WebSphere software

IBM

e business software

XML Tools

WebSphere Application Server - Express Beta

IBM Software Group

Agenda

- Features
- Different Tools
- XML Signature

XML Tool Features

- Comprehensive visual XML development environment.
- Tool set includes components for building DTDs, XML schemas, XML, and XSL files
  - XML Editor
    - Assigning an XSL, XML schema or DTD file to an XML file
    - New Java Bean XML/XSL client wizard
    - XML signature wizard
  - XML Schema Editor
    - Generate HTML from XSD
  - XSL Debugging and Transformation Tool
    - Local and Remote Transformation
  - XSL Editor
    - New Wizards
  - XML and SQL query wizard
  - XML to XML mapping editor
  - RDB to XML mapping editor

XML Authoring Tools

- Integrated set of visual tools for editing
  - XML Editor
  - XML Schema Editor (XSD)
  - DTD Editor
  - XSL Editor
- XML resource opened in an editor registered for editing the resource (DTD file has DTD editor)
- Outline view to show the content tree
  - add, remove operations
- Design view for structural editing with choice list
- Source view for editing source with intelligent assist
- Supports W3C standard
- Validation

Useful XML Utilities

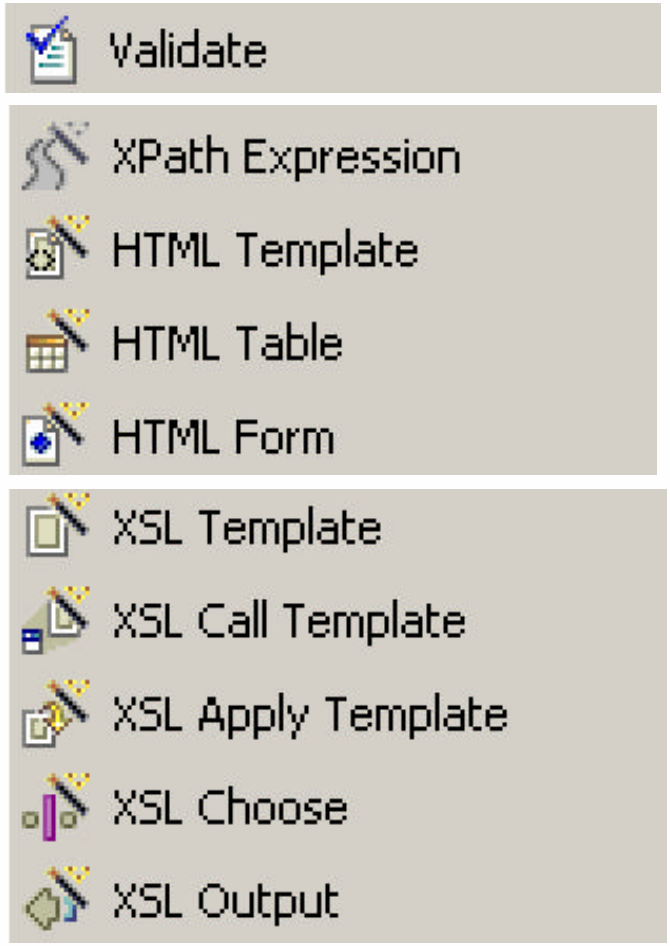
- Schema conversion
  - XSD to DTD
  - DTD to XSD
- HTML form from DTD
  - Creates fields in HTML form
  - Specify Servlet to handle Submit button for testing
- Java beans from DTD, XSD
  - Allows you to code directly to instance rather than DOM APIs
  - Bean created for each element in DTD, XSD
  - A Factory bean for creation of a new XML document
  - Sample program for using the beans created

XSL Debug Perspective

- Current XSL Element View
  - Property
  - XSL Expression
  - Actual Value
- Breakpoints View
- Template Call Stack
  - Name Template
  - Match Template
  - Priority
  - Mode
- Sessions View
  - Step forward through the result document
  - Step backward through the result document
  - Restart from the beginning
  - Run to breakpoint
  - Open the browser on the transformation result

Enhanced XSL Editor

1. Validate XSL file
2. Create XPath expression using XPath Wizard
3. Create a Default template for HTML header
4. Create a Table in XSL
5. Create a form in XSL
6. Add a template rule
7. Add a Call-Template
8. Add an Apply Template
9. Add Conditional Logic to the document
10. Add the output method
11. Debug the XSL stylesheet
12. Run transformation on the XSL



XPath Wizard

- XPath is a language for addressing parts of an XML document
- A new wizard helps users build XPath in an XML/XSD/XSL document.
- You can have arithmetic expressions and string, number, and Boolean expressions used to locate XML elements and attributes in a document
- Can be used as a simple query language (prior to XQuery)
- Examples
  - /PersonnelRec/Person/Name/Family
  - /PersonnelRec/Person/@Salary
- W3C Recommendation

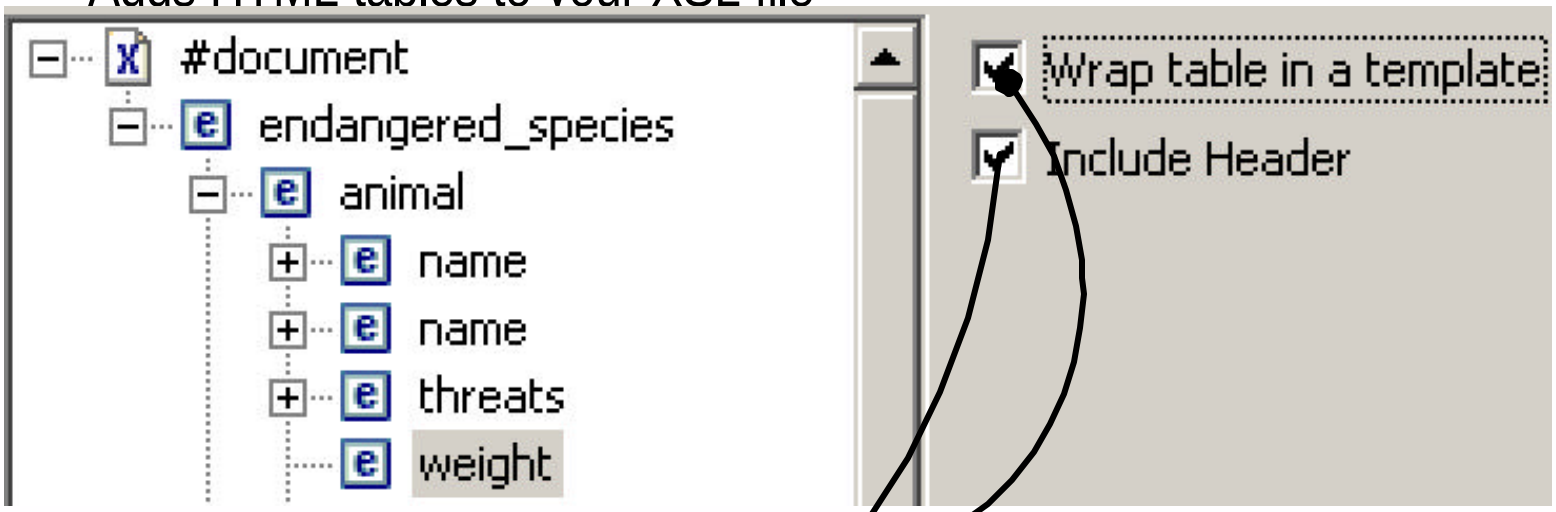
Default Template Wizard

- Creates a template that will generate an header together with the appropriate <xsl:output> element

```
<xsl:output method="html" encoding="UTF-8"/>
<xsl:template match="/">
  <html>
    <head>
      <title>Untitled</title>
    </head>
    <body>
      <xsl:apply-templates/>
    </body>
  </html>
</xsl:template>
```

XSL Table Wizard

- Adds HTML tables to your XSL file

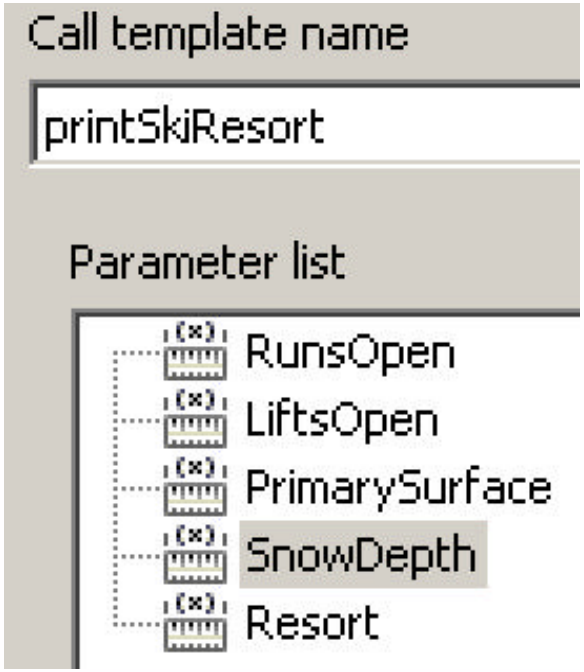



```
<xsl:template match="endangered_species">
  <table>
    <tr>
    </tr>
    <xsl:for-each select="/endangered_species/animal/weight">
      <tr>
      </tr>
    </xsl:for-each>
  </table>
</xsl:template>
```

Call Template Wizard

- The Call Template wizard adds <xsl:call-template> elements to your XSL file.
- This element calls a named template
- Call a template named "description" when the processor finds a customer element:

```
<xsl:template match="customer">
  <xsl:call-template name="description"/>
</xsl:template>
```



XSL Output Wizard

- The XSL Output wizard adds <xsl:output> elements to your XSL file

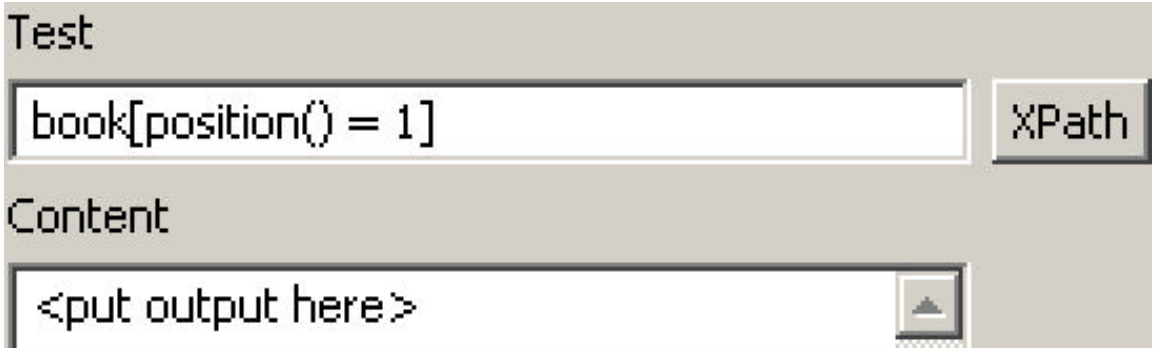
```
<xsl:output method="html" encoding="UTF-8" indent="no"/>
```

- Attributes
  - cdata-section-elements
  - doctype-public
  - encoding
  - indent
  - media-type
  - indent
  - method
  - omit-xml-declaration
  - standalone

XSL Condition Wizard

- XSL Condition Wizard
  - adds <xsl: choose> elements for XSL files
  - adds <xsl: otherwise>
  - example:

```
<xsl:choose>
  <xsl:when Books="book[position() =1]"">
    ... some code i.e.: HTML ...
  </xsl:when>
  <xsl:otherwise>
    ... some code ....
  </xsl:otherwise>
</xsl:choose>
```

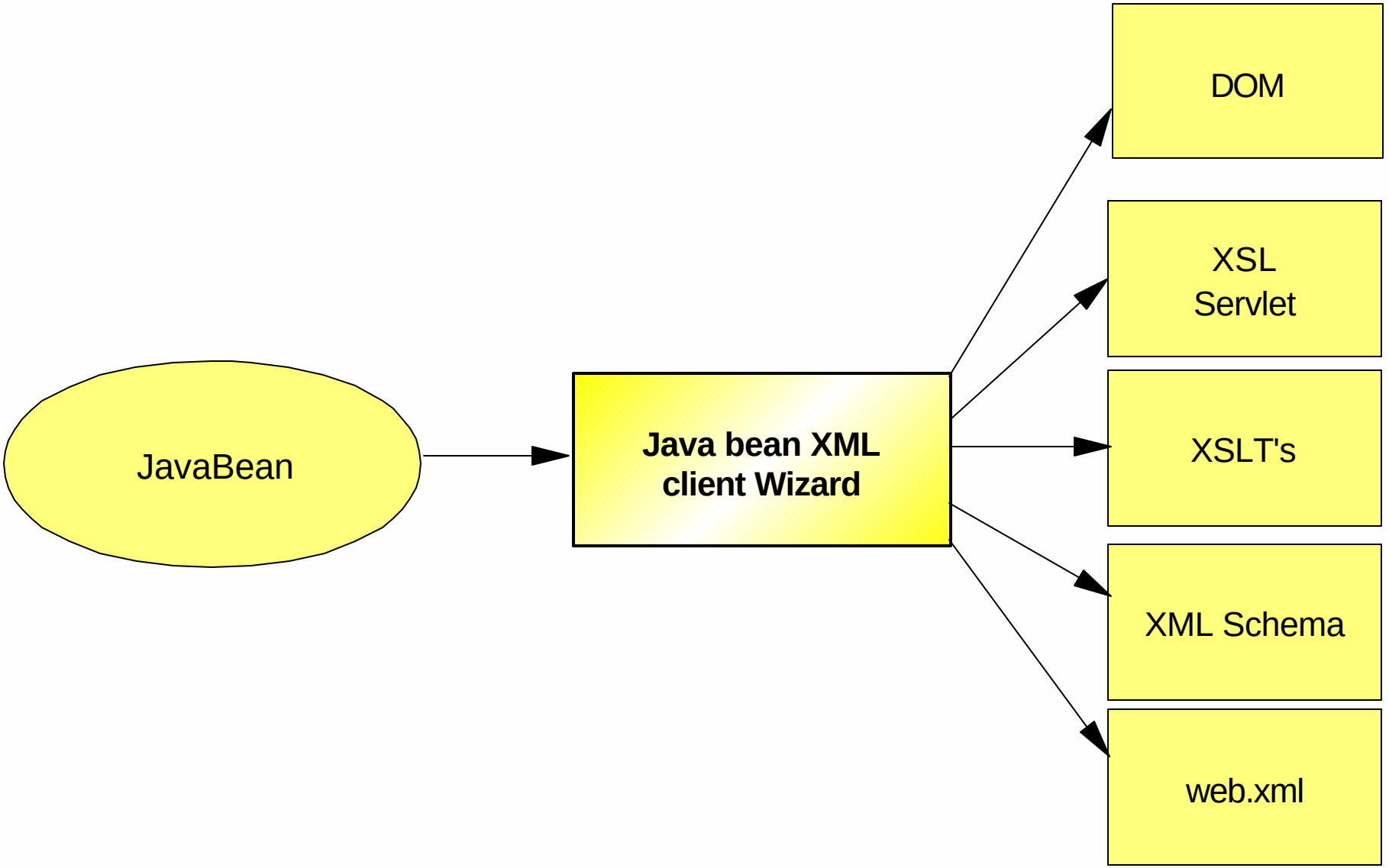



```
<xsl:choose>
  <xsl:when test="book[position() = 1]">
    <put output here>
  </xsl:when>
</xsl:choose>
```

XSL Support *(continued)*

- The XSL Stylesheet Editor plugin
  - Is a visual editor for XSL stylesheets providing a WYSIWYG editor window and supporting views that allow a user to manipulate the output of an XML/XSL stylesheet transformation to make changes to the stylesheet. Specifically it provides a multipage editor part containing a
    - source and tree views of the XML samples source
    - rules (similar to outline) views of the XSL stylesheet
  - A new release is currently under development that adds a customization API, better tool support, Xalan 2.3 support and NLS and Linux support.

Java Bean XML Client Wizard

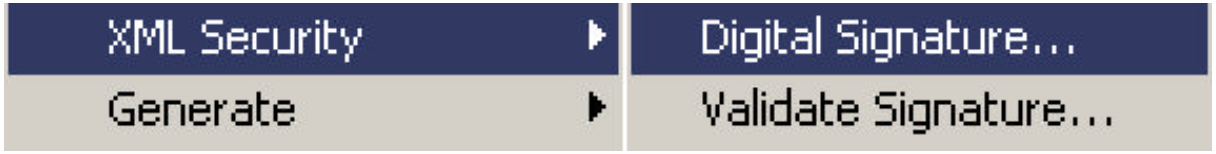


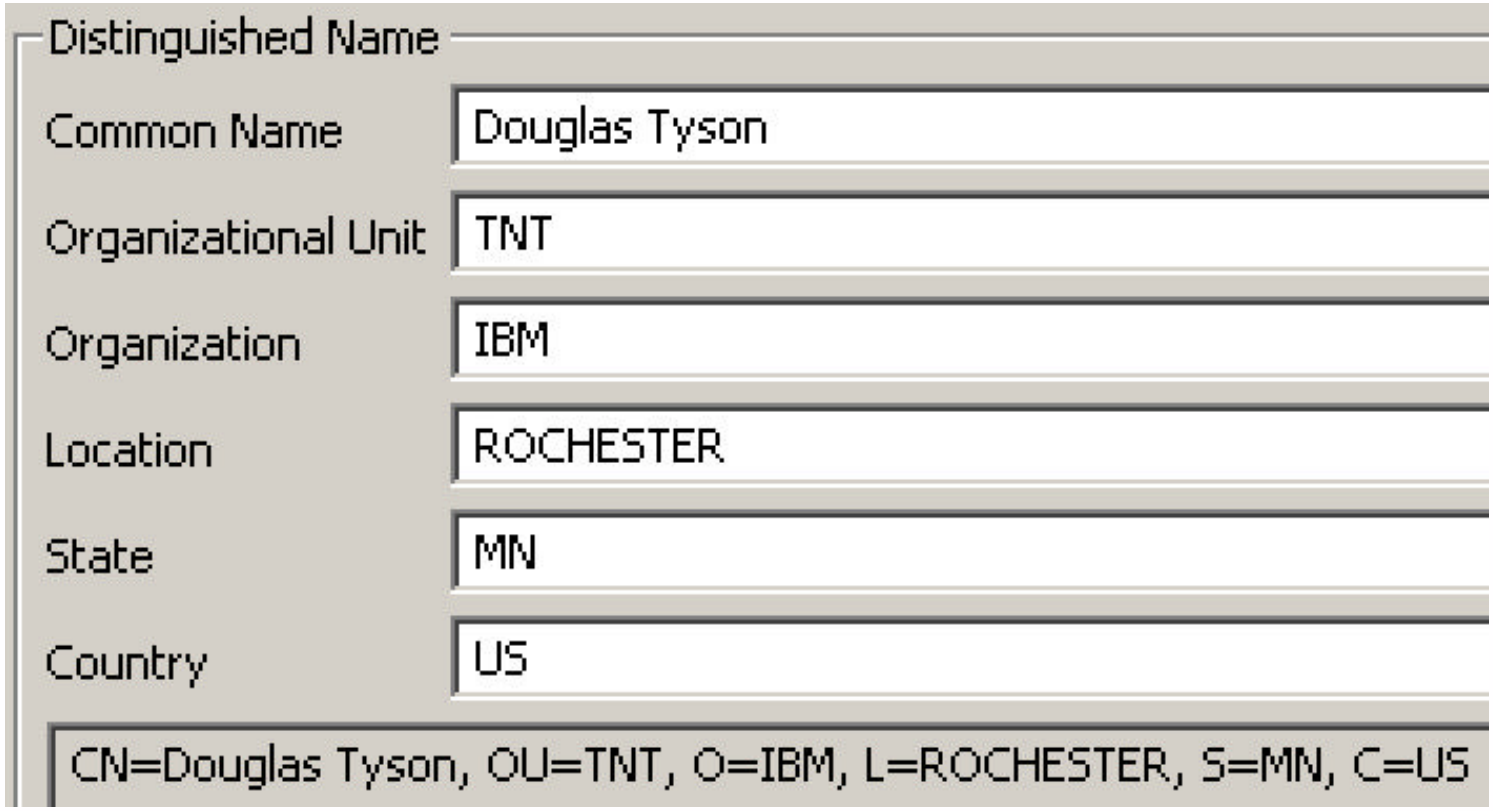
Java Bean XML Client Wizard Steps

- Specify the folder and Java package
- Browse to bean and choose bean properties
- Design the input form by specifying the page properties and the bean properties (fields) that the generated web page will expose to the user for input
- Design the results form
- Optional: Specify the prefix for generated files
- Files generated:
  - Input XML Form (Prefix.xsl)
  - Result XSL stylesheet (PrefixResult.xsl)
  - Splash screen (Prefix.html)
  - XSL Servlet (PrefixXSLServlet.java)
  - XML DOM (PrefixXML.java)
  - XML schema (Prefix.xsd)

XML Signatures

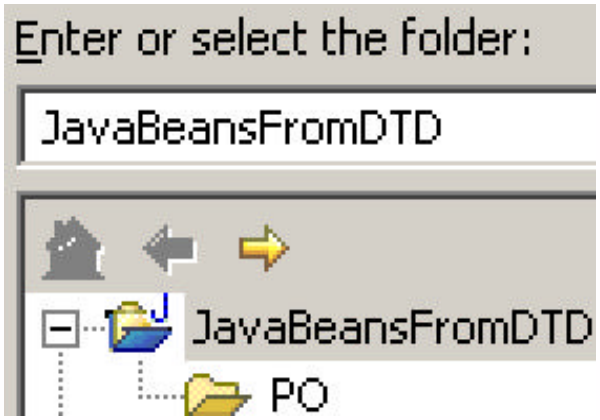
- Create Digital Signatures for an XML file
- Manually edit signature after creation

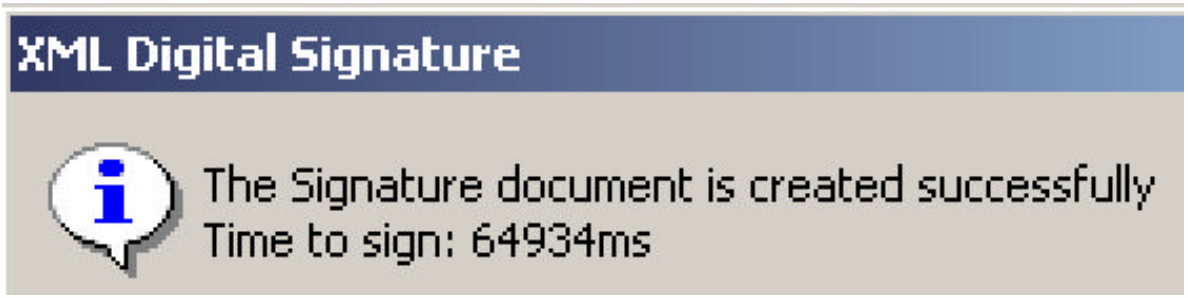




XML Signatures *(continued)*

- Create an embedded signature or external signature
- Specify the filename that contains the XML signature





WebSphere software

IBM

Generating and Validating Digital Signatures

- Case 1: Generate Signature and validate
  - Valid
- Case 2: If the user modifies the data, such as adding a space, then when they run Validate, they should get this.
  - Invalid
- Case 3: If the user changes one of the signature key values (those cryptic numbers)
  - Invalid

Feature Summary

- The XSL Source Editor now has a validate button to validate .xsl files (XSL Source Editor)
- A new wizard has been added to build XPath expression (XSL source, XSL trace-debug)
- XSL Source Editor now has 2 content outline views - template only and default XML tree view (XSL source editor)
- New look & functions for XSL trace. It is more like the Java debugger, with breakpoints, call stack, current XSL element, session views etc. Also change the name to XSL Debugger. We have a new XSL Debug Perspective too. (XSL trace/debug)
- XSL generation from XHTML (XSL Source editor)
- Generating HTML files from XML schema files (XML schema editor)
- Java Bean XML Client Wizard - similar to Java bean web client - but produces XML & XSL from a bean instead of JSP
- Assign XSL, DTD, and XSD to an XML document (XML editor)

Section

Appendix

Appendix - Other XML-related technologies

- Xerces
  - IBM developed and made available a parser known as XML4J that was a complimentary download. IBM donated this parser to the Apache Open Source initiative, and the current version of the IBM parser is known as Xerces. Xerces (the name comes from the Xerces blue butterfly) is a set of parsers compatible with XML. Xerces parsers are available for the Java language and C++, implementing the DOM and SAX standards.
- XHTML
  - The W3C redefined the HTML standard based on XML. The W3C describes XHTML as a reformulation of HTML 4.0 as an application of the XML. XML is a structured set of rules for how to define any kind of data; anyone can invent a particular set of markup for a particular purpose. As long as everyone uses it (the writer and an application program at the receiver's end), it can be adapted and used for many purposes, including describing the appearance of a Web page. Consequently, reframing HTML in terms of XML seemed reasonable. The result is XHTML, a particular application of XML for expressing Web pages.

Appendix (continued)

- XIS
  - XML Information Set (Infoset) is a candidate specification of the W3C. It provides a consistent set of definitions for use in other specifications that need to refer to the information in a well-formed XML document. An information item is an abstract representation of some part of an XML document; each information item has a set of associated named properties.
- XLink
  - XML Linking Language (XLink) allows elements to be inserted into XML documents to create and describe links between resources. It uses XML syntax to create structures that can describe links similar to the simple unidirectional hyperlinks of today's HTML, as well as more sophisticated links. The Extended Markup Language Pointer Language (XPointer) may be used with XLink. An XLink link is an explicit relationship between resources or portions of resources. It is made explicit by a linking element, which is an XLink-conforming XML element that asserts the existence of a link.
- XML Base
  - XML Base is an XML construct to describe links between resources. Base allows authors to explicitly specify a document's base Uniform Resource Identifier (URI) for resolving relative URIs in links to external images, applets, form-processing programs, stylesheets, and so on.

Appendix (continued)

- XML Schema
  - XML Schema is an XML language for describing and constraining the content of XML documents. It is a W3C proposal to resolve the XML DTD limitations in defining complex relationships between data elements and their usage. XML Schema lets you define the number of possible occurrences of an element with some precision. It is a scoped language, each definition being visible only within the schema element where it is defined along with all its descendants.
- XMLNS
  - XML Namespaces (XMLNS) is a recommendation to extend the data model to allow element type names and attribute names to be qualified with a URI. The XSL specifications differentiate between different elements using the concept of namespaces. Formatting tags typically start with the prefix "xmlns:fo=". Transformation tags typically start with the prefix "xmlns:xsl=". Namespaces are a mechanism to differentiate a series of tags. Namespaces typically are specified as an attribute of the root element so that they apply to the entire XML document.

Appendix (continued)

- XML-QL
  - Extensible Markup Language-Query Language (XML-QL) is one of two W3C initiatives to describe an XML query language. XML-QL draws from database technology traditions with focus on large repositories, integration, creation of new views of existing data, and transforming data into common data exchange formats.
- XPath
  - XPath Language (XPath) allows a particular part of an XML document to be identified. An XPath addresses parts of an XML document using a compact syntax to model an XML document as a tree of nodes. During the development of XSL and XSLT, the designers realized that both specifications needed some form of mechanism to define tree structures along with the ability to specify subtrees for processing. To accommodate both specifications, the tree expression component was combined into another specification known as XPath.
- XPointer
  - XPointer is an XML-linking specification. It references a fragment of a document that can be passed to the server as a Uniform Resource Locator (URL), thus making the process application load the chunk of data rather than the entire document. XPointer is an extension to XPath for addressing points, ranges, and nodes to locate information by string matching.

Appendix (continued)

- XQL
  - Extensible Query Language (XQL) is one of two W3C initiatives for describing an XML query language. XQL focuses on integrating full text and structured queries, describing the structured search, and creating multiple presentations from a single document.
- XSL
  - XSL is defined by W3C to add formatting information to XML data. It is based on two standards: Cascading Stylesheets (CSS) and Document Style Semantics and Specification Language (DSSSL). XSL started as a single specification. During its development, it was determined that the formatting portion should be uncoupled from the transformation portion of the specification. Thus, the formatting components remained as part of the XSL spec, and the transformation component was split out into a separate XSLT specification.

Appendix (continued)

- XSLT
  - XSLT is a W3C transformation language that defines a common language for transforming one XML document into an XML document with a different structure. XSLT can be thought of as an extension of XSL. XSLT shows how the XML document should be reorganized into another data structure that could be presented by following an XSL stylesheet. It is used to describe how to transform the source tree or data structure of an XML document into the result tree for a new XML document, which can be different in structure. The coding for the XSLT also is referred to as a stylesheet and can be combined with an XSL stylesheet or used independently, thus enabling interoperability.