

How to develop XSLT stylesheets for Saxon-CE

Radu PISOI

radu_pisoi@oxygenxml.com

A few words about Saxon-CE processor

- Saxon-CE is a Saxonica implementation of XSLT 2.0 processor designed to work within a web browser
- Saxon-CE can handle the user input and interactions directly in the XSLT code
- Saxon-CE is a JavaScript library obtained by converting the Java processor into highly-optimized JavaScript using GWT library

'Book list' sample

<http://saxonica.com/ce/user-doc/1.1/index.html#!samples/booklist>

Saxonica ▾ Saxon ▾ Sample Applications ▾ Book List

Saxon 🔍 ⤴

- About Saxon-CE
- Change Log for Saxon-CE
- Getting started
- Developing Applications
- JavaScript API
- Conformance
- ▾ Sample Applications
 - **Book List**
 - Knight's Tour
 - JavaScript Interoperability
- ▾ Demonstrations
- System Programming Interfaces

Book List

To try out the book list application, visit [here](#).

Wait for the list of books to appear (it may take a while if you are on a slow connection, as Saxon is being downloaded into your browser cache from the server). When it appears, try clicking on column headings to sort the tables by a chosen column, or clicking the check boxes to filter which books are displayed.

To see how it works, view the source of the HTML page ("view source" in your browser), and examine [the stylesheet](#).

Note how the HTML page contains two `script` elements, one to identify the Saxon Javascript library, the other to identify the stylesheet and source document.

Note how the XSLT code addresses an area of the HTML page to be updated using `xs1:result-document`.

Note the event handlers in the stylesheet - the template rules with `mode="ixsl:onclick"` or `mode="ixsl:onmouseover"`. Note how these template rules update attributes of the HTML page using `ixsl:set-attribute`. Also see how user-defined attributes such as `data-order` and `data-type` are used to retain state information in the HTML tree, so that the sorting process can decide whether to sort ascending or descending, and whether to sort alphabetically or numerically.

Note how the `onmouseover` event handler reads the mouse coordinates by a call on the `ixsl:event()` extension function which returns the current mouse event, and then uses these coordinates to position the tooltip.

XSLT with Saxon-CE support add-on

<http://www.oxygenxml.com/InstData/Addons/Saxon-CE/updateSite.xml>

- an XSLT module
- an implementation of the Java extension SchemaManagerFilter
- a Schematron module
- two default validation scenarios

XSLT with Saxon-CE support add-on

<http://www.oxygenxml.com/InstData/Addons/Saxon-CE/updateSite.xml>

- an *XSLT module* containing declarations of the extension functions specific to Saxon-CE processor

Unknown Saxon-CE extension function

```
<xsl:variable  
  name="styleLeftVal"  
  select="concat(ixsl:get(ixsl:event(), 'clientX') + 30, 'px')"/>
```

XSLT module snippet

```
<xsl:function name="ixsl:event" override="no"/>  
  
<xsl:function name="ixsl:get" override="no">  
  <xsl:param name="object"/>  
  <xsl:param name="property"/>  
</xsl:function>
```

XSLT with Saxon-CE support add-on

<http://www.oxygenxml.com/InstData/Addons/Saxon-CE/updateSite.xml>

- an implementation of the Java extension *SchemaManagerFilter* to improve content completion proposals

```

/**
 * Interface for objects used to filter the editor content completion schema manager proposals.
 * This should be implemented if the list of content completion proposals must be filtered based on some
 criteria or
 * some new entries need to be added.
 */
public interface SchemaManagerFilter extends Extension {
    List<CIElement> filterElements(List<CIElement> elements, WhatElementsCanGoHereContext context);
    List<CIValue> filterElementValues(List<CIValue> elementValues, Context context);
    List<CIAttribute> filterAttributes(List<CIAttribute> attributes, WhatAttributesCanGoHereContext context);
    List<CIValue> filterAttributeValues(List<CIValue> attributeValues, WhatPossibleValuesHasAttributeContext
 context);
}

```

XSLT with Saxon-CE support add-on

<http://www.oxygenxml.com/InstData/Addons/Saxon-CE/updateSite.xml>

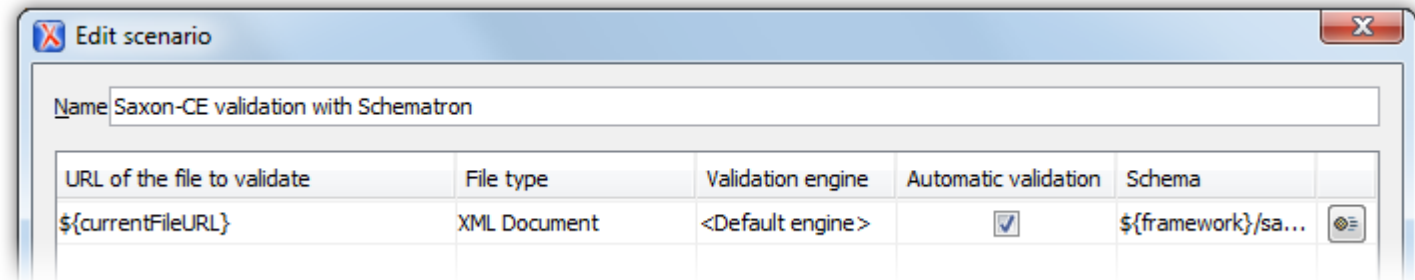
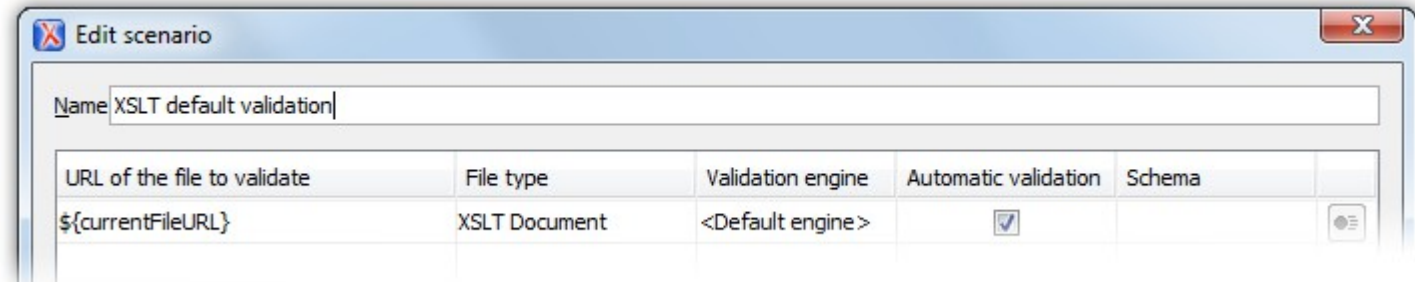
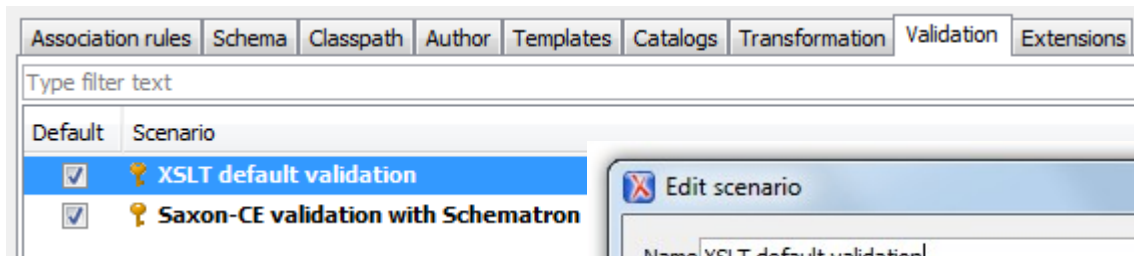
- a *Schematron module* to verify the name of Saxon-CE extension elements and attributes

```
<pattern id="ixsl_element_names">
  <p>Test the names of Saxon-CE extension elements.</p>
  <rule context="ixsl:*">
    <let name="elemName" value="local-name()"/>
    <assert test="$elemName = 'set-attribute' or $elemName = 'set-property' or
      $elemName = 'schedule-action' or $elemName = 'remove-attribute'" role="warning">
      Unknown Saxon CE extension element: '<value-of select="$elemName"/>'.
      Expected one of: 'set-attribute', 'set-property', 'remove-attribute' and 'schedule-action'.
    </assert>
  </rule>
</pattern>
```

XSLT with Saxon-CE support add-on

<http://www.oxygenxml.com/InstData/Addons/Saxon-CE/updateSite.xml>

- two default validation scenarios



Conclusions

- Installing *XSLT with Saxon-CE* add-on helps you to develop Saxon-CE stylesheets
- A document type can be extended to improve content completion or document validation
- A document type can be distributed as an add-on

Thank you!

Location for *XSLT with Saxon-CE support* add-on:

<http://www.oxygenxml.com/InstData/Addons/Saxon-CE/updateSite.xml>

Questions?

<oXygen/> XML Editor
<http://www.oxygenxml.com>
radu_pisoi@oxygenxml.com