

Chapter 1: XML Fundamentals	
XML Theory	2-3
What Can XML Offer that HTML Cannot?	4
XML and XSL	5
Power of Tags	6
XML Modeling	7-8
Database versus XML	9
Document Type Definition	10-11
DAD - Document Access Definition	12
XML Schemas	13-15
XSL	16
XSLT	17
Xpath	18
Xpointer	19
Xlink	20
Namespaces	21
DOM	22-23
SAX	24
JAXP	25
Database Argument	26-27
DB2 RDBMS	28
DB2 XML Extender Features	29-31
Features	32-33
DTD: Document Type Definition	34
DAD: Document Access Definition	35-36
Composition	37
Decomposition	38
SQL Mapping	39
RDB Node Mapping	40

Chapter 2: XML Planning and Design

Default Namespace - targetNamespace or XMLSchema?	2
Scenario: Library.xsd Reusing BookCatalogue.xsd	4
3 Ways to Design the Library Schema	5
Version 1: XMLSchema = Default Namespace	6
Observations about Version 1	7-8
Version 2: targetNamespace = Default Namespace	9
Observations about Version 2	10-11
Version 3 - No Default Namespace	12
Observations about Version 3	13
Default Namespace Best Practice	14

Chapter 3: Element versus Type

Issue	2
Guidelines	3-4

Non-determinism and the <any> Element	24-25
<any> -> Quite Restricted	26
Embed Additional Elements with an <other> Element	27
Commentary	28

Chapter 7: Global versus Local

Issue	2-3
Three Design Approaches	4
Russian Doll Design	5
Salami Slice Design	6
Venetian Blind Design	7
Characteristics of Russian Doll Design	8
Characteristics of Salami Slice Design	9
Characteristics of Venetian Blind Design	10
Guidelines	11

Chapter 8: Hide / Localize Namespace versus Expose Namespaces

Multiple Schemas, each with a Different Namespace	2
Hide / Localize Namespaces	3
Expose Namespaces	4
Hide / Localize Namespaces	5
Expose Namespaces	6
elementFormDefault - the Exposure "Switch"	7
Designing Schemas	8
How Do We Design Schemas	9
Instance Document	10-11
Requirements for Hiding / Localizing Namespaces	12-13
Instance Document	14
Requirements for either Hiding or Exposing Namespaces	15
Instance Document	17
Guidelines: Hide / Localize Namespaces within the Schema	18
Guidelines: Exposing Namespaces in Instance Documents	19
Best Practices	20

Chapter 9: Implementing Variable Content Containers

Issue	2
4 Methods	3
Method 1 - Use an Abstract Element and Element Substitution	4-7
Method 1 Advantages - Extensibility	11-13
Method 1 Advantages - Semantic Cohesion	14
Method 1 Disadvantages - No Independent Elements	15
Method 1 Disadvantages - Limited Structural Variability	16
Method 1 Disadvantages - Nonscalable Processing	17-18
Method 1 Disadvantages - No Control over Namespace Exposure	19
Method 2 - Use a <choice> Element	20-21
Method 2 Advantages - Independent Elements	22
Method 2 Disadvantages - Nonextensible	23
Method 2 Disadvantages - No Semantic Coherence	24-25

Chapter 4: Extending XML Schemas

Not "All Powerful"	2
Two Approaches to Extending XML Schemas	3
Enhancing XML Schemas using XSLT/Xpath	4
Enhancing XML Schemas Using Schematron	5
First Example: Verify that A > B	6-9

Chapter 5: Creating Extensible Content Models

Introduction to Schematron	2-3
The <assert> Element	4
The <rule> Element	5
The <diagnostic> Element	6-7
Schematron Schema	8-10
Inform Schematron of the Namespace	11-13
Example 1	14
XSLT Implementation of Checking the Constraints	15
Output	16
Schematron Implementation of Checking the Constraint	17
Example 2	18-19
XSLT Implementation of Checking the Constraints	20-21
Output	22
Schematron Implementation of Checking the Constraints	23
Alternative Schematron Implementation	24-25
Observation	26
Example 3	27-29
XSLT Implementation of Checking the Constraint	30
Output	31
Schematron Implementation of the Constraint	32
Advantages of using XSLT/Xpath	33
Disadvantages of Using XSLT/XPath	34
Advantages of Using Schematron	35
Disadvantages of Using Schematron	36
Inside Schematron	37

Chapter 6: Extending XML Schemas

Definition	2
Static, Fixed Content Model	3-4
Extensibility via Type Substitution	5-7
Extend a Schema without Touching it	8-10
Disadvantages of using Type Substitution	11
Disadvantages of using Type Substitution 1	12
Requirements for Extending Content Models	13
Extensibility via the <any> Element	14-17
Alternate Schema for Book	18
Controlling Extensibility using the Block Attribute	19
Control Over Where and How Much Extensibility	20
Recognizing our Limitations	21
Non-determinism and the <any> Element	22
Definition of Non-determinism	23

Method 3 - Use an Abstract Type and Type Substitution	26-28
Method 3 Advantages - Extensibility	29-30
Method 3 Advantages - Minimal Dependencies	31
Method 3 Advantages - Uniform, Scalable Processing	32
Method 3 Advantages - Semantic Cohesion	33
Method 3 Advantages - Control over Namespace Exposure	34
Method 3 Disadvantages - No Independent Elements	35
Method 3 Disadvantages - Limited Structural Variability	36
Method 4 - Motivation	37-38
Method 4 - Use a Dangling Type	39-46
Method 4 - Summary	47
Method 4 Advantages - Dynamic	48
Method 4 Advantages - Dynamic	49
Method 4 Advantages	50
Method 4 Disadvantages - Must be in Another Namespace	51-53
Best Practices	54

Chapter 10: Multi-Schema Project: Zero, One, or Many Namespaces

Managing Multiple Schemas - Same or Different Namespaces	2
Multi-Schema Project: Zero, One, or Many Namespaces?	3
Heterogeneous Namespaces Design	4-5
Homogenous Namespaces Design	6-7
No Namespace Design - Chameleon Design	8-9
Homogeneous and Chameleon Designs - Reuse and Customize	10
Example Showing <redefine>	11
Advantages/Disadvantages Heterogeneous Namespace Design	12
Advantages/Disadvantages Homogeneous Namespace Design	13
Advantages/Disadvantages: No Namespace Design - Chameleon Design	14
Using Proxy Schemas in the Chameleon Design	15
Proxy Schema	16
Using Proxy Schemas for Name Collision Avoidance	17
Advantages/Disadvantages - No Namespace Design Chameleon Design	18-19
Creating Tools to Process Chameleon Components	20
Chameleon Components	21
Chameleon Component Tools	22
Identifier for Each Component - Schema id Attribute	23
Chameleon Component Tools	24
Traceability: Schema id versus Namespaces	25
Guidelines	26
Best Practices	27