

PI871.3 WORKING GROUP

MAY 2, 2024

24/2



AGENDA

- Investigate tools that support RFC 8259
- Continue discussion about incompatibilities/mapping
 - XML->JSON

RFC 8259

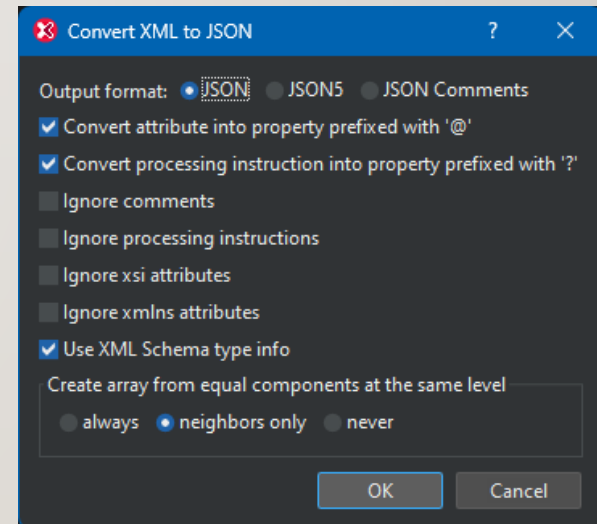
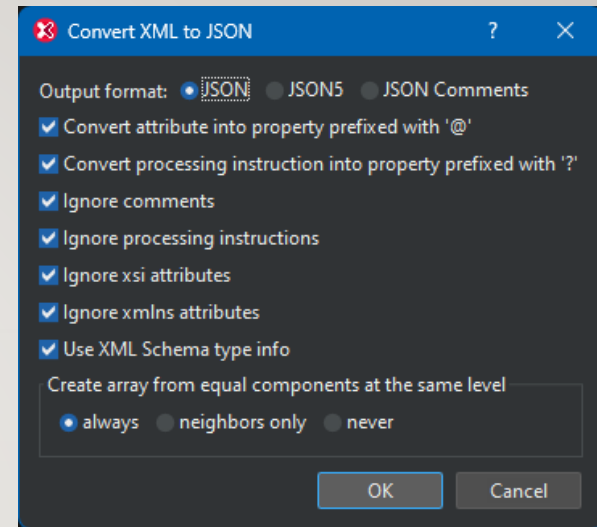
- Unifying JSON standard
 - JSON defines a small set of formatting rules for the portable representation of structured data. This document removes inconsistencies with other specifications of JSON, repairs specification errors, and offers experience-based interoperability guidance.
- December 2017
- Tools
 - C# (System.Text.Json) (~~partially compliant – Newtonsoft.Json~~)
 - Go
 - Python
 - XMLSpy

TOOLS

- XMLSpy
- C# - System.Text.Json namespace
- Python

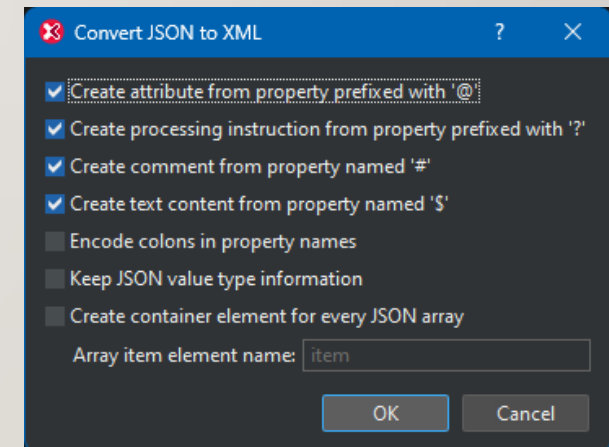
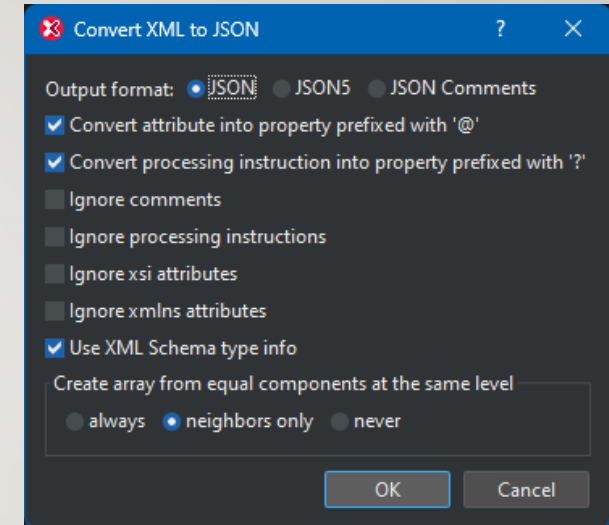
XMLSPY

- Built-in functionality to convert XML files to JSON
 - Tested with ~205 files
 - Examples from IEEE standards, test cases etc
 - Tried 2 different configurations
- Files converted to JSON without issues



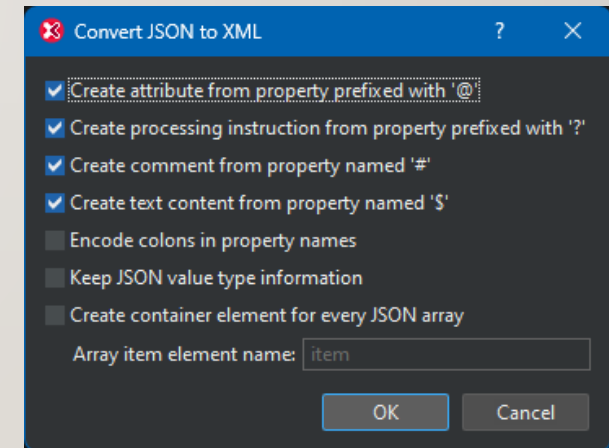
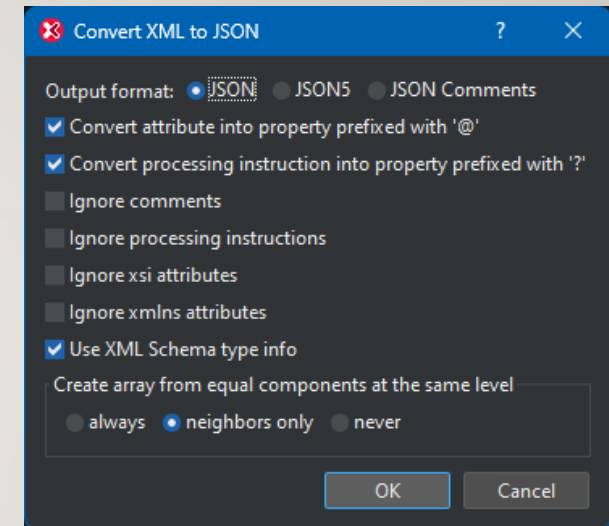
XMLSPY – XML<->JSON

- To ensure close to valid/complete XML, we had to encode the following information in the JSON
 - Differentiate between attributes and elements
 - Option 1
 - Also helped keeping namespace from the root element



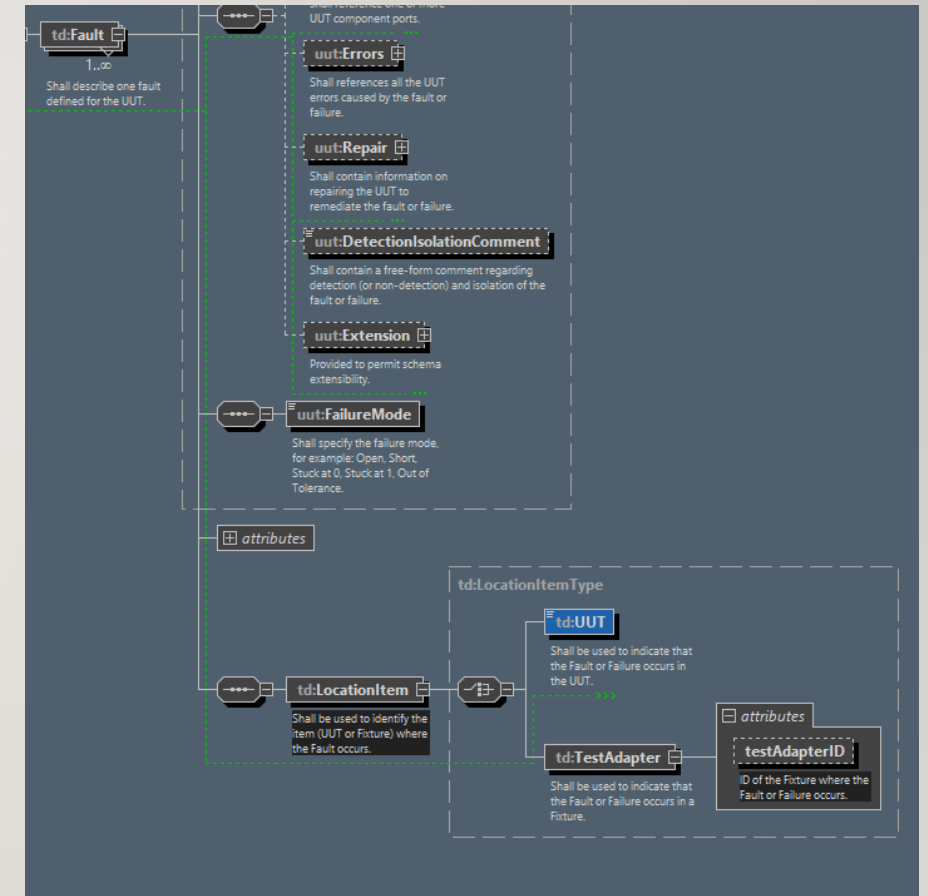
XMLSPY – CONVERSION ISSUES

- `<xs:restriction base="PlaneAngle" minInclusive="0" maxInclusive="2*pi rad"/>`
 - `xmlns:std="urn:IEEE-1641:2010:STDBSC"`
`xmlns:std1="urn:IEEE-1641:2010:STDBSC"`
 - `"@xmlns": "urn:IEEE-1641:2010:STDBSC",`
`"@xmlns": "urn:IEEE-1641:2010:STDBSC",`
`"@xmlns": "urn:IEEE-1641:2010:STDBSC"`



C# - SYSTEM.TEXT.JSON NAMESPACE

- Read the files generated using second set of options from C#
- Ensured that I was able to open the generated JSON files without issues in C#
- Additional items in JSON
 - {"@ID": "cIfI", "@componentID": "cI", "@detectability": "DetectableInsertable", "uut:FailureMode": "Unnamed", "td:LocationItem": {"td:UUT": {}},
 - Default values



PYTHON

- Read the files generated using second set of options from Python
- Test suite - [nst/JSONTestSuite: A comprehensive test suite for RFC 8259 compliant JSON parsers \(github.com\)](#)
 - [Parsing JSON is a Minefield \(seriot.ch\)](#)
- Initial results look promising
- In work

CONCLUSION

- Unless the rest of the testing and any additional test cases prove otherwise the standard should use RFC 8259
- Thoughts?

CONVERSION ISSUES (XML TO JSON)

- Known issues
 - Large numbers
 - Numbers should be within IEEE 754 binary64 range
 - RFC 7493 recommends converting them to strings?
 - `<[CDATA[]]>`
 - encode binary data as base64
 - Will require customization to the JSON conversion library
 - Encoding
 - Always generate UTF-8?
 - Comments
 - Will be removed
 - DateTime
 - Date and times should be strings in ISO 8601 format
 - Element vs attributes
 - Attributes are prefixed with an @ at the start of the object
 - Namespace
 - Will be added due to rule for attributes

CONVERSION ISSUES (XML TO JSON)

- XML strings including encoded strings
- XML declaration and processing instructions
 - prefixed with ?.
 - `<?xml version='1.0' standalone='no'?>` -> `"?xml": {"@version": "1.0", "@standalone": "no" }`
- Multiple nodes with the same name at the same level
 - Always force grouping into a JSON array (See Role element)
 - `<role>AdminI</role>` -> `"role": "AdminI"`
 - `<role>AdminI</role>` -> `"role": ["AdminI"]`
 - Order of elements
 - Recommendation - force order of nodes
- Empty elements
 - Force empty elements to have null
 - `<element />` -> `{ "element": null }`

CONVERSION ISSUES (XML TO JSON)

- Issues with element names and JSON key name?
 - Potentially invalid characters for JSON
 - Quotation mark (!) \!
 - Backslash (\) \\\
 - Slash (/) \/
 - Backspace \b
 - Form feed \f
 - New line \n
 - Carriage return \r
 - Horizontal tab \t
 - Some control characters like \u2019
- Mixed content
 - `<p>This is an example!</p>`
 - `{ "p": { "#text": "This ", "b": "is", "#text1": " an ", "b#1": "example", "#text2": "!" } }`

NEXT STEPS AND DATES

- **Continue analysis of generated and transformed files**
- **Continue discussion about incompatibilities/mapping**
 - **XML->JSON**
 - JSON->XML
- Define Conversion Rules between XML and JSON formats (Future)
- Next electronic meeting dates
 - June 18, 2024 – 10 AM CST