Apache Xalan-J’s, XSLT 3.0 specification implementation status

Document modified : 2024-06-25

Document author : Apache Xalan-J team

(1) XSLT 3.0 & XPath 3.1

The XSLT 3.0 specification defines various conformance features as follows, and the level to which Xalan-J implements them.

a)	Basic XSLT processor	Supported XSLT 3.0 instructions and XPath language features, whose implementations are available are described in subsequent sections of this document, below.
b)	Schema-aware XSLT processor	Not supported There is support for, minor use cases of XSLT instruction xsl:import-schema. There are few working examples for this, available within tests folder ‘xsl_import_schema’.
c)	Serialization feature	Supported
d)	Streaming feature	Not supported
e)	Dynamic evaluation feature	Not supported
f)	XPath 3.1 feature, for arrays	Supported
g)	Higher-order functions feature	Supported

Following are details of Xalan-J, XSL 3.0 family of language features, whose working implementation is available on Xalan-J XSL 3.0 dev repos branch ‘xalan-j_xslt3.0’.

(1.1) XSLT 3.0 features

XSLT 3.0 language specification : https://www.w3.org/TR/xslt-30/

1) xsl:for-each-group instruction

2) xsl:analyze-string instruction

3) xsl:iterate instruction

4) xsl:for-each instruction implementation is modified, to handle few XSLT 3.0 requirements.

5) xsl:function instruction

6) xsl:sequence instruction

7) xsl:attribute element can now have "select" attribute as well in addition to mutually exclusive child content as well, as specified by XSLT 3.0 spec.

8) xsl:import-schema instruction

   Currently, the XML Schema simple types imported via xsl:import-schema instruction within an XSLT stylesheet, can be used with “as” attribute of XSLT xsl:variable elements to enforce schema type constraints on xsl:variable data contents.

9) xsl:variable instruction evaluation to node set instead of result tree fragment (RTF). This is a XSLT spec change first introduced within XSLT 2.0 language, as compared to XSLT 1.0.

10) The sequence type expression "as" attribute on XSLT elements xsl:variable, xsl:template, xs:function, xsl:param, xsl:with-param.

.

11) Function implementations

    a) New function implementations : fn:current-grouping-key, fn:current-group, fn:regex-group

    b) Function implementation enhancements : fn:system-property

(1.2) XPath 3.1 expression language features

XPath 3.1 language specification : https://www.w3.org/TR/xpath-31/

1) Range "to" expression

2) Value comparison operators eq, ne, lt, le, gt, ge

3) Function item "inline function expression"

4) Dynamic function calls

5) "if" expression

6) "for" expression

7) Quantified expressions 'some', 'every'

8) "let" expression

9) Sequence constructor expression, using comma operator

For e.g, XPath expressions like (1, 2, 3) etc.

10) String concatenation operator "||"

11) Node comparison operators "is", "<<", ">>"

12) Simple map operator '!'

13) Instance Of expression

14) Implementation of various new XML Schema built-in data types for use within XSLT 3.0 stylesheets and XPath 3.1 expressions. Implementation of, XPath constructor function calls (for e.g, xs:string('hello'), xs:date('2005-10-07') etc) for these supported XML Schema data types.

Currently, following XML Schema built-in data types are supported (depicted with XML Schema data type and subtype hierarchy as specified by "W3C XML Schema" data types specification), for this work :

```
xs:anyType
    xs:anySimpleType
        xs:anyAtomicType
            xs:anyURI
            xs:boolean
            xs:date
            xs:dateTime
            xs:decimal
               xs:integer
               xs:long
               xs:int
            xs:double
            xs:duration
               xs:dayTimeDuration
               xs:yearMonthDuration
            xs:float
            xs:QName
            xs:string
               xs:normalizedString
               xs:token
                  xs:Name
                     xs:NCName
            xs:time
```

In addition to above mentioned XML Schema built-in data types, an XML Schema type xs:untyped specified by XPath 3.1 specification has also been implemented.

15) Collation support

Within the context of XSL languages, a collation is a method by which text information is compared and sorted.

As specified by XPath 3.1 F&O spec, implementations of following collations are available:

15.1) The Unicode Codepoint Collation

15.2) The Unicode Collation Algorithm

Support for following collation uri query parameters is available : 'fallback', 'lang', 'strength'

For the collation's query "lang" parameter, all languages as those supported by Java's 'java.util.Locale' class are available within Xalan-J's XSLT 3.0 implementation (ref, https://docs.oracle.com/javase/8/docs/api/java/util/Locale.html).

For the collation's query "strength" parameter, following values are supported : 'primary', 'secondary', 'tertiary', 'identical'.

15.3) The HTML ASCII Case-Insensitive Collation

16) Sequence type expressions

17) Map expressions

18) Array expressions

19) Cast expression

20) Castable expression

21) Named function reference

22) Arrow operator (=>)

(1.3) XPath 3.1 functions

XPath 3.1 F&O specification : https://www.w3.org/TR/xpath-functions-31/

Implementation of built-in functions namespace uri : http://www.w3.org/2005/xpath-functions

Implementation of built-in math functions namespace uri : http://www.w3.org/2005/xpath-functions/math

1) Functions on numeric values

fn:abs	
fn:round	(implementation of an optional second argument, that's used to specify 'precision')

2) Context functions

fn:current-dateTime
fn:current-date

fn:current-time
fn:implicit-timezone
fn:default-collation

3) Functions giving access to external information

fn:doc
fn:unparsed-text

4) Functions on strings

fn:string-join
fn:upper-case
fn:lower-case
fn:codepoints-to-string
fn:string-to-codepoints
fn:compare (with support for collation argument)
fn:codepoint-equal
fn:contains-token (with support for collation argument)

5) String functions that use regular expressions

fn:matches
fn:replace
fn:tokenize
fn:analyze-string

6) Functions that compare values in sequences

fn:distinct-values (with support for collation argument)
fn:index-of (with support for collation argument)
fn:deep-equal (with support for collation argument)

7) Mathematical trigonometric and exponential functions

math:pi
math:exp
math:exp10
math:log
math:log10
math:pow
math:sqrt
math:sin
math:cos
math:tan
math:asin
math:acos
math:atan
math:atan2

8) Component extraction functions on durations

fn:years-from-duration
fn:months-from-duration
fn:days-from-duration
fn:hours-from-duration
fn:minutes-from-duration
fn:seconds-from-duration

9) Constructing xs:dateTime value

fn:dateTime

10) Component extraction functions on dates and times

fn:year-from-dateTime
fn:month-from-dateTime
fn:day-from-dateTime
fn:hours-from-dateTime
fn:minutes-from-dateTime
fn:seconds-from-dateTime
fn:timezone-from-dateTime
fn:year-from-date
fn:month-from-date
fn:day-from-date
fn:timezone-from-date
fn:hours-from-time
fn:minutes-from-time
fn:seconds-from-time
fn:timezone-from-time

11) Built-in higher-order functions

fn:for-each
fn:filter
fn:fold-left
fn:fold-right
fn:for-each-pair
fn:sort (with support for collation argument)
fn:apply

12) Functions on sequences

12.1 General functions on sequences
fn:empty
fn:exists
fn:head
fn:tail

fn:insert-before
fn:remove
fn:reverse
fn:subsequence
fn:unordered

12.2 Aggregate functions
fn:avg
fn:max
fn:min

13) Parsing and serializing

fn:parse-xml
fn:parse-xml-fragment

14) Accessors

fn:node-name
fn:data
fn:base-uri
fn:document-uri

15) Functions related to QNames

fn:resolve-QName
fn:QName

16) Functions related to maps

map:merge
map:size
map:keys
map:contains
map:get
map:put
map:entry
map:remove
map:for-each

17) Functions related to arrays

array:size
array:get
array:put
array:append
array:subarray
array:remove
array:insert-before

array:head
array:tail
array:reverse
array:join
array:for-each
array:filter
array:fold-left
array:fold-right
array:for-each-pair
array:sort (with support for collation argument)

18) Functions on JSON data

fn:parse-json
fn:json-doc
fn:json-to-xml
fn:xml-to-json

Other than the above mentioned newly implemented XPath 3.1 functions, all the functions that are already available within XPath 1.0 (all of them are common with XPath 3.1 function library as well) are available within Xalan-J's XPath 3.1 implementation as well.

Please refer to the web link https://www.w3.org/TR/1999/REC-xpath-19991116/ (section "4 Core Function Library"), for XPath 1.0 functions that shall work with Xalan-J's XSLT 3.0 implementation as well.

(2) Xalan-J XSLT 3.0 & XPath 3.1 test suite

For the Xalan-J XSLT 3.0 & XPath 3.1 implementations described within this document, a working test suite is available at the location : https://github.com/apache/xalan-java/tree/xalan-j_xslt3.0/tests, and the results of these Xalan-J XSL tests are available at the location : https://xalan.apache.org/xalan-j/xsl3/tests/AllXsl3Tests_20240625-173731.xml.

Apache Xalan-J site
https://xalan.apache.org/xalan-j/