

Install and configure the Xerces2 Java parser

Parse XML from Java code

Skill Level: Introductory

[Nicholas Chase](#)

Author

20 Nov 2002

This tutorial is for developers who need to parse XML documents from within their Java applications. It describes the process of installing, configuring, and testing the Xerces2 Java parser, version 2.2.1. This tutorial addresses only the installation and configuration details for Xerces2, not its use.

Section 1. Introduction

Should I take this tutorial?

This tutorial is for developers who need to parse XML documents from within their Java applications. It describes the process of installing, configuring, and testing the Xerces2 Java parser, version 2.2.1.

This tutorial addresses only the installation and configuration details for Xerces2, not its use. The reader need not be familiar with Java, or even XML, to benefit from the material discussed.

What is this tutorial about?

This tutorial describes the process necessary for installing, configuring, and testing the Xerces2 Java parser, which is maintained by the Apache Project. To install

Xerces-J you need to prepare the environment by obtaining an appropriate Java Virtual Machine (JVM), setting environment variables, and installing the files. After installation, you may set other environment variables such as the `CLASSPATH`.

Tools

The tools necessary for this tutorial depend greatly on the platform and originally installed tools.

- A JVM, such as the IBM Java machine or Sun's JDK must be installed and working on the target machine. Links to JVMs for various platforms are listed in [Resources](#).
- The Xerces2 Java 2.2.1 binary files: Apache provides pre-compiled files, so you don't need the source files. Download the binaries from <http://apache.osuosl.org/xml/xalan-j/>.

Section 2. What is Xerces2 Java?

Extensible Markup Language (XML)

XML is a generic tagging language similar to HTML; however, it is much more flexible, allowing the developer to choose the tags and how they represent data. For example, order information may be stored in a structure similar to the following:

```
<?xml version="1.0"?>
<orders>
  <order orderid="A2234q">
    <customerid>GOLD94</customerid>
    <orderdate>10/28/2001</orderdate>
    <item itemid="WCHAIR">
      <itemdesc>Standard Wheelchair</itemdesc>
      <itemprice>$487</itemprice>
      <itemqty>1</itemqty>
    </item>
    <item itemid="WCUSH">
      <itemdesc>Wheelchair Cushion</itemdesc>
      <itemprice>$23</itemprice>
      <itemqty>1</itemqty>
    </item>
  </order>
</orders>
```

In general, these structures contain elements (such as `<itemqty>1</itemqty>`)

and attributes (such as `orderid="A2234q"`) that convey information to an application or to a person reading the file.

Why parse XML files?

To use the data that may be contained in an XML structure, an application must parse the file, converting it into a form where it is useful. Depending on the method used, parsing may create an in-memory representation of the data or a stream of events.

The Xerces2 Java parser supports both types of parsing. Using a Document Object Model (DOM) parser, Xerces2 creates an in-memory representation of all of the elements, attributes, and other data within the file. This representation is known as a `Document` object, and can be both read and modified.

Xerces2 also supports the use of a Simple API for XML (SAX) parser, which instead provides a series of events, such as `startElement` and `characters`. A SAX stream can be faster and more useful than a DOM object in certain situations, but it is read-only.

Supported standards

Xerces2 Java version 2.2.1 (Xerces-J) is based on the World Wide Web Consortium's (W3C) XML 1.0 recommendation and provides support for DOM Level 1, DOM Level 2 Core, Traversal and Range, and Events. It also provides support for SAX versions 1 and 2. Xerces-J was originally created by IBM (where it was called XML4J) and donated to the Apache project.

Xerces-J can be used as a validating parser, with support for both Document Type Definitions (DTDs) and the W3C's XML Schema recommendation version 1.0. Xerces-J 2.2.1 is a fully-conforming XML Schema processor.

Section 3. Prepare the environment

Install a JVM

Xerces-J is written entirely in the Java language, and should work with any recent Java 1 or Java 2 virtual machine, specifically:

- Java 1 -- JDK 1.1.6, 1.1.7, 1.1.8
- Java 2 -- JDK 1.2.2, 1.3

Users of Java 2 version 1.4.x need to make use of the [The Endorsed Standards Override Mechanism](#).

Download locations depend on the target platform:

- Download Windows, Linux, AIX, AS/400, OS/2, OS390, and VM/ESA versions from <http://www.ibm.com/developerworks/java/jdk/index.html>.
- Download Windows, Linux x86, and Solaris/SPARC x86 versions from <http://java.sun.com/j2se/1.3/>.
- Download a Mac OS X version from <http://devworld.apple.com/java/>.
- For other platforms, see <http://java.sun.com/cgi-bin/java-ports.cgi>.

Obtain the binaries

Different versions of the Xerces2 files are available, depending on the developer's tolerance for instability. The current release at the time of this writing can be downloaded at <http://apache.osuosl.org/xml/xalan-j/>.

Download the appropriate archive and place it in the location where you ultimately want the Xerces2 directory to go.

Setting environment variables

To most efficiently use Xerces-J, it helps to have the Java executable on the system path, as determined by the `PATH` environment variable. (An environment variable is a variable that is available to all applications.) How this is accomplished is determined by your operating system.

Windows: Open a DOS-Prompt window and type:

```
set path=%PATH%;c:\jdk1.3.1\bin
```

Make sure to use the actual location of the `java.exe` file. To make this setting apply to every opened window, add this command to `autoexec.bat` and restart. Alternatively, on Windows NT, choose `Start/Settings/Control Panel/System` and select the `Environment` tab. In the upper box, click `Path` and add the new location in the bottom text field. Click `Set`, then `OK`.

UNIX/Linux: The proper way to set a variable depends on the type of UNIX and shell. To add the Java binaries to the path, type:

```
setenv path=$PATH:/path/to/add
```

or

```
set PATH=$PATH:/path/to/add
```

```
EXPORT PATH=$PATH:/path/to/add
```

For the value to be present after a reboot, make sure to add the path within the `config` file in `/etc`.

Section 4. Install the files

Choose a location

The actual location of the Xerces-J installation is not necessarily important, as long as it is not inconvenient. If a structure of Java applications already exists, however, it may be convenient to install the files into this structure. All Xerces2 Java files are installed within a single folder; no information is written to the registry. They are simply Java classes that may be used by other applications.

Unpack the files

The Xerces-J classes are distributed within a ZIP file, `Xerces-J-bin.2.2.1.zip`. To unpack this file, place it in the same location where you ultimately want the Xerces2 directory to be. For example, for an eventual installation of `c:\xerces-2_2_1`, place the distribution file at `c:\Xerces-J-bin.2.2.1.zip`.

A separate zip utility is not required to unpack the files, though it may be used. The `jar` utility, included with the Java Development Kit, will also unpack the files using:

```
jar xf Xerces-J-bin.2.2.1.zip
```

The result is a directory called `xerces-2_2_1`. This directory may be renamed, but

it is often better to leave it as-is to avoid confusion about installed versions.

Included files

The Xerces2 Java installation includes several directories and files:

- **docs:** Contains documentation on all of the relevant APIs, including both Xerces2-specific and general XML classes such as the DOM and SAX classes.
- **samples:** Contains source code for the sample applications. This code may be examined and modified for the developer's own use.
- **data:** Contains several data files that can be used by the sample applications.
- **License:** This text file contains the Apache license for Xerces-J, which permits both use and redistribution.
- **xmlParserAPIs.jar:** This file contains the basic XML interfaces, such as the DOM and SAX interfaces.
- **xercesImpl.jar:** This file contains the Xerces2 implementations of the interfaces in `xmlParserAPIs.jar` and other Xerces2-specific classes.
- **xercesSamples.jar:** This file contains the compiled classes used by the sample applications.
- **Readme.html:** This file simply redirects the user to the actual "home page" for the Xerces-J documentation, `docs/html/index.html`.

Set the CLASSPATH

Before a Java application can use a class, it must be able to find it. Earlier versions of Java required the setting of a `CLASSPATH` environment variable that told the `java` executable where to find these files.

Current versions no longer require the `CLASSPATH` variable, but it remains useful in situations where files are contained in other directories, such as the `xerces-2_2_1` directory. Using the same method used to set the `PATH` variable for the Java executable (in [Setting environment variables](#)), set the `CLASSPATH` variable so it includes the `xmlParserAPIs.jar` and `xercesImpl.jar` files. The following example code was split into two lines due to format limitations. In reality, it is a single line of code:

```
. ;c:\xerces-2_2_1\xmlParserAPIs.jar;  
c:\xerces-2_2_1\xercesImpl.jar;c:\proj\base.jar;
```

For Java 1.3, however, it doesn't end there. The `java` executable automatically uses any classes located in `lib/ext` before it uses classes specified in the `CLASSPATH`, so be sure to check for outdated versions of files such as `xerces.jar` (a previous version which included the classes in both `xercesImpl.jar` and `xmlParserAPIs.jar`), `xercesImpl.jar`, `crimson.jar`, or `xml.jar`. Either remove these files, or overwrite them with more current versions.

The Endorsed Standards Override Mechanism

Many of the DOM- and SAX-related classes included in Xerces-J are also part of Java 1.4, and under normal circumstances, the versions included with Java would be used instead of those provided with Xerces2.

Fortunately, these classes are *endorsed standards*, so you can override the Java version using the Endorsed Standards Override Mechanism by placing the relevant jar files (in this case, `xercesImpl.jar`) in the appropriate directory. The default location is:

```
%JAVA_HOME%\lib\endorsed
```

on Windows or:

```
%JAVA_HOME%/lib/endorsed
```

on a Linux system.

Developers can also choose a new location by setting the `java.endorsed.dirs` system property.

Users of older versions of Java technology need not be concerned with this issue.

Section 5. Test the installation

The sample applications

The best way to test the Xerces2 Java installation is to attempt to run one of the sample applications that are included with the distribution. If these applications run successfully, the installation is installed and configured correctly.

Xerces-J 2.2.1 includes five different types of samples:

- **DOM samples:** These samples include simple (but useful) applications such as `dom.Counter`, `dom.GetElementsByTagName`, and `dom.Writer`, which can be used to test the installation or to see how to accomplish basic DOM tasks. It also includes `dom.DOMAddLines`, which demonstrates more advanced DOM programming, and two experimental samples based on DOM Level 3, `dom.ASBuilder` and `dom.DOM3`.
- **SAX samples:** These samples include `sax.Counter`, `sax.DocumentTracer`, and `sax.Writer`, which demonstrate usage patterns for a SAX parser.
- **Socket samples:** These samples, `socket.DelayedInput` and `socket.KeepSocketOpen`, demonstrate some solutions to difficulties that arise when parsing data from a remote location.
- **User Interface samples:** These samples, `ui.DefaultImages`, `ui.DOMParserSaveEncoding`, `ui.DOMTree`, `ui.DOMTreeFull`, `ui.TreeView`, and `ui.TreeViewer`, demonstrate the use of the Xerces2 parser in the context of a GUI application.
- **XNI samples:** These 12 samples are for hard-core programmers who want to develop applications using the Xerces2 Native Interface, the internal API against which Xerces2 itself is built.

These samples are valuable for several reasons:

- They allow testing of the Xerces-J installation.
- The source code is included with the distribution and you may use it as the basis for new applications.
- In particular, they can be useful for learning how to perform specific actions using Xerces-J.

Running the samples

Running the samples involves making sure the Java executable knows where to find both the Xerces2 base classes (in the `xmlParserAPIs.jar` and `xercesImpl.jar` files) and the compiled sample applications (in the `xercesSamples.jar` file). The base classes should already be part of the `CLASSPATH` variable, so there are two options for making the sample applications available.

The first option is to simply add `xercesSamples.jar` to the `CLASSPATH` variable. This is the simplest option, but has the potential to lead to confusion later, if any new

applications share class names with any of the samples.

The second option is to specify the `CLASSPATH` at the time of execution. If the Java executable sees the `-cp` switch, it uses that classpath information. Consequently, to run, for example, the `dom.Counter` sample, go to the `xerces-2_1_0` directory and type:

```
java -cp %CLASSPATH%;xercesSamples.jar dom.Counter data/personal.xml
```

The sample applications accept any XML file as input.

For more information on other available parameters for the sample applications, check the `docs/samples.html` file included with the distribution.

Troubleshooting

The most common problem encountered in running the sample applications is the inability to locate a particular class. If problems arise, take the following steps:

1. Make sure the `java` executable is on the path, or that it is directly specified.
2. Make sure that `xmlParserAPIs.jar` and `xercesImpl.jar` are part of the `CLASSPATH` by typing `set`.
3. Make sure that `xercesSamples.jar` is part of the `CLASSPATH`, or is specified using the `-cp` switch.
4. If using the `-cp` switch, make sure that it contains an accurate reference to the appropriate `*.jar` files. In other words, simply specifying `xercesSamples.jar` only works when running the application from the directory where `xercesSamples.jar` is located.
5. Make sure the package names (such as `sax.*` or `dom.*`) are specified.
6. Make sure that the names are spelled correctly, including case sensitivity.
7. Make sure to specify a file to process, and that the file is available in the specified location.

Section 6. Summary

Xerces2 Java installation summary

Xerces2 Java is a Java technology-based validating XML parser that supports both DOM and SAX. Installing Xerces-J involves obtaining the distribution, unpacking it into the desired location, and setting the appropriate environment variables.

The Xerces-J distribution also includes sample files handy for both testing the installation and for serving as the basis of new applications.

Resources

Learn

- Xerces2 API documentation is included with the distribution, or at <http://xml.apache.org/xerces2-j/api.html>.
- For a good understanding of the underlying XML recommendations Xerces2 uses, see the [Understanding DOM](#) (*developerWorks*, August 2001) and [Understanding SAX](#) (*developerWorks*, September 2001) tutorials.
- Take the [Validating XML](#) tutorial to find out how to use Xerces to validate an XML document using the W3C's XML Schemas (*developerWorks*, September 2001).
- Find out how you can become an [IBM Certified Developer in XML and related technologies](#).
- Explore many more XML resources on the [developerWorks XML zone](#).
- Stay current with [developerWorks technical events and Webcasts](#).

Get products and technologies

- Download Xerces2 Java from the Apache project at <http://xml.apache.org/xerces2-j/index.html>.
- Download versions of Xerces for [C++](#) and [Perl](#).
- Download versions of Java for various OSes from <http://www.ibm.com/developerworks/java/jdk/index.html>.
- Download Windows, Linux x86, and Solaris/SPARC x86 versions of Java from <http://java.sun.com/j2se/1.3/>.
- Download a Mac OS X version of Java from <http://devworld.apple.com/java/>.
- Build your next development project with [IBM trial software](#), available for download directly from developerWorks.

Discuss

- [Participate in the discussion forum for this content](#).

About the author

Nicholas Chase

Nicholas Chase has been involved in Web site development for companies such as Lucent Technologies, Sun Microsystems, Oracle,

and the Tampa Bay Buccaneers. Nick has been a high school physics teacher, a low-level radioactive waste facility manager, an online science fiction magazine editor, a multimedia engineer, and an Oracle instructor. More recently, he was the Chief Technology Officer of Site Dynamics Interactive Communications in Clearwater, Florida, USA, and is the author of three books on Web development, including *XML Primer Plus* (Sams). He loves to hear from readers and can be reached at nicholas@nicholaschase.com.