

What's New in Oxygen XML Editor 20

- Development -

Oxygen Team

support@oxygenxml.com

@oxygenxml

© 2018 Syncro Soft SRL. All rights reserved.



XSLT

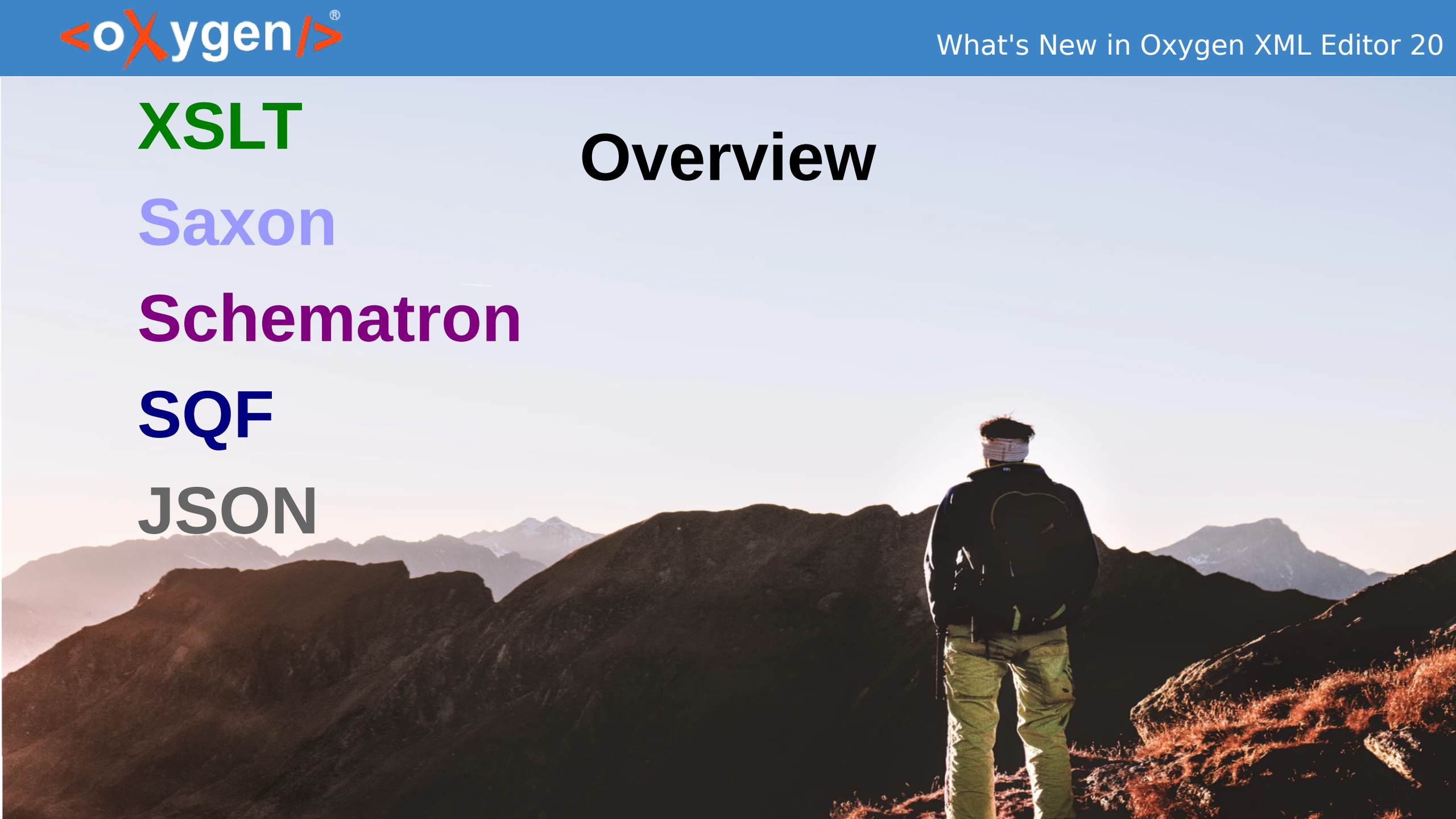
Saxon

Schematron

SQF

JSON

Overview

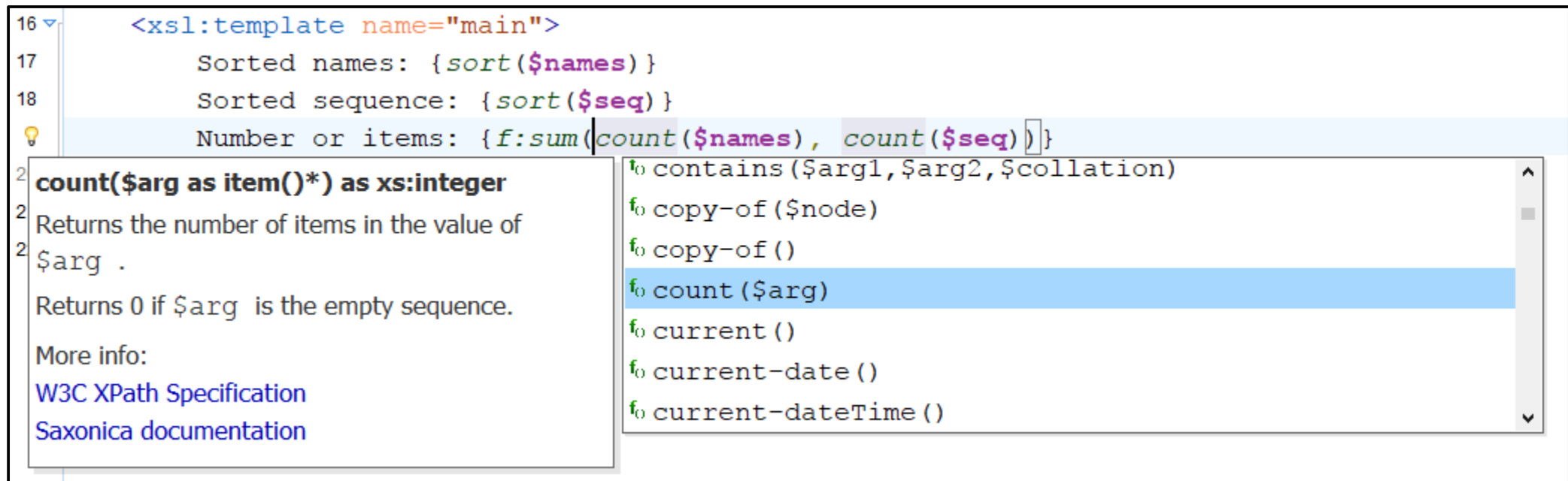


XSLT Improvements

- XPath content completion in TVTs
- Search and refactoring in TVTs
- Documentation for functions and template presented in the content completion window
- ...

XPath Content Completion in TVTs

- Full content completion support was added for text value templates



The screenshot shows the Oxygen XML Editor interface. The main editor displays an XSLT template with the following code:

```
16 <xsl:template name="main">
17     Sorted names: {sort($names)}
18     Sorted sequence: {sort($seq)}
19     Number or items: {f:sum(count($names), count($seq))}
```






A light blue tooltip is visible over the `count($names)` function in the third line of code. The tooltip is divided into two panes:



- Left Pane:** Contains the signature `count($arg as item(*) as xs:integer)`, a description: "Returns the number of items in the value of \$arg . Returns 0 if \$arg is the empty sequence.", and links for "More info:" including "W3C XPath Specification" and "Saxonica documentation".
- Right Pane:** A scrollable list of XPath functions. The `count($arg)` function is currently selected and highlighted in blue. Other visible functions include `contains($arg1, $arg2, $collation)`, `copy-of($node)`, `copy-of()`, `current()`, `current-date()`, and `current-dateTime()`.

Search and Refactoring in TVTs

- Added support to search and rename variable, parameters, or functions in text value templates

```
<xsl:template name="main">
  Sorted names: {sort($names)}
  Sorted sequence: {sort($seq)}
  Number or items: {f:sum(count($names), count($seq))}
</xsl:template>
```

Function: 'sum' Scope: Project	
 Rename Component in...	
 Search Declarations	Ctrl+Shift+D
 Search References	
 Component Dependencies	Ctrl+Shift+F4
 Change scope...	

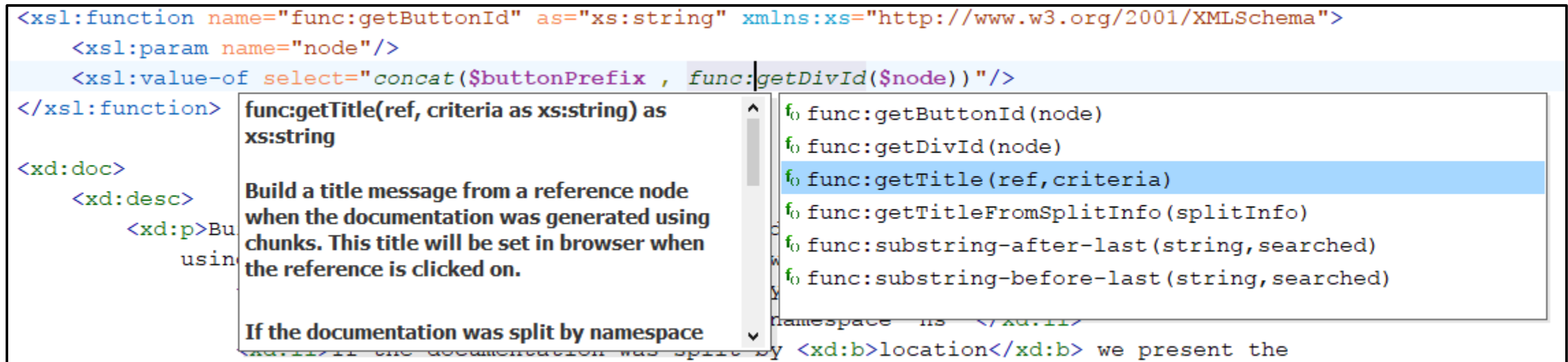
Function: 'sum' Scope: Current File	
 Rename Component	Alt+Shift+R
 Search Occurrences	Ctrl+Shift+U

Renames the component and updates all its references. Scope: Project

Functions and Templates Documentation

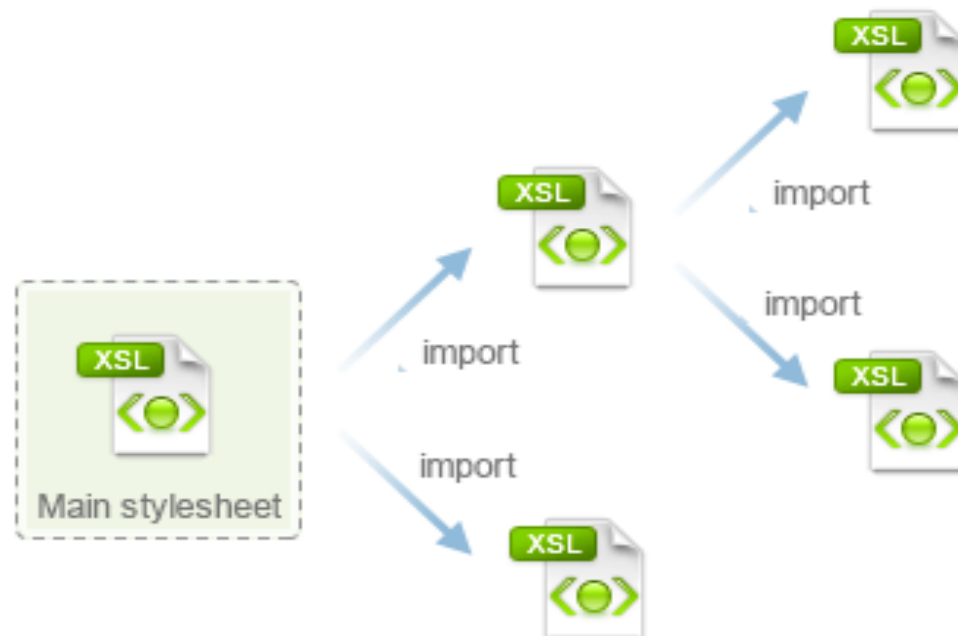
- Documentation for current function and template presented in the content completion window

```
<xsl:function name="func:getButtonId" as="xs:string" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xsl:param name="node"/>
  <xsl:value-of select="concat($buttonPrefix , func:getDivId($node))"/>
</xsl:function>
<xd:doc>
  <xd:desc>
    <xd:p>Build a title message from a reference node when the documentation was generated using chunks. This title will be set in browser when the reference is clicked on.
    </xd:p>
    <xd:p>If the documentation was split by namespace
  </xd:desc>
</xd:doc>
```



Override Parameters in XSLT Modules

- When editing XSLT that references modules, if you try to define a new parameter, the content completion list shows the names of the parameters defined in the modules



Convert xsl:attribute to Inline Attributes


- You can now convert xsl:attribute elements to inline attributes for elements outside the XSL namespace

```
<person>  
  <xsl:attribute name="id">john.doe</xsl:attribute>  
  <xsl:attribute name="email"><xsl:text>john.doe@example.com</xsl:text></xsl:attribute>  
  <xsl:attribute name="manager"><xsl:value-of select="person[@id='boss']/name"/></xsl:attribute>  
</person>
```



```
<person id="john.doe" email="john.doe@example.com" manager="{person[@id='boss']/name}"/>
```


Improved Refactoring Support

 **Improved extract template** - the action that extracts a fragment as a named template now adds type information to the variables or parameters used in the new template.

oxygenxml.com/doc/ug-editor/topics/xslt-refactoring-actions.html

Saxon Update

- We keep updating Saxon
 - Saxon **9.8.0.8** in **<oxygen/>** 20.0
 - Saxon **9.6, 9.7, 9.8** available as **<oxygen/>** add-ons



Saxon

- Saxon HE/PE/EE is builtin in <oxygen/>
- Used for:
 - XSLT validation, execution, and debugging
 - Compile XSL stylesheet for Saxon JS or Saxon HE/PE/EE
 - XQuery validation, execution, and debugging
 - Schematron and SQF execution
 - XML Schema validation

Schematron

- Added content completion and syntax highlighting for the ISO Schematron @documents attributes



SQF Update

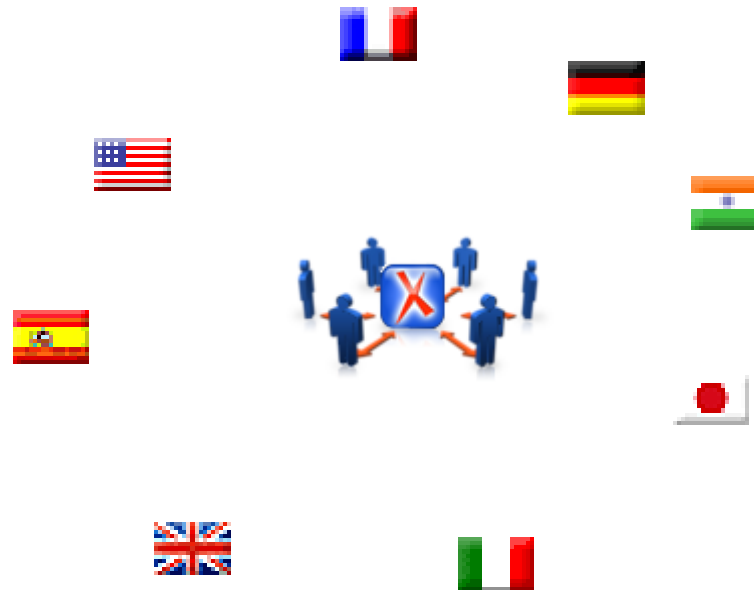
- Multilingual support for quick fixes
- Generate Multiple Similar Fixes
- Changed sqf:keep in sqf:copy-of
- Added @flags for sqf:stringReplace



<http://schematron-quickfix.github.io/sqf>

Multilingual Support for Quick Fixes

- Impose same rules but present them in different languages



Multilingual Support in SQF

- The name and description of a quick fix are defined by sqf:title and sqf:p elements
- The @ref attribute specifies IDs or keys for alternative localization

```
<sqf:fix id="addBone">  
  <sqf:description>  
    <sqf:title ref="fix_jp fix_de">Add a bone</sqf:title>  
    <sqf:p ref="fix_d_jp fix_d_de">Add a bone as child</sqf:p>  
  </sqf:description>  
  <sqf:add node-type="element" target="bone"/>  
</sqf:fix>
```

The diagram illustrates the localization of a quick fix for the 'dog' element in three languages. Each language is represented by a flag icon and a lightbulb icon. The quick fix is shown as a blue box with a plus sign and a text description. The description is localized into the respective language. The action is shown as a white box with a text description.

Language	Quick Fix Description	Action
English (UK)	A dog should have a bone. + Add a bone	Add a bone element as child
German	Ein Hund sollte ein Bein haben. + Fügen Sie einen Knochen hinzu	Fügen Sie ein Knochelement als untergeordnetes Element hinzu
Japanese	犬は骨を持っている必要があります。 + 骨を追加する	子要素としてボーン要素を追加する

Localization Using Diagnostics

- Implementation of quick fix localization using Schematron diagnostics
 - A diagnostic element is used for each language
 - The @ref attribute refers diagnostic IDs

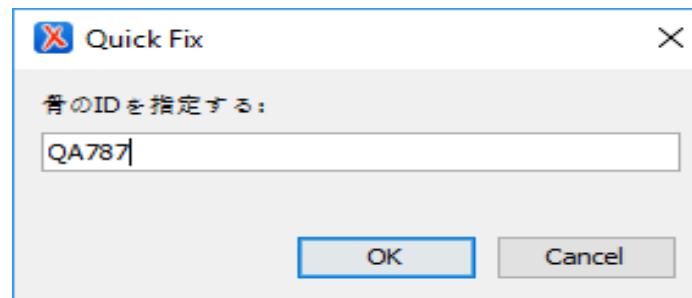
```
<sqf:fix id="addBone">
  <sqf:description>
    <sqf:title ref="fix_fr fix_de">Add a bone</sqf:title>
  </sqf:description>
  <sqf:add node-type="element" target="bone"/>
</sqf:fix>
....
<sch:diagnostics>
  <sch:diagnostic id="fix_fr" xml:lang="fr"> Ajouter un os </sch:diagnostic>
  <sch:diagnostic id="fix_de" xml:lang="de"> Fügen Sie einen Knochen hinzu</sch:diagnostic>
</sch:diagnostics>
```


SQF Localization

- Separate files for each language
- Include files from the main Schematron
- Support for included diagnostics
 - Validation
 - Content completion
 - Search and rename

User Entry Multilingual Messages

- Support to present the user entry messages in multiple languages



```
<sqf:user-entry name="boneId">  
  <sqf:description>  
    <sqf:title ref="ue_en ue_fr ue_de ue_ja">Specify bone ID:</sqf:title>  
  </sqf:description>  
</sqf:user-entry>  
<sqf:add node-type="element" target="bone" select="concat('ID: ', $boneId)"/>
```

Language Options

- Uses Schematron messages options:
 - Use the application language
 - Use the “xml:lang” attribute set on the Schematron root
 - Ignore the language and show all messages
 - Use a custom language

Generate Multiple Similar Fixes

- Added support to generate quick fixes dynamically using the @use-for-each attribute
- Generate a quick fix for each match of the @use-for-each attribute

```
<sqf:fix id="removeAnyItem" use-for-each="1 to count(li)">  
  ....  
</sqf:fix>
```

Use-for-each Attribute

- XPath content completion
- Present `$sqf:current` in content completion list

```
<sqf:fix id="removeAnyItem" use-for-each="1 to count(li)">
  <sqf:description>
    <sqf:title>Remove item #<sch:value-of select="$sqf:current"/></sqf:title>
  </sqf:description>
  <sqf:delete match="li[$sqf:current]"/>
</sqf:fix>
```

Changed sqf:keep in sqf:copy-of

- `sqf:copy-of` – copies the nodes selected by the `select` attribute, similar with `xsl:copy-of`

```
<report test="//doc:footnote" sqf:fix="parentBrackets">Footnote in footnote is forbidden.</report>
<sqf:fix id="parentBrackets">
  <sqf:description><sqf:title>Resolve as text in brackets.</sqf:title></sqf:description>
  <sqf:replace >
    (<sqf:copy-of select="//doc:para/node()" />)
  </sqf:replace>
</sqf:fix>
```

Added @flags for sqf:stringReplace

- `sqf:stringReplace` operation specifies also the flags for the regular expression
- Similar with the flags of the `xsl:analyze-string` instruction

```
<sqf:stringReplace regex="dialog(?:\s+box)" flags="i" select="dialog box"/>
```

Improvements for `sqf:user-entry`

- Content completion
- Search
- Refactoring

```
<sqf:fix id="editTitle">
  <sqf:description>
    <sqf:title>Edit the jurnal title</sqf:title>
  </sqf:description>
  <sqf:user-entry name="newTitle" default="@title">
    <sqf:description><sqf:title>Edit the title:</sqf:title></sqf:description>
  </sqf:user-entry>
  <sqf:replace match="@title" target="title" node-type="keep" select="$newTitle"/>
</sqf:fix>
```


JSON

- Improved the syntax highlighting for JSON scripts embedded in XHTML documents
- Added JSON-specific syntax highlights for DITA codeblock elements
- JSON arrays can now be folded using the collapse/expand arrow
- When using the action to convert XML to JSON, the generated JSON file is automatically formatted and indented

THANK YOU!

Any questions?

<oXygen/> XML Editor

<http://www.oxygenxml.com>

octavian_nadolu@oxygenxml.com

@OctavianNadolu