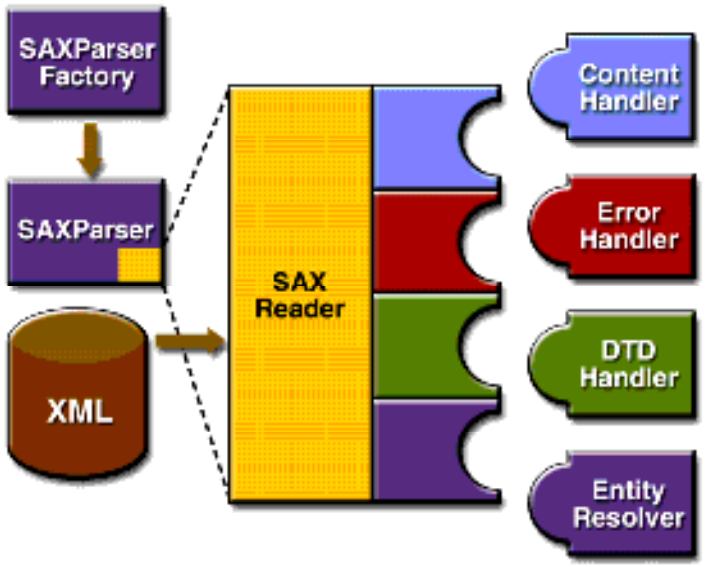
SAX

Sax stands for Simple API for XML processing.
SAX is acommon interface implemented for many different XML parsers.

This is the Basic outline of the SAX parsing API



To start the process, an instance of the SAXParserFactory class is used to generate an instance of the parser.

The parser wraps a SAXReader object. When the parsers parse() method is invoked, the reader invokes one of several callback methods implemented in the application. Those methods are defined by the interfaces ContentHandler, ErrorHandler, DTDHandler, and EntityResolver.

Summary of the Key SAX APIs:

SAXParserFactory

A SAXParserFactory object creates an instance of the parser determined by the system property, javax.xml.parsers.SAXParserFactory.

SAXParser

The SAXParser interface defines several kinds of parse() methods. In general, you pass an XML data source and a DefaultHandler object to the parser, which processes the xml and invokes the appropriate methods in the handler object.

SAXReader

The SAXParser wraps a SAXReader. Typically, you don't care about that, but once in a while you need to get hold of it using SAXParsers getXMLReader(), so you can configure it. it is the SAXReader which carries on the conversation with the SAX event handlers you define.

DefaultHandler

Not shown in the diagram, a DefaultHandler implements the ContentHandler, ErrorHandler, DTDHandler, and EntityResolver interfaces (with null methods), so you can override only the one's you are interested in.

ContentHandler

Methods like startDocument, endDocument, startElement, and endElement are invoked then an xml tag is recognised. This interface also defines methods characters and processingInstruction, which are invoked when the parser encounters the text in anXML element or an inline processing instruction, respectively.

ErrorHandler

Methods error, fatalError, and warning are invoked in response to various parsing errors. The deafult error handler throws an exception for fatal errors and ignores(including validation errors). That's one reason you need to know something about SAX parser, even if you are using the DOM. Sometimes, the application may be able to recover from a validation error. Other times, it may need to generate an exception. To ensure the correct handling, you'll need to supply your own error handler to the parser.

DTDHandler

Defines methods you will generally never be called upon to use. Used when processing a DTD to recognise and act on declarations for an unparsed entity.

EntityResolver

The resolverEntity method is invoked when the parser must identify data identified by a URI. In most cases, a URI is simply a URL, which specifies the location of a document, but in some cases the document may be identified by a URN-a public identifier, or name, that is unique in the web space. The public identifier may be specifiedin addition to the URL. The EntityResolver can then use the public identifier instead of the URL to find the document, for examle to access a local copy of the document if one exists.

The SAX Packages

org.xml.sax
Defines the SAX interfaces. The name org.xml is the package prefix that was settled on by the group that defined the SAX API

org.xml.sax.ext
Defines SAX extensions that are ysed when doing more sphisticated SAX processing, for example, to process a document type definitions (DTD) or to see the detailed syntax for a file.

org.xml.sax.helpers
Contains helper classes that make it easier to use SAX--for example, by defining a default handler that has null-methods for all of the interfaces, so you only need to overrdie the ones you actually want to implement.

javax.xml.parsers
Defines the SAXParserFactory class which returns the SAXParser. Also defines exception classas for reporting errors.