Product Description for Saxon-PE (Professional Edition)

Version 9.9 released Sep 2018

This document lists the features provided by Saxon 9.9 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 3.0 Basic Processor	Provides all mandatory features from the XSLT 3.0 specification (including try/catch, iterate, accumulators, maps, named modes, content value templates, and extended patterns).
1.2 XSLT 3.0 Serialization	Provides the serialization feature.
1.3 XSLT 3.0 Compatibility	Provides XSLT 1.0 compatibility mode.
1.4 XSLT 3.0 Dynamic Evaluation	Provides use of the XSLT 3.0 instruction `xsl:evaluate` which allows dynamic evaluation of XPath expressions.
1.5 XSLT 3.0 XPath 3.1 Feature	Provides full use of XPath 3.1 features, including XPath 3.1 functions, and maps and arrays.
1.6 XSLT 3.0 Higher-Order Functions	Provides higher-order functions: specifically, the ability to use functions as values, including dynamic function calls, inline functions, partial function application, and the XPath 3.1 higher-order functions.

Optional features not provided: XSLT 3.0 Schema Awareness, XSLT 3.0 Streaming.

For more details see: XSLT 3.0 conformance.

Relevant W3C Specification: XSLT 3.0 Recommendation (08 June 2017).

2. XPath

2.1 XPath 3.1 Basic	Provides all XPath 3.1 features which do not require schema-awareness or higher-order functions. This includes an implementation of maps and arrays, and support for JSON.

2.2 XPath 3.1 Higher-Order Functions	Provides 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.

Optional features not provided: XPath 3.1 Schema Aware.

For more details see: XPath 3.1 conformance.

Relevant W3C Specification: XPath 3.1 Recommendation (21 March 2017).

3. XQuery

3.1 XQuery 3.1 Minimal Conformance	Provides Minimal Conformance (including try/catch and "group-by") as defined in section 5 of the XQuery 3.1 specification.
3.2 XQuery 3.1 Modules	Provides the Module feature.
3.3 XQuery 3.1 Serialization	Provides the Serialization feature.
3.4 XQuery 3.1 Higher-Order Functions	Provides the Higher-Order Function feature.

Optional features not provided: XQuery 3.1 Schema Aware, XQuery 3.1 Typed Data, XQuery 3.1 Static Typing, XQuery Update 1.0.

For more details see: XQuery 3.1 conformance.

Relevant W3C Specification: XQuery 3.1 Recommendation (21 March 2017).

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. Note that this edition does not provide the ability to create compiled stylesheet packages.

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 (Java and .NET)

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.

For more details see: Saxon extension functions.

13. SQL extension

XSLT extension functions and 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)

Run-time localization support for formatting of dates and numbers, and sorting and comparison of strings, building on the capabilities of the ICU-J library. The Advanced level also includes APIs which allow additional languages to be supported.

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.