Product Description for Saxon-PE (Professional Edition)

Version 9.7 released Nov 2015

This document lists the features provided by Saxon 9.7 Professional Edition (Saxon-PE).

This document does not form part of any contract unless expressly incorporated.

Language Support

1. XSLT (Transformation Processing)

1.1 XSLT 2.0 (Basic)

Provides a basic XSLT 2.0 processor as defined in section 21 of the XSLT 2.0 Recommendation: it is a conformance level that includes all features of the language other than those that involve schema processing.

For more details see: XSLT 2.0 conformance.

Relevant W3C Specification: XSLT 2.0 Recommendation (23 January 2007).

1.2 XSLT 3.0 (Basic)

Provides selected features from the XSLT 3.0 specification - including try/catch, evaluate, iterate, accumulators, maps, named modes, content value templates, and extended patterns, as well as XPath 3.0 and XPath 3.1 features such as higher-order functions; with the notable exceptions of schema-awareness, streaming, and packages.

For more details see: XSLT 3.0 conformance.

Relevant W3C Specification: XSLT 3.0 Candidate Recommendation (19 November 2015).

2. XPath

2.1 XPath 2.0 (Basic)

Provides all XPath 2.0 features other than schema-awareness.

For more details see: XPath 2.0 conformance.

Relevant W3C Specification: XPath 2.0 Recommendation (14 December 2010).

2.2 XPath 3.0 (Higher-Order Functions)

Provides all XPath 3.0 features, other than schema-awareness. This includes features that require higher-order functions: specifically, the ability to use functions as values, including dynamic function calls, inline functions, partial function application, and specific higher-order functions such as `fn:filter` and `fn:fold-left`. Saxon XPath 3.0

processing also allows selected XPath 3.1 features to be used: the main feature being the implementation of maps.

For more details see: XPath 3.0 conformance.

Relevant W3C Specification: XPath 3.0 Recommendation (08 April 2014).

2.3 XPath 3.1 (Higher-Order Functions)

Provides all XPath 3.1 features, other than schema-awareness. This includes an implementation of maps and arrays, and support for JSON; and features that require higher-order functions - specifically, the ability to use functions as values, including dynamic function calls, inline functions, partial function application, and specific higher-order functions.

For more details see: XPath 3.1 conformance.

Relevant W3C Specification: XPath 3.1 Candidate Recommendation (18 December 2014).

3. XQuery

3.1 XQuery 1.0 (Basic)

Provides the following features, as defined in section 5 of the XQuery 1.0 specification: Minimal Conformance; Full Axis Feature; Module Feature; and Serialization Feature.

For more details see: XQuery 1.0 conformance.

Relevant W3C Specification: XQuery 1.0 Recommendation (14 December 2010).

3.2 XQuery 3.0 (Higher-Order Functions)

Provides the following features, as defined in section 5 of the XQuery 3.0 specification: Minimal Conformance (including try/catch and "group-by"); Module Feature; Serialization Feature; and Higher-Order Function Feature.

For more details see: XQuery 3.0 conformance.

Relevant W3C Specification: XQuery 3.0 Recommendation (08 April 2014).

3.3 XQuery 3.1 (Higher-Order Functions)

Provides the following features, as defined in section 5 of the XQuery 3.1 specification: Minimal Conformance (including try/catch and "group-by"); Module Feature; Serialization Feature; and Higher-Order Function Feature.

For more details see: XQuery 3.1 conformance.

Relevant W3C Specification: XQuery 3.1 Candidate Recommendation (18 December 2014).

Performance Features

4. Binary XML

Saxon's PTree format is a serialized binary representation of Saxon's internal tree format. It occupies around the same amount of disk space as the original XML, but is faster to serialize and faster to reparse.

For more details see: The PTree file format.

5. Import stylesheet packages

Allows the importing of stylesheet packages in compiled form. Possible with all editions, provided the package only uses features available in that edition.

For more details see: Compiling a Stylesheet.

6. Optimizer (Basic)

The Basic optimizer provided with all Saxon editions provides a wide range of static and dynamic optimizations including full pipelining of list operations, lazy evaluation of variables, elimination of redundant sorting operations, etc.

7. Reading W3C schemas and DTDs

The W3C web server now routinely rejects requests for commonly-referenced files such as the DTD for XHTML, causing parsing failures. In response to this, Saxon now includes copies of these documents within the issued JAR file, and recognizes requests for these documents, satisfying the request using the local copy.

Extensibility

8. EXSLT extension functions

A selection of EXSLT extension functions are provided (in the modules Common, Dates and Times, Math, Random, and Sets), as listed in the documentation.

For more details see: EXSLT extensions.

9. EXPath extension functions

A selection of EXPath extension functions are provided (in the modules Archive, Binary, and File), as listed in the documentation.

For more details see: EXPath extensions.

10. Extensibility using custom classes

Ability to write extension functions (for use in XSLT, XQuery, or XPath) by implementing a Saxon-defined interface and registering the implementation with the Saxon Configuration.

For more details see: Extensibility.

11. Extensibility using reflexion

Ability to access existing Java or .NET methods dynamically and invoke them as extension functions by means of dynamic loading and reflexion.

For more details see: Extensibility.

12. Saxon extension functions (Basic)

Extension functions, as listed in the documentation, in the Saxon namespace. The Basic level excludes extension functions that depend on streaming or schema-awareness. Many Saxon extension functions have over the years become superseded by facilities in the W3C standards or in EXPath libraries, but some such as the ability to access the line number of an element remain available only through this library.

For more details see: Saxon extension functions.

13. SQL extension

XSLT extension instructions providing access to SQL databases. Available on the Java platform only (not .NET).

For more details see: Saxon SQL extension.

14. XSLT element extensibility

Ability to implement XSLT extension instructions by implementing a Saxon-defined interface and registering the implementation with the Saxon Configuration.

For more details see: Writing XSLT extension instructions.

Localization

15. Localization (Advanced)

All the interfaces for developers are in English, but there is some localization support in transformation to enable dates and numbers to be formatted and collations performed in other languages (so end-user output can be localized, but developer output cannot). The Advanced level includes number and date formatting for a variety of languages, as well as support for the Unicode Collation Algorithm using features from the ICU - International Components for Unicode. The Advanced level also includes APIs which allow support for additional languages to be developed.

For more details see: Unicode collation, Localizing numbers and dates.

Interfaces and APIs

16. JAXP API

Implementations of the standard JAXP interfaces for XSLT transformation, XPath evaluation, and XML Schema validation. Applies to the Java platform only.

For more details see: JAXP API conformance.

17. S9API API

Saxon's native interface for processing XSLT, XQuery, XPath, and XML Schema. Available in slightly different forms on the Java and .NET platforms.

18. Support for DOM

Ability to use a DOM (Document Object Model) for the input and output of transformations and queries. On the .NET platform this includes the System.XML DOM classes.

For more details see: Object models.

19. Support for JDOM, JDOM2, AXIOM, DOM4J, and XOM

Ability to use a JDOM, JDOM2, AXIOM, DOM4J, and XOM for the input or output of transformations and queries. Applies to the Java platform only. Note that the code for these interfaces is open source and can be compiled to work with Saxon-HE, but it does not come packaged with the Saxon-HE download.

For more details see: Object models.

20. XQJ API

Implementations of the standard XQJ interfaces for XQuery processing. Applies to the Java platform only. Note that the XQJ interfaces have been removed from the standard download of Saxon-HE because the Oracle specification license is not open source, but they are available on request.

For more details see: XQJ API conformance.