

XSLT Development with <oXygen/>

Part3: Debugger, Profiler and Unit Testing

Octavian Nadolu

octavian_nadolu@oxygenxml.com
[@OctavianNadolu](#)



Generate HTML



```
<books>
  <book targeturi="bookTargetUri">
    <title>XSLT 2.0 Programmer's Reference</title>
    <author>Michael Kay</author>
    <image description="XSLT 2.0" src="XSLT.jpg"/>
    <paperback>955 pages</paperback>
    <publisher>Wrox</publisher>
    <isbn>0764</isbn>
    <description>XSLT 2.0 Programmer's Reference,
      3rd Edition, is the authoritative reference
      guide to the language</description>
  </book>
</books>
```



[XSLT 2.0 Programmer's Reference](#)

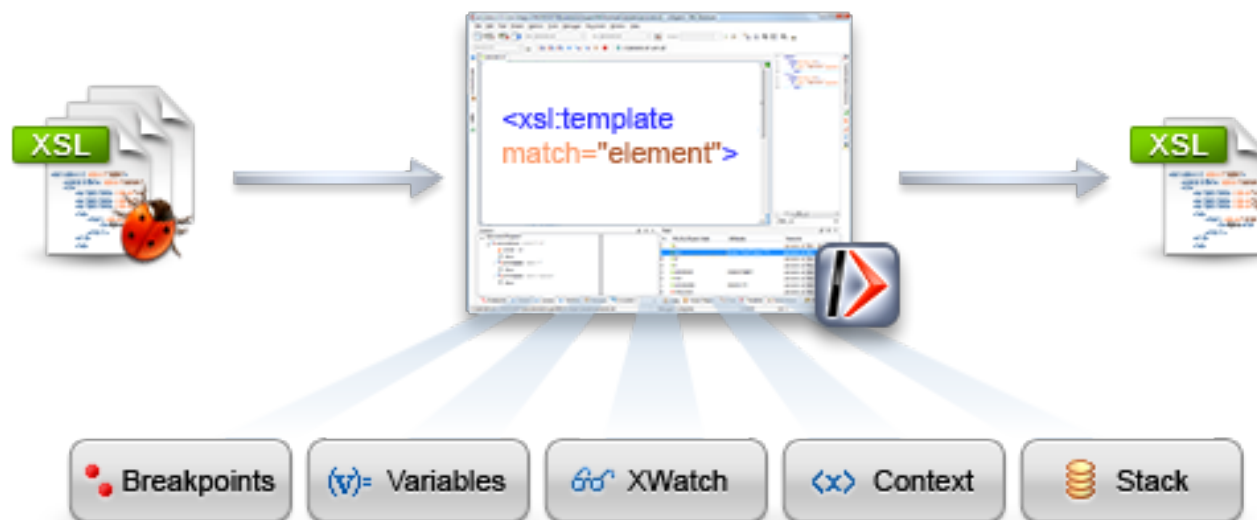
by Michael Kay

Paperback: 955 pages | **Publisher:** Wrox | **Isbn:** 0764

XSLT 2.0 Programmer's Reference, 3rd Edition, is the authoritative reference guide to the language.

Demo – step 1

- Find and fix problems in stylesheet

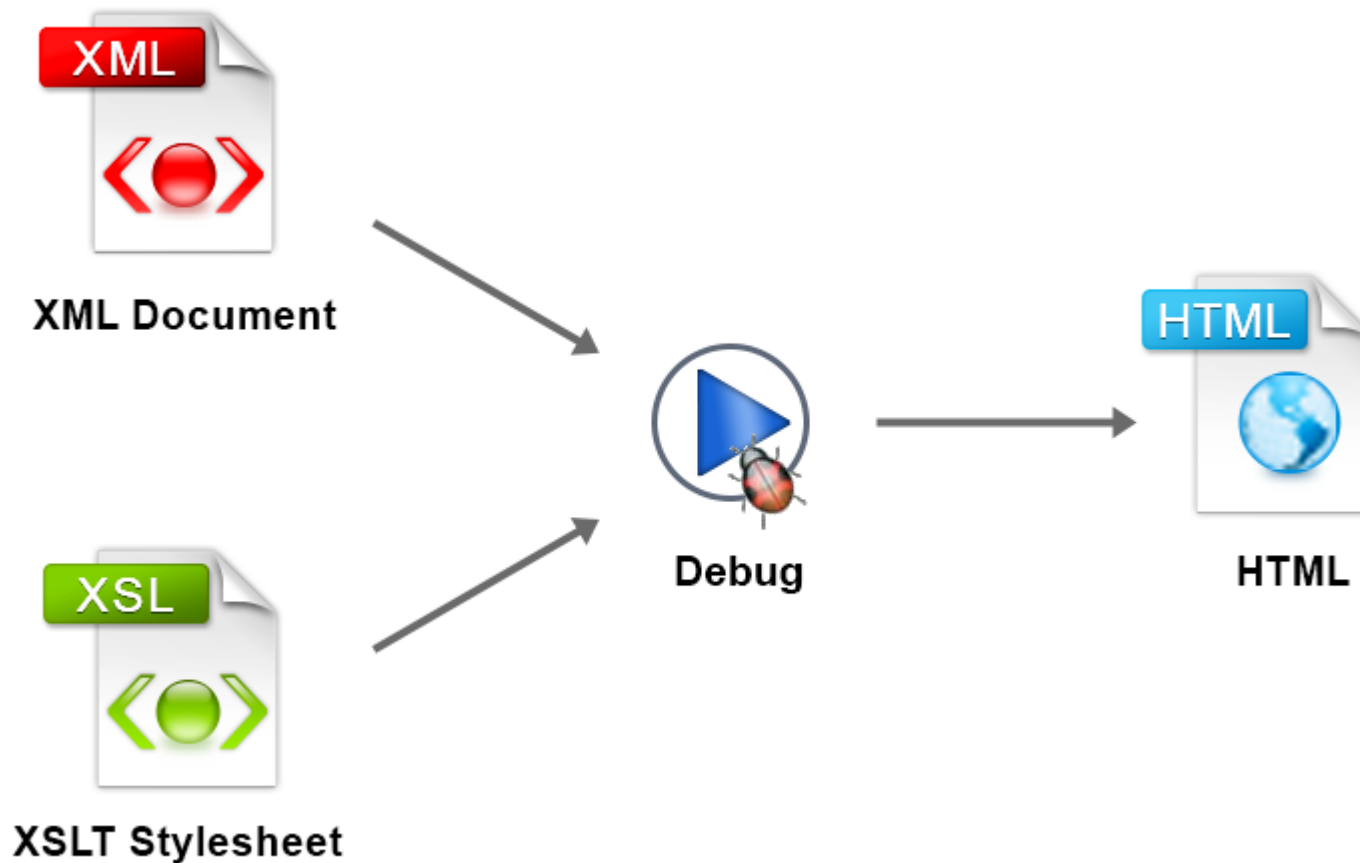


Conclusion

- Debugger perspective
- Support for the mostly used XSLT processors
- XSLT 1.0, 2.0, 3.0

Demo – step 2

- Save and reuse debugging configuration

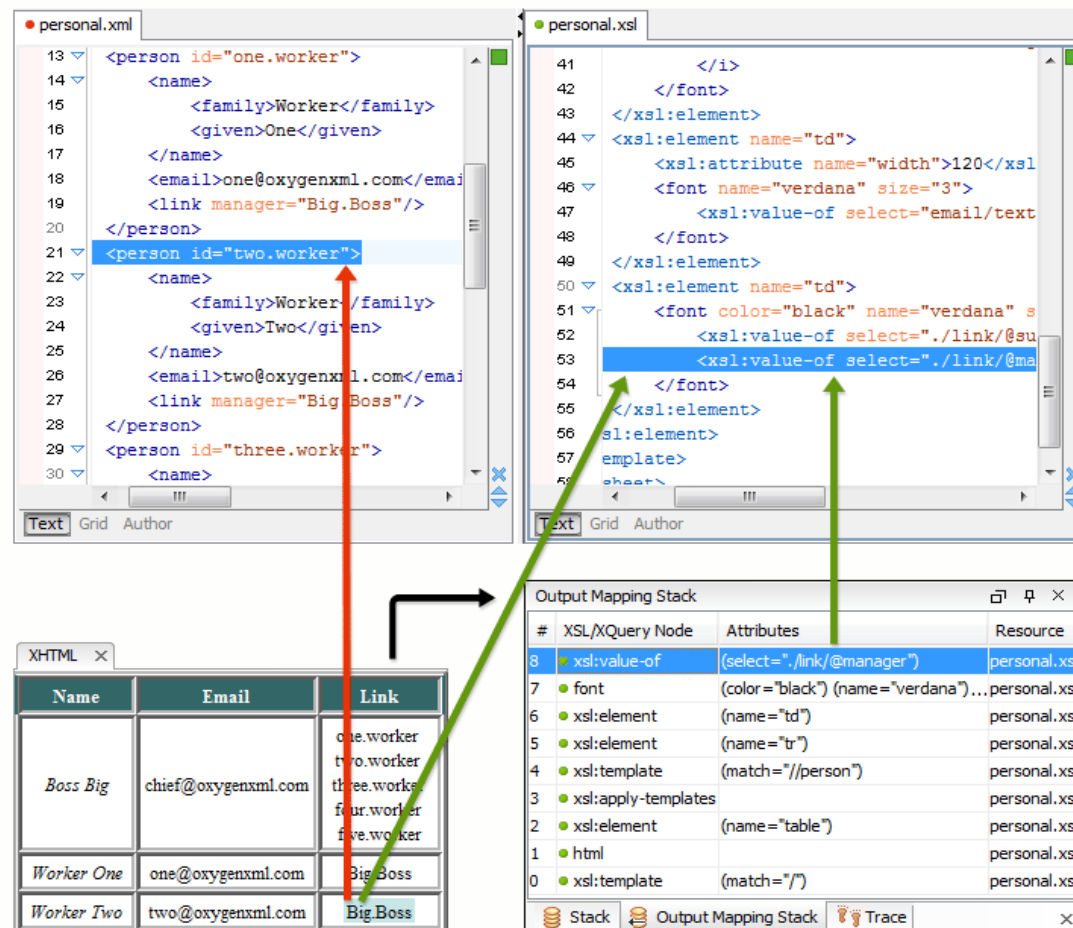


Conclusion

- Save configuration
- Reuse debugging environment

Demo – step 3

- Determining what XSLT expression generated particular output



The screenshot illustrates the process of tracing an XSLT expression to its output in the oXygen XML Editor. It consists of three main components:

- personal.xml (Source XML):** Contains three person records. The second record, `<person id="two.worker">`, is selected. It has a `<link manager="Big.Boss"/>` element.
- personal.xsl (XSLT Template):** Shows the transformation logic. A `<xsl:value-of select="email/text">` expression is highlighted, which is linked to the `email` attribute of the `td` element in the output table.
- Output Mapping Stack:** A table showing the mapping of XSLT nodes to the output. The stack is as follows:

#	XSL/XQuery Node	Attributes	Resource
8	xsl:value-of	(select="./link/@manager")	personal.xml
7	font	(color="black") (name="verdana")...	personal.xml
6	xsl:element	(name="td")	personal.xml
5	xsl:element	(name="tr")	personal.xml
4	xsl:template	(match="//person")	personal.xml
3	xsl:apply-templates		personal.xml
2	xsl:element	(name="table")	personal.xml
1	html		personal.xml
0	xsl:template	(match="/")	personal.xml

Arrows indicate the flow of data: a red arrow points from the `link` element in the XML to the `link` attribute in the `td` element of the output table; a green arrow points from the `link` attribute in the `td` element to the `link` attribute in the `td` element of the output table.

Conclusion

- Output to source mapping
- Output Mapping Stack view

Demo – step 4

- Using breakpoints to find and fix problems

```
31      <xsl:with-param name="conditional" select="0"/>
32    </xsl:call-template>
33  <xsl:choose>
34    <xsl:when test="$placement = 'before'">
35      <xsl:call-template name="formal.object.heading"
36        <div class="{ $class }-contents">
37        <xsl:apply-templates/>
38      </div>
39    <!-- This doesn't belong inside formal.object;
```

Conclusion

- Breakpoints view
- Persistent breakpoints
- Conditional breakpoints
- Conditions evaluated at each step in the current debugger context

Demo – step 5

- Performance profiling of XSLT Stylesheets



Conclusion

- XSLT Profiler
- Invocation Tree View
- Hotspots View

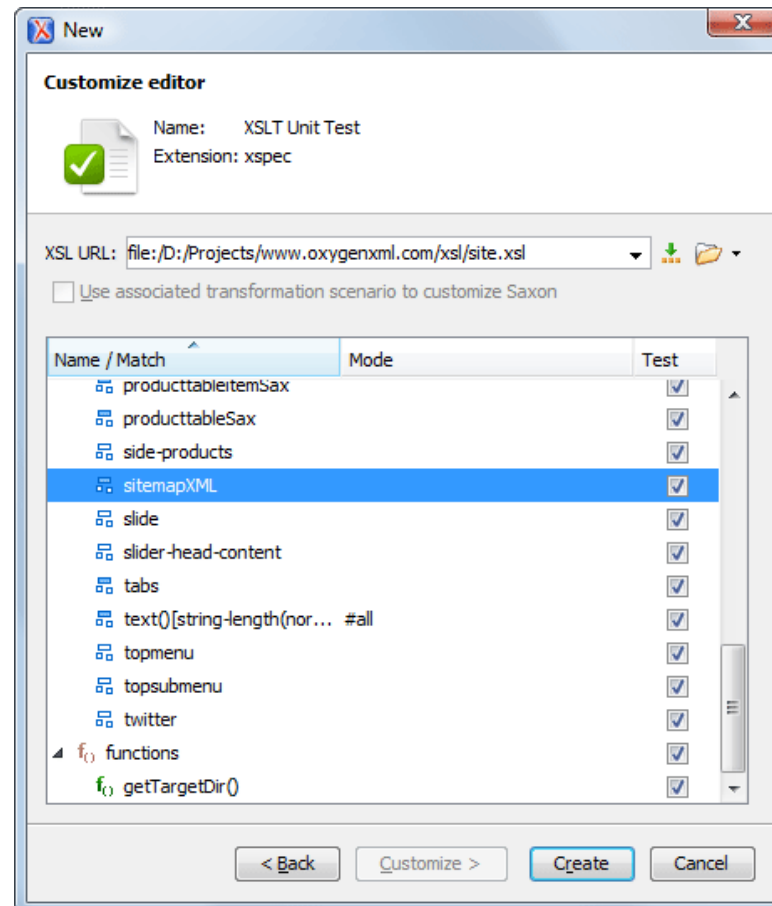
Demo – step 6

- Test your XSLT Stylesheets



Conclusion

- <oXygen/> provides XSLT Unit Test support based on XSpec



<https://code.google.com/p/xspec/wiki/WritingScenarios>

Thank you!

Questions?

<oXygen/> XML Editor
<http://www.oxygenxml.com>
octavian_nadolu@oxygenxml.com
[@OctavianNadolu](#)

Q&A

Answers to your questions

If your question is not answered now, we will be in touch with an answer by email after the webinar.

The recording of the webinar will be available from our events page:

http://www.oxygenxml.com/events_programme.html#pastevents