



# Technical Track Session

## XML Techie Tools

Tim Bornholt





# XML Parsers

- Many types of XML Parsers are available today to meet specific business needs.
- Generic parsers
  - SAX and DOM
- Data Binding





# SAX and DOM

- Streaming Api for Xml
  - Very fast
  - One and only one linear pass through the doc
  - Uses callback methods to handle events
- DOM
  - Flexible; can walk the tree up and down
  - Memory intensive
  - Method calls like getParent(), getChildNodes()





# Data Binding Toolkits

- Take XML Schemas or XML instances and parse into objects used by the language.
- Usually create arrays for recurring nodes.
- Objects and method names match the tag names.
  - getSchool()
  - addAddress()





# Common Uses of XSL

- Two main scenarios for using XSL transformations:
  - Data conversion
  - Publishing





# Data Conversion Applications

- Data conversion will not go away just because XML has been invented.
  - Different data models
  - Different ways of representing things
  - Different subsets needed by different people





# Sorting XSL

- Within a `<apply-templates>` or `<call-template>` block

```
<xsl:sort  
  select="i:PersonalIdentifiers/i:SSN" data-  
  type="number" order="ascending"/>
```

```
<xsl:sort select="i:Birth/i:BirthDate" data-  
  type="text" order="descending"/>
```





# Convert XML to XML

- XSLT is an obvious choice for converting XML from one schema to XML in a different schema.
- Convert a COD 2.0c document to COD 2.0d.
  - New namespace declaration
  - Two new optional fields added







# COD 2.0c to COD 2.0d

- XSL is namespace aware.

```
<xsl:template match="*">  
  <xsl:element name="{local-name()}">  
    <xsl:copy-of select="@*" />  
    <xsl:value-of select="text()" />  
    <xsl:apply-templates select="*" />  
  </xsl:element>  
</xsl:template>
```



**Demo**





# XML to a Different Format

- May have internal legacy systems without access to XML parser.
- XML can be translated into flat file.





# ISIR to Flat File (CSV)

- Tried:

```
<xsl:for-each select="*">
```

```
  <xsl:value-of select="."/>
```

```
  <xsl:text>,</xsl:text>
```

```
</xsl:for-each>
```

- Doesn't print anything if tag is missing.
- This doesn't work for csv or fixed length.





# ISIR to Flat File

- Manually pick every column to be displayed.

```
<xsl:value-of select="i:LastName"/>
```

```
<xsl:text>,</xsl:text>
```

```
<xsl:value-of select="i:FirstName"/>
```

```
<xsl:text>,</xsl:text>
```

```
<xsl:value-of select="i:MiddleInitial",
```

```
<xsl:text>,</xsl:text>
```



**Demo**

12





# Different Format to XML

- This can be a bit more cumbersome.
- Implement a SAX interface or a DOM interface that parses your file.
- Then treat your alternate format as XML.
  - Translate to HTML or PDF with XSLT





# More than One Input Document

- Using the `XSLT document()` function, additional source files can be loaded.





# XSLT Document() Function

```
<xsl:apply-templates  
  select="document('disb.xml')"/>
```

- Can insert the whole document or partial based on Xpath.
- Different than xsl:include and xsl:import.
  - Those let you insert one stylesheet in another





# Comparing Two XML Files

- Want to compare published schemas with data stored in the registry.
- Pass one file on the command line.
- Load the other file with document().
- For all elements in the first file, find a corresponding element in the second.







# Comparing Two XML Files

```
<xsl:variable name="elementName"  
  select="@name"/>
```

```
<xsl:value-of select="$elementName"/>
```

```
<xsl:apply-templates select="." mode="xmlverb"/>
```

```
<xsl:variable name="ad"  
  select="document('registry/aid-  
disbursement.xsd')"/>
```

```
<xsl:apply-templates  
  select="$ad//*[ @name=$elementName]"  
  mode="xmlverb"/>
```



**Demo**



# Publishing with XSL

- Designed for human consumption.
- Most common uses:
  - HTML – Web pages
  - PDF – Printable reports





# XML to HTML

- Can be translated several ways:
  - Transformation handled by the browser
  - Transform entire XML document displaying the fields necessary
  - Convert the document to contain only the fields necessary and transform that.





# Translation by the Browser

- Can be accomplished by all modern browsers
  - Internet Explorer 4+
  - Mozilla Firefox 0.9, 1.0+
  - Opera 6.0+
- However!!! Slightly different syntax for different browsers and different versions.





# Server Side Translation

- Java Servers can use Servlets or JSPs.
  - Requires a minimal amount of code
- Microsoft IIS Servers can use XSL ISAPI extension.





# XML to HTML

```
<xsl:template match="i:CommonRecordISIR">
```

```
<HTML>
```

```
<HEAD><title>ISIR HTML Transformation</title>
```

```
</HEAD>
```

```
<BODY>
```

```
<xsl:call-template name="start-table"/>
```

```
<xsl:apply-templates select="i:Student"/>
```

```
<xsl:call-template name="end-table"/>
```

```
</BODY>
```

```
</HTML>
```

```
</xsl:template>
```





# XML to HTML

- Start Table:

```
<xsl:template name="start-table">
  <xsl:text disable-output-
    escaping="yes">&lt;table
      border="1"&gt;</xsl:text>
  <tr><th>Name</th><th>SSN</th><th>Birth
    Date</th></tr>
</xsl:template>
```





# XML to HTML

- Outputting data elements

```
<xsl:template match="i:Student">
```

```
<tr>
```

```
<td><xsl:apply-templates
```

```
  select="i:Name"/></td>
```

```
<td><xsl:value-of
```

```
  select="i:PersonalIdentifiers/i:SSN"/></td>
```

```
<td><xsl:value-of
```

```
  select="i:Birth/i:BirthDate"/></td>
```

```
</tr>
```

```
</xsl:template>
```



**Demo**







# XML to PDF

- Can use XSL Formatting Objects – XSL:FO
  - Much more control than printing an HTML page
  - Able to specify page headers and footers, page numbers, footnotes, etc.





# XML to PDF

- Different namespace than XSL
- Many special tags for functionality
  - fo:region-body
  - fo:static-content
  - fo:block
  - fo:page-sequence
  - fo:external-graphic



**Demo**





# Caveats

- Most all XSLT processors hold the entire XML tree in memory.
- Can split XML document into chunks and process independently.
  - If this isn't practical, XSLT probably isn't a good choice for large documents.





# Contact Information

We appreciate your feedback and comments.  
We can be reached at:

Tim Bornholtz

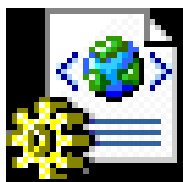
Phone: (202) 377-3465

Email: [tim.bornholtz@ed.gov](mailto:tim.bornholtz@ed.gov)

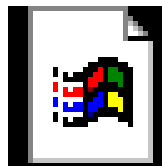




# COD 2.0c to COD 2.0d Demo



Convert.xsl

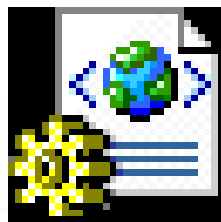


CommonRecord2.0c.xml

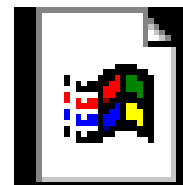
All files are provided as samples to demonstrate the concept only



# ISIR to Flat File Demo



Csv.xsl

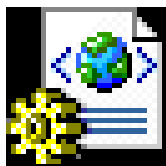


isir-xml-80.xml

All files are provided as samples to demonstrate the concept only



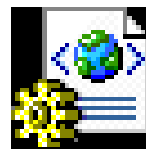
# Comparing Two XML Files Demo



transform.xsl



xmlverbatim.css



xmlverbatim.xsl



aid-disbursement.xsd

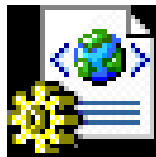


FSA\_ApplicationSubmission\_CC\_1.0.2.xsd

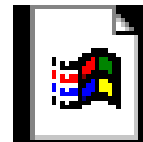
All files are provided as samples to demonstrate the concept only  
This demo requires XSLT 2



# XML to HTML Demo



Tohtml.xsl



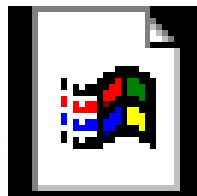
isir-xml-80.xml

All files are provided as samples to demonstrate the concept only





# XML to PDF Demo



Isir-pdf.xsl



isir-xml-80.xml

All files are provided as samples to demonstrate the concept only