

PROGRAMMER TO PROGRAMMER™



PROFESSIONAL

XML

Databases

Kevin Williams

with Michael Brundage, Patrick Dengler, Jeff Gabriel, Andy Hoskinson, Michael Kay, Thomas Maxwell,  
Marcelo Ochoa, Johnny Papa, Mohan Varmane



# Table of Contents

<b>Introduction</b>	<b>1</b>
<b>Why XML and Databases</b>	<b>2</b>
<b>What This Book is About</b>	<b>3</b>
<b>Who Should Use This Book?</b>	<b>3</b>
Data Analysts	3
Relational Database Developers	3
XML Developers	4
<b>Understanding the Problems We Face</b>	<b>4</b>
Structure of the Book	4
Design Techniques	5
Technologies	5
Data Access	6
Common Tasks	6
Case Studies	7
Appendices	7
<b>Technologies Used in the Book</b>	<b>7</b>
<b>Conventions</b>	<b>8</b>
<b>Customer Support</b>	<b>8</b>
Source Code and Updates	8
Errata	9
<b>Chapter 1: XML Design for Data</b>	<b>11</b>
<b>XML for Text Versus XML for Data</b>	<b>12</b>
XML for Text	12
XML for Data	12
<b>Representing Data in XML</b>	<b>14</b>
Element Content Models	14
Element-only Content	14
Mixed Content	14
Text-only Content	15
EMPTY Content	15
ANY Content	16
Using Attributes	16
Other Considerations	17
Audience	17
Performance	17
Data Modeling Versus Representation Modeling	18
XML Data Structures – A Summary	20

<b>Mapping Between RDBMS and XML Structures</b>	<b>20</b>
Structure	21
Elements	21
Attributes	22
Data Points	22
Our Invoice Using Elements	24
Our Invoice Using Attributes	25
Comparing the Two Approaches	26
Elements or Attributes – The Conclusion	30
Relationships	30
Containment	31
More Complex Relationships – Pointers	32
More Complex Relationships – Containment	34
Relationships – Conclusion	36
<b>Sample Modeling Exercise</b>	<b>37</b>
Before We Begin	37
What is the Scope of the Document?	38
Which Structures Are We Modeling?	38
What Are the Relationships Between Entities?	39
Which Data Points Need to be Associated with Each Structure?	39
Creating the XML DTD	40
Start With the Structures	41
Add the Data Points to the Elements	41
Incorporate the Relationships	41
Sample XML Documents	43
<b>Summary</b>	<b>45</b>

---

<b>Chapter 2: XML Structures for Existing Databases</b>	<b>47</b>
---	-----------

<b>Migrating a Database to XML</b>	<b>48</b>
Scoping the XML Document	49
Creating the Root Element	51
Model the Tables	51
Model the Nonforeign Key Columns	52
Adding ID Attributes	54
Handling Foreign Keys	55
Add Enumerated Attributes for Lookup Tables	55
Add Element Content to the Root Element	57
Walk the Relationships	58
Add Missing Elements to the Root Element	60
Discard Unreferenced ID attributes	61
An Example XML Document	62
<b>Summary</b>	<b>63</b>

---

<b>Chapter 3: Database Structures for Existing XML</b>	<b>67</b>
--	-----------

<b>How to Handle the Various DTD Declarations</b>	<b>68</b>
Element Declarations	68
The Element-only (Structured Content) Model	68
The Text-only Content Model	75
The EMPTY Content Model	78
The Mixed Content Model	80
The ANY Content Model	83

Attribute List Declarations	84
CDATA	84
Enumerated Lists	85
ID and IDREF	87
IDREFS	91
NMTOKEN and NMTOKENS	94
ENTITY and ENTITIES	95
Entity Declarations	96
Notation Declarations	96
Avoid Name Collisions!	96
Summary	97
Example	97
Modeling the Attributes	102
<b>Summary</b>	<b>107</b>
<b>The Rules</b>	<b>107</b>
<b>Chapter 4: Standards Design</b>	<b>111</b>
<b>Scoping the Solution</b>	<b>111</b>
Types of Standards	112
System Internal Standards	112
Cross-system Standards	112
Industry-level Standards	112
Document Usage	113
Archival Documents	113
Transactional Data Documents	115
Presentation Layer Documents	117
<b>Before Diving In: Ground Rules</b>	<b>120</b>
Implementation Assumptions	121
Elements vs. Attributes	121
Restricting Element Content	122
Don't Allow the ANY Element Type	122
Don't Allow the Mixed-content Element Type	123
Constrain Elements that have Structured Content	123
Capturing Strong Typing Information	124
Naming Conventions	125
<b>Understanding the Impact of Design Decisions</b>	<b>126</b>
Performance	126
Document Size	126
Overnormalization	127
Too Many Pointing Relationships	127
Coding Time	129
Document Complexity	130
Pointing Relationships	130
Levels of Abstraction	130
Developer Ramp-up Time	131
Extensibility	131
<b>During the Development</b>	<b>131</b>
Subdividing the Workload	131
Data issues	132
General vs. Specific	132
Required vs. Optional	133
"Tag soup"	133
Keeping the Structure Representation-Independent	133

<b>Pulling It all Together</b>	<b>135</b>
Peer Review	135
Resolving Disputes	135
The Implementation Guide	135
Statement of Purpose	135
Dictionary	136
Document Structure Diagram	138
Sample Documents	139

<b>Summary</b>	<b>140</b>
----------------	------------

---

<b>Chapter 5: XML Schemas</b>	<b>143</b>
-------------------------------	------------

<b>Introducing Schemas</b>	<b>144</b>
----------------------------	------------

A Note Before We Start – Other Schema Technologies	144
Jumping Right In	145
Schemas are Just XML Files	145
Our First Simple Schema	145
Why Schemas	147
XML Parsers with Schema Support	149
Learning the New Vocabulary	150
Datatypes	150
Structures	151
XML Schema Preamble – The <schema> Element	151
Attribute Declarations	152
Element Definitions	153
Complex Type Definitions	155
Content From Another Schema – <any>	157
Named Model Groups – <group>	157
Attribute Groups	158
Notation declarations	158
Annotations	159
Using Other Namespaces	160
Summary of Structures	161

<b>Datatypes</b>	<b>162</b>
------------------	------------

Primitive Datatypes	162
Derived Datatypes	163
Built-in Derived Types for XML Schema	163
Atomic, List and Union Datatypes	164
Atomic Datatypes	164
List Datatypes	165
Union Datatypes	166
Aspects of Datatypes	166
Value Spaces	166
Lexical Spaces	167
Facets	167
Fundamental Facets	167
Constraining Facets	169
Simple Type Definitions	175
Scope of Simple Type Definitions	178
Global Scope of Simple Type Definitions	178
Local Scope of Simple Type Definitions	178
ID, IDREF and IDREFS	179
Using ID as a Primary Key and IDREF for Foreign Keys	179
Identity-Constraints	180
Unique Values	180
Key and KeyRef	181

<b>Example Schemas</b>	<b>181</b>
Example 1 – Name Details	182
Example 2 – Using an Attribute Group to Represent Rows	185
Example 3 – Mixed Content Models	185
Example 4 – Enumeration	188
<b>Summary</b>	<b>189</b>

---

## **Chapter 6: DOM** **191**

### **What is the DOM?** **192**

How Does the DOM Work?	192
DOMString	193
DOM Implementations	193

### **DOM Interfaces** **194**

The Structure Model	195
Inheritance and Flattened Views	197
The DOM Core	197

### **Fundamental Interfaces** **198**

Node	200
Getting Node Information	201
Traversing the Tree	203
Adding, Updating, and Removing Nodes	205
Document	208
DOMImplementation	210
DocumentFragment	211
NodeList	212
Element	213
NamedNodeMap	216
Attr	216
CharacterData and Text	217
Handling Complete Strings	217
Handling Substrings	218
Modifying Strings	218
Splitting Text	219
Comments	220
DOMException	220

### **Extended Interfaces** **220**

CDATA Sections	221
DocumentType	221
Notation	221
EntityReference	221
Entity	222
Processing Instructions	222

### **Working With Our Data** **222**

Accessing the DOM from JavaScript	222
Retrieving the Data from an XML Document using the DOM	223
Adding to the Contents of the Document Using the DOM	228
Adding Information from Another DOM Tree	230

<b>When To Use or Not Use the DOM</b>	<b>232</b>
Size of Document	233
How the DOM Processes a Document	234
<b>The Task in Hand</b>	<b>235</b>
Reading and Filtering	235
Add, Update, Delete	236
Transformations of Structure	236
Creating Documents	237
<b>Summary</b>	<b>238</b>
<b>Chapter 7: SAX – The Simple API for XML</b>	<b>241</b>
<b>SAX Parsers</b>	<b>242</b>
Understanding SAX	242
<b>Example 1 – A Simple SAX Application</b>	<b>244</b>
Preparing the XMLReader Class	245
Catching Events from the XMLReader	246
The Error Handler	248
The Result	249
A Moment with DOM	249
Choosing Between SAX and DOM	251
Best Uses of the DOM	252
Best Uses of SAX	252
<b>The Stateless World of SAX</b>	<b>252</b>
Example 2 – Creating Attribute Centric Content from Element Centric Content	253
The frmAttsCentric Form	254
The XML Document	255
The ContentHandler Class	256
The Error Handler Class	260
The Result	260
Context Summary	261
<b>Handling Large Documents</b>	<b>261</b>
Example 3 – Creating an Efficient XML Document from a Large Verbose One	262
Prepare the XMLReader Class	263
Begin Parsing Events	265
The Results	266
<b>SAX Filters</b>	<b>267</b>
Example 4 – Using an Implementation of the XMLFilter Class	267
Preparing to Use a Chain of XMLFilters	268
Using the XMLFilter Implementation Class	270
The saxContentHandler Class	273
The Result	274
Filters Summary	274
<b>Take What You Need – Storing Result Data</b>	<b>275</b>
Example 5 – Complex State	275
The XML	276
The ContentHandler	276
Writing to the Database	280
The Result	282
<b>Summary</b>	<b>283</b>

<b>What is XSLT?</b>	<b>286</b>
<b>The Transformation Process</b>	<b>287</b>
<b>XSLT as a Programming Language</b>	<b>289</b>
XML Syntax	289
Rule-based	289
XPath Queries	290
Functional Programming	291
Data Types	292
<b>The XPath Data Model</b>	<b>293</b>
Names and Namespaces	297
<b>XPath Expressions</b>	<b>298</b>
Context	298
Primaries	299
Operators	299
Comparing Node-sets	301
Location Paths	301
Example Location Paths	301
Syntax Rules for Location Paths	302
Steps	302
Axis Names	303
Node-tests	304
Predicates	304
Abbreviations	305
XPath Functions	306
<b>Stylesheets, Templates, and Patterns</b>	<b>314</b>
The <xsl:stylesheet> Element	314
The <xsl:template> Element	315
Patterns	317
Selecting a Template Rule	318
Parameters	319
<b>The Contents of a Template</b>	<b>320</b>
Attribute Value Templates	320
XSLT Instructions	321
Literal Result Elements	329
Sorting	329
<b>Top-level Elements</b>	<b>330</b>
<b>Some Working Examples</b>	<b>336</b>
Example: Displaying Soccer Results	336
Source	336
The First Stylesheet	337
Stylesheet 2	339
Selecting a Stylesheet Dynamically	342
<b>XSLT and Databases</b>	<b>343</b>
<b>Summary</b>	<b>344</b>



<b>Resource Linking</b>	<b>348</b>
Invoice	349
Item	349
InvoiceItem	349
<b>Simple XLink Elements</b>	<b>349</b>
role	351
arcrole	351
title	351
show	351
actuate	352
Simplify the Simple Link with a DTD	352
<b>Extended XLink Elements</b>	<b>353</b>
The Elements of Extended Style Links	356
The arc-type Element	358
The resource-type Element	361
The locator-type Element	363
Using Extended Links	366
Extended Link Summary	369
<b>Extended Links and Relational Data</b>	<b>369</b>
Making the Relationship with XLink	370
<b>Summary</b>	<b>373</b>
<b>Additional Resources</b>	<b>373</b>

---

**Chapter 10: Other Technologies (XBase, XPointer, XInclude, XHTML, XForms) 375**

<b>XBase</b>	<b>376</b>
"Advanced" XBase	378
Determining the Base URI and Relative URIs	380
XBase Summary	381
<b>XPointer</b>	<b>381</b>
Technology Status	382
Locations and Targets	382
Identifiers Using XPointer and XLink	382
Full Form	383
Bare Names	383
Child Sequences	384
Extensions to XPath	385
Points	385
Ranges	386
How Do We Select Ranges?	387
Querying with XPointer	389
Other Points	391
XPointer Function Extensions to XPath	391
Rules and Errors	393
XPointer Summary	393
<b>XInclude</b>	<b>393</b>
Modular Development	393
An XInclude Example	394
Syntax	395
XInclude Advantages	397
XInclude Summary	397

<b>XHTML</b>	<b>398</b>
How XHTML differs from HTML	398
XHTML Flavors	401
Transitional	401
Strict	402
Frameset	402
XHTML Summary	402
<b>XForms</b>	<b>402</b>
How Do XForms Differ from HTML Forms?	402
The Model	403
The Data Model	403
The Logic Layer	404
The Presentation Layer	404
Current Status	404
A Few More Features	404
DataTypes	404
Common Facets	405
Data Models	406
XForms Summary	407
<b>Summary</b>	<b>407</b>

---

## **Chapter 11: The XML Query Language** **409**

<b>Requirements and Use Cases</b>	<b>410</b>
Selection	410
Transformation	411
Data Centricity	412
Types and Operators	412
<b>Data Model</b>	<b>413</b>
Collections	413
References	413
Nodes	413
Scalars and ValueNodes	414
Node Constructors and Accessors	415
Information Items	415
Types	415
Implementation Issues	416
<b>Algebra</b>	<b>416</b>
<b>Syntax</b>	<b>417</b>
XPath	421
XPath Overview	421
Evaluating XPath as a Query Language	421
XPath Examples	422
XSLT	423
XSLT Overview	424
XSLT as a Query Language	424
XSLT Examples	424
Quilt	426
Quilt Overview	426
Quilt as a Query Language	427
Quilt Examples	427
XSQL	428
XSQL Overview	428
XSQL as a Query Language	428
XSQL Examples	428
<b>Summary</b>	<b>429</b>

<b>Types of flat files</b>	<b>432</b>
Delimited	432
Fixed-width	432
Tagged record	432
<b>Issues</b>	<b>433</b>
Level of Normalization	433
Data Formatting	433
<b>Plan of Attack</b>	<b>433</b>
Transform the XML Document to name-value Pairs	433
Transform the Flat File to name-value Pairs	436
Map the Transform	439
<b>Transforming from Flat Files to XML</b>	<b>442</b>
Programming Approaches	442
Manual Serialization	442
SAX	442
The DOM	442
Handling Different Flat File Types	442
Delimited	443
Fixed-width	448
Tagged Record	456
<b>Transforming from XML to Flat Files</b>	<b>461</b>
Programming Approaches	461
Parse and Serialize	461
XSLT	461
Handling Different File Types	461
Delimited	462
Fixed-width	470
Tagged Record	476
<b>Summary</b>	<b>479</b>

<b>XML Support in ADO 2.6 and SQL Server 2000</b>	<b>482</b>
What is a Stream?	482
Obtaining a Stream	483
Persistence of XML	484
Running XML Queries	484
Queries Using FOR XML	484
Queries Using Annotated Schema	485
Merging XML with Relational Data	485
New XML Properties	485
Mapping Schema	485
Base Path	486
Output Stream	486
<b>XML Persistence</b>	<b>486</b>
Persisting to a File	486
Opening a Recordset from a File	488
Persisting to a Stream	490
Opening a Recordset from a Stream	491
Persisting to the Response Object	492
Writing a Stream to a Client	493
Persisting a Stream to a Client	495

<b>XML Queries via ADO</b>	<b>496</b>
FOR XML Usage	497
FOR XML Template Queries	498
FOR XML URL Queries	498
FOR XML via ADO	499
Running a FOR XML Command	499
<b>Non-SQL Queries via ADO</b>	<b>504</b>
Mapping Schema File	504
Mapping Schema & Base Path Properties	506
Code Examples	506
<b>Joining XML and SQL Data</b>	<b>509</b>
The ASP Page	511
Returning XML	513
Inserting Data	515
<b>ADO+</b>	<b>517</b>
Evolution	517
The Idea	517
The ADO+ Framework	518
DataSet	518
Connection	518
DataSetCommand	519
DataSet	519
Schema	519
The DataSetCommand and the Connection Objects	524
VB	524
Typeness of DataSet	528
<b>Summary</b>	<b>530</b>

---

## **Chapter 14: Storing and Retrieving XML in SQL Server 2000** **533**

<b>Retrieving XML from SQL Server 2000: FOR XML</b>	<b>534</b>
New SQL Server Query Support	535
FOR XML: General Syntax	536
The RAW Mode	537
The AUTO Mode	538
The Explicit Mode	540
Specifying Column Aliases	541
Specifying the Metadata Columns	542
Specifying the Directive in the Column Alias	542
<b>Generating XML from the Rowset (Universal Table)</b>	<b>547</b>
Hierarchy Generation	548
The First Rowset: Representing the <customer> Element	549
The Second Rowset: Representing the <order> Element	549
The SELECT Query	550
Processing the Rowset (Universal Table)	550
Further Examples	551
Example 1 – Using idrefs to Create Attributes	551
Example 2 – Producing XML Containing Siblings	555
<b>Alternative Ways to Retrieve XML</b>	<b>556</b>

<b>Storing XML in SQL Server 2000: OPENXML</b>	<b>556</b>
Using OPENXML in SQL Statements	557
Creating the In-Memory Representation of the Document	557
Understanding OPENXML	559
OPENXML: Attribute-centric and Element-centric Mapping	561
Additional Mapping Information for Specifying the Rowset Schema	562
The Edge Table Schema for the Rowset	566

<b>Bulk Loading XML</b>	<b>572</b>
How the Database is Generated from XML	574

<b>Summary</b>	<b>578</b>
----------------	------------

---

<b>Chapter 15: XML Views in SQL Server 2000</b>	<b>581</b>
---	------------

Fragments and Documents	582
-------------------------	-----

<b>XDR Schemas</b>	<b>582</b>
The Default Mapping	584
Querying SQL Server	585
Names and Other Restrictions	586

<b>Annotated XDR Schemas</b>	<b>586</b>
Tables and Columns (sql:relation and sql:field)	586
Null Values	588
Join Relationships (sql:relationship)	588
Multi-column Joins	589
Link Tables	590
Qualified Joins (sql:limit-field and sql:limit-value)	591
Keys, Nesting and Ordering (sql:key-fields)	592
Alternative Mappings	593
Unmapped Data (sql:map-field)	593
Constant Elements (sql:is-constant)	594
XML Text (sql:overflow-field)	595
Data Types (sql:datatype, dt:type, sql:id-prefix)	596
Booleans and Binaries	597
Strings	598
Numbers	598
Identifiers	599
Dates and Times	600
Mapping Between XDR Types and SQL Types	601
CDATA Sections	601
Embedded Binary Values	602

<b>Table 1. Annotation Summary</b>	<b>603</b>
------------------------------------	------------

<b>Templates</b>	<b>604</b>
------------------	------------

<b>XPath</b>	<b>605</b>
Introduction	607
Navigation	607
Predication	608
XPath Types	609
XPath Expressions	610
XPath and the XML View	611
Default Schema	612
XPath Parameters	612
Additional XPath Axes	613

<b>Updategrams</b>	<b>613</b>
Introduction	614
Values, Absence and NULL	616
Insert/Update/Delete Heuristics	617
Parameters	618
Default Schema	618
Server-Generated Identities	619
Data Types	619
Overflow	620

<b>Advanced Topics</b>	<b>620</b>
Namespaces and External Schemas	620
Structural Recursion	622

<b>Summary</b>	<b>624</b>
----------------	------------

---

<b>Chapter 16: JDBC</b>	<b>627</b>
-------------------------	------------

How XML Enhances JDBC (and vice versa)	627
Software Needed for This Chapter	628
JDK 1.3	628
Xalan XSLT Processor for Java	628
Software Needed to Implement Rowsets	629
Tomcat 3.1	630
A JDBC Data Source and Driver (Oracle, SQL Server, etc.)	631

<b>Generating XML from JDBC</b>	<b>632</b>
---------------------------------	------------

Our Simple XML Gateway Architecture for JDBC	632
The JDBC2XML Class	634
Applying XML Encoding Rules to the Result Set Data	634
Serializing the Result Set's Metadata and Data as XML	635
Executing the Query	637
The XMLDataGateway Servlet	637
Implementing the Servlet	638
Developing an HTML Form Interface	639
Using the XMLDataGateway Servlet	640
Summary	646
The JDBC2HTML Servlet	646
Implementing the JDBC2HTML Servlet	647
Writing an XSL Stylesheet	649
Using the JDBC2HTML Servlet	651
Summary	652

<b>Using XML for distributed JDBC Applications</b>	<b>652</b>
--	------------

The Rowset Interface	653
The <code>sun.jdbc.rowset.WebRowSet</code> Class	653
Implementing a Distributed JDBC Application Using the <code>WebRowSet</code> Class	654
Setting up the Sample Database	655
Fetching a Rowset Via HTTP: The <code>WebRowSetFetchServlet</code> Class	655
Performing a Batch Update Via HTTP: The <code>WebRowSetUpdateServlet</code> Class	658
Inserting, Updating, and Deleting Data at the Client: The <code>WebRowSetHTTPClient</code> Class	659

<b>Packaging the Code for J2EE Deployment</b>	<b>664</b>
---	------------

The Web Application Deployment Descriptor	665
Building the Application	666
Compile the Java Classes	666
Package the Bytecode into a JAR	667
Package the Application into a WAR	667

<b>Summary</b>	<b>667</b>
----------------	------------

<b>Data Warehousing</b>	<b>670</b>
The Two Roles of Data	670
Detailed Information Gathering	670
Information Querying and Summarization	671
The Traditional Solution	672
The Data Warehousing Solution	675
On-Line Transaction Processing (OLTP)	675
On-Line Analytical Processing (OLAP)	676
Parts of an OLAP Database	677
The Role of XML	683
Summary	691
<b>Data Archival</b>	<b>692</b>
Classical Approaches	692
Using XML for Data Archival	693
Summary	696
<b>Data Repositories</b>	<b>696</b>
Classical Approaches	696
Using XML for Data Repositories	697
Summary	698
<b>Summary</b>	<b>699</b>

<b>Executing a Data Transmission</b>	<b>701</b>
Agree on a Format	702
Transport	702
Routing	702
Request-Response Processing	702
<b>Classic Strategies</b>	<b>702</b>
Selecting on a Format	702
Delimited Files	703
Fixed-width Files	703
Proprietary/Tagged Record Formats	703
Problems with Classic Structures	705
Transport	706
Physical Media	706
E-mail	706
FTP	707
Socket Code	707
Virtual Private Network (VPN)	707
Leased-Line	707
<b>How Can XML Help?</b>	<b>708</b>
XML Documents are Self-Documenting	708
XML Documents are Flexible	709
XML Documents are Normalized	710
XML Documents can Utilize Off-The-Shelf XML Tools	710
Routing and Requesting	710

<b>SOAP</b>	<b>712</b>
The SOAP Envelope	713
The SOAP Header	713
The SOAP Body	714
The SOAP Fault Element	716
<b>Transmission Over HTTP</b>	<b>717</b>
HTTP SOAP Request	718
HTTP SOAP Response	718
An HTTP Transmission Example	718
Biztalk	720
<b>Compressing XML</b>	<b>720</b>
<b>Summary</b>	<b>721</b>

---

## **Chapter 19: Marshalling and Presentation** **723**

<b>Marshalling</b>	<b>723</b>
Custom Structures	724
Recordsets	725
XML	726
The Long-Term Solution: Built-In Methods	726
The Manual Approach	726
Example	727
Conclusion	737
<b>Information Gathering</b>	<b>737</b>
The long-term solution: XForms	737
Manual Approach	738
Example 1: Transforming Form Data to XML on the Client	738
Example 2: Transforming Form Submissions to XML on the Server	743
Example 3: Creating HTML Forms with XSLT for Updates	744
Conclusion	753
<b>Presentation</b>	<b>754</b>
Why Use XML?	754
Example: XML Detail – HTML Detail	754
Example: XML Detail – WML Detail	756
Example: XML Detail – WML Summary	757
Summary	760
<b>Conclusion</b>	<b>760</b>

---

## **Chapter 20: SQL Server 2000 XML Sample Applications** **763**

<b>Project 1 – Accessing SQL Server 2000 Directly Over HTTP</b>	<b>764</b>
XML Templates – Getting XML from SQL Server Across the Web	764
Templates Featuring XPath Queries	767
Applying XSLT to a Template	768
Updategrams	769
Posting a Template Using an HTML Form	770
Posting a Template to the Server	771
Passing Parameters to the Query	773
Executing Template Files	774
Sample ASP Application	776
Creating the Application	778
Sample ADO Applications	781
Executing a Command	781
Executing an XML Template	782
Passing Parameters	784



## **Building an Empire: an eLemonade Company**

The Internet Lemonade Stand – Project Requirements

Database Design

External XML Sources

**786**

786

787

789

## **The Two Different Solutions**

Prototyping with OpenXML and FOR XML

Displaying News and Expenses

Place Orders and Create Customer Accounts

Prototyping with XPath and Updategrams

Annotate a Schema

Improve the Schema

Display News and Expenses

Create and Edit Customer Accounts

Personalized Customer Page

**789**

789

790

791

792

793

794

798

800

803

## **Summary**

Helpful Web Sites

**805**

805

## **Chapter 21: DB Prism: A Framework to Generate Dynamic XML from a Database 807**

### **Cocoon Architecture**

**809**

### **DB Prism Architecture**

**811**

DB Prism Class Diagram

811

DB Prism: Benefits Provided to the Cocoon Framework

814

### **Part I. DB Prism Internals**

**814**

Who is who?

814

Wrappers

814

Engine

815

Common Issues with Writing a New Adapter

818

Writing the New Adapter

818

DBPrism.java

819

DBConnection.java

822

DBFactory.java

825

SPProc.java

825

### **Part II: Putting DB Prism to Work**

**826**

Before Start

826

Cocoon Setup

829

Downloading Cocoon

829

Create a Write-enabled Directory

829

DB Prism Setup

831

Downloading DB Prism

831

Database Setup

833

Configuring DB Prism to Access the Database

835

Making the Hello World Example

836

Making a Content Management System

838

Brief Introduction to a CMS

838

Design of the Meta Model

842

Writing the Java Code

844

Building Pages

844

Related Procedures for Making The Page

848

Installing the CMS

858

Deployment

859

### **Conclusions and Beyond**

**860**

### **Summary**

**861**

**What Is a Markup Language?**Tags and Elements  
Attributes**863**

864

864

**What Is XML?**

How XML Documents are Processed

The Basic Building Blocks of XML

The XML Declaration

Elements

Attributes

Character Data

CDATA Section

Entity References

Processing Instructions

Comments

Namespaces

**865**

869

869

870

871

872

873

874

875

876

877

877

**Document Type Definitions**

The Standalone Declaration

Element Declarations

Attribute Declarations

Notation Declarations

Entity Declarations

Conditional Sections

Thinking in Trees

**878**

880

880

885

892

892

895

896

**Technology Overview**

XML Schemas

The Document Object Model (DOM)

The Simple API for XML (SAX)

XSLT/XPath

XML Query

XLink

XPointer

XBase

XForms

XML Fragment Interchange

XInclude

**897**

897

897

897

898

898

898

898

898

898

899

899

**Summary****899****Appendix B: Relational Database Primer****901****Types of Database****902****SQL****902****Designing Relational Databases****902**

Logical Data Design

Entities

Attributes

Relationships

Physical Data Design

Tables

Columns

903

903

903

904

905

905

906

<b>Creating Our Tables</b>	<b>908</b>
Primary Keys	909
Foreign Keys	910
Indexes	911
Triggers	912
Stored Procedures	913
<b>Summary</b>	<b>913</b>
<b>References</b>	<b>913</b>

---

## **Appendix C: XML Schema Datatypes** **915**

<b>Primitive Types</b>	<b>915</b>
string	916
boolean	916
decimal	916
float	917
double	917
timeDuration	917
recurringDuration	918
binary	919
uriReference	919
ID	920
Constraining Facets for Primitive Types	921

<b>Built-in Derived Types</b>	<b>922</b>
CDATA	923
token	923
language	923
NMTOKEN, NMTOKENS	924
ENTITIES	924
IDREFS	924
name, NCName	924
integer, negativeInteger, positiveInteger, nonNegativeInteger, nonPositiveInteger	925
byte, short, int, long	925
unsignedByte, unsignedShort, unsignedInt, unsignedLong	925
century, year, month, date	925
recurringDate, recurringDay	925
time, timeInstant, timePeriod	926
Constraining Facets for Derived Types	926

---

## **Appendix D: SAX 2.0: The Simple API for XML** **929**

Class org.xml.sax.SAXException	<b>964</b>
--------------------------------	------------

---

## **Appendix E: Setting Up a Virtual Directory for SQL Server 2000** **975**

Introduction	<b>975</b>
Setting up Your Virtual Directory	<b>975</b>

---

## **Appendix F: Support, Errata and P2P.Wrox.Com** **985**

<b>Index</b>	<b>991</b>
--------------	------------

---