HOWTO

Configure an External XSLT Transformer

Table of Contents

Introduction	1
Installation and Configuration	1

This article explains how to install and configure an external XSLT transformer in Oxygen XML Editor 3.0 or higher.

Introduction

Oxygen comes with several preconfigured XSLT transformers like Apache Xalan, Saxon and MSXML. Each is a processor able to interpret the instructions of an XSLT Stylesheet conforming to XSL Transformations Version 1.0 W3C Recommendation [http://www.w3.org/TR/1999/REC-xslt-19991116].

To enable the use of other XSLT transformers - faster, providing special XSLT extensions or a newer versions of Xalan or Saxon - the user can integrate and use a preferred transformer without switching between Oxygen and a command line prompt at run time.

Installation and Configuration

The transformer we will be installing is Saxon version 8.9 SA. If you are installing a different transformer, please exchange names, syntax and parameters to match your application.

Before you proceed, please ensure that you have copied and installed the new XSLT transformer as per the instructions of the vendor. It's also a good idea to test that it works from the command line or by using a utility that may have been provided by the vendor.

There are two ways for configuring an external XSLT engine as XSLT transformer in Oxygen: as JAXP transformer invoked through the Java JAXP interface or as custom XSLT transformer invoked with a configurable command line.

Procedure 1. Configuring an External XSLT Engine as JAXP Transformer

- 1. Exit Oxygen if it is started.
- 2. Copy the JAR file(s) of the transformer (just saxon8sa.jar in the case of Saxon 8.9 SA) to the lib subdirectory of your Oxygen installation directory.
- 3. Start Oxygen.
- 4. From the main menu select Options → Preferences → XML → XSLT / FO / XQuery → XSLT. The XSLT preferences panel is displayed in the Preferences dialog.

ype filter text	Q	XML / XSLT-FO-XQuery / XSLT
	-	
		JAXP XSLT Transformer
Elements by Prefix		To use your own transformer, set the value of the
Open/Save		system property "javax.xml.transform.TransformerFactory"
Code Templates		Value
Diagram		value jcom.saxonica.scriemaAwareTransformerFactory
Spell Check		
Document Checking		Engine used for XSLT validation (Scenario transformer takes precedence)
Default Schema Association		VELT 1. 0 Velidete with Covers F. F.
Control Custom Validation		
CSS Validator		X5LT 2.0 Validate with Saxon8B
XMI Catalog		,
XML Catalog		
MIL Instances Generator		
⊡-XSLT-FO-XQuery		
⊡…Saxon		
Saxon6		
⊡Saxon8		
Advanced		
XSLTProc		
MSXML NET		
MISAME.NET		
Advanced		
Debugger		
Profiler		
FO Processors		
XPath		
LCustom Engines		
Import		
Data Sources		
i able Filters		Global Options C. Project Options Restore Defaulty
	-	Result provide provide a resource personal and rescure personal and resc

- 5. Enter the name of the transformer factory, **com.saxonica.SchemaAwareTransformerFactory** for Saxon 8.9 SA, as the value of the JAXP property **javax.xml.transform.TransformerFactory**. Oxygen sets this value for the property and uses it next time it creates a JAXP transformation.
- 6. Click the OK button to save your changes and return to the editor.
- 7. Test the new XSLT transformer by configuring and running a transformation scenario using the JAXP transformer.

Edit scenario				
Name: JAXP XSLT transformer - test				
XSLT FO Processor Output				
XML URL: \${currentFileURL}	£	ø	B	N
XSL URL: rojects/userguide/frameworks/docbook/xsl/fo/docbook.xsl 💌	£	ø	B	<u>s</u>
More about \${currentFileURL}				
Use "xml-stylesheet" declaration				
Transformer: Saxon6.5.5				
MSXML3.0				
Append MSXML4.0				
Saxon.NET Additional 3				
Saxon8B				
Saxon8SA				
Xalan			Ca	ncel

The processing messages and results will be shown in the Message Pane of the editor.

Info	Description - 2 items		Location	Resource	System ID
-	E Failed to compile stylesheet. 1 error detected. F XPath syntax error at char 4 on line 4 in {'a' < '}:	Unmatched quote in expression	4:0	personal.xsl personal.xsl	file:/D:/projec
					×
					*
4					
Erro	rs				×

Procedure 2. Configuring an External XSLT Engine as Custom XSLT Engine

- 1. Start Oxygen if it is not started.
- 2. Go to the Custom Engines preferences panel from menu Options \rightarrow Preferences \rightarrow XML \rightarrow XSLT / FO / XQuery \rightarrow XSLT \rightarrow Custom Engines

Preferences				
Type filter text	XN	1L / XSLT-FO-XQuery /	⁷ Custom Engines	
		Name	Description	Engine type
		New	Test Saxon 8.9 SA as custom engin	Engine type
	- - -	Global Options (C Project	Options 😭	Restore Defaults
Apply				Can

3. Press the *New* button to create a new custom XSLT engine.

Command line			X
Engine type			
XSLT			-
Name			
Saxon 8.9 SA			
Description			
Test running Saxon 8.9 SA engine as external transformer	r		
Output - Encoding			
Default encoding			-
Error - Encoding			
Default encoding			-
Working directory			
\${pd}//saxon8.95A		±.	\bowtie
Command line			
java -cp .;saxon8sa.jar;saxon8-xpath.jar;saxon8-xqj.ja r;saxon8-sql.jar com.saxonica.Transform -t -o \${out} \${ xml} \${xsl}	*	Ø	2
ОК		Cano	el

4. Complete the dialog by selecting *XSLT* in the *Engine type* combo box, entering the name of the engine that will appear in the scenario edit dialog, the output stream encoding, the error stream encoding, a working directory (usually the directory where the custom XSLT engine is installed) and a command line executed when the engine is applied in a transformation scenario. The working directory and the command line can include editor variables like \${cfd} (meaning current file directory), \${pd} (meaning project directory), \${xml} (meaning the input XML document configured in the transformation scenario), \${out} (meaning the output file configured in the transformation scenario), etc.

🗴 Insert Edi	tor Variables	
You can use th	ne following editor variables:	
\${xml}	XML Document	
\${xmlu}	XML Document (URL)	
\${xsl}	XSLT/XQuery Document	
\${xslu}	XSLT/XQuery Document (URL)	
\${out}	The output file	
\${outu}	The output file (URL)	
		Close

- 5. Click the OK button to save your changes and return to the editor.
- 6. Test the new XSLT transformer by configuring and running a transformation scenario using the Saxon 8.9 SA transformer.

Edit scenario	 	×
Name: Custom XSLT engine - test	 	
XSL1 FO Processor Output		
XML URL: \${currentFileURL}	\ge	😂 🕓
XSL URL: file:/D:/projects/eXml/samples/personal.xsl	D	🔗 🕒
More about \${currentFileURL}		
Use "xml-stylesheet" declaration		
Transformer: Saxon6.5.5		
PSaxon.NET		
Saxon6.5.5		
Append Daxonob		
Xalan		
Ex Saxon 8.9 SA		
Saxon B 8.6		
OK		Cancel
	_	Cancer