



XML Pipeline Language (XPL)

*W3C XML Processing Working Group
Cannes, February 2006*

Erik Bruchez

Sw. Arch. / Co-Founder
ebruchez@orbeon.com

History

- 2002: birth of XPL
 - Reaction to needs + vision
 - Do better than Cocoon pipelines
- 2004: Open source implementation
 - <http://forge.objectweb.org/projects/ops/>
- 2005: W3C submission (formal draft spec)
 - <http://www.w3.org/Submission/xpl/>

XPL Builds on What You Know

- XML-based syntax
- XML infosets
- Pipelines
- Components (“processors”)
- “Steps” (processor instances)
- Inputs and outputs

Simplicity: A Driving Principle

- Consume and produce only XML infosets
- 0..n inputs
- 0..n outputs
- Pipelines are components too

Advanced Features

- Conditionals
- Iterations
- Validation of any input / output
 - W3C Schema
 - Relax NG
- Embedded infosets

A Simple Syntax

- p:pipeline
- p:processor
- p:input / p:output
- p:choose / p:when / p:otherwise
- p:for-each

Functional Processing Model

- Pipelines and components as functions
- Lazy evaluation
 - dependencies trickle from outputs required
 - processors decide what inputs they need
- Very natural model to implement

OPS XPL Engine

- Matured over 3 years
- Implements XPL pre-1.0
- SAX-based
- Built-in caching
 - XSLT transformers
 - Compiled schemas
 - Etc.

Cascaded Transforms with XPL

```
<p:pipeline>
  <p:input name="my-input"/>
  <p:output name="my-output" infosetref="#my-output"/>

  <p:processor name="oxf:xslt">
    <p:input name="data" infosetref="#my-input"/>
    <p:input name="config" infosetref="my-stylesheet1.xsl"/>
    <p:output name="data" infoset="first-result"/>
  </p:processor>

  <p:processor name="oxf:xslt">
    <p:input name="data" infosetref="#first-result"/>
    <p:input name="config" infosetref="my-stylesheet2.xsl"/>
    <p:output name="data" infoset="my-output"/>
  </p:processor>

</p:pipeline>
```

Adding Schema Validation

```
<p:pipeline>
  <p:input name="my-input" schema-href="my-input.xsd"/>
  <p:output name="my-output" infosetref="my-output" schema-href="my-output.rng"/>

  <p:processor name="oxf:xslt">
    <p:input name="data" infosetref="#my-input"/>
    <p:input name="config" infosetref="my-stylesheet1.xsl"/>
    <p:output name="data" infoset="first-result" schema-href="result1.xsd"/>
  </p:processor>

  <p:processor name="oxf:xslt">
    <p:input name="data" infosetref="#first-result"/>
    <p:input name="config" infosetref="my-stylesheet2.xsl"/>
    <p:output name="data" infoset="my-output"/>
  </p:processor>
</p:pipeline>
```



Other Examples

[Source code of more complex examples]



The End!