***** 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 2023-12-30 *****

Author : Apache XalanJ team

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,

(1) XSLT 3.0 features

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

1.1 xsl:for-each-group instruction

1.2 xsl:analyze-string instruction

1.3 xsl:iterate instruction

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

1.5 xsl:function instruction

1.6 xsl:sequence instruction

1.7 xsl:attribute instruction can now have "select" attribute as well, as specified by XSLT 3.0 spec.

1.8 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.

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

.

1.10 Function implementations : fn:current-grouping-key, fn:current-group, fn:regex-group

(2) XPath 3.1 features

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

2.1 Range "to" expression

2.1 Value comparison operators eq, ne, lt, le, gt, ge

2.3 Function item "inline function expression"

2.4 Dynamic function calls

2.5 "if" expression

2.6 "for" expression

2.7 Quantified expressions 'some', 'every'

2.8 "let" expression

2.9 Sequence constructor expression, using comma operator

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

2.10 String concatenation operator "||"

2.11 Node comparison operators "is", "<<", ">>"

2.12 Simple map operator '!'

2.13 'instance of' expression

2.14 Implementation of various new XML Schema data types for use within XSLT 3.0 stylesheets and XPath 3.1 expressions. Implementation of, XPath constructor function calls for supported XML Schema data types.

Currently, following XML Schema data types are supported, for this work : xs:boolean, xs:string, xs:normalizedString, xs:token, xs:date, xs:dateTime, xs:duration, xs:dayTimeDuration, xs:yearMonthDuration, xs:time, xs:decimal, xs:double, xs:float, xs:int, xs:integer, xs:long.

2.15 Collation support

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

1) The Unicode Codepoint Collation

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 supported by Java's 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'.

3) The HTML ASCII Case-Insensitive Collation

2.16 Sequence type expressions

(3) 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

3.1 String functions that use regular expressions

fn:matches
fn:replace
fn:tokenize

3.2 Functions on numeric values

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

3.3 Functions giving access to external information

fn:doc
fn:unparsed-text

3.4 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)

3.5 Context functions

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

3.6 Functions that compare values in sequences

fn:distinct-values
fn:index-of
fn:deep-equal

3.7 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

3.8 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

3.9 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.

3.10 Functions on sequences

fn:empty
fn:exists
fn:head
fn:tail
fn:insert-before
fn:remove
fn:reverse
fn:subsequence
fn:unordered
fn:avg
fn:max

fn:min

3.11 Parsing and serializing

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

3.12 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.

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

XalanJ contact information : https://xalan.apache.org/xalan-j/contact_us.html