xmlprague

Schematron & SQF Improvements

Octavian Nadolu
octavian_nadolu@oxygenxml.com
@OctavianNadolu

syncro soft

<oXygen/> XML Editor

Overview

- Schematron 2016 updates

- SQF Update*

(*) Available in <oXygen/> XML Editor version 20.0

Schematron 2016 Update

- Added support for Schematron 2016 standard
  - Updated the schema and stylesheets
- The new enhancements are:
  - Properties - add arbitrary sch:property elements to the SVRL succeeded-report or failed-assert
  - sch:pattern/@documents – run the pattern over a subordinate documents not the main one
  - sch:let – the @value is optional, can specify the value in the content

Support for Properties

- Content Completion
- Search
- Refactoring

```
<sch:report test="true()" properties="australianDollar precision3a">
    The element assetValue should be a non-negative decimal number
</sch:report>
…
<sch:properties>
   <sch:property id="australianDollar" role="currency">AUD</sch:property>
   <sch:property id="precision3a">
      <orecision>3</orecision>
   <sch:property>
</sch:properties>
```

sch:pattern/@documents

- Support to execute the pattern over the subordinate documents

```
<sch:let name="URIs" value="('section1.xml', 'section2.xml', 'section3.xml')"/>
<sch:pattern documents="$URIs">
   <sch:rule context="doc:sect1/doc:title">
      <sch:assert test="@id"> The section title should have an ID</sch:assert>
   </sch:rule>
</sch:pattern>
```

sch:let

- The sch:let/@value is optional, you can also specify the value in the content

```
<sch:let name="idValue">
   <i>
      <sch:value-of select="concat('id_', $currentID)"/>
   </i>
</sch:let>
```

SQF Update*

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

(*) Available in <oXygen/> XML Editor version 20.0

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

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

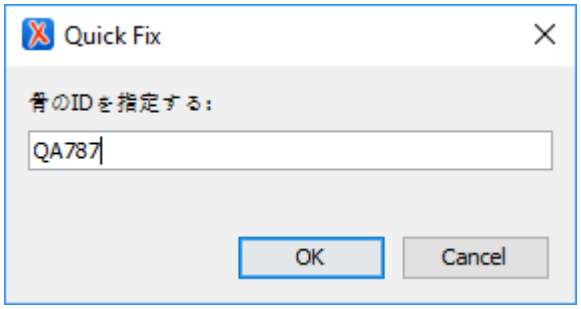
```
<sqf:fix id="addBone">
   <sqf:description>
      <sqf:title ref="fix_en fix_de">Add a bone</sqf:title>
   </sqf:description>
   <sqf:add node-type="element" target="bone"/>
</sqf:fix>
....
<sch:diagnostics>
   <sch:diagnostic id="fix_en" xml:lang="en">Add a bone</sch:diagnostic>
   <sch:diagnostic 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

Localization Using Diagnostics

- Pro
  - It is conform with the Schematron standard
  - Easier to translate because both Schematron messages and quick fix messages are kept together
  - Same implementation
- Cons
  - You cannot have IDs with the same name
  - SQF will depend on the Schematron language and cannot be encapsulated separately
  - A diagnostic does not represent a quick fix title or description

Generate Quick Fixes Dynamically

- 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>
```

Thank you!

Questions?

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