

## **Course Summary**

### Description

This comprehensive XML training class teaches students to create well-formed XML documents, to build sophisticated XML Schema for validating XML documents and to build eXtensible Stylesheets (XSLs) for transforming XML documents into XHTML, and other XML structures.

#### **Objectives**

At the end of this course, students will be able to:

- Learn to create well-formed XML documents.
- Learn the difference between HTML and XHTML.
- Learn to create basic DTDs.
- Learn to create basic XML schemas.
- Learn to validate XML documents against DTDs and XML schemas.
- Learn to create basic XSLTs to transform XML documents for output as text, HTML, and other XML structures.
- Learn the purpose and power of XML Schema
- Learn to declare simple-type and complextype elements
- Learn to derive custom types

#### Topics

- XML Basics
- DTDs
- Simple-Type Elements
- XSLT Basics
- XML Schema Basics
- Simple-Type Elements
- Complex-Type Elements
- Attributes
- Reusing Schema Components
- Annotating a Schema

#### Audience

- Learn to declare attributes
- Learn to annotate schemas
- Learn to understand namespaces
- Learn to work with multiple XML schema documents
- Learn to use XPath to access XML elements and attributes
- Learn to loop through XML elements
- Learn to use conditionals in XSLT
- Learn to create and call templates
- Learn to work with multiple XSLTs
- Learn to reference external XML documents
- Namespaces
- XPath
- Flow Control in XSLT
- XSLT Templates, Parameters and Variables
- Multiple XML and XSLT Documents
- Working with Keys
- Advanced XSLT Techniques
- Tying It All Together

This course is designed for those who need to write XML documents and incorporate them into their Java-based applications.

#### **Prerequisites**

Experience in the following would be useful for this XML class: HTML

## Duration

Five days

Due to the nature of this material, this document refers to numerous hardware and software products by their trade names. References to other companies and their products are for informational purposes only, and all trademarks are the properties of their respective companies. It is not the intent of ProTech Professional Technical Services, Inc. to use any of these names generically



# Comprehensive XML

## **Course Outline**

## I. XML Basics

- A. What is XML?
- B. XML Benefits
  - 1. XML Holds Data, Nothing More
  - 2. XML Separates Structure from Formatting
  - 3. XML Promotes Data Sharing
  - 4. XML is Human-Readable
  - 5. XML is Free
- C. XML in Practice
  - 1. Content Management
  - 2. Web Services
  - 3. RDF / RSS Feeds
  - XML Documents
    - 1. The Prolog
    - 2. Elements
    - 3. Attributes
    - 4. CDATA
    - 5. Whitespace
    - 6. XML Syntax Rules
    - 7. Special Characters
- E. Creating a Simple XML File

## II. DTDs

D.

- A. Well-formed vs. Valid
- B. The Purpose of DTDs
- C. Creating DTDs
  - 1. The Document Element
  - 2. Other Elements
  - 3. Choice of Elements
  - 4. Empty Elements
  - 5. Mixed Content
  - 6. Location of Modifier
  - 7. Using Parentheses for Complex Declarations
  - 8. Declaring Attributes
- D. Validating an XML Document with a DTD

## III. XML Schema Basics

- A. The Purpose of XML Schema
- B. The Power of XML Schema
- C. A First Look
  - 1. A Simple XML Schema
- D. Validating an XML Instance Document

## IV. Simple-Type Elements

- A. Overview
- B. Built-in Simple Types
  - 1. 19 Primitive Data Types
  - 2. Built-in Derived Data Types
  - 3. Defining a Simple-type Element
- C. User-derived Simple Types
  - 1. Applying Facets
    - 2. Controlling Length
    - 3. Specifying Patterns
    - 4. Working with Numbers
    - 5. Enumerations
    - 6. Whitespace-handling
  - Specifying Element Type Locally
- E. Nonatomic Types
  - 1. Lists
  - 2. Unions
- F. Declaring Global Simple-Type Elements
  - 1. Global vs. Local Simple-Type Elements
- G. Default Values
- H. Fixed Values
- I. Nil Values

## V. XSLT Basics

Β.

C.

D.

- A. eXtensible Stylesheet Language
  - An XSLT Stylesheet
    - 1. xsl:template
    - 2. xsl:value-of
  - 3. Whitespace and xsl:text
  - Output Types
  - 1. Text
  - 2. XML
  - 3. HTML
- D. Elements and Attributes
  - 1. xsl:element
  - 2. xsl:attribute
  - 3. Attributes and Curly Brackets

## VI. XML Schema Basics

- A. The Purpose of XML Schema
- B. The Power of XML Schema
- C. A First Look
  - 1. A Simple XML Schema
- D. Validating an XML Instance Document

Due to the nature of this material, this document refers to numerous hardware and software products by their trade names. References to other companies and their products are for informational purposes only, and all trademarks are the properties of their respective companies. It is not the intent of ProTech Professional Technical Services, Inc. to use any of these names generically



# Comprehensive XML

## Course Outline (cont'd)

## VII. Simple-Type Elements

- A. OverviewB. Built-in Simple
  - Built-in Simple Types
    - 1. 19 Primitive Data Types
    - 2. Built-in Derived Data Types
    - 3. Defining a Simple-type Element
- C. User-derived Simple Types
  - 1. Applying Facets
  - 2. Controlling Length
  - 3. Specifying Patterns
  - 4. Working with Numbers
  - 5. Enumerations
  - 6. Whitespace-handling
- D. Specifying Element Type Locally
  - Nonatomic Types
    - 1. Lists

F

- 2. Unions
- F. Declaring Global Simple-Type Elements
  - 1. Global vs. Local Simple-Type
  - Elements 2. Default Values
  - Default Values
    Fixed Values
  - Fixed Value
    Nil Values
- VIII. Complex-Type Elements
  - A. Overview B Content Mc
    - 6. Content Models
      - 1. xs:sequence
      - 2. xs:all
      - 3. xs:choice
  - C. Complex Model Groups
  - D. Occurrence Constraints
  - E. Declaring Global Complex-Type Elements
  - F. Mixed Content
  - G. Defining Complex Types Globally

#### IX. Attributes

- A. Overview
- B. Empty Elements
- C. Adding Attributes to Elements with Complex Content
- D. Adding Attributes to Elements with Simple Content
- E. Restricting Attribute Values
- F. Default and Fixed Values

- 1. Default Values
- 2. Fixed Values
- G. Requiring Attributes

#### X. Reusing Schema Components

- A. Overview
- B. Groups
  - 1. Element Groups
  - 2. Attribute Groups
- C. Extending Complex Types

#### XI. Tying It All Together

- A. Workshop: Creating and reusing XML schemas
- B. Annotating XML Schemas
- C. Overview

#### XII. Annotating a Schema

A. Transforming an XML Schema for Documentation

## XIII. Namespaces

- A. Overview
- B. Purpose of Namespaces
- C. Target Namespaces
- D. Default Namespaces
- E. Locally Declared Elements and Attributes
- F. Qualified Locals
- G. The XMLSchema-instance Namespace
- H. Using Multiple Namespaces

## XIV. XSLT Basics

C.

- A. eXtensible Stylesheet Language
- B. An XSLT Stylesheet
  - 1. xsl:template
  - 2. xsl:value-of
  - 3. Whitespace and xsl:text
  - Output Types
  - 1. Text
    - 2. XML
    - 3. HTML
- D. XSLT Elements and Attributes

Due to the nature of this material, this document refers to numerous hardware and software products by their trade names. References to other companies and their products are for informational purposes only, and all trademarks are the properties of their respective companies. It is not the intent of ProTech Professional Technical Services, Inc. to use any of these names generically



# Comprehensive XML

## Course Outline (cont'd)

#### XV. XPath

- A. XPath in XSLT
- B. XPath Expression
- C. XPath Terminology
- D. Context Node
  - 1. Current Node
  - 2. Context Size
  - 3. Proximity Position
- E. Location Paths
- F. Axis
- G. Node Test
- H. Predicate
- I. Accessing Nodes
- J. Abbreviated Syntax
- K. XPath Functions
- L. XPath Operators

#### XVI. Flow Control in XSLT

- A. Looping in XSLT
- B. Sorting with XSLT
- C. Looping and Sorting
- D. Conditions with XSLT

#### XVII. XSLT Templates, Parameters and Variables

- A. xsl:apply-templates
- B. xsl:call-template
- C. Passing Parameters
- D. Removing Content
- E. Template Modes
- F. Template Priority
  - 1. Default Priorities
  - 2. Assigning Priorities
- G. XSLT Variables

#### XVIII. Multiple XML and XSLT Documents

- A. Including XSLTs
- B. Importing XSLTs
- C. Conflict resolution
- D. The document() Function

## XIX. Working with Keys

- A. Key Basics
  - 1. <xsl:key/>
    - 2. The key() Function
    - 3. Improving Performance with Keys
    - 4. Cross References
    - 5. The Key Way
    - 6. Grouping
    - 7. The generate-id() Function
    - 8. Using generate-id() for Grouping

#### XX. Advanced XSLT Techniques

- A. Working with Numbered Lists
  - 1. The position() function
  - 2. xsl:number
- B. Outputting Processing Instructions
  - Copying Nodes

C.

- 1. xsl:copy
- 2. xsl:copy-of

#### XXI. Tying It All Together

A. Workshop: Sharing data and transforming it for the Web