

# Contents

<b>Introduction</b> . . . . .	<b>xxix</b>
<b>List of Examples</b> . . . . .	<b>.xxxix</b>

## **Part I: Foundations**

### **Chapter 1: XSLT in Context** **3**

---

<b>What Is XSLT?</b> . . . . .	<b>3</b>
Why Version 2.0? . . . . .	5
A Scenario: Transforming Music . . . . .	5
<b>How Does XSLT Transform XML?</b> . . . . .	<b>7</b>
XSLT and SQL: An Analogy . . . . .	8
XSLT Processors . . . . .	9
A Simple XSLT Stylesheet . . . . .	10
An XSLT 2.0 Stylesheet . . . . .	19
<b>The Place of XSLT in the XML Family</b> . . . . .	<b>21</b>
XSLT and XSL Formatting Objects . . . . .	22
XSLT and XPath . . . . .	22
XSLT and XML Namespaces . . . . .	23
XSLT and CSS . . . . .	24
XSLT and XML Schemas . . . . .	24
XSLT and XQuery . . . . .	26
<b>The History of XSL</b> . . . . .	<b>26</b>
Prehistory . . . . .	26
The First XSL Proposal . . . . .	28
Saxon . . . . .	30
Beyond XSLT 1.0 . . . . .	30
Convergence with XQuery . . . . .	31
The Development of XSLT 2.0 and XPath 2.0 . . . . .	32
<b>XSLT 2.0 as a Language</b> . . . . .	<b>33</b>
Use of XML Syntax . . . . .	33
No Side Effects . . . . .	34
Rule-Based . . . . .	35
Types Based on XML Schema . . . . .	38
A Two-Language System: XSLT and XPath . . . . .	39
<b>Summary</b> . . . . .	<b>40</b>

# Contents

---

<b>Chapter 2: The XSLT Processing Model</b>	<b>41</b>
<b>XSLT: A System Overview</b>	<b>41</b>
A Simplified Overview	41
Trees, Not Documents	42
Different Output Formats	43
Multiple Inputs and Outputs	44
<b>The XDM Tree Model</b>	<b>45</b>
XML as a Tree	45
Completing the UML Class Diagram	56
Document Order	57
Names and Namespaces	58
IDs and IDREFs	61
Characters in the Data Model	62
What Does the Tree Leave Out?	64
From Textual XML to a Data Model	65
Controlling Serialization	67
<b>The Transformation Process</b>	<b>67</b>
Invoking a Transformation	67
Template Rules	68
Push Processing	74
Controlling Which Nodes to Process	76
Modes	78
Built-In Template Rules	78
Conflict Resolution Policy	79
<b>Error Handling</b>	<b>80</b>
<b>Variables and Expressions</b>	<b>80</b>
Variables	81
Parameters	82
Expressions	82
Context	84
Temporary Documents	85
<b>Summary</b>	<b>88</b>
<b>Chapter 3: Stylesheet Structure</b>	<b>89</b>
<b>Changes in XSLT 2.0</b>	<b>90</b>
<b>The Modular Structure of a Stylesheet</b>	<b>90</b>
<b>The &lt;xsl:stylesheet&gt; Element</b>	<b>98</b>
<b>The &lt;?xml-stylesheet?&gt; Processing Instruction</b>	<b>99</b>
<b>Embedded Stylesheets</b>	<b>102</b>

---

<b>Declarations</b> .....	<b>104</b>
XSLT-Defined Declarations .....	105
Implementor-Defined Declarations .....	106
User-Defined Top-Level Elements .....	107
<b>Instructions</b> .....	<b>108</b>
XSLT Instructions .....	108
Extension Instructions .....	111
Literal Result Elements .....	112
Attribute Value Templates .....	122
<b>Simplified Stylesheets</b> .....	<b>125</b>
<b>Writing Portable Stylesheets</b> .....	<b>127</b>
Conditional Compilation .....	127
Version Compatibility .....	128
Extensibility .....	134
<b>Whitespace</b> .....	<b>141</b>
The Effect of Stripping Whitespace Nodes .....	146
Whitespace Nodes in the Stylesheet .....	146
Solving Whitespace Problems .....	147
<b>Summary</b> .....	<b>148</b>
<b>Chapter 4: Stylesheets and Schemas</b> .....	<b>151</b>
<b>XML Schema: An Overview</b> .....	<b>151</b>
Simple Type Definitions .....	152
Elements with Attributes and Simple Content .....	154
Elements with Mixed Content .....	155
Elements with Element-Only Content .....	156
Defining a Type Hierarchy .....	157
Substitution Groups .....	158
<b>Declaring Types in XSLT</b> .....	<b>161</b>
<b>Validating the Source Document</b> .....	<b>165</b>
<b>Validating the Result Document</b> .....	<b>170</b>
<b>Validating a Temporary Document</b> .....	<b>174</b>
<b>Validating Individual Elements</b> .....	<b>176</b>
<b>Validating Individual Attributes</b> .....	<b>179</b>
<b>The default-validation Attribute</b> .....	<b>180</b>
<b>Importing Schemas</b> .....	<b>180</b>
<b>Using xsi:type</b> .....	<b>181</b>
<b>Nullability</b> .....	<b>182</b>
<b>Summary</b> .....	<b>183</b>

## Contents

---

<b>Chapter 5: Types</b>	<b>185</b>
<b>What Is a Type System?</b>	<b>185</b>
<b>Changes in 2.0</b>	<b>186</b>
<b>Sequences</b>	<b>187</b>
<b>Atomic Values</b>	<b>189</b>
<b>Atomic Types</b>	<b>191</b>
The Major Atomic Types	193
The Minor Atomic Types	206
Derived Numeric Types	210
Derived String Types	213
Untyped Atomic Values	215
xs:NMTOKENS, xs:IDREFS, and xs:ENTITIES	217
<b>Schema Types and XPath Types</b>	<b>217</b>
<b>The Type Matching Rules</b>	<b>219</b>
<b>Static and Dynamic Type Checking</b>	<b>221</b>
<b>Summary</b>	<b>224</b>

## Part II: XSLT and XPath Reference

<b>Chapter 6: XSLT Elements</b>	<b>227</b>
<b>xsl:analyze-string</b>	<b>230</b>
<b>xsl:apply-imports</b>	<b>237</b>
<b>xsl:apply-templates</b>	<b>240</b>
<b>xsl:attribute</b>	<b>254</b>
<b>xsl:attribute-set</b>	<b>266</b>
<b>xsl:call-template</b>	<b>271</b>
<b>xsl:character-map</b>	<b>280</b>
<b>xsl:choose</b>	<b>282</b>
<b>xsl:comment</b>	<b>285</b>
<b>xsl:copy</b>	<b>287</b>
<b>xsl:copy-of</b>	<b>292</b>
<b>xsl:decimal-format</b>	<b>298</b>
<b>xsl:document</b>	<b>303</b>
<b>xsl:element</b>	<b>306</b>
<b>xsl:fallback</b>	<b>316</b>
<b>xsl:for-each</b>	<b>322</b>
<b>xsl:for-each-group</b>	<b>326</b>
<b>xsl:function</b>	<b>344</b>
<b>xsl:if</b>	<b>353</b>

---

<b>xsl:import</b> . . . . .	<b>357</b>
<b>xsl:import-schema</b> . . . . .	<b>368</b>
<b>xsl:include</b> . . . . .	<b>372</b>
<b>xsl:key</b> . . . . .	<b>376</b>
<b>xsl:matching-substring</b> . . . . .	<b>386</b>
<b>xsl:message</b> . . . . .	<b>386</b>
<b>xsl:namespace</b> . . . . .	<b>390</b>
<b>xsl:namespace-alias</b> . . . . .	<b>394</b>
<b>xsl:next-match</b> . . . . .	<b>399</b>
<b>xsl:non-matching-substring</b> . . . . .	<b>402</b>
<b>xsl:number</b> . . . . .	<b>403</b>
<b>xsl:otherwise</b> . . . . .	<b>420</b>
<b>xsl:output</b> . . . . .	<b>420</b>
<b>xsl:output-character</b> . . . . .	<b>424</b>
<b>xsl:param</b> . . . . .	<b>425</b>
<b>xsl:perform-sort</b> . . . . .	<b>437</b>
<b>xsl:preserve-space</b> . . . . .	<b>439</b>
<b>xsl:processing-instruction</b> . . . . .	<b>442</b>
<b>xsl:result-document</b> . . . . .	<b>445</b>
<b>xsl:sequence</b> . . . . .	<b>452</b>
<b>xsl:sort</b> . . . . .	<b>455</b>
<b>xsl:strip-space</b> . . . . .	<b>465</b>
<b>xsl:stylesheet</b> . . . . .	<b>465</b>
<b>xsl:template</b> . . . . .	<b>483</b>
<b>xsl:text</b> . . . . .	<b>492</b>
<b>xsl:transform</b> . . . . .	<b>495</b>
<b>xsl:value-of</b> . . . . .	<b>495</b>
<b>xsl:variable</b> . . . . .	<b>500</b>
<b>xsl:when</b> . . . . .	<b>515</b>
<b>xsl:with-param</b> . . . . .	<b>517</b>
<b>Summary</b> . . . . .	<b>519</b>
<b>Chapter 7: XPath Fundamentals</b> . . . . .	<b>521</b>
<b>Notation</b> . . . . .	<b>522</b>
<b>Where to Start</b> . . . . .	<b>523</b>
<b>Expressions</b> . . . . .	<b>524</b>
Examples . . . . .	527
<b>Lexical Constructs</b> . . . . .	<b>527</b>
Comments . . . . .	529
Numeric Literals . . . . .	530

## Contents

---

String Literals .....	532
Names .....	534
Operators .....	537
<b>Primary Expressions .....</b>	<b>539</b>
Examples .....	540
<b>Variable References .....</b>	<b>540</b>
Usage .....	540
Examples .....	541
<b>Parenthesized Expressions .....</b>	<b>542</b>
Changes in XPath 2.0 .....	543
<b>Context Item Expressions .....</b>	<b>543</b>
Changes in XPath 2.0 .....	544
Usage .....	544
<b>Function Calls .....</b>	<b>544</b>
Identifying the Function to be Called .....	545
Converting the Arguments and the Result .....	547
Changes in XPath 2.0 .....	549
Side Effects .....	549
Examples .....	550
<b>Conditional Expressions .....</b>	<b>551</b>
Changes in XPath 2.0 .....	552
Examples .....	553
<b>The XPath Evaluation Context .....</b>	<b>553</b>
The Static Context .....	554
The Dynamic Context .....	563
<b>Summary .....</b>	<b>568</b>
<b>Chapter 8: XPath: Operators on Items .....</b>	<b>571</b>
<b>Arithmetic Operators .....</b>	<b>571</b>
Syntax .....	571
Type Promotion .....	572
Changes in XPath 2.0 .....	573
Effect .....	573
Arithmetic Using Numbers .....	574
Examples of Numeric Arithmetic .....	576
Arithmetic Using Durations .....	577
<b>Value Comparisons .....</b>	<b>581</b>
Permitted Operand Types .....	582
Type Checking for Value Comparisons .....	586
Examples of Value Comparisons .....	587

<b>General Comparisons</b> .....	<b>588</b>
Changes in XPath 2.0 .....	588
Rules for General Comparisons .....	589
Existential Comparison .....	590
Examples of General Comparisons .....	592
<b>Node Comparisons</b> .....	<b>593</b>
The «is»Operator .....	593
The operators «<<» and «<<» .....	594
Changes in XPath 2.0 .....	594
<b>Boolean Expressions</b> .....	<b>594</b>
Shortcut Semantics .....	595
Examples .....	596
<b>Summary</b> .....	<b>596</b>
<b>Chapter 9: XPath: Path Expressions</b> .....	<b>599</b>
<b>Examples of Path Expressions</b> .....	<b>600</b>
<b>Changes in XPath 2.0</b> .....	<b>601</b>
<b>Document Order and Duplicates</b> .....	<b>602</b>
<b>The Binary «/» Operator</b> .....	<b>602</b>
Syntax .....	602
Effect .....	603
Examples of the Binary «/» Operator .....	604
Associativity of the «/» Operator .....	605
<b>Axis Steps</b> .....	<b>606</b>
Syntax of Axis Steps .....	606
Effect .....	607
Examples of Axis Steps .....	608
Axes .....	609
Node Tests .....	613
Name Tests .....	614
Kind Tests .....	616
Predicates .....	617
Abbreviated Axis Steps .....	621
<b>Rooted Path Expressions</b> .....	<b>625</b>
Syntax .....	625
Examples of Rooted Paths .....	626
<b>The «//» Abbreviation</b> .....	<b>626</b>
Examples Using «//» .....	627
Comparing «//» with «/descendant::» .....	628
<b>Combining Sets of Nodes</b> .....	<b>628</b>
Syntax .....	629

## Contents

---

Examples .....	630
Usage .....	630
Set Intersection and Difference in XPath 1.0 .....	631
Sets of Atomic Values .....	631
<b>Summary .....</b>	<b>632</b>
<b>Chapter 10: XPath: Sequence Expressions .....</b>	<b>633</b>
<b>The Comma Operator .....</b>	<b>634</b>
Examples .....	635
<b>Numeric Ranges: The «to» Operator .....</b>	<b>636</b>
Examples .....	637
<b>Filter Expressions .....</b>	<b>638</b>
Examples .....	640
<b>The «for» Expression .....</b>	<b>640</b>
Mapping a Sequence .....	641
Examples .....	642
The Context Item in a «for» Expression .....	642
Combining Multiple Sequences .....	643
Example .....	644
<b>Simple Mapping Expressions .....</b>	<b>644</b>
<b>The «some» and «every» Expressions .....</b>	<b>646</b>
Examples .....	648
Quantification and the «=» Operator .....	649
Errors in «some» and «every» Expressions .....	649
<b>Summary .....</b>	<b>651</b>
<b>Chapter 11: XPath: Type Expressions .....</b>	<b>653</b>
<b>Converting Atomic Values .....</b>	<b>654</b>
Converting between Primitive Types .....	656
Converting between Derived Types .....	664
<b>Sequence Type Descriptors .....</b>	<b>668</b>
Matching Atomic Values .....	669
Matching Nodes .....	670
Matching Elements and Attributes .....	672
<b>The «instance of» Operator .....</b>	<b>677</b>
<b>The «treat as» Operator .....</b>	<b>678</b>
<b>Summary .....</b>	<b>680</b>
<b>Chapter 12: XSLT Patterns .....</b>	<b>681</b>
<b>Patterns and Expressions .....</b>	<b>681</b>
<b>Changes in XSLT 2.0 .....</b>	<b>682</b>

---

<b>The Formal Definition</b> .....	<b>683</b>
Applying the Definition in Practice .....	684
An Algorithm for Matching Patterns .....	685
Patterns Containing Predicates .....	685
<b>An Informal Definition</b> .....	<b>685</b>
<b>Conflict Resolution</b> .....	<b>686</b>
<b>Matching Parentless Nodes</b> .....	<b>688</b>
<b>The Syntax of Patterns</b> .....	<b>689</b>
Pattern .....	689
PathPattern .....	690
RelativePathPattern .....	693
PatternStep .....	694
IdKeyPattern .....	704
<b>Summary</b> .....	<b>708</b>
<b>Chapter 13: The Function Library</b> .....	<b>709</b>
<b>A Word about Naming</b> .....	<b>710</b>
<b>Functions by Category</b> .....	<b>710</b>
<b>Notation</b> .....	<b>712</b>
<b>Code Samples</b> .....	<b>714</b>
<b>Function Definitions</b> .....	<b>714</b>
abs .....	714
adjust-date-to-timezone, adjust-dateTime-to-timezone, adjust-time-to-timezone .....	715
avg .....	718
base-uri .....	719
boolean .....	721
ceiling .....	723
codepoint-equal .....	724
codepoints-to-string .....	725
collection .....	726
compare .....	727
concat .....	729
contains .....	730
count .....	733
current .....	734
current-date, current-dateTime, current-time .....	738
current-group .....	739
current-grouping-key .....	740
current-Time .....	741
data .....	741
dateTime .....	743
day-from-date, day-from-dateTime .....	744

## Contents

---

days-from-duration .....	745
deep-equal .....	745
default-collation .....	748
distinct-values .....	749
doc, doc-available .....	750
document .....	754
document-uri .....	764
element-available .....	764
empty .....	770
encode-for-uri .....	771
ends-with .....	773
error .....	774
escape-html-uri .....	775
exactly-one .....	777
exists .....	778
false .....	779
floor .....	779
format-date, format-dateTime, format-time .....	781
format-number .....	788
format-time .....	792
function-available .....	792
generate-id .....	797
hours-from-dateTime, hours-from-time .....	800
hours-from-duration .....	801
id .....	802
idref .....	804
implicit-timezone .....	806
index-of .....	807
in-scope-prefixes .....	808
insert-before .....	810
iri-to-uri .....	811
key .....	812
lang .....	819
last .....	820
local-name .....	824
local-name-from-QName .....	826
lower-case .....	827
matches .....	828
max, min .....	830
min .....	832
minutes-from-dateTime, minutes-from-time .....	832
minutes-from-duration .....	832

## Contents

---

minutes-from-time	833
month-from-date, month-from-dateTime	833
months-from-duration	834
name	835
namespace-uri	837
namespace-uri-for-prefix	839
namespace-uri-from-QName	841
nilled	842
node-name	843
normalize-space	845
normalize-unicode	847
not	850
number	851
one-or-more	853
position	854
prefix-from-QName	857
QName	858
regex-group	860
remove	861
replace	862
resolve-QName	864
resolve-uri	867
reverse	869
root	870
round	870
round-half-to-even	872
seconds-from-dateTime, seconds-from-time	873
seconds-from-duration	874
seconds-from-time	875
starts-with	875
static-base-uri	876
string	877
string-join	879
string-length	880
string-to-codepoints	881
subsequence	882
substring	883
substring-after	885
substring-before	887
sum	889
system-property	890
timezone-from-date, timezone-from-dateTime, timezone-from-time	893

## Contents

---

tokenize . . . . .	894
trace . . . . .	896
translate . . . . .	897
true . . . . .	899
type-available . . . . .	899
unordered . . . . .	901
unparsed-entity-public-id, unparsed-entity-uri . . . . .	902
unparsed-text, unparsed-text-available . . . . .	904
upper-case . . . . .	910
year-from-date, year-from-dateTime . . . . .	911
years-from-duration . . . . .	911
zero-or-one . . . . .	912
<b>Summary . . . . .</b>	<b>913</b>
<b>Chapter 14: Regular Expressions . . . . .</b>	<b>915</b>
<b>Branches and Pieces . . . . .</b>	<b>916</b>
<b>Quantifiers . . . . .</b>	<b>916</b>
<b>Atoms . . . . .</b>	<b>917</b>
<b>Subexpressions . . . . .</b>	<b>918</b>
<b>Back-References . . . . .</b>	<b>918</b>
<b>Character Groups . . . . .</b>	<b>919</b>
<b>Character Ranges . . . . .</b>	<b>919</b>
<b>Character Class Escapes . . . . .</b>	<b>920</b>
<b>Character Blocks . . . . .</b>	<b>922</b>
<b>Character Categories . . . . .</b>	<b>924</b>
<b>Flags . . . . .</b>	<b>925</b>
The «i» flag . . . . .	925
The «m» flag . . . . .	926
The «s» flag . . . . .	926
The «x» flag . . . . .	926
<b>Disallowed Constructs . . . . .</b>	<b>927</b>
<b>Summary . . . . .</b>	<b>927</b>
<b>Chapter 15: Serialization . . . . .</b>	<b>929</b>
<b>The XML Output Method . . . . .</b>	<b>929</b>
<b>The HTML Output Method . . . . .</b>	<b>936</b>
<b>The XHTML Output Method . . . . .</b>	<b>939</b>
<b>The Text Output Method . . . . .</b>	<b>940</b>

<b>Using the &lt;xsl:output&gt; declaration</b>	<b>940</b>
<b>Character Maps</b>	<b>941</b>
Usage	942
Choosing Characters to Map	942
Limitations of Character Maps	944
<b>Disable Output Escaping</b>	<b>945</b>
Reasons to Disable Output Escaping	945
Why disable-output-escaping Is Deprecated	946
Using disable-output-escaping to Wrap HTML in CDATA	947
Character Maps as a Substitute for disable-output-escaping	948
<b>Summary</b>	<b>949</b>

## Part III: Exploitation

### **Chapter 16: Extensibility** **953**

<b>What Vendor Extensions Are Allowed?</b>	<b>954</b>
<b>Extension Functions</b>	<b>955</b>
When Are Extension Functions Needed?	955
When Are Extension Functions Not Needed?	956
Calling Extension Functions	956
What Language Is Best?	957
Client-Side Script	957
Binding Extension Functions	957
XPath Trees and the DOM	963
Calling External Functions within a Loop	965
Functions with Uncontrolled Side Effects	967
<b>Keeping Extensions Portable</b>	<b>970</b>
<b>Summary</b>	<b>971</b>

### **Chapter 17: Stylesheet Design Patterns** **973**

<b>Fill-in-the-Blanks Stylesheets</b>	<b>973</b>
<b>Navigational Stylesheets</b>	<b>976</b>
<b>Rule-Based Stylesheets</b>	<b>980</b>
<b>Computational Stylesheets</b>	<b>985</b>
Programming without Assignment Statements	985
So Why Are They Called Variables?	989
Avoiding Assignment Statements	989
<b>Summary</b>	<b>1000</b>

## Contents

---

<b>Chapter 18: Case Study: XMLSpec</b>	<b>1001</b>
<b>Formatting the XML Specification</b>	<b>1002</b>
<b>Preface</b>	<b>1004</b>
<b>Creating the HTML Outline</b>	<b>1008</b>
<b>Formatting the Document Header</b>	<b>1012</b>
<b>Creating the Table of Contents</b>	<b>1019</b>
<b>Creating Section Headers</b>	<b>1023</b>
<b>Formatting the Text</b>	<b>1024</b>
<b>Producing Lists</b>	<b>1028</b>
<b>Making Cross-References</b>	<b>1029</b>
<b>Setting Out the Production Rules</b>	<b>1033</b>
<b>Overlay Stylesheets</b>	<b>1041</b>
diffspec.xsl	1041
REC-xml.xsl	1044
<b>Stylesheets for Other Specifications</b>	<b>1044</b>
xslt.xsl	1045
xsltdiff.xsl	1046
funcproto.xsl	1046
xsl-query.xsl	1047
xmlspec.xsl	1047
<b>Summary</b>	<b>1047</b>
<b>Chapter 19: Case Study: A Family Tree</b>	<b>1049</b>
<b>Modeling a Family Tree</b>	<b>1050</b>
The GEDCOM Data Model	1050
Creating a Schema for GEDCOM 6.0	1053
The GEDCOM 6.0 Schema	1054
<b>Creating a Data File</b>	<b>1058</b>
Converting GEDCOM Files to XML	1059
Converting from GEDCOM 5.5 to 6.0	1063
<b>Displaying the Family Tree Data</b>	<b>1072</b>
The Stylesheet	1073
Putting It Together	1085
<b>Summary</b>	<b>1098</b>
<b>Chapter 20: Case Study: Knight's Tour</b>	<b>1099</b>
<b>The Problem</b>	<b>1099</b>
<b>The Algorithm</b>	<b>1100</b>
<b>Placing the Knight</b>	<b>1104</b>
<b>Displaying the Final Board</b>	<b>1105</b>

<b>Finding the Route</b> .....	<b>1106</b>
Finding the Possible Moves .....	1107
Trying the Possible Moves .....	1109
Selecting the Best Move .....	1111
<b>Running the Stylesheet</b> .....	<b>1112</b>
<b>Observations</b> .....	<b>1112</b>
<b>Summary</b> .....	<b>1113</b>

## Part IV: Appendices

### **Appendix A: XPath 2.0 Syntax Summary** **1117**

<b>Whitespace and Comments</b> .....	<b>1118</b>
<b>Tokens</b> .....	<b>1118</b>
<b>Syntax Productions</b> .....	<b>1119</b>
<b>Operator Precedence</b> .....	<b>1122</b>

### **Appendix B: Error Codes** **1123**

<b>Functions and Operators (FO)</b> .....	<b>1124</b>
<b>XPath Errors (XP)</b> .....	<b>1126</b>
<b>XSLT Errors (XT)</b> .....	<b>1127</b>

### **Appendix C: Backward Compatibility** **1139**

<b>Stage 1: Backward-compatibility Mode</b> .....	<b>1140</b>
Deprecated Facilities .....	1140
Error Handling .....	1140
Comparing Strings .....	1141
Numeric Formats .....	1141
Other XPath Changes .....	1142
Serialization Changes .....	1142
<b>Stage 2: Setting version="2.0"</b> .....	<b>1142</b>
The First Node Rule .....	1142
Type Checking of Function Arguments .....	1143
Comparison Operators .....	1143
Arithmetic .....	1143
The Empty Sequence .....	1143
Error Semantics for «and» and «or» .....	1144
Other XSLT Differences .....	1144
<b>Stage 3: Adding a Schema</b> .....	<b>1145</b>
<b>Summary</b> .....	<b>1145</b>

## Contents

---

<b>Appendix D: Microsoft XSLT Processors</b>	<b>1147</b>
<b>MSXML</b>	<b>1147</b>
Objects	1148
IXMLDOMDocument and IXMLDOMDocument2	1148
IXMLDOMNode	1150
IXMLDOMNodeList	1151
IXMLDOMParseError	1151
IXMLDOMSelection	1152
IXSLProcessor	1152
IXSLTemplate	1153
Putting it Together	1154
Restrictions	1158
<b>System.Xml</b>	<b>1158</b>
XPathDocument	1158
XmlNode	1159
XPathNavigable	1159
XPathNavigator	1159
XsltTransform	1159
<b>Summary</b>	<b>1161</b>
<b>Appendix E: JAXP: The Java API for Transformation</b>	<b>1163</b>
<b>The JAXP Parser API</b>	<b>1164</b>
JAXP Support for SAX	1164
JAXP Support for DOM	1166
<b>The JAXP Transformation API</b>	<b>1169</b>
<b>Examples of JAXP Transformations</b>	<b>1187</b>
Example 1: Transformation Using Files	1187
Example 2: Supplying Parameters and Output Properties	1188
Example 3: Holding Documents in Memory	1188
Example 4: Using the <?xml-stylesheet?> Processing Instruction	1189
Example 5: A SAX Pipeline	1190
<b>Summary</b>	<b>1193</b>
<b>Appendix F: Saxon</b>	<b>1195</b>
<b>Using Saxon from the Command Line</b>	<b>1196</b>
<b>Using Saxon from a Java Application</b>	<b>1199</b>
Using Saxon via JAXP Interfaces	1199
The s9api Interface	1202
<b>Using Saxon from a .NET Application</b>	<b>1203</b>
<b>Saxon Tree Models</b>	<b>1205</b>

---

<b>Extensibility</b> .....	<b>1205</b>
Writing Extension Functions in Java .....	1206
Writing Extension Functions under .NET .....	1206
Collations .....	1207
<b>Extensions</b> .....	<b>1208</b>
Serialization Extensions .....	1208
Extension Attributes .....	1209
Extension Instructions .....	1209
Extension Functions .....	1209
<b>The evaluate() Extension</b> .....	<b>1210</b>
<b>Summary</b> .....	<b>1214</b>
<b>Appendix G: Altova</b> .....	<b>1215</b>
<b>Running from within XMLSpy</b> .....	<b>1215</b>
<b>Conformance</b> .....	<b>1216</b>
<b>Extensions and Extensibility</b> .....	<b>1217</b>
<b>The Command Line Interface</b> .....	<b>1217</b>
<b>Using the API</b> .....	<b>1218</b>
The COM API .....	1218
The Java API .....	1219
The .NET API .....	1220
<b>Summary</b> .....	<b>1220</b>
<b>Appendix H: Glossary</b> .....	<b>1221</b>
<b>Index</b> .....	<b>1233</b>

