

**\*\*\* XalanJ, XSL 3.0 family of languages (XSLT 3.0, XPath 3.1, XPath 3.1 F&O [functions and operators]) implementation status, as on 2024-02-05 \*\*\***

**Author : Apache XalanJ team**

**(I) XSLT 3.0 and XPath 3.1**

Following are details of "XalanJ, XSL 3.0 family of languages" features, whose working implementation is available on XalanJ XSLT 3.0 development repository branch 'xalan-j\_xslt3.0'.

**(A) XSLT 3.0 features**

XSLT 3.0 language home page : <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 : `fn:current-grouping-key`, `fn:current-group`, `fn:regex-group`

**(B) XPath 3.1 expression language features**

XPath 3.1 language home page : <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:boolean
      xs:date
      xs:dateTime
      xs:decimal
      xs:integer
      xs:long
      xs:int
      xs:double
      xs:duration
        xs:dayTimeDuration
        xs:yearMonthDuration
```

- xs:float
- xs:string
  - xs:normalizedString
  - xs:token
- xs:time

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

#### 15) Collation support

As specified by XPath 3.1 F&O spec, following collation implementations are supported,

- a) The Unicode Codepoint Collation
- b) 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 XalanJ'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'.

- c) The HTML ASCII Case-Insensitive Collation

#### 16) Sequence type expressions

### (C) XPath 3.1 functions

XPath 3.1 F&O home page : <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) String functions that use regular expressions

- fn:matches
- fn:replace
- fn:tokenize

#### 2) Functions on numeric values

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

### 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) Context functions

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

### 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) 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) Basic 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)

The function implementations for these, yet doesn't support type declarations on parameters and return type.

## 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

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 XalanJ's XPath 3.1 implementation as well.

## (II) XalanJ XSLT 3.0 test suite

For the XalanJ XSLT 3.0 implementation described within this document, a working test suite is available at [https://github.com/apache/xalan-java/tree/xalan-j\\_xslt3.0/tests](https://github.com/apache/xalan-java/tree/xalan-j_xslt3.0/tests) (with an entry point Java test suite class [https://github.com/apache/xalan-java/blob/xalan-j\\_xslt3.0/tests/org/apache/xalan/xslt3/AllXsl3Tests.java](https://github.com/apache/xalan-java/blob/xalan-j_xslt3.0/tests/org/apache/xalan/xslt3/AllXsl3Tests.java)).

XalanJ home page : <https://xalan.apache.org/xalan-j/index.html>

XalanJ contact information : [https://xalan.apache.org/xalan-j/contact\\_us.html](https://xalan.apache.org/xalan-j/contact_us.html)