



apache
Xerces
xml parser

tutorialspoint
SIMPLY EASY LEARNING

www.tutorialspoint.com

 <https://www.facebook.com/tutorialspointindia>

 <https://twitter.com/tutorialspoint>

About the Tutorial

Apache Xerces is a Java-based processor that provides standard interfaces and implementations for DOM, SAX and StAX XML parsing API standards.

This tutorial will teach you the basic XML concepts and also explain how to use the various types of Apache Xerces XML parsers in a simple and intuitive way.

Audience

This tutorial has been designed for the beginners to help them understand the fundamentals related to XML parsing using Java Programming language.

After completing this tutorial you will find yourself at a moderate level of expertise in knowledge of XML parsing using Java from where you can take yourself to the next levels.

Prerequisites

Knowledge of computers is not a prerequisite to follow the contents of this tutorial. This tutorial assumes no background in computers or computer programming, though basic knowledge of computer terminologies will help in understanding the given concepts very easily.

Copyright & Disclaimer

© Copyright 2017 by Tutorials Point (I) Pvt. Ltd.

All the content and graphics published in this e-book are the property of Tutorials Point (I) Pvt. Ltd. The user of this e-book is prohibited to reuse, retain, copy, distribute or republish any contents or a part of contents of this e-book in any manner without written consent of the publisher.

We strive to update the contents of our website and tutorials as timely and as precisely as possible, however, the contents may contain inaccuracies or errors. Tutorials Point (I) Pvt. Ltd. provides no guarantee regarding the accuracy, timeliness or completeness of our website or its contents including this tutorial. If you discover any errors on our website or in this tutorial, please notify us at contact@tutorialspoint.com

Table of Contents

About the Tutorial	i
Audience.....	i
Prerequisites.....	i
Copyright & Disclaimer	i
Table of Contents	ii
1. XML – Overview	1
2. Apache Xerces – Environment Setup.....	3
3. Apache Xerces – XML Parsers.....	6
XERCES DOM PARSER.....	7
4. Apache Xerces – DOM Parser Overview	8
5. Apache Xerces DOM Parser – Parse XML Document	10
6. Apache Xerces Parser – Query XML Document	15
7. Apache Xerces Parser – Create XML Document.....	18
8. Apache Xerces DOM Parser – Modify XML Document.....	21
XERCES SAX PARSER.....	24
9. Apache Xerces SAX Parser – Overview	25
ContentHandler Interface.....	25
Attributes Interface	26
10. Apache Xerces SAX Parser – Parse XML Document	27
11. Apache Xerces SAX Parser – Query XML Document	33
12. Apache Xerces SAX Parser – Create XML Document.....	39
13. Apache Xerces SAX Parser – Modify XML Document.....	40
XERCES STAX PARSER	46
14. Apache Xerces StAX Parser – Overview.....	47
15. Apache Xerces StAX Parser – Parse XML Document	49
16. Apache Xerces StAX Parser – Query XML Document	53
17. Apache Xerces StAX Parser – Create XML Document	57
18. Apache Xerces StAX Parser – Modify XML Document	60

1. XML – OVERVIEW

What is XML?

XML is a simple text-based language which was designed to store and transport data in plain text format. It stands for Extensible Markup Language. Following are some of the salient features of XML.

- XML is a markup language.
- XML is a tag based language like HTML.
- XML tags are not predefined like HTML.
- You can define your own tags which is why it is called extensible language.
- XML tags are designed to be self-descriptive.
- XML is a W3C Recommendation for data storage and transport.

Example

```
<?xml version="1.0"?>
<Class>
  <Name>First</Name>
  <Sections>
    <Section>
      <Name>A</Name>
      <Students>
        <Student>Rohan</Student>
        <Student>Mohan</Student>
        <Student>Sohan</Student>
        <Student>Lalit</Student>
        <Student>Vinay</Student>
      </Students>
    </Section>
    <Section>
      <Name>B</Name>
      <Students>
        <Student>Robert</Student>
```

```
        <Student>Julie</Student>
        <Student>Kalie</Student>
        <Student>Michael</Student>
    </Students>
</Section>
</Sections>
</Class>
```

Advantages

Following are the advantages provided by XML:

- **Technology agnostic** - Being plain text, XML is technology independent. It can be used by any technology for data storage and transmission purpose.
- **Human readable** - XML uses simple text format. It is human readable and understandable.
- **Extensible** - In XML, custom tags can be created and used very easily.
- **Allow Validation** - Using XSD, DTD and XML structure can be validated easily.

Disadvantages

Following are the disadvantages of XML usage:

- **Redundant Syntax** - Normally XML file contains numerous repetitive terms.
- **Verbose** - Being a verbose language, XML file size increases the transmission and storage costs.

2. APACHE XERCES – ENVIRONMENT SETUP

This chapter takes you through the process of setting up Apache Xerces on Windows and Linux based systems. Apache Xerces can be easily installed and integrated with your current Java environment following a few simple steps without any complex setup procedures. User administration is required while installation.

System Requirements

JDK	Java SE 2 JDK 1.5 or above
Memory	1 GB RAM (recommended)
Disk Space	No minimum requirement
Operating System	Windows XP or above, Linux

Let us now proceed with the steps to install Apache Xerces.

Step 1: Verify your Java Installation

First of all, you need to have the Java Software Development Kit (SDK) installed on your system. To verify this, execute any of the two commands depending on the platform you are working on.

If the Java installation is successful, then it will display the current version and specification of your Java installation. A sample output is given in the following table.

Platform	Command	Sample Output
Windows	Open Command Console and type: java -version	Java version "1.7.0_60" Java (TM) SE Run Time Environment (build 1.7.0_60-b19) Java Hotspot (TM) 64-bit Server VM (build 24.60- b09,mixed mode)
Linux	Open command terminal and type: \$java -version	Java version "1.7.0_25" Open JDK Runtime Environment (rhel- 2.3.10.4.el6_4-x86_64) Open JDK 64-Bit Server VM (build 23.7-b01, mixed mode)

- We assume the readers of this tutorial have Java SDK version 1.7.0_60 installed on their system.

- In case you do not have Java SDK, download its current version from <http://www.oracle.com/technetwork/java/javase/downloads/index.html> and have it installed.

Step 2: Set your Java Environment

Set the environment variable JAVA_HOME to point to the base directory location where Java is installed on your machine. For example,

Platform	Description
Windows	Set JAVA_HOME to C:\ProgramFiles\java\jdk1.7.0_60
Linux	Export JAVA_HOME=/usr/local/java-current

Append the full path of the Java compiler location to the System Path.

Platform	Description
Windows	Linux
Append the String "C:\Program Files\Java\jdk1.7.0_60\bin" to the end of the system variable PATH.	Export PATH=\$PATH:\$JAVA_HOME/bin/

Execute the command java version from the command prompt as explained above.

Step 3: Install Apache Xerces Library

Download the latest version of Apache Xerces from <http://xerces.apache.org/mirrors.cgi> and unzip its contents to a folder from where the required libraries can be linked to your Java program. Let us assume the files are collected in a folder xerces-2_11_0 on the C drive.

Add the complete path of the five jars as highlighted in the above image to the CLASSPATH.

Platform	Description
----------	-------------

Windows	Append the following strings to the end of the user variable CLASSPATH: C:\xerces-2_11_0\resolver.jar; C:\xerces-2_11_0\serializer.jar; C:\xerces-2_11_0\xercesImpl.jar; C:\xerces-2_11_0\xercesSamples.jar; C:\xerces-2_11_0\xml-apis.jar;
Linux	Export CLASSPATH=\$CLASSPATH: /usr/share/xerces-2_11_0\resolver.jar: /usr/share/xerces-2_11_0\serializer.jar: /usr/share/xerces-2_11_0\xercesImpl.jar: /usr/share/xerces-2_11_0\xercesSamples.jar: /usr/share/xerces-2_11_0\xml-apis.jar:

3. APACHE XERCES – XML PARSERS

What is Apache Xerces2?

Xerces2 is a Java based processor and provides standard interfaces and implementations for following XML parsing API standards:

- Document Object Model (DOM) Level 3
- Simple API for XML (SAX) 2.0.2
- Streaming API for XML (StAX) 1.0 Event API
- Java APIs for XML Processing (JAXP) 1.4

What is XML Parsing?

Parsing XML refers to going through the XML document to access data or to modify data in one or the other way.

What is XML Parser?

XML Parser provides a way to access or modify the data present in an XML document. Java provides multiple options to parse XML document. Following are various types of parsers which are commonly used to parse XML documents.

- **Dom Parser** - Parses the document by loading the complete contents of the document and creating its complete hierarchical tree in memory.
- **SAX Parser** - Parses the document on event based triggers. Does not load the complete document into the memory.
- **StAX Parser** - Parses the document in similar fashion to SAX parser but in a more efficient way.

Now, we will elaborate each parser using the Apache Xerces library in our subsequent chapters.

Xerces DOM Parser

4. APACHE XERCES – DOM PARSER OVERVIEW

The Document Object Model is an official recommendation of the World Wide Web Consortium (W3C). It defines an interface that enables programs to access and update the style, structure, and contents of the XML documents. XML parsers that support the DOM, implement that interface.

When to use?

You should use a DOM parser when:

- You need to know a lot about the structure of a document.
- You need to move parts of the document around (you might want to sort certain elements, for example).
- You need to use the information in the document more than once.

What you get?

When you parse an XML document with a DOM parser, you get back a tree structure that contains all of the elements of your document. The DOM provides a variety of functions you can use to examine the contents and structure of the document.

Advantages

The DOM is a common interface for manipulating document structures. One of its design goals is that the Java code written for one DOM-compliant parser should run on any other DOM-compliant parser without changes.

DOM Interfaces

The DOM defines several Java interfaces. Here are the most common interfaces:

- **Node** - The base datatype of the DOM.
- **Element** - The vast majority of the objects you will deal with are Elements.
- **Attr** - Represents an attribute of an element.
- **Text** - The actual content of an Element or Attr.
- **Document** - Represents the entire XML document. A Document object is often referred to as a DOM tree.

Common DOM methods

When you are working with the DOM, there are several methods that are used often:

- **Document.getDocumentElement()** - Returns the root element of the document.

- **Node.getFirstChild()** - Returns the first child of a given Node.
- **Node.getLastChild()** - Returns the last child of a given Node.
- **Node.getNextSibling()** - These methods return the next sibling of a given Node.
- **Node.getPreviousSibling()** - These methods return the previous sibling of a given Node.
- **Node.getAttribute(attrName)** - For a given Node, returns the attribute with the requested name.

End of ebook preview
If you liked what you saw...
Buy it from our store @ <https://store.tutorialspoint.com>