly formed pseudo-attributes (missing or unbalanced quotes, for example), the template may silently return erroneous results. <xsl:template name="pi-attribute"> <xsl:param name="pis" select="processing-instruction('BOGUS_PI')"></xsl:param> <xsl:param name="attribute">filename</xsl:param> <xsl:param name="count">1</xsl:param> <xsl:choose> <xsl:when test="$count>count($pis)"> <!-- not found --> </xsl:when> <xsl:otherwise> <xsl:variable name="pi"> <xsl:value-of select="$pis[$count]"></xsl:value-of> </xsl:variable> <xsl:variable name="pivalue"> <xsl:value-of select="concat(' ', normalize-space($pi))"></xsl:value-of> </xsl:variable> <xsl:choose> <xsl:when test="contains($pivalue,concat(' ', $attribute, '='))"> <xsl:variable name="rest" select="substring-after($pivalue,concat(' ', $attribute,'='))"></xsl:variable> <xsl:variable name="quote" select="substring($rest,1,1)"></xsl:variable> <xsl:value-of select="substring-before(substring($rest,2),$quote)"></xsl:value-of> </xsl:when> <xsl:otherwise> <xsl:call-template name="pi-attribute"> <xsl:with-param name="pis" select="$pis"></xsl:with-param> <xsl:with-param name="attribute" select="$attribute"></xsl:with-param> <xsl:with-param name="count" select="$count + 1"></xsl:with-param> </xsl:call-template> </xsl:otherwise> </xsl:choose> </xsl:otherwise> </xsl:choose> </xsl:template> Namelookup.key — Retrieve the value associated with a particular key in a table Description Given a table of space-delimited key/value pairs, the lookup.key template extracts the value associated with a particular key. <xsl:template name="lookup.key"> <xsl:param name="key" select=""></xsl:param> <xsl:param name="table" select=""></xsl:param> <xsl:if test="contains($table, ' ')"> <xsl:choose> <xsl:when test="substring-before($table, ' ') = $key"> <xsl:variable name="rest" select="substring-after($table, ' ')"></xsl:variable> <xsl:choose> <xsl:when test="contains($rest, ' ')"> <xsl:value-of select="substring-before($rest, ' ')"></xsl:value-of> </xsl:when> <xsl:otherwise> <xsl:value-of select="$rest"></xsl:value-of> </xsl:otherwise> </xsl:choose> </xsl:when> <xsl:otherwise> <xsl:call-template name="lookup.key"> <xsl:with-param name="key" select="$key"></xsl:with-param> <xsl:with-param name="table" select="substring-after(substring-after($table,' '), ' ')"></xsl:with-param> </xsl:call-template> </xsl:otherwise> </xsl:choose> </xsl:if> </xsl:template> Namexpath.location — Calculate the XPath child-sequence to the current node Description The xpath.location template calculates the absolute path from the root of the tree to the current element node. <xsl:template name="xpath.location"> <xsl:param name="node" select="."></xsl:param> <xsl:param name="path" select=""></xsl:param> <xsl:variable name="next.path"> <xsl:value-of select="local-name($node)"></xsl:value-of> <xsl:if test="$path != ''">/</xsl:if> <xsl:value-of select="$path"></xsl:value-of> </xsl:variable> <xsl:choose> <xsl:when test="$node/parent::*"> <xsl:call-template name="xpath.location"> <xsl:with-param name="node" select="$node/parent::*"></xsl:with-param> <xsl:with-param name="path" select="$next.path"></xsl:with-param> </xsl:call-template> </xsl:when> <xsl:otherwise> <xsl:text>/</xsl:text> <xsl:value-of select="$next.path"></xsl:value-of> </xsl:otherwise> </xsl:choose> </xsl:template> Namecomment-escape-string — Prepare a string for inclusion in an XML comment Description The comment-escape-string template returns a string that has been transformed so that it can safely be output as an XML comment. Internal occurrences of "--" will be replaced with "- -" and a leading and/or trailing space will be added to the string, if necessary. <xsl:template name="comment-escape-string"> <xsl:param name="string" select=""></xsl:param> <xsl:if test="starts-with($string, '-')"> <xsl:text> </xsl:text> </xsl:if> <xsl:call-template name="comment-escape-string.recursive"> <xsl:with-param name="string" select="$string"></xsl:with-param> </xsl:call-template> <xsl:if test="substring($string, string-length($string), 1) = '-'"> <xsl:text> </xsl:text> </xsl:if> </xsl:template> Namecomment-escape-string.recursive — Internal function used by comment-escape-string Description The comment-escape-

PDF BY DBLATEX

string.recursive template is used by comment-escape-string. <xsl:template name="comment-escape-string.recursive"> <xsl:param name="string" select="""></xsl:param> <xsl:choose> <xsl:when test="contains($string, '--')"> <xsl:value-of select="substring-before($string, '--')"></xsl:value-of> <xsl:value-of select="'- -'"></xsl:value-of> <xsl:call-template name="comment-escape-string.recursive"> <xsl:with-param name="string" select="substring-after($string, '--')"></xsl:with-param> </xsl:call-template> </xsl:when> <xsl:otherwise> <xsl:value-of select="$string"></xsl:value-of> </xsl:otherwise> </xsl:choose> </xsl:template> Nametrim.text — Trim leading and trailing whitespace from a text node Description Given a text node, this function trims leading and trailing whitespace from it and returns the trimmed contents. <xsl:template name="trim.text"> <xsl:param name="contents" select="."></xsl:param> <xsl:variable name="contents-left-trimmed"> <xsl:call-template name="trim-left"> <xsl:with-param name="contents" select="$contents"></xsl:with-param> </xsl:call-template> </xsl:variable> <xsl:variable name="contents-trimmed"> <xsl:call-template name="trim-right"> <xsl:with-param name="contents" select="$contents-left-trimmed"></xsl:with-param> </xsl:call-template> </xsl:variable> <xsl:value-of select="$contents-trimmed"></xsl:value-of> </xsl:template> <xsl:template name="trim-left"> <xsl:param name="contents"></xsl:param> <xsl:choose> <xsl:when test="starts-with($contents,' ') or starts-with($contents,' ') or starts-with($contents,' ') or starts-with($contents,' ')"> <xsl:call-template name="trim-left"> <xsl:with-param name="contents" select="substring($contents, 2)"></xsl:with-param> </xsl:call-template> </xsl:when> <xsl:otherwise> <xsl:value-of select="$contents"></xsl:value-of> </xsl:otherwise> </xsl:choose> </xsl:template> <xsl:template name="trim-right"> <xsl:param name="contents"></xsl:param> <xsl:variable name="last-char"> <xsl:value-of select="substring($contents, string-length($contents), 1)"></xsl:value-of> </xsl:variable> <xsl:choose> <xsl:when test="($last-char = ' ') or ($last-char = ' ') or ($last-char = ' ') or ($last-char = ' ')"> <xsl:call-template name="trim-right"> <xsl:with-param name="contents" select="substring($contents, 1, string-length($contents) - 1)"></xsl:with-param> </xsl:call-template> </xsl:when> <xsl:otherwise> <xsl:value-of select="$contents"></xsl:value-of> </xsl:otherwise> </xsl:choose> </xsl:template> Namestr.tokenize.keep.delimiters — Tokenize a string while preserving any delimiters Description Based on the occurrence of one or more delimiter characters, this function breaks a string into a list of tokens and delimiters, marking up each of the tokens with a token element and preserving the delimiters as text nodes between the tokens. NoteThis function is a very slightly modified version of a function from the EXSLT site11 http://www.exslt.org/. The original is available at: http://www.exslt.org/str/functions/tokenize/str.tokenize.template.xsl The str.tokenize.keep.delimiters function differs only in that it preserves the delimiters instead of discarding them. <xsl:template name="str.tokenize.keep.delimiters"> <xsl:param name="string" select="""></xsl:param> <xsl:param name="delimiters" select="' '"></xsl:param> <xsl:choose> <xsl:when test="not($string)"></xsl:when> <xsl:when test="not($delimiters)"> <xsl:call-template name="str.tokenize.keep.delimiters-characters"> <xsl:with-param name="string" select="$string"></xsl:with-param> </xsl:call-template> </xsl:when> <xsl:otherwise> <xsl:call-template name="str.tokenize.keep.delimiters-delimiters"> <xsl:with-param name="string" select="$string"></xsl:with-param> <xsl:with-param name="delimiters" select="$delimiters"></xsl:with-param> </xsl:call-template> </xsl:otherwise> </xsl:choose> </xsl:template> <xsl:template name="str.tokenize.keep.delimiters-characters"> <xsl:param name="string"></xsl:param> <xsl:if test="$string"> <ssb:token><xsl:value-of select="substring($string, 1, 1)"></xsl:value-of></ssb:token> <xsl:call-template name="str.tokenize.keep.delimiters-characters"> <xsl:with-param name="string" select="substring($string, 2)"></xsl:with-param> </xsl:call-template> </xsl:if> </xsl:template> <xsl:template name="str.tokenize.keep.delimiters-delimiters"> <xsl:param name="string"></xsl:param> <xsl:param name="delimiters"></xsl:param> <xsl:variable name="delimiter" select="substring($delimiters, 1, 1)"></xsl:variable> <xsl:choose> <xsl:when test="not($delimiter)"> <ssb:token><xsl:value-of select="$string"></xsl:value-of></ssb:token> </xsl:when> <xsl:when test="contains($string, $delimiter)"> <xsl:if test="not(starts-with($string, $delimiter))"> <xsl:call-template name="str.tokenize.keep.delimiters-delimiters"> <xsl:with-param name="string" select="substring-before($string, $delimiter)"></xsl:with-param> <xsl:with-param name="delimiters" select="substring($delimiters, 2)"></xsl:with-param> </xsl:call-template> </xsl:if> <!-- output each delimiter --> <xsl:value-of select="$delimiter"></xsl:value-of> <xsl:call-template name="str.tokenize.keep.delimiters-delimiters"> <xsl:with-param name="string" select="substring-after($string, $delimiter)"></xsl:with-param> <xsl:with-param name="delimiters" select="$delimiters"></xsl:with-param> </xsl:call-template> </xsl:when> <xsl:otherwise> <xsl:call-template name="str.tokenize.keep.delimiters-delimiters"> <xsl:with-param name="string" select="$string"></xsl:with-param> <xsl:with-param name="delimiters" select="substring($delimiters, 2)"></xsl:with-param> </xsl:call-template> </xsl:otherwise> </xsl:choose> </xsl:template> Nameapply-string-subst-map — Apply a string-substitution map Description This function applies a "string substitution" map. Use it when you want to do multiple string substitutions on the same target content. It reads in two things: content, the content on which to perform the substitution, and map.contents, a node set of elements (the names of the elements don't matter), with each element having the following attributes: •oldstring, a string to be replaced •newstring, a string with which to replace oldstring The function uses map.contents to do substitution on content, and then returns the modified contents. NoteThis function is a very slightly modified version of Jeni Tennison's replace_strings function in the multiple string replacements11 http://www.dpawson.co.uk/xsl/sect2/StringReplace.html#d9351e13 section of Dave Pawson's XSLT FAQ22 http://www.dpawson.co.uk/xsl/index.html. The apply-string-subst-map function is es-

PDF BY DBLATEX

sentially the same function as the apply-character-map function; the only difference is that in the map that apply-string-subst-map expects, oldstring and newstring attributes are used instead of character and string attributes. <xsl:template name="apply-string-subst-map"> <xsl:param name="content"></xsl:param> <xsl:param name="map.contents"></xsl:param> <xsl:variable name="replaced_text"> <xsl:call-template name="string.subst"> <xsl:with-param name="string" select="$content"></xsl:with-param> <xsl:with-param name="target" select="$map.contents[1]/@oldstring"></xsl:with-param> <xsl:with-param name="replacement" select="$map.contents[1]/@newstring"></xsl:with-param> </xsl:call-template> </xsl:variable> <xsl:choose> <xsl:when test="$map.contents[2]"> <xsl:call-template name="apply-string-subst-map"> <xsl:with-param name="content" select="$replaced_text"></xsl:with-param> <xsl:with-param name="map.contents" select="$map.contents[position() > 1]"></xsl:with-param> </xsl:call-template> </xsl:when> <xsl:otherwise> <xsl:value-of select="$replaced_text"></xsl:value-of> </xsl:otherwise> </xsl:choose> </xsl:template> IntroductionIntroductionRelative URI FunctionsIntroduction These functions manipulate relative URI references. The following assumptions must hold true: 1.All URIs are relative. 2.No URI contains the "../" sequence which would effectively move "up" the hierarchy. If these assumptions do not hold, the results are unpredictable. Table of Contentscount.uri.path.depthtrim.common.uri.pathsresolve.pathdirnameNamecount.uri.path.depth — Count the number of path components in a relative URI Description This function counts the number of path components in a relative URI. <xsl:template name="count.uri.path.depth"> <xsl:param name="filename" select="""></xsl:param> <xsl:param name="count" select="0"></xsl:param> <xsl:choose> <xsl:when test="contains($filename, '/')"> <xsl:call-template name="count.uri.path.depth"> <xsl:with-param name="filename" select="substring-after($filename, '/')"></xsl:with-param> <xsl:with-param name="count" select="$count + 1"></xsl:with-param> </xsl:call-template> </xsl:when> <xsl:otherwise> <xsl:value-of select="$count"></xsl:value-of> </xsl:otherwise> </xsl:choose> </xsl:template> Nametrim.common.uri.paths — Trim common leading path components from a relative URI Description This function trims common leading path components from a relative URI. <xsl:template name="trim.common.uri.paths"> <xsl:param name="uriA" select="""></xsl:param> <xsl:param name="uriB" select="""></xsl:param> <xsl:param name="return" select="'A'"></xsl:param> <!-- Resolve any ../ in the path --> <xsl:variable name="trimmed.uriA"> <xsl:call-template name="resolve.path"> <xsl:with-param name="filename" select="$uriA"></xsl:with-param> </xsl:call-template> </xsl:variable> <xsl:variable name="trimmed.uriB"> <xsl:call-template name="resolve.path"> <xsl:with-param name="filename" select="$uriB"></xsl:with-param> </xsl:call-template> </xsl:variable> <xsl:choose> <xsl:when test="contains($trimmed.uriA, '/') and contains($trimmed.ur '/') and substring-before($trimmed.uriA, '/') = substring-before($trimmed.uriB, '/')"> <xsl:call-template name="trim.common.uri.paths"> <xsl:with-param name="uriA" select="substring-after($trimmed.uriA, '/')"></xsl:with-param> <xsl:with-param name="uriB" select="substring-after($trimmed.uriB, '/')"></xsl:with-param> <xsl:with-param name="return" select="$return"></xsl:with-param> </xsl:call-template> </xsl:when> <xsl:otherwise> <xsl:choose> <xsl:when test="$return = 'A'"> <xsl:value-of select="$trimmed.uriA"></xsl:value-of> </xsl:when> <xsl:otherwise> <xsl:value-of select="$trimmed.uriB"></xsl:value-of> </xsl:otherwise> </xsl:choose> </xsl:otherwise> </xsl:choose> </xsl:template> Nameresolve.path — Resolve any ../ in path, except leading ../ Description This function resolves any relative ../ parts of a file path. <xsl:template name="resolve.path"> <xsl:param name="filename" select="""></xsl:param> <xsl:choose> <!-- Leading .. are not eliminated --> <xsl:when test="starts-with($filename, '../')"> <xsl:value-of select="'../'"></xsl:value-of> <xsl:call-template name="resolve.path"> <xsl:with-param name="filename" select="substring-after($filename, '../')"></xsl:with-param> </xsl:call-template> </xsl:when> <xsl:when test="contains($filename, '/../')"> <xsl:call-template name="resolve.path"> <xsl:with-param name="filename"> <xsl:call-template name="dirname"> <xsl:with-param name="filename" select="substring-before($filename, '/../')"></xsl:with-param> </xsl:call-template> <xsl:value-of select="substring-after($filename, '/../')"></xsl:value-of> </xsl:with-param> </xsl:call-template> </xsl:when> <xsl:otherwise> <xsl:value-of select="$filename"></xsl:value-of> </xsl:otherwise> </xsl:choose> </xsl:template> Namedirname — Return directory part of a path Description This function returns the directory part of a path. <xsl:template name="dirname"> <xsl:param name="filename" select="""></xsl:param> <xsl:if test="contains($filename, '/')"> <xsl:value-of select="substring-before($filename, '/')"></xsl:value-of> <xsl:text>/</xsl:text> <xsl:call-template name="dirname"> <xsl:with-param name="filename" select="substring-after($filename, '/')"></xsl:with-param> </xsl:call-template> </xsl:if> </xsl:template> Part II. Common Template ReferenceCommon Template ReferenceCommon Template ReferenceTable of ContentsIII. Common » Base Template Referenceis.componentis.sectionsection.levelqanda.section.levelselect.mediaobjectselect.mediaobject.indexis.acceptable.mediaob typeIV. Common » Refentry Metadata Template Referenceget.refentry.metadataget.refentry.titleget.refentry.sectionget.refentry.dateget.r Common » Utility Template Referencelog.messageget.doc.titlepad-stringVI. Common » Character-Map Template Referenceapply-character-mapread-character-mapIntroductionIntroductionCommon » Base Template Reference$Id: common.xsl 9347 2012-05-11 03:49:49Z bobstayton $ Introduction This is technical reference documentation for the "base" set of common templates in the DocBook XSL Stylesheets. This is not intended to be user documentation. It is provided for developers writing customization layers for the stylesheets. Table of Contentsis.componentis.sectionsection.levelqanda.section.levelselect.mediaobjectselect.mediaobject.ind typeNameis.component — Tests if a given node is a component-level element Synopsis<xsl:template name="is.component"> <xsl:param name="node" select="."/> ... </xsl:template> Description This template returns '1' if the specified node is a

PDF BY DBLATEX

component (Chapter, Appendix, etc.), and ’0’ otherwise. Parameters nodeThe node which is to be tested. Returns This template returns ’1’ if the specified node is a component (Chapter, Appendix, etc.), and ’0’ otherwise. Nameis.section — Tests if a given node is a section-level element Synopsis<xsl:template name="is.section"> <xsl:param name="node" select="."/> ... </xsl:template> Description This template returns ’1’ if the specified node is a section (Section, Sect1, Sect2, etc.), and ’0’ otherwise. Parameters nodeThe node which is to be tested. Returns This template returns ’1’ if the specified node is a section (Section, Sect1, Sect2, etc.), and ’0’ otherwise. Namesection.level — Returns the hierarchical level of a section Synopsis<xsl:template name="section.level"> <xsl:param name="node" select="."/> ... </xsl:template> Description This template calculates the hierarchical level of a section. The element sect1 is at level 1, sect2 is at level 2, etc. Recursive sections are calculated down to the fifth level. Parameters nodeThe section node for which the level should be calculated. Defaults to the context node. Returns The section level, “1”, “2”, etc. Nameqanda.section.level — Returns the hierarchical level of a QandASet Synopsis<xsl:template name="qanda.section.level"/> Description This template calculates the hierarchical level of a QandASet. Returns The level, “1”, “2”, etc. Nameselect.mediaobject — Selects and processes an appropriate media object from a list Synopsis<xsl:template name="select.mediaobject"> <xsl:param name="olist" select="imageobject|imageobjectco|videoobject|audioobject|textobject"/> ... </xsl:template> Description This template takes a list of media objects (usually the children of a mediaobject or inlinemediaobject) and processes the "right" object. This template relies on a template named "select.mediaobject.index" to determine which object in the list is appropriate. If no acceptable object is located, nothing happens. Parameters olistThe node list of potential objects to examine. Returns Calls <xsl:apply-templates> on the selected object. Nameselect.mediaobject.index — Selects the position of the appropriate media object from a list Synopsis<xsl:template name="select.mediaobject.index"> <xsl:param name="olist" select="imageobject|imageobjectco|videoobject|audioobject|textobject"/> <xsl:param name="count">1</xsl:param> ... </xsl:template> Description This template takes a list of media objects (usually the children of a mediaobject or inlinemediaobject) and determines the "right" object. It returns the position of that object to be used by the calling template. If the parameter use.role.for.mediaobject is nonzero, then it first checks for an object with a role attribute of the appropriate value. It takes the first of those. Otherwise, it takes the first acceptable object through a recursive pass through the list. This template relies on a template named "is.acceptable.mediaobject" to determine if a given object is an acceptable graphic. The semantics of media objects is that the first acceptable graphic should be used. If no acceptable object is located, no index is returned. Parameters olistThe node list of potential objects to examine. countThe position in the list currently being considered by the recursive process. Returns Returns the position in the original list of the selected object. Nameis.acceptable.mediaobject — Returns ’1’ if the specified media object is recognized Synopsis<xsl:template name="is.acceptable.mediaobject"> <xsl:param name="object"/> ... </xsl:template> Description This template examines a media object and returns ’1’ if the object is recognized as a graphic. Parameters objectThe media object to consider. Returns 0 or 1 Namecheck.id.unique — Warn users about references to non-unique IDs Synopsis<xsl:template name="check.id.unique">
<xsl:param name="linkend"/> ... </xsl:template> Description If passed an ID in linkend, check.id.unique prints a warning message to the user if either the ID does not exist or the ID is not unique. Namecheck.idref.targets — Warn users about incorrectly typed references Synopsis<xsl:template name="check.idref.targets">
<xsl:param name="linkend"/> <xsl:param name="element-list"/> ... </xsl:template> Description If passed an ID in linkend, check.idref.targets makes sure that the element pointed to by the link is one of the elements listed in element-list and warns the user otherwise. Namecopyright.years — Print a set of years with collapsed ranges Synopsis<xsl:template name="copyright.years">
<xsl:param name="years"/> <xsl:param name="print.ranges" select="1"/> <xsl:param name="single.year.ranges" select="0"/> <xsl:param name="firstyear" select="0"/>
<xsl:param name="nextyear" select="0"/> ... </xsl:template> Description This template prints a list of year elements with consecutive years printed as a range. In other words: <year>1992</year> <year>1993</year> <year>1994</year> is printed “1992-1994”, whereas: <year>1992</year> <year>1994</year> is printed “1992, 1994”. This template assumes that all the year elements contain only decimal year numbers, that the elements are sorted in increasing numerical order, that there are no duplicates, and that all the years are expressed in full “century+year” (“1999” not “99”) notation. Parameters yearsThe initial set of year elements. print.rangesIf non-zero, multi-year ranges are collapsed. If zero, all years are printed discretely. single.year.rangesIf non-zero, two consecutive years will be printed as a range, otherwise, they will be printed discretely. In other words, a single year range is “1991-1992” but discretely it’s “1991, 1992”. Returns This template returns the formatted list of years. Namefind.path.params — Search in a table for the "best" match for the node Synopsis<xsl:template name="find.path.params">
<xsl:param name="node" select="."/> <xsl:param name="table" select=""”"/> <xsl:param name="location">
<xsl:call-template name="xpath.location">
<xsl:with-param name="node" select="$node"/>
</xsl:call-template>
</xsl:param>
...
</xsl:template> Description This template searches in a table for the value that most-closely (in the typical best-match sense of XSLT) matches the current (element) node location. Namestring.upper — Converts a string to all uppercase letters Synopsis<xsl:template name="string.upper">
<xsl:param name="string" select="”"”/> ... </xsl:template> Description Given a string, this template does a language-aware conversion of that string to all uppercase letters, based on the values of the lowercase.alpha and uppercase.alpha gentext keys for the current locale. It affects only those characters found in the values of lowercase.alpha and uppercase.alpha. All other characters are left unchanged. Parameters stringThe string to convert to uppercase. Namestring.lower — Converts a string to all lowercase letters Synopsis<xsl:template name="string.lower"> <xsl:param name="string" select="”"”/> ... </xsl:template> Description Given a string, this template does a language-aware conversion

PDF BY DBLATEX

of that string to all lowercase letters, based on the values of the uppercase.alpha and lowercase.alpha gentext keys for the current locale. It affects only those characters found in the values of uppercase.alpha and lowercase.alpha. All other characters are left unchanged. Parameters stringThe string to convert to lowercase. Nameselect.choice.separator — Returns localized choice separator Synopsis<xsl:template name="select.choice.separator"/> Description This template enables auto-generation of an appropriate localized "choice" separator (for example, "and" or "or") before the final item in an inline list (though it could also be useful for generating choice separators for non-inline lists). It currently works by evaluating a processing instruction (PI) of the form <?dbchoice choice="foo"?> : •if the value of the choice pseudo-attribute is "and" or "or", returns a localized "and" or "or" •otherwise returns the literal value of the choice pseudo-attribute The latter is provided only as a temporary workaround because the locale files do not currently have translations for the word or. So if you want to generate a a logical "or" separator in French (for example), you currently need to do this: <?dbchoice choice="ou"?> WarningThe dbchoice processing instruction is an unfortunate hack; support for it may disappear in the future (particularly if and when a more appropriate means for marking up "choice" lists becomes available in DocBook). Nameevaluate.info.profile — Evaluates an info profile Synopsis<xsl:template name="evaluate.info.profile"> <xsl:param name="profile"/> <xsl:param name="info"/> - ... </xsl:template> Description This template evaluates an "info profile" matching the XPath expression given by the profile parameter. It relies on the XSLT evaluate() extension function. The value of the profile parameter can include the literal string $info. If found in the value of the profile parameter, the literal string $info string is replaced with the value of the info parameter, which should be a set of *info nodes; the expression is then evaluated using the XSLT evaluate() extension function. Parameters profileA string representing an XPath expression infoA set of *info nodes Returns Returns a node (the result of evaluating the profile parameter) Namegraphic.format.content-type — Returns mimetype for media format Synopsis<xsl:template - name="graphic.format.content-type"> <xsl:param name="format"/> ... </xsl:template> Description This takes as input a 'format' param and returns a mimetype string. It uses an xsl:choose after first converting the input to all uppercase. IntroductionIntroductionCommon » Refentry Metadata Template Reference$Id: refentry.xsl 7867 2008-03-07 09:54:25Z xmldoc $ Introduction This is technical reference documentation for the "refentry metadata" templates in the DocBook XSL Stylesheets. This is not intended to be user documentation. It is provided for developers writing customization layers for the stylesheets. NoteCurrently, only the manpages stylesheets make use of these templates. They are, however, potentially useful elsewhere. Table of Contentsget.refentry.metadataget.refentry.titleget.refentry.sectionget.refentry.dateget.refentry.sourceget.refentry.source.nameget.refentry.version — Gathers metadata from a refentry and its ancestors Synopsis<xsl:template name="get.refentry.metadata"> <xsl:param name="refname <xsl:param name="info"/> <xsl:param name="prefs"/> ... </xsl:template> Description Reference documentation for particular commands, functions, etc., is sometimes viewed in isolation from its greater "context". For example, users view Unix man pages as, well, individual pages, not as part of a "book" of some kind. Therefore, it is sometimes necessary to embed "context" information in output for each refentry. However, one problem is that different users mark up that context information in different ways. Often (usually), the context information is not actually part of the content of the refentry itself, but instead part of the content of a parent or ancestor element to the refentry. And even then, DocBook provides a variety of elements that users might potentially use to mark up the same kind of information. One user might use the productnumber element to mark up version information about a particular product, while another might use the releaseinfo element. Taking all that in mind, the get.refentry.metadata template tries to gather metadata from a refentry element and its ancestor elements in an intelligent and user-configurable way. The basic mechanism used in the XPath expressions throughout this stylesheet is to select the relevant metadata from the *info element that is closest to the actual refentry – either on the refentry itself, or on its nearest ancestor. NoteThe get.refentry.metadata template is actually just sort of a "driver" template; it calls other templates that do the actual data collection, then returns the data as a set. Parameters refnameThe first refname in the refentry infoA set of info nodes (from a refentry element and its ancestors) prefsA node containing user preferences (from global stylesheet parameters) Returns Returns a node set with the following elements. The descriptions are verbatim from the man(7) man page. titlethe title of the man page (e.g., MAN) sectionthe section number the man page should be placed in (e.g., 7) datethe date of the last revision sourcethe source of the command manualthe title of the manual (e.g., Linux Programmer's Manual) Nameget.refentry.title — Gets title metadata for a refentry Synopsis<xsl:template - name="get.refentry.title"> <xsl:param name="refname"/> ... </xsl:template> Description The man(7) man page describes this as "the title of the man page (e.g., MAN). This differs from refname in that, if the refentry has a refentrytitle, we use that as the title; otherwise, we just use first refname in the first refnamediv in the source. Parameters refnameThe first refname in the refentry Returns Returns a title node. Nameget.refentry.section — Gets section metadata for a refentry Synopsis<xsl:template - name="get.refentry.section"> <xsl:param name="refname"/> <xsl:param name="quiet" select="0"/> ... </xsl:template> Description The man(7) man page describes this as "the section number the man page should be placed in (e.g., 7)". If we do not find a manvolnum specified in the source, and we find that the refentry is for a function, we use the section number 3 ["Library calls (functions within program libraries)"]; otherwise, we default to using 1 ["Executable programs or shell commands"]. Parameters refnameThe first refname in the refentry quietIf non-zero, no "missing" message is emitted Returns Returns a string representing a section number. Nameget.refentry.date — Gets date metadata for a refentry Synopsis<xsl:template name="get.refentry.date"> <xsl:param name="refname"/> <xsl:param name="info"/> <xsl:param name="prefs"/> ... </xsl:template> Description The man(7) man page describes this as "the date of the last revision". If we cannot find a date in the source, we generate one. Parameters refnameThe first refname in the refentry infoA set of info nodes (from a refentry element and its ancestors) prefsA node

containing users preferences (from global stylesheet parameters) Returns Returns a date node. Nameget.refentry.source — Gets source metadata for a refentry Synopsis<xsl:template name="get.refentry.source"> <xsl:param name="refname"/> <xsl:param -name="info"/> <xsl:param name="prefs"/> ... </xsl:template> Description The man(7) man page describes this as "the source of the command", and provides the following examples: •For binaries, use something like: GNU, NET-2, SLS Distribution, MCC Distribution. •For system calls, use the version of the kernel that you are currently looking at: Linux 0.99.11. •For library calls, use the source of the function: GNU, BSD 4.3, Linux DLL 4.4.1. The solbook(5) man page describes something very much like what man(7) calls "source", except that solbook(5) names it "software" and describes it like this: This is the name of the software product that the topic discussed on the reference page belongs to. For example UNIX commands are part of the SunOS x.x release. In practice, there are many pages that simply have a version number in the "source" field. So, it looks like what we have is a two-part field, Name Version, where: Nameproduct name (e.g., BSD) or org. name (e.g., GNU) Versionversion name Each part is optional. If the Name is a product name, then the Version is probably the version of the product. Or there may be no Name, in which case, if there is a Version, it is probably the version of the item itself, not the product it is part of. Or, if the Name is an organization name, then there probably will be no Version. Parameters refnameThe first refname in the refentry infoA set of info nodes (from a refentry element and its ancestors) prefsA node containing users preferences (from global stylesheet parameters) Returns Returns a source node. Nameget.refentry.source.name — Gets source-name metadata for a refentry Synopsis<xsl:template name="get.refentry.source.name"> <xsl:param name="refname"/> <xsl:param -name="info"/> <xsl:param name="prefs"/> ... </xsl:template> Description A "source name" is one part of a (potentially) two-part Name Version source field. For more details, see the documentation for the get.refentry.source template. Parameters refnameThe first refname in the refentry infoA set of info nodes (from a refentry element and its ancestors) prefsA node containing users preferences (from global stylesheet parameters) Returns Depending on what output method is used for the current stylesheet, either returns a text node or possibly an element node, containing "source name" data. Nameget.refentry.version — Gets version metadata for a refentry Synopsis<xsl:template name="get.refentry.version"> <xsl:param name="refname"/> <xsl:param name="info"/> <xsl:param name="prefs"/> ... </xsl:template> Description A "version" is one part of a (potentially) two-part Name Version source field. For more details, see the documentation for the get.refentry.source template. Parameters refnameThe first refname in the refentry infoA set of info nodes (from a refentry element and its ancestors) prefsA node containing users preferences (from global stylesheet parameters) Returns Depending on what output method is used for the current stylesheet, either returns a text node or possibly an element node, containing "version" data. Nameget.refentry.manual — Gets source metadata for a refentry Synopsis<xsl:template name="get.refentry.manual"> <xsl:param name="refname"/> <xsl:param name="info"/> <xsl:param name="prefs"/> ... </xsl:template> Description The man(7) man page describes this as "the title of the manual (e.g., Linux Programmer's Manual)". Here are some examples from existing man pages: •dpkg utilities (dpkg-name) •User Contributed Perl Documentation (GET) •GNU Development Tools (ld) •Emperor Norton Utilities (ddate) •Debian GNU/Linux manual (faked) •GIMP Manual Pages (gimp) •KDOC Documentation System (qt2kdoc) The solbook(5) man page describes something very much like what man(7) calls "manual", except that solbook(5) names it "sectdesc" and describes it like this: This is the section title of the reference page; for example User Commands. Parameters refnameThe first refname in the refentry infoA set of info nodes (from a refentry element and its ancestors) prefsA node containing users preferences (from global stylesheet parameters) Returns Returns a manual node. Nameget.refentry.metadata.prefs — Gets user preferences for refentry metadata gathering Synopsis<xsl:template name="get.refentry.metadata.prefs"/> Description The DocBook XSL stylesheets include several user-configurable global stylesheet parameters for controlling refentry metadata gathering. Those parameters are not read directly by the other refentry metadata-gathering templates. Instead, they are read only by the get.refentry.metadata.prefs template, which assembles them into a structure that is then passed to the other refentry metadata-gathering templates. So the, get.refentry.metadata.prefs template is the only interface to collecting stylesheet parameters for controlling refentry metadata gathering. Parameters There are no local parameters for this template; however, it does rely on a number of global parameters. Returns Returns a manual node. Nameset.refentry.metadata — Sets content of a refentry metadata item Synopsis<xsl:template name="set.refentry.metadata"> <xsl:param name="refname"/> <xsl:param name="info"/> <xsl:param name="contents"/> <xsl:param name="context"/> <xsl:param name="preferred"/> - ... </xsl:template> Description The set.refentry.metadata template is called each time a suitable source element is found for a certain metadata field. Parameters refnameThe first refname in the refentry infoA single *info node that contains the selected source element. contentsA node containing the selected source element. contextA string describing the metadata context in which the set.refentry.metadata template was called: either "date", "source", "version", or "manual". Returns Returns formatted contents of a selected source element. IntroductionIntroductionCommon » Utility Template Reference$Id: utility.xsl 7101 2007-07-20 15:32:12Z xmldoc $ Introduction This is technical reference documentation for the miscellaneous utility templates in the DocBook XSL Stylesheets. NoteThese templates are defined in a separate file from the set of "common" templates because some of the common templates reference DocBook XSL stylesheet parameters, requiring the entire set of parameters to be imported/included in any stylesheet that imports/includes the common templates. The utility templates don't import or include any DocBook XSL stylesheet parameters, so the utility templates can be used without importing the whole set of parameters. This is not intended to be user documentation. It is provided for developers writing customization layers for the stylesheets. Table of Contentslog.messageget.doc.titlepad-stringNamelog.message — Logs/emits format-

PDF BY DBLATEX

ted notes and warnings Synopsis<xsl:template name="log.message"> <xsl:param name="level"/> <xsl:param name="source"/> <xsl:param name="context-desc"/> <xsl:param name="context-desc-field-length">12</xsl:param> <xsl:param name="context-desc-padded"> <xsl:if test="not($context-desc = '')"> <xsl:call-template name="pad-string"> <xsl:with-param name="leftRight">right</xsl:with-param> <xsl:with-param name="padVar" select="substring($context-desc, 1, $context-desc-field-length)"/> <xsl:with-param name="length" select="$context-desc-field-length"/> </xsl:call-template> </xsl:if> </xsl:param> <xsl:param name="message"/> <xsl:param name="message-field-length" select="45"/> <xsl:param name="message-padded"> <xsl:variable name="spaces-for-blank-level"> <!-- * if the level field is blank, we'll need to pad out --> <!-- * the message field with spaces to compensate --> <xsl:choose> <xsl:when test="$level = ''"> <xsl:value-of select="4 + 2"/> <!-- * 4 = hard-coded length of comment text ("Note" or "Warn") --> <!-- * + 2 = length of colon-plus-space separator ": " --> </xsl:when> <xsl:otherwise> <xsl:value-of select="0"/> </xsl:otherwise> </xsl:choose> </xsl:variable> <xsl:variable name="spaces-for-blank-context-desc"> <!-- * if the context-description field is blank, we'll need --> <!-- * to pad out the message field with spaces to compensate --> <xsl:choose> <xsl:when test="$context-desc = ''"> <xsl:value-of select="$context-desc-field-length + 2"/> <!-- * + 2 = length of colon-plus-space separator ": " --> </xsl:when> <xsl:otherwise> <xsl:value-of select="0"/> </xsl:otherwise> </xsl:choose> </xsl:variable> <xsl:variable name="extra-spaces" select="$spaces-for-blank-level + $spaces-for-blank-context-desc"/> <xsl:call-template name="pad-string"> <xsl:with-param name="leftRight">right</xsl:with-param> <xsl:with-param name="padVar" select="substring($message, 1, ($message-field-length + $extra-spaces))"/> <xsl:with-param name="length" select="$message-field-length + $extra-spaces"/> </xsl:call-template> </xsl:param> ... </xsl:template> Description The log.message template is a utility template for logging/emitting formatted messages – that is, notes and warnings, along with a given log "level" and an identifier for the "source" that the message relates to. Parameters levelText to log/emit in the message-level field to indicate the message level (Note or Warning) sourceText to log/emit in the source field to identify the "source" to which the notification/warning relates. This can be any arbitrary string, but because the message lacks line and column numbers to identify the exact part of the source document to which it relates, the intention is that the value you pass into the source parameter should give the user some way to identify the portion of their source document on which to take potentially take action in response to the log message (for example, to edit, change, or add content). So the source value should be, for example, an ID, book/chapter/article title, title of some formal object, or even a string giving an XPath expression. context-descText to log/emit in the context-description field to describe the context for the message. context-desc-field-lengthSpecifies length of the context-description field (in characters); default is 12 If the text specified by the context-desc parameter is longer than the number of characters specified in context-desc-field-length, it is truncated to context-desc-field-length (12 characters by default). If the specified text is shorter than context-desc-field-length, it is right-padded out to context-desc-field-length (12 by default). If no value has been specified for the context-desc parameter, the field is left empty and the text of the log message begins with the value of the message parameter. messageText to log/emit in the actual message field message-field-lengthSpecifies length of the message field (in characters); default is 45 Returns Outputs a message (generally, to standard error). Nameget.doc.title — Gets a title from the current document Synopsis<xsl:template name="get.doc.title"/> Description The get.doc.title template is a utility template for returning the first title found in the current document. Returns Returns a string containing some identifying title for the current document . Namepad-string — Right-pads or left-pads a string out to a certain length Synopsis<xsl:template name="pad-string"> <xsl:param name="padChar" select="' '"/> <xsl:param name="leftRight">left</xsl:param> <xsl:param name="padVar"/> <xsl:param name="length"/> ... </xsl:template> Description This function takes string padVar and pads it out in the direction rightLeft to the string-length length, using string padChar (a space character by default) as the padding string (note that padChar can be a string; it is not limited to just being a single character). NoteThis function began as a copy of Nate Austin's prepend-pad function in the Padding Content11 http://www.dpawson.co.uk/xsl/sect2/padding.html section of Dave Pawson's XSLT FAQ22 http://www.dpawson.co.uk/xsl/index.html. Returns Returns a (padded) string. IntroductionIntroductionCommon » Character-Map Template Reference$Id: charmap.xsl 7266 2007-08-22 11:58:42Z xmldoc $ Introduction This is technical reference documentation for the character-map templates in the DocBook XSL Stylesheets. NoteThese templates are defined in a separate file from the set of "common" templates because some of the common templates reference DocBook XSL stylesheet parameters, requiring the entire set of parameters to be imported/included in any stylesheet that imports/includes the common templates. The character-map templates don't import or include any DocBook XSL stylesheet parameters, so the character-map templates can be used without importing the whole set of parameters. This is not intended to be user documentation. It is provided for developers writing customization layers for the stylesheets. Table of Contentsapply-character-mapread-character-mapNameapply-character-map — Applies an XSLT character map Synopsis<xsl:template name="apply-character-map"> <xsl:param name="content"/> <xsl:param name="map.contents"/> ... </xsl:template> Description This template applies an XSLT character map11 http://www.w3.org/TR/xslt20/#character-maps; that is, it causes certain individual characters to be substituted with strings of one or more characters. It is useful mainly for replacing multiple "special" characters or symbols in the same target content. It uses the value of map.contents to do substitution on content, and then returns the modified contents. NoteThis template is a very slightly modified version of Jeni Tennison's replace_strings template in the multiple string replacements22 http://www.dpawson.co.uk/xsl/sect2/StringReplace.html#d9351e13 section of Dave Pawson's XSLT

PDF BY DBLATEX

FAQ33 http://www.dpawson.co.uk/xsl/index.html. The apply-string-subst-map template is essentially the same template as the apply-character-map template; the only difference is that in the map that apply-string-subst-map expects, oldstring and newstring attributes are used instead of character and string attributes. Parameters contentThe content on which to perform the character-map substitution. map.contentsA node set of elements, with each element having the following attributes: •character, a character to be replaced •string, a string with which to replace character Nameread-character-map — Reads in all or part of an XSLT character map Synopsis<xsl:template name="read-character-map"> <xsl:param name="use.subset"/> <xsl:param name="subset.profile"/> <xsl:param name="uri"/> ... </xsl:template> Description The XSLT 2.0 specification describes character maps11 http://www.w3.org/TR/xslt20/#character-maps and explains how they may be used to allow a specific character appearing in a text or attribute node in a final result tree to be substituted by a specified string of characters during serialization. The read-character-map template provides a means for reading and using character maps with XSLT 1.0-based tools. This template reads the character-map contents from uri (in full or in part, depending on the value of the use.subset parameter), then passes those contents to the apply-character-map template, along with content, the data on which to perform the character substitution. Using the character map “in part” means that it uses only those output-character elements that match the XPath expression given in the value of the subset.profile parameter. The current implementation of that capability here relies on the evaluate extension XSLT function. Parameters use.subsetSpecifies whether to use a subset of the character map instead of the whole map; boolean 0 or 1 subset.profileXPath expression that specifies what subset of the character map to use uriURI for a character map Part III. Formatting Object Table ReferenceIntroduction This is technical reference documentation for the FO table-processing templates in the DocBook XSL Stylesheets. This is not intended to be user documentation. It is provided for developers writing customization layers for the stylesheets. calc.column.widthcalc.column.widthNamecalc.column.width — Calculate an XSL FO table column width specification from a CALS table column width specification. Synopsis<xsl:template name="calc.column.width"> <xsl:param name="colwidth">1*</xsl:param> ... </xsl:template> Description CALS expresses table column widths in the following basic forms: •99.99units, a fixed length specifier. •99.99, a fixed length specifier without any units. •99.99*, a relative length specifier. •99.99*+99.99units, a combination of both. The CALS units are points (pt), picas (pi), centimeters (cm), millimeters (mm), and inches (in). These are the same units as XSL, except that XSL abbreviates picas "pc" instead of "pi". If a length specifier has no units, the CALS default unit (pt) is assumed. Relative length specifiers are represented in XSL with the proportional-column-width() function. Here are some examples: •"36pt" becomes "36pt" •"3pi" becomes "3pc" •"36" becomes "36pt" •"3*" becomes "proportional-column-width(3)" •"3*+2pi" becomes "proportional-column-width(3)+2pc" •"1*+2" becomes "proportional-column-width(1)+2pt" Parameters colwidthThe CALS column width specification. Returns The XSL column width specification. Part IV. Titlepage Template Stylesheet ReferenceIntroduction This is technical reference documentation for the “titlepage” templates in the DocBook XSL Stylesheets. This is not intended to be user documentation. It is provided for developers writing customization layers for the stylesheets. t:templatest:templatesNamet:templates — Construct a stylesheet for the templates provided Synopsis<xsl:template match="t:templates"/> Description The t:templates element is the root of a set of templates. This template creates an appropriate xsl:stylesheet for the templates. If the t:templates element has a base-stylesheet attribute, an xsl:import statement is constructed for it. xsl:*xsl:*Namexsl:* — Copy xsl: elements straight through Synopsis<xsl:template match="xsl:*"/> Description This template simply copies the xsl: elements straight through into the result tree. t:titlepaget:titlepageNamet:titlepage — Create the templates necessary to construct a title page Synopsis<xsl:template match="t:titlepage"/> Description The t:titlepage element creates a set of templates for processing the titlepage for an element. The “root” of this template set is the template named “wrapper.titlepage”. That is the template that should be called to generate the title page. The t:titlepage element has three attributes: elementThe name of the source document element for which these templates apply. In other words, to make a title page for the article element, set the element attribute to “article”. This attribute is required. wrapperThe entire title page can be wrapped with an element. This attribute identifies that element. classIf the class attribute is set, a class attribute with this value will be added to the wrapper element that surrounds the entire title page. Any other attributes are copied through literally to the wrapper element. The content of a t:titlepage is one or more t:titlepage-content, t:titlepage-separator, and t:titlepage-before elements. Each of these elements may be provided for the “recto” and “verso” sides of the title page. @* (in copy.literal.atts mode)@* (in copy.literal.atts mode)Name@* (in copy.literal.atts mode) — Copy t:titlepage attributes Synopsis<xsl:template match="@*" mode="copy.literal.atts"/> Description This template copies all of the “other” attributes from a t:titlepage element onto the specified wrapper. t:titlepage-contentt:titlepage-contentNamet:titlepage-content — Create templates for the content of one side of a title page Synopsis<xsl:template match="t:titlepage-content"/> Description The title page content, that is, the elements from the source document that are rendered on the title page, can be controlled independently for the recto and verso sides of the title page. The t:titlepage-content element has two attributes: sideIdentifies the side of the page to which this title page content applies. The side attribute is required and must be set to either “recto” or “verso”. In addition, you must specify exactly one t:titlepage-content for each side within each t:titlepage. orderIndicates how the order of the elements presented on the title page is determined. If the order is “document”, the elements are presented in document order. Otherwise (if the order is “stylesheet”), the elements are presented in the order that they appear in the template (and consequently in the stylesheet). The content of a t:titlepage-content element is a list of element names. These names should be unqualified. They identify the elements in the source document that should appear on the title page. Each element may have a single attribute: predicate. The value of this attribute is used as a predicate for the expression that

PDF BY DBLATEX

matches the element on which it occurs. In other words, to put only the first three authors on the recto-side of a title page, you could specify: <t:titlepage-contents side="recto"> <!-- other titlepage elements --> <author predicate="[count(previous-sibling::author)<2]"/> <!-- other titlepage elements --> </t:titlepage-contents> Usually, the elements so named are empty. But it is possible to make one level of selection within them. Suppose that you want to process authorgroup elements on the title page, but you want to select only proper authors, editors, or corporate authors, not collaborators or other credited authors. In that case, you can put a t:or group inside the authorgroup element: <t:titlepage-contents side="recto"> <!-- other titlepage elements --> <authorgroup> <t:or> <author/> <editor/> <corpauthor/> </t:or> </authorgroup> <!-- other - titlepage elements --> </t:titlepage-contents> This will have the effect of automatically generating a template for processing authorgroups in the title page mode, selecting only the specified children. If you need more complex processing, you'll have to construct the templates by hand. t:titlepage-separatort:titlepage-separatorNamet:titlepage-separator — Create templates for the separator Synopsis<xsl:template match="t:titlepage-separator"/> Description The title page is separated from the content which follows it by the markup specified in the t:titlepage-separator element. t:titlepage-beforet:titlepage-beforeNamet:titlepage-before — Create templates for what precedes a title page Synopsis<xsl:template match="t:titlepage-before"/> Description Each side of the title page is preceded by the markup specified in the t:titlepage-before element for that side. * (in copy mode)* (in copy mode)Name* (in copy mode) — Copy elements Synopsis<xsl:template match="*" mode="copy"/> Description This template simply copies the elements that it applies to straight through into the result tree. @* (in copy mode)@* (in copy mode)Name@* (in copy mode) — Copy attributes Synopsis<xsl:template match="@*" mode="copy"/> Description This template simply copies the attributes that it applies to straight through into the result tree. * (in document.order mode)* (in document.order mode)Name* (in document.order mode) — Create rules to process titlepage elements in document order Synopsis<xsl:template match="*" mode="document.order"/> Description This template is called to process all of the children of the t:titlepage-content element. It creates the hairy select expression necessary to process each of those elements in the title page. Note that this template automatically handles the case where some DocBook elements, like title and subtitle, can occur both inside the *info elements where metadata is usually stored and outside. It also automatically calculates the name for the *info container and handles elements that have historically had containers with different names. * (in document.order mode)* (in document.order mode)Name* (in document.order mode) — Create rules to process titlepage elements in stylesheet order Synopsis<xsl:template match="*" mode="document.order"/> Description This template is called to process all of the children of the t:titlepage-content element. It creates the set of xsl:apply-templates elements necessary process each of those elements in the title page. Note that this template automatically handles the case where some DocBook elements, like title and subtitle, can occur both inside the *info elements where metadata is usually stored and outside. It also automatically calculates the name for the *info container and handles elements that have historically had containers with different names. * (in titlepage.specialrules mode)* (in titlepage.specialrules mode)Name* (in titlepage.specialrules mode) — Create templates for special rules Synopsis<xsl:template match="*" mode="titlepage.specialrules"/> Description This template is called to process all of the descendants of the t:titlepage-content element that require special processing. At present, that's just t:or elements. * (in titlepage.subrules mode)* (in titlepage.subrules mode)Name* (in titlepage.subrules mode) — Create template for individual special rules Synopsis<xsl:template match="*" mode="titlepage.subrules"/> Description This template is called to process the children of special template elements. t:ort:orNamet:or — Process the t:or special rule Synopsis<xsl:template match="t:or"/><xsl:template - match="t:or" mode="titlepage.subrules"/> Description This template processes t:or. t:or (in titlepage.subrules mode)t:or (in titlepage.subrules mode)Namet:or (in titlepage.subrules mode) — Process the t:or special rule in titlepage.subrules mode Synopsis<xsl:template match="t:or" mode="titlepage.subrules"/> Description The titlepage.subrules mode doesn't apply to t:or, so just reprocess this node in the normal mode. element-or-listelement-or-listNameelement-or-list — Construct the "or-list" used in the select attribute for special rules. Synopsis<xsl:template name="element-or-list"> <xsl:param name="elements" select="*"/> <xsl:param name="element.count" select="count($elements)"/> <xsl:param name="count" select="1"/> <xsl:param name="orlist"/> ... </xsl:template> Description Walk through each of the children of t:or, producing the text of the select attribute.

PDF BY DBLATEX