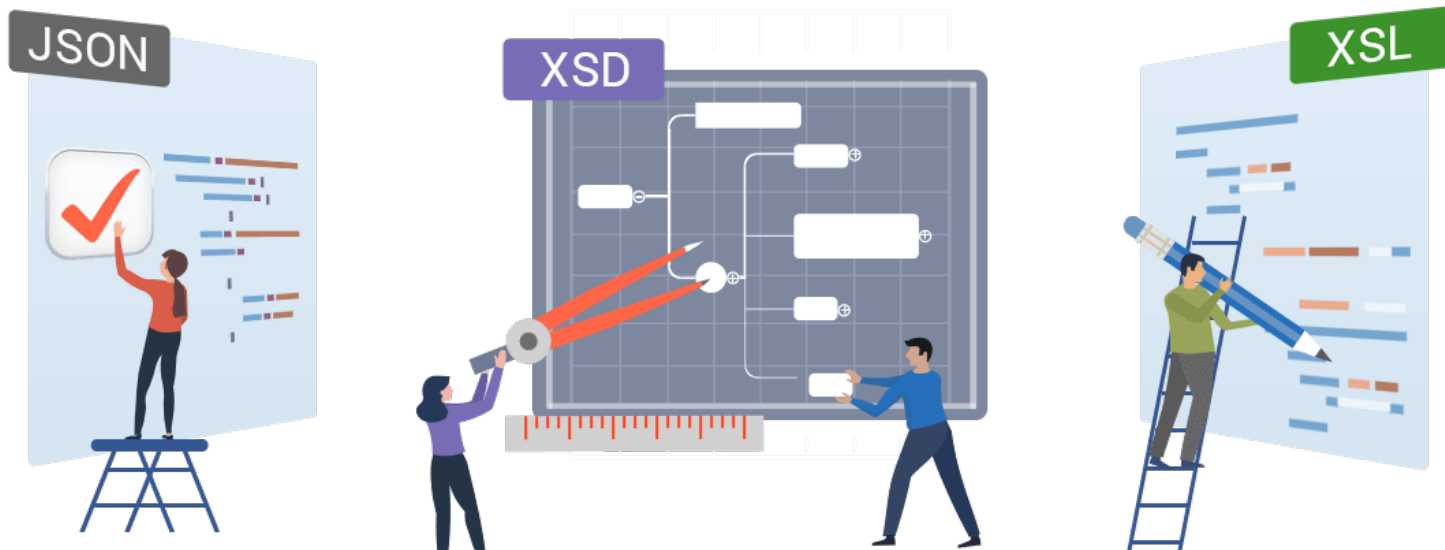


Development

- Tools for Designing XML and JSON Schemas
- XML and JSON Validation
- All XML and JSON Standards Support
- XSLT & XQuery Editing, Transformation and Debugging



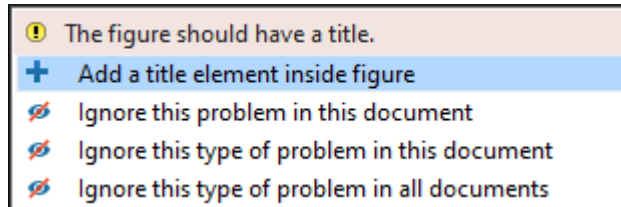
New Features

- Ignore Validation Problems
- JSON Schema Documentation
- YAML Editor
- Improvements on XSLT and XQuery
- New Command-line Scripts for Comparing Files or Directories



Ignore Validation Problems

- Validation problems that have an ID can now be **ignored**
- You can **activate/deactivate** this feature
- Quick fix actions to **ignore/unignore** a problem
- Ignore **Schematron** validation problems
- **Show/hide** the ignored problems.



JSON Schema Documentation

- Specify the **included components** in JSON Schema documentation
- **Split output into multiple files** for the JSON Schema documentation generator
- Source code for JSON Schema documentation presented with **syntax highlights**
- JSON Schema documentation shows **schema location**

The screenshot shows the Oxygen JSON Schema documentation interface. On the left is a 'Table of Contents' with three sections: 'Main Schema Schema', 'Properties personnel', and 'Definitions' (listing personType, idType, nameType, nameValueType, and linkType). Below this is a footer: 'JSON Schema documentation generated by <oxygen/>'. The main area is titled 'Main Schema personal-schema.json' and contains a table of metadata (Document version, Document location, Annotations, Title, Properties, Type, Additional Properties) and a 'Source' section with a JSON code block. A 'Showing:' control on the right has checkboxes for Annotations, Properties, Constraints, Enumerations, Combined, and Source, all of which are checked. Below the main area is a 'Property personnel' section with its own 'Annotations' table.

Name	Occurrence
personnel	optional

```
{
  "type": "object",
  "title": "Personal schema.",
  "$schema": "http://json-schema.org/draft-07/schema#",
  "properties": {
    "personnel": {
      "type": "object",
      "additionalProperties": false,
      "title": "The 'personnel' property",
      "description": "Defines the personnel.",
      "properties": {}
    }
  }
}
```

Title	Description
The 'personnel' property	Defines the personnel as a collection of person entries.

XSLT and XQuery

- Generate XSLT Profiling Reports Using a New Saxon Option
- Generate XQuery Profiling Reports Using a New Saxon Option
- Easily Create an XSLT 3.0 Package (xsl:package)

module	line	instruction	count
iso-schematron-quick-fixes.xsl	392	template process-user-entry-and-activity-elements	11
iso-schematron-message.xsl	127	template generate-message	40
iso-schematron-quick-fixes.xsl	940	template generate-add-node-operation	8
iso-schematron-message.xsl	214	template getMessage	80
iso-schematron-quick-fixes.xsl	353	template process-quick-fix	11
iso_schematron_skeleton.xsl	716	template generate-default-rules	1
iso-schematron-quick-fixes.xsl	159	template process-message-end	8
iso-schematron-quick-fixes.xsl	251	template process-quick-fixes	6
iso_schematron_skeleton.xsl	1835	template text()	42
iso-schematron-quick-fixes.xsl	91	template process-rule	6
iso-schematron-quick-fixes.xsl	592	template generate-system-id-attr	58
iso-schematron-quick-fixes.xsl	118	template end-process-rule	6
iso-schematron-quick-fixes.xsl	775	template process-replace-operation	5

YAML Editor

- Syntax highlighting
- Automatic validation
- Batch Validation
- Automatic indenting
- Format and Indent (Pretty Print) YAML files
- Convert between YAML and JSON



```
personal.yaml* X
1 ---
2 $schema: personal-schema.yaml
3 personnel:
4   person:
5     - id: Big.Boss
6       name:
7         family: Boss
8         given: Big
9         email: chief@oxygenxml.com
10        email: big_boss@oxygenxml.com
11       link:
12         subordinates:
13           - one.worker
14           - two.worker
15           - three.worker
16           - four.worker
17           - five.worker
18     - id: one.worker
19       name:
20         family: Worker
21         given: One
22         email: one@oxygenxml.com
```

Validation:
❗ Duplicate key email: "email: big_boss"
Press F2 for focus

❗ Duplicate key email: "email: big_boss"

Compare Directory Script

- New Command-Line Script for **Comparing Directories**
 - compare two directories and get the comparison results in various formats (yam1, json, or xml)
 - numerous options as arguments and filter the results according to your needs

```
differences:
diffsNo: 7
onlyFoundInDir1: 2
onlyFoundInDir2: 2
modified: 3
files:
- M : concepts\winterFlowers.dita
- 01: images\Narcissus.jpg
- 02: images\Rose.jpg
- M : topics\flowers\gerbera.dita
- 01: topics\flowers\narcissus.dita
- 02: topics\flowers\roses.dita
- M : flowers.ditamap
```

Example - compares the directories and redirects the console output to a JSON file named "results.json", with "raw" mode formatting:

```
sh scripts/compareDirs.sh dir1 dir2 -out json/raw > results.json
```


Compare Files Script

- New Command-Line Script for **Comparing Files**
 - compare two files and get the comparison results in various formats (yaml, json, or xml)
 - numerous options (ignoreWS, ignorePI, ignoreComments, ignoreAttrOrder ...)

```
differences:
  diffsNo: 31
  diffs:
    - firstFile:
      startOffset: 153
      endOffset: 159
      secondFile:
        startOffset: 153
        endOffset: 162
  ...
```

Example - compares the files (ignoring the namespaces and prefixes) and redirects the console output to the results.xml file (XML-formatted)

```
sh scripts/compareFiles file1 file2 -ins -inp -ind -out xml > results.xml
```

Future Plans

- JSON Schema Diagram
- YAML Validation with JSON/YAML Schema
- Improve XSLT 3.0 Support
- Schematron and SQF Improvements
- Diff Scripting 3-way and Merge
- and more





Questions?

Octavian Nadolu
Software Architect at Syncro Soft

octavian.nadolu@oxygenxml.com

Twitter: [@OctavianNadolu](https://twitter.com/OctavianNadolu)

LinkedIn: [octaviannadolu](https://www.linkedin.com/in/octaviannadolu)