

---

# 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

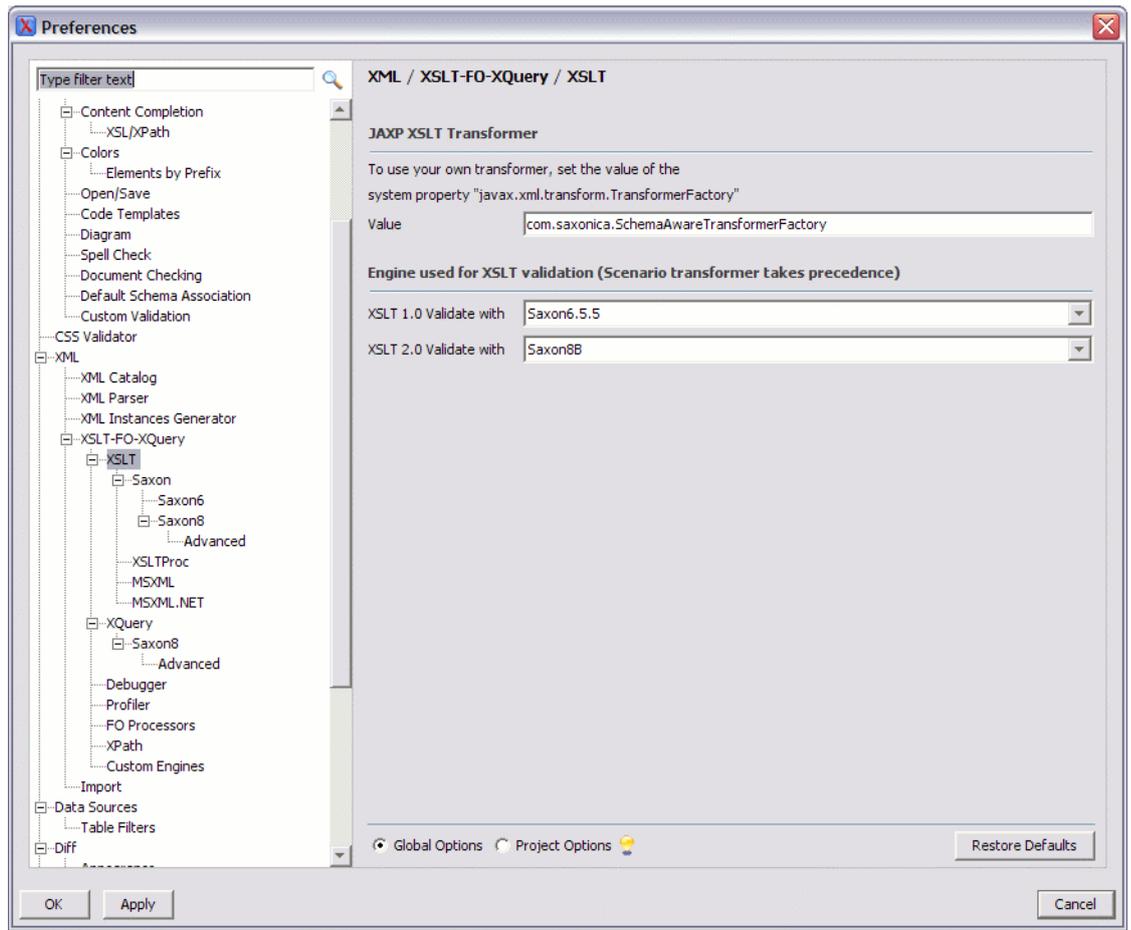
The transformer we will be installing is Saxon version 8.9 SA. If you are installing a different transformer, please exchange names, syntax and parameters to match your application.

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.

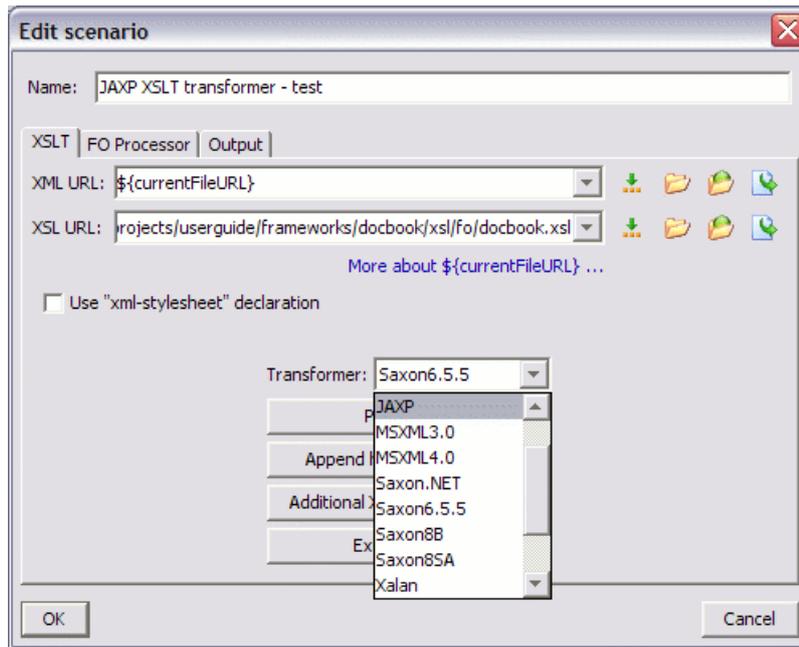
There are two ways for configuring an external XSLT engine as XSLT transformer in Oxygen: as JAXP transformer invoked through the Java JAXP interface or as custom XSLT transformer invoked with a configurable command line.

### Procedure 1. Configuring an External XSLT Engine as JAXP Transformer

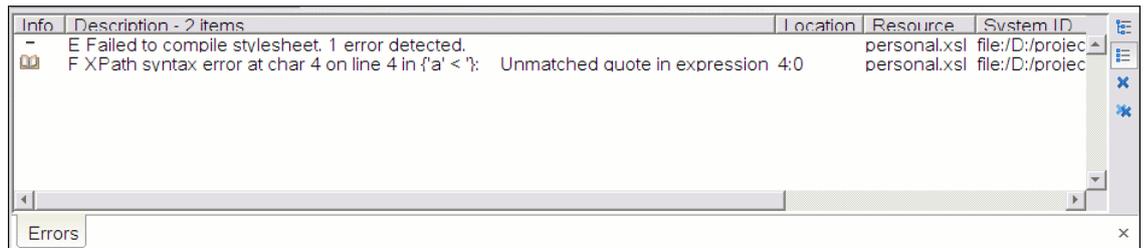
1. Exit Oxygen if it is started.
2. Copy the JAR file(s) of the transformer (just `saxon8sa.jar` in the case of Saxon 8.9 SA) to the `lib` subdirectory of your Oxygen installation directory.
3. Start Oxygen.
4. From the main menu select Options → Preferences → XML → XSLT / FO / XQuery → XSLT. The XSLT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



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.

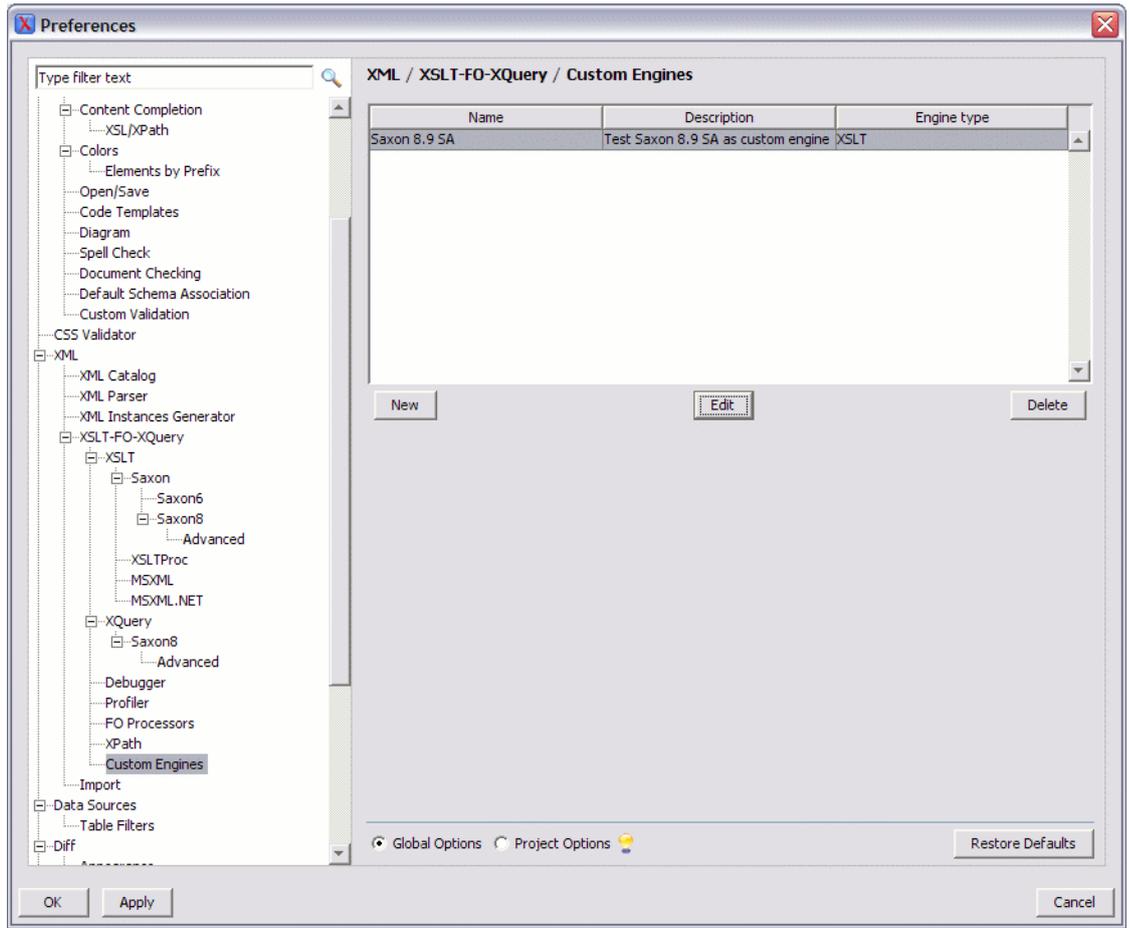


The processing messages and results will be shown in the Message Pane of the editor.

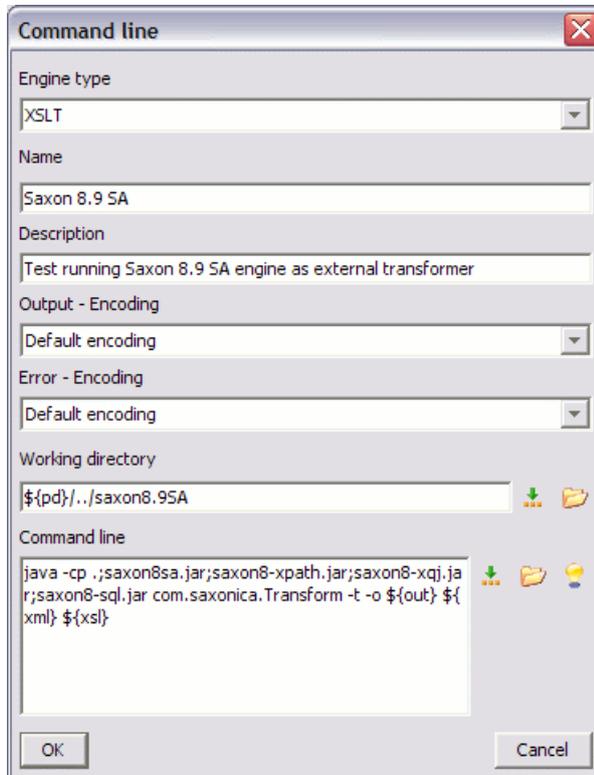


## Procedure 2. Configuring an External XSLT Engine as Custom XSLT Engine

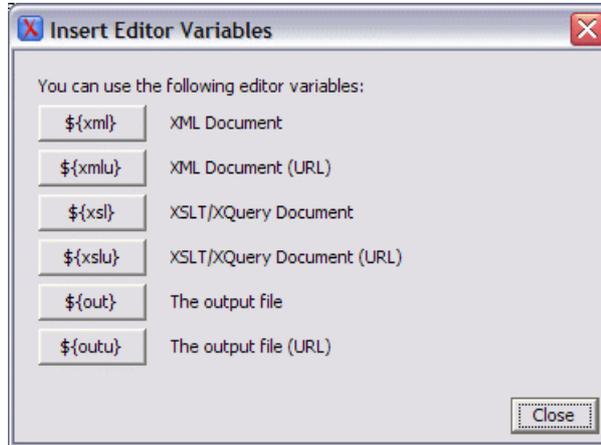
1. Start Oxygen if it is not started.
2. Go to the Custom Engines preferences panel from menu Options → Preferences → XML → XSLT / FO / XQuery → XSLT → Custom Engines



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



- 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.



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

