Transforming XML with XPATH, XSLT and XQuery

Duration:	3 days
Type:	intermediate

Description

This course enables developers to extract information from XML documents and transform that information into another format. Typical examples are converting a business document from an external to an internal schema, preparing a publication for printing and converting XML program output to HTML for manual inspection. XSLT is the traditional technology used for this task, but it has recently been complimented by the XQuery standard, which is aimed specifically at XML stored in databases.

The course provides an in-depth guide to each of these technologies, along with the XPath addressing language that both are dependant on. The nature of the technologies allows the course to be very hands-on, with delegates working on sample transformations all the way through the course.

Prerequisites

Developers should be experienced at writing HTML and XML documents.

List of Modules

Review of Core Concepts

- Creating well-formed documents
- Validating documents with DTD's
- Validating documents with Schemas
- Declaring namespaces in XML
- Using several namespaces at once

Introduction to XPath

- The need for an addressing language
- Why the XSL specification was split
- The relationships between XPath, XSLT, XSL-FO and XQuery
- Node types in the XPath data model
- The axes of the XPath data model
- Changes made to the data model in XPath 2.0

Creating XPath Expressions

- Components of a single step
- Combining steps in expressions
- Absolute vs. relative expressions
- The different kinds of node test
- Abbreviations allowed in XPath
- Using the built in functions
- Applying multiple predicates
- Understanding how positioning works for forward and reverse axes
- Extra features introduced in XPath 2.0

Introduction to XSLT

- The process of XML transformation
- Where to obtain an XSLT Engine
- Useful scenarios for applying XSLT
- Scenarios where XSLT is detrimental

Writing XSLT Stylesheets

- Declaring new rules in stylesheets
- Understanding the built in rules
- Matching against the root node
- Matching the document element
- Pulling nodes from the input document using *value-of* and *for-each*
- Pushing nodes through the stylesheet using the *apply-templates* instruction
- Choosing between push and pull
- Combining the two approaches

Advanced XSLT Stylesheets

- Dynamically building static content
- Controlling the transformation with the *output* and *message* instructions
- Adding conditions via *if* and *choose*
- Calling templates by name
- Using parameters and variables
- Numbering and sorting nodes
- Using Key Tables for grouping
- Modularising your stylesheets with *include*, *import*, and *apply-imports*
- Extra features introduced in XSLT 2.0

Introduction to XQuery

- The evolution of XML support in databases
- XML as a native RDBMS type
- Using XPath to search XML within the DB
- Why XSLT is not a natural fit for database admins
- Comparing XQuery and XSLT

Basic XQuery Expressions

- The FLWOR structure of an expression
- Beginning an expression with *for*
- Beginning an expression with *let*
- Using *let* to bind variables
- Using the *where* clause to filter results
- Ordering and positioning in XQuery

Practical Uses of XQuery

- Producing an HTML document from a database query
- Producing an XML document from a database query
- Using *distinct-values* to eliminate duplicates
- Dynamically generating the names of elements and attributes