

FAQ

This document contains a list of Frequently Asked Questions (FAQ) about SAX. If you have questions about SAX that aren't answered here, try sending them to the sax-users@lists.sourceforge.net mailing list, or to xml-dev.

Last Modified: 28 November 2001

SAX in Java

How do I learn to use SAX?

Start with the QuickStart link at the left. Current books covering Java and XML also address SAX2.

What's a SAX driver, and how do I find one?

A driver implements the SAX2 *XMLReader* interface. It either parses XML directly, or repackages the parser so you can talk to it using SAX interfaces like *ContentHandler*. Although you can have the SAX2 interfaces without a driver, that's not useful; it'd be like using JDBC without a database.

Most current Java programming environments include SAX2 drivers, along with the interfaces, in the core of their XML support. That includes servlet environments and JDK 1.4 (from Sun). See the "links" page (in this website's menu) for some parser distributions you can get if you're temporarily SAX-deprived. Read the documentation with that distribution; you may need to know the name of the driver class to make *XMLReaderFactory.createXMLReader()* behave. (See the "quickstart" link for a table of some common names for SAX drivers.)

The *ContentHandler.characters()* callback is missing data!

Please read the JavaDoc for this method. A parser may split text into any number of separate chunks, and some characters may be reported using *ignorableWhitespace()* instead of this callback.

If you want all the text inside an element, you need to collect the text from the various *characters* callbacks into a buffer. Only when you see the *endElement* event can you be sure that you have seen all the text, and some of it may really "belong" to child elements.

Why doesn't this SAX parser report the XML declaration with *ContentHandler.processingInstruction()*?

Your parser is correct. The XML and text declarations look like processing instructions for historical reasons (to avoid breaking legacy SGML parsers) but they are *not* processing instructions. See production 23 in the XML 1.0 Recommendation. (A SAX2 Extensions 1.1 API will expose the information in these declarations, although not all parsers will support it.)

Does SAX support comments/CDATA sections/DOCTYPE declarations, etc.?

Not in the core API. These kinds of things are pure lexical details, and are not relevant to most kinds of XML processing, so it doesn't make sense to put them in the core and force all implementors to support them.

However, SAX2 is designed to be extensible, and the [LexicalHandler](#) interface is supported by most SAX parsers. SAX2 parsers are *not* required to support this handler, but they are required to report an error if you try to use handlers they don't support.

Should I use SAX or DOM?

Yes! SAX and DOM are appropriate for different situations. If you're interested in the advantages and disadvantages of each, see the link at the left contrasting event based APIs to tree based ones. If you're interested in socio-political aspects, remember that SAX was designed without requiring people to drive or fly to any face-to-face meetings or conferences, so it causes less pollution than the DOM. It was also designed fully in the open, not behind closed doors.

J2SE 1.4 bundles an old version of SAX2. How do I make SAX2 r2 or later available?

Use the new [Endorsed Standards Override Mechanism](#) and copy the new `sax.jar` into the directory specified there. It'll be something like `$JAVA_HOME/jre/lib/endorsed` for the JDK (or drop the "jre" for the JRE). Notice that SAX is on the list of standards it's OK to do this with, right there in alphabetical order. Using this mechanism should let you redistribute a JRE with current SAX support.

SAX2 r2 is API compatible with the older SAX2 version used in the JDK, but it's got better documentation and some bugfixes. The "SAX2 Extensions 1.1" is where new features get added.

Are there SAX2 Conformance Tests?

Some; they're not hosted on this website, since they're under GPL. See the links page for the xmlconf project. There are two kinds of tests. The older tests make sure that SAX2 parsers all do the Right Thing in terms of parsing XML and reporting errors or document data. That's essentially an issue of conforming to the XML 1.0 spec, as its requirements map to SAX2. There are some newer *sax2unit* tests covering SAX2 APIs that don't relate so directly to XML 1.0 conformance requirements.

Those tests are mostly important as a way that you can be sure that different SAX2 parsers do the things described in the API specification. If they don't, you'd probably end up writing code that depended on some particular parser, which is just what SAX is trying to prevent!

SAX in Other Languages

Where's the formal language-independent SAX2 Specification?

There *isn't* any, and probably there won't ever be one. SAX2 in Java is defined by its interfaces and by the base of running code -- it's more like English Common Law rather than the heavily codified Civil Code of ISO or W3C specifications. Outside of Java, SAX is whatever programmers in that language decide it should be.

Where can I find SAX for a language other than Java?

See the link at the left; there are bindings in programming languages environments such as Python, Perl, Pascal, C/C++, and COM.

I'm having trouble using SAX with COM/Visual Basic/C/C++. Can you help?

Sorry, no. Microsoft and other organizations and individuals have released their own software under the name 'SAX', but every one is slightly different. They are free to use the name, but if you need help, you'll need to get in touch with the authors directly.

Licensing

These answers are from David Megginson, who made the original decision to put SAX into the Public Domain.

Why is SAX in the Public Domain? Why not LGPL or another open-source license?

There are two reasons:

1. A license is a threat -- follow the terms or I'll sue you. I don't like to make threats because (a) it's rude, and (b) I know that I could never afford to sue a big company like Sun, Microsoft, Oracle, or IBM anyway, so it would be undignified to pretend.
2. Open source licenses cause big headaches for project managers, and not only because of the recent anti-GPL FUD coming out of Redmond -- including an LGPL or MPL component in a private system may delay a project for weeks trying to get approval from the legal department and senior management, at least until the company adapts its culture to an open-source world.

I respect and use the GPL and other open-source licenses when I work on other projects, of course, and I appreciate all of the good that the GPL has done for the world.

Is the SAX name trademarked?

No: I (David Megginson) assert no intellectual property rights over it. You can use the names *SAX* or *Simple API for XML* for anything you want, anywhere you want. That doesn't mean that you can use *my* name any way you want.

May we include part or all of the SAX code and/or documentation in a book or on a CD?

See the previous answers. SAX is in the Public Domain, so you can do whatever you want with it. There is *no* need for clearance editors at publishing companies to ask for permission.

Why do so many Canadians work with XML?

It's the only international career open to us if we're not good skaters.