

Java

The infrastructure project is basing its code on Java 5.0 (aka 1.5).

Java XML processing

Java 1.4 added native support for XML, but didn't have XPath support, so was limited. Java 5.0 adds the XPath support and some other improvements, but the class/method names are still inconsistent, which makes it a bit clunky to use. The original DOM and SAX libraries were developed by different groups, and the Java developers haven't completely abstracted everything away yet. Even so, it should be possible to do everything you want.

DOM is an object-oriented model based on the concepts of Documents and Nodes.
Sax is a stream-processing model based on an `InputSource`.

Java provides the `Source` and `Result` interfaces as an abstraction layer. These allow the XSLT processors (Transformers) to work with objects that start out as `DOM`, `Sax`, or pure `Stream` objects.

`SAXSource.sourceToInputSource()` attempts to convert any `Source` object into a `Sax` object.
`DocumentBuilder.parse(InputSource is)` will create `DOM Documents` from `SAXSources`.

`XPath.evaluate(String expression, InputSource source)` works with `SAX` objects, while the poorly-described `XPath.evaluate(String expression, Object item)` *should* be able to handle `DOM` objects.

An oddity: Since both `DOM` and `SAX` implement `Source` and `Result`, theoretically you should be able to use `Transformer.transform(Source xmlSource, Result outputTarget)` to convert between types *while* running an XSLT.

Troubleshooting

If duplicate XML nodes appear, you need to use `cloneNode()` more often.

If you're having trouble adding a node to a document, make sure you `importNode()` to convert the node to work with the new document.

If you've having trouble running `importNode()`, make sure you're running on an element, and not the document's root node.

See also [Java Troubleshooting](#) or [Java RMI Troubleshooting](#).