

Simple API for XML

In this chapter we focus on the Simple API for XML (SAX), an event-driven, serial-access mechanism for accessing XML documents. This protocol is frequently used by servlets and network-oriented programs that need to transmit and receive XML documents, because it's the fastest and least memory-intensive mechanism that is currently available for dealing with XML documents, other than StAX.

Note: In a nutshell, SAX is oriented towards *state independent* processing, where the handling of an element does not depend on the elements that came before. StAX, on the other hand, is oriented towards *state dependent* processing. For a more detailed comparison, see SAX and StAX in [Basic Standards](#) and [When to Use SAX](#).

Setting up a program to use SAX requires a bit more work than setting up to use the Document Object Model (DOM). SAX is an event-driven model (you provide the callback methods, and the parser invokes them as it reads the XML data), and that makes it harder to visualize. Finally, you can't "back up" to an earlier part of the document, or rearrange it, any more than you can back up a serial data stream or rearrange characters you have read from that stream.

For those reasons, developers who are writing a user-oriented application that displays an XML document and possibly modifies it will want to use the DOM mechanism described in [Chapter 6](#).

However, even if you plan to build DOM applications exclusively, there are several important reasons for familiarizing yourself with the SAX model:

- **Same Error Handling:** The same kinds of exceptions are generated by the SAX and DOM APIs, so the error handling code is virtually identical.
 - **Handling Validation Errors:** By default, the specifications require that validation errors (which you'll learn more about in this part of the tutorial) are ignored. If you want to throw an exception in the event of a validation error (and you probably do), then you need to understand how SAX error handling works.
 - **Converting Existing Data:** As you'll see in [Chapter 6](#), there is a mechanism you can use to convert an existing data set to XML. However, taking advantage of that mechanism requires an understanding of the SAX model.
-

Note: The XML files used in this chapter can be found in `<INSTALL>/j2eetutorial14/examples/xml/samples/`.
The programs and output listings can be found in `<INSTALL>/j2eetutorial14/examples/jaxp/sax/samples/`.

All of the material in *The J2EE(TM) 1.4 Tutorial* is [copyright](#)-protected and may not be published in other works without express written permission from Sun Microsystems.