DOM XML Parser

The DOM is the easiest to use Java XML Parser. It parses an entire XML document and load it into memory, modeling it with Object for easy nodel traversal. DOM Parser is slow and consume a lot memory if it load a XML document which contains a lot of data.

* [Read a XML file](http://www.mkyong.com/java/how-to-read-xml-file-in-java-dom-parser/)  
  Read a XML file and print it out each elements.
* [Modify existing XML file](http://www.mkyong.com/java/how-to-modify-xml-file-in-java-dom-parser/)  
  Modify an existing XML file, by update the element and attribute, and also how to delete a element.
* [Create a new XML file](http://www.mkyong.com/java/how-to-create-xml-file-in-java-dom/)  
  Create a XML file with new document, element and attribute.
* [Count XML Elements](http://www.mkyong.com/java/how-to-count-xml-elements-in-java-dom-parser/)  
  Search and count total number of elements in a XML file.

<?xml version="1.0"?>

<company>

<staff id="1001">

<firstname>yong</firstname>

<lastname>mook kim</lastname>

<nickname>mkyong</nickname>

<salary>100000</salary>

</staff>

<staff id="2001">

<firstname>low</firstname>

<lastname>yin fong</lastname>

<nickname>fong fong</nickname>

<salary>200000</salary>

</staff>

</company>

ReadXMLFile.java

package com.mkyong.seo;

import javax.xml.parsers.DocumentBuilderFactory;

import javax.xml.parsers.DocumentBuilder;

import org.w3c.dom.Document;

import org.w3c.dom.NodeList;

import org.w3c.dom.Node;

import org.w3c.dom.Element;

import java.io.File;

public class ReadXMLFile {

public static void main(String argv[]) {

try {

File fXmlFile = new File("/Users/mkyong/staff.xml");

DocumentBuilderFactory dbFactory = DocumentBuilderFactory.newInstance();

DocumentBuilder dBuilder = dbFactory.newDocumentBuilder();

Document doc = dBuilder.parse(fXmlFile);

//optional, but recommended

//read this - http://stackoverflow.com/questions/13786607/normalization-in-dom-parsing-with-java-how-does-it-work

doc.getDocumentElement().normalize();

System.out.println("Root element :" + doc.getDocumentElement().getNodeName());

NodeList nList = doc.getElementsByTagName("staff");

System.out.println("----------------------------");

for (int temp = 0; temp < nList.getLength(); temp++) {

Node nNode = nList.item(temp);

System.out.println("\nCurrent Element :" + nNode.getNodeName());

if (nNode.getNodeType() == Node.ELEMENT\_NODE) {

Element eElement = (Element) nNode;

System.out.println("Staff id : " + eElement.getAttribute("id"));

System.out.println("First Name : " + eElement.getElementsByTagName("firstname").item(0).getTextContent());

System.out.println("Last Name : " + eElement.getElementsByTagName("lastname").item(0).getTextContent());

System.out.println("Nick Name : " + eElement.getElementsByTagName("nickname").item(0).getTextContent());

System.out.println("Salary : " + eElement.getElementsByTagName("salary").item(0).getTextContent());

}

}

} catch (Exception e) {

e.printStackTrace();

}

}

}

Copy

Result

Root element :company

----------------------------

Current Element :staff

Staff id : 1001

First Name : yong

Last Name : mook kim

Nick Name : mkyong

Salary : 100000

Current Element :staff

Staff id : 2001

First Name : low

Last Name : yin fong

Nick Name : fong fong

Salary : 200000

Copy

2. Looping the Node

This example reads the same "staff.xml", and showing you how to loop the node one by one, and print out the node name and value, and also the attribute if any.

ReadXMLFile2.java

package com.mkyong.seo;

import java.io.File;

import javax.xml.parsers.DocumentBuilder;

import javax.xml.parsers.DocumentBuilderFactory;

import org.w3c.dom.Document;

import org.w3c.dom.NamedNodeMap;

import org.w3c.dom.Node;

import org.w3c.dom.NodeList;

public class ReadXMLFile2 {

public static void main(String[] args) {

try {

File file = new File("/Users/mkyong/staff.xml");

DocumentBuilder dBuilder = DocumentBuilderFactory.newInstance()

.newDocumentBuilder();

Document doc = dBuilder.parse(file);

System.out.println("Root element :" + doc.getDocumentElement().getNodeName());

if (doc.hasChildNodes()) {

printNote(doc.getChildNodes());

}

} catch (Exception e) {

System.out.println(e.getMessage());

}

}

private static void printNote(NodeList nodeList) {

for (int count = 0; count < nodeList.getLength(); count++) {

Node tempNode = nodeList.item(count);

// make sure it's element node.

if (tempNode.getNodeType() == Node.ELEMENT\_NODE) {

// get node name and value

System.out.println("\nNode Name =" + tempNode.getNodeName() + " [OPEN]");

System.out.println("Node Value =" + tempNode.getTextContent());

if (tempNode.hasAttributes()) {

// get attributes names and values

NamedNodeMap nodeMap = tempNode.getAttributes();

for (int i = 0; i < nodeMap.getLength(); i++) {

Node node = nodeMap.item(i);

System.out.println("attr name : " + node.getNodeName());

System.out.println("attr value : " + node.getNodeValue());

}

}

if (tempNode.hasChildNodes()) {

// loop again if has child nodes

printNote(tempNode.getChildNodes());

}

System.out.println("Node Name =" + tempNode.getNodeName() + " [CLOSE]");

}

}

}

}

Copy

Result :

Root element :company

Node Name =company [OPEN]

Node Value =

yong

mook kim

mkyong

100000

low

yin fong

fong fong

200000

Node Name =staff [OPEN]

Node Value =

yong

mook kim

mkyong

100000

attr name : id

attr value : 1001

Node Name =firstname [OPEN]

Node Value =yong

Node Name =firstname [CLOSE]

Node Name =lastname [OPEN]

Node Value =mook kim

Node Name =lastname [CLOSE]

Node Name =nickname [OPEN]

Node Value =mkyong

Node Name =nickname [CLOSE]

Node Name =salary [OPEN]

Node Value =100000

Node Name =salary [CLOSE]

Node Name =staff [CLOSE]

Node Name =staff [OPEN]

Node Value =

low

yin fong

fong fong

200000

attr name : id

attr value : 2001

Node Name =firstname [OPEN]

Node Value =low

Node Name =firstname [CLOSE]

Node Name =lastname [OPEN]

Node Value =yin fong

Node Name =lastname [CLOSE]

Node Name =nickname [OPEN]

Node Value =fong fong

Node Name =nickname [CLOSE]

Node Name =salary [OPEN]

Node Value =200000

Node Name =salary [CLOSE]

Node Name =staff [CLOSE]

Node Name =company [CLOSE]

Copy

# How to modify XML file in Java – (DOM Parser)

By [mkyong](http://www.mkyong.com/author/mkyong/" \o "mkyong class=) | August 4, 2011 | Updated : August 29, 2012 | Viewed : 361,086 | +881 pv/w

In this example, we demonstrate the use of **DOM parser** to modify an existing XML file :

1. Add a new element
2. Update existing element attribute
3. Update existing element value
4. Delete existing element

## 1. XML file

See before and after XML file.

File : file.xml – Original XML file.

<?xml version="1.0" encoding="UTF-8" standalone="no" ?>

<company>

<staff id="1">

<firstname>yong</firstname>

<lastname>mook kim</lastname>

<nickname>mkyong</nickname>

<salary>100000</salary>

</staff>

</company>

Copy

Later, update above XML file via DOM XML Parser.

1. Update the staff attribute id = 2
2. Update salary value to 200000
3. Append a new “age” element under staff
4. Delete “firstname” element under staff

File : file.xml – Newly modified XML file.

<?xml version="1.0" encoding="UTF-8" standalone="no" ?>

<company>

<staff id="2">

<lastname>mook kim</lastname>

<nickname>mkyong</nickname>

<salary>2000000</salary>

<age>28</age>

</staff>

</company>

Copy

## 2. DOM Parser

DOM XML parser to update an above XML file.

import java.io.File;

import java.io.IOException;

import javax.xml.parsers.DocumentBuilder;

import javax.xml.parsers.DocumentBuilderFactory;

import javax.xml.parsers.ParserConfigurationException;

import javax.xml.transform.Transformer;

import javax.xml.transform.TransformerException;

import javax.xml.transform.TransformerFactory;

import javax.xml.transform.dom.DOMSource;

import javax.xml.transform.stream.StreamResult;

import org.w3c.dom.Document;

import org.w3c.dom.Element;

import org.w3c.dom.NamedNodeMap;

import org.w3c.dom.Node;

import org.w3c.dom.NodeList;

import org.xml.sax.SAXException;

public class ModifyXMLFile {

public static void main(String argv[]) {

try {

String filepath = "c:\\file.xml";

DocumentBuilderFactory docFactory = DocumentBuilderFactory.newInstance();

DocumentBuilder docBuilder = docFactory.newDocumentBuilder();

Document doc = docBuilder.parse(filepath);

// Get the root element

Node company = doc.getFirstChild();

// Get the staff element , it may not working if tag has spaces, or

// whatever weird characters in front...it's better to use

// getElementsByTagName() to get it directly.

// Node staff = company.getFirstChild();

// Get the staff element by tag name directly

Node staff = doc.getElementsByTagName("staff").item(0);

// update staff attribute

NamedNodeMap attr = staff.getAttributes();

Node nodeAttr = attr.getNamedItem("id");

nodeAttr.setTextContent("2");

// append a new node to staff

Element age = doc.createElement("age");

age.appendChild(doc.createTextNode("28"));

staff.appendChild(age);

// loop the staff child node

NodeList list = staff.getChildNodes();

for (int i = 0; i < list.getLength(); i++) {

Node node = list.item(i);

// get the salary element, and update the value

if ("salary".equals(node.getNodeName())) {

node.setTextContent("2000000");

}

//remove firstname

if ("firstname".equals(node.getNodeName())) {

staff.removeChild(node);

}

}

// write the content into xml file

TransformerFactory transformerFactory = TransformerFactory.newInstance();

Transformer transformer = transformerFactory.newTransformer();

DOMSource source = new DOMSource(doc);

StreamResult result = new StreamResult(new File(filepath));

transformer.transform(source, result);

System.out.println("Done");

} catch (ParserConfigurationException pce) {

pce.printStackTrace();

} catch (TransformerException tfe) {

tfe.printStackTrace();

} catch (IOException ioe) {

ioe.printStackTrace();

} catch (SAXException sae) {

sae.printStackTrace();

}

}

}

Copy

# How to create XML file in Java – (DOM Parser)

By [mkyong](http://www.mkyong.com/author/mkyong/" \o "mkyong class=) | April 2, 2010 | Updated : August 4, 2011 | Viewed : 752,669 | +925 pv/w

DOM provides many handy classes to create XML file easily. Firstly, you have to create a Document with **DocumentBuilder**class, define all the XML content – node, attribute with **Element** class. In last, use **Transformer** class to output the entire XML content to stream output, typically a File.

In this tutorial, we show you how to use DOM XML parser to create a XML file.

## DOM Parser Example

At the end of the example, following XML file named “file.xml” will be created.

<?xml version="1.0" encoding="UTF-8" standalone="no" ?>

<company>

<staff id="1">

<firstname>yong</firstname>

<lastname>mook kim</lastname>

<nickname>mkyong</nickname>

<salary>100000</salary>

</staff>

</company>

Copy

File : WriteXMLFile.java – Java class to create a XML file.

package com.mkyong.core;

import java.io.File;

import javax.xml.parsers.DocumentBuilder;

import javax.xml.parsers.DocumentBuilderFactory;

import javax.xml.parsers.ParserConfigurationException;

import javax.xml.transform.Transformer;

import javax.xml.transform.TransformerException;

import javax.xml.transform.TransformerFactory;

import javax.xml.transform.dom.DOMSource;

import javax.xml.transform.stream.StreamResult;

import org.w3c.dom.Attr;

import org.w3c.dom.Document;

import org.w3c.dom.Element;

public class WriteXMLFile {

public static void main(String argv[]) {

try {

DocumentBuilderFactory docFactory = DocumentBuilderFactory.newInstance();

DocumentBuilder docBuilder = docFactory.newDocumentBuilder();

// root elements

Document doc = docBuilder.newDocument();

Element rootElement = doc.createElement("company");

doc.appendChild(rootElement);

// staff elements

Element staff = doc.createElement("Staff");

rootElement.appendChild(staff);

// set attribute to staff element

Attr attr = doc.createAttribute("id");

attr.setValue("1");

staff.setAttributeNode(attr);

// shorten way

// staff.setAttribute("id", "1");

// firstname elements

Element firstname = doc.createElement("firstname");

firstname.appendChild(doc.createTextNode("yong"));

staff.appendChild(firstname);

// lastname elements

Element lastname = doc.createElement("lastname");

lastname.appendChild(doc.createTextNode("mook kim"));

staff.appendChild(lastname);

// nickname elements

Element nickname = doc.createElement("nickname");

nickname.appendChild(doc.createTextNode("mkyong"));

staff.appendChild(nickname);

// salary elements

Element salary = doc.createElement("salary");

salary.appendChild(doc.createTextNode("100000"));

staff.appendChild(salary);

// write the content into xml file

TransformerFactory transformerFactory = TransformerFactory.newInstance();

Transformer transformer = transformerFactory.newTransformer();

DOMSource source = new DOMSource(doc);

StreamResult result = new StreamResult(new File("C:\\file.xml"));

// Output to console for testing

// StreamResult result = new StreamResult(System.out);

transformer.transform(source, result);

System.out.println("File saved!");

} catch (ParserConfigurationException pce) {

pce.printStackTrace();

} catch (TransformerException tfe) {

tfe.printStackTrace();

}

}

}

Copy

A new XML file is created in “*C:\\file.xml*“, with default UTF-8 encoded.

**Note**  
For debugging, you can change the **StreamResult** to output the XML content to your console.

StreamResult result = new StreamResult(System.out);

transformer.transform(source, result);

# How to count XML Elements in Java – (DOM Parser)

By [mkyong](http://www.mkyong.com/author/mkyong/" \o "mkyong class=) | August 4, 2011 | Viewed : 56,914 | +152 pv/w

In this example, we show you how to use **DOM Parser** to count the total number of elements in a XML file. First, search the element name, and then you can use NodeList.getLength() to get the total number of available elements.

NodeList list = doc.getElementsByTagName("staff");

System.out.println("Total of elements : " + list.getLength());

Copy

File : file.xml

<?xml version="1.0" encoding="UTF-8" standalone="no"?>

<company>

<staff id="1">

<firstname>yong</firstname>

<lastname>mook kim</lastname>

<nickname>mkyong</nickname>

<salary>2000000</salary>

<age>29</age>

</staff>

<staff id="2">

<firstname>low</firstname>

<lastname>yin fong</lastname>

<nickname>fong fong</nickname>

<salary>1000000</salary>

</staff>

<staff id="3">

<firstname>Ali</firstname>

<lastname>Baba</lastname>

<nickname>Alibaba</nickname>

<salary>199000</salary>

<age>40</age>

</staff>

</company>

Copy

File : CountXMLElement.java – Search total number of available “**staff**” elements.

import java.io.IOException;

import javax.xml.parsers.DocumentBuilder;

import javax.xml.parsers.DocumentBuilderFactory;

import javax.xml.parsers.ParserConfigurationException;

import org.w3c.dom.Document;

import org.w3c.dom.NodeList;

import org.xml.sax.SAXException;

public class CountXMLElement {

public static void main(String argv[]) {

try {

String filepath = "c:\\file.xml";

DocumentBuilderFactory docFactory = DocumentBuilderFactory.newInstance();

DocumentBuilder docBuilder = docFactory.newDocumentBuilder();

Document doc = docBuilder.parse(filepath);

NodeList list = doc.getElementsByTagName("staff");

System.out.println("Total of elements : " + list.getLength());

} catch (ParserConfigurationException pce) {

pce.printStackTrace();

} catch (IOException ioe) {

ioe.printStackTrace();

} catch (SAXException sae) {

sae.printStackTrace();

}

}

}

Copy

Output

Total of elements : 3

Copy

SAX XML Parser

SAX parser is work differently with DOM parser, it does not load any XML document into memory and create some object representation of the XML document. Instead, the SAX parser use callback function (org.xml.sax.helpers.DefaultHandler) to informs clients of the XML document structure.

* [Read a XML file](http://www.mkyong.com/java/how-to-read-xml-file-in-java-sax-parser/)  
  Read a XML file via SAX callback methods.
* [Read a UTF-8 XML file](http://www.mkyong.com/java/how-to-read-utf-8-xml-file-in-java-sax-parser/)  
  Read a Unicode XML file via SAX callback methods.
* [SAX Error – Invalid byte 1 of 1-byte UTF-8 sequence](http://www.mkyong.com/java/sax-error-malformedbytesequenceexception-invalid-byte-1-of-1-byte-utf-8-sequence/)  
  Common SAX error for XML file which contains Unicode character.
* [SAX Error – Content is not allowed in prolog](http://www.mkyong.com/java/sax-error-content-is-not-allowed-in-prolog/)  
  Common SAX error for invalid XML file content.

*SAX Parser is faster and uses less memory than DOM parser.*

# How to read XML file in Java – (SAX Parser)

By [mkyong](http://www.mkyong.com/author/mkyong/" \o "mkyong class=) | December 7, 2008 | Updated : May 18, 2015 | Viewed : 789,987 | +914 pv/w

**SAX parser** is working differently with a DOM parser, it neither load any XML document into memory nor create any object representation of the XML document. Instead, the SAX parser use callback function (org.xml.sax.helpers.DefaultHandler) to informs clients of the XML document structure.

SAX Parser is faster and uses less memory than DOM parser.

Copy

See following SAX callback methods :

* **startDocument()** and **endDocument()** – Method called at the start and end of an XML document.
* **startElement()** and **endElement()** – Method called at the start and end of a document element.
* **characters()** – Method called with the text contents in between the start and end tags of an XML document element.

## 1. XML file

Create a simple XML file.

<?xml version="1.0"?>

<company>

<staff>

<firstname>yong</firstname>

<lastname>mook kim</lastname>

<nickname>mkyong</nickname>

<salary>100000</salary>

</staff>

<staff>

<firstname>low</firstname>

<lastname>yin fong</lastname>

<nickname>fong fong</nickname>

<salary>200000</salary>

</staff>

</company>

Copy

## 2. Java file

Use SAX parser to parse the XML file.

import javax.xml.parsers.SAXParser;

import javax.xml.parsers.SAXParserFactory;

import org.xml.sax.Attributes;

import org.xml.sax.SAXException;

import org.xml.sax.helpers.DefaultHandler;

public class ReadXMLFile {

public static void main(String argv[]) {

try {

SAXParserFactory factory = SAXParserFactory.newInstance();

SAXParser saxParser = factory.newSAXParser();

DefaultHandler handler = new DefaultHandler() {

boolean bfname = false;

boolean blname = false;

boolean bnname = false;

boolean bsalary = false;

public void startElement(String uri, String localName,String qName,

Attributes attributes) throws SAXException {

System.out.println("Start Element :" + qName);

if (qName.equalsIgnoreCase("FIRSTNAME")) {

bfname = true;

}

if (qName.equalsIgnoreCase("LASTNAME")) {

blname = true;

}

if (qName.equalsIgnoreCase("NICKNAME")) {

bnname = true;

}

if (qName.equalsIgnoreCase("SALARY")) {

bsalary = true;

}

}

public void endElement(String uri, String localName,

String qName) throws SAXException {

System.out.println("End Element :" + qName);

}

public void characters(char ch[], int start, int length) throws SAXException {

if (bfname) {

System.out.println("First Name : " + new String(ch, start, length));

bfname = false;

}

if (blname) {

System.out.println("Last Name : " + new String(ch, start, length));

blname = false;

}

if (bnname) {

System.out.println("Nick Name : " + new String(ch, start, length));

bnname = false;

}

if (bsalary) {

System.out.println("Salary : " + new String(ch, start, length));

bsalary = false;

}

}

};

saxParser.parse("c:\\file.xml", handler);

} catch (Exception e) {

e.printStackTrace();

}

}

}

Copy

Result

Start Element :company

Start Element :staff

Start Element :firstname

First Name : yong

End Element :firstname

Start Element :lastname

Last Name : mook kim

End Element :lastname

Start Element :nickname

Nick Name : mkyong

End Element :nickname

Start Element :salary

Salary : 100000

End Element :salary

End Element :staff

Start Element :staff

Start Element :firstname

First Name : low

End Element :firstname

Start Element :lastname

Last Name : yin fong

End Element :lastname

Start Element :nickname

Nick Name : fong fong

End Element :nickname

Start Element :salary

Salary : 200000

End Element :salary

End Element :staff

End Element :company

Copy

**Warning**  
This example may encounter exceptions for **UTF-8** XML file, please read this article about [how to read the XML “UTF-8” file in SAX](http://www.mkyong.com/java/how-to-read-utf-8-xml-file-in-java-sax-parser/)

**Note**  
You may interest to read this [How to read XML file in Java – (DOM Parser)](http://www.mkyong.com/java/how-to-read-xml-file-in-java-dom-parser)

# How to read UTF-8 XML file in Java – (SAX Parser)

By [mkyong](http://www.mkyong.com/author/mkyong/" \o "mkyong class=) | December 17, 2009 | Updated : August 4, 2011 | Viewed : 183,267 | +251 pv/w

In previous [Java SAX XML example](http://www.mkyong.com/java/how-to-read-xml-file-in-java-sax-parser/), there is no problem if you use SAX to parse a plain text (ANSI) XML file, however, if you parse a XML file which contains some special UTF-8 characters, it will prompts “[Invalid byte 1 of 1-byte UTF-8 sequence](http://www.mkyong.com/java/sax-error-malformedbytesequenceexception-invalid-byte-1-of-1-byte-utf-8-sequence/)” exception.

com.sun.org.apache.xerces.internal.impl.io.MalformedByteSequenceException:

Invalid byte 1 of 1-byte UTF-8 sequence.

Copy

See following xml file which contain a special UTF-8 characters “§” (press [Alt + 789](http://tools.oratory.com/altcodes.html))

<?xml version="1.0"?>

<company>

<staff>

<firstname>yong</firstname>

<lastname>mook kim</lastname>

<nickname>§</nickname>

<salary>100000</salary>

</staff>

</company>

Copy

To fix it, just override the SAX input source like this :

File file = new File("c:\\file-utf.xml");

InputStream inputStream= new FileInputStream(file);

Reader reader = new InputStreamReader(inputStream,"UTF-8");

InputSource is = new InputSource(reader);

is.setEncoding("UTF-8");

saxParser.parse(is, handler);

Copy

See a full example of using SAX parser to parse a Unicode XML file.

package com.mkyong.test;

import java.io.File;

import java.io.FileInputStream;

import java.io.InputStream;

import java.io.InputStreamReader;

import java.io.Reader;

import javax.xml.parsers.SAXParser;

import javax.xml.parsers.SAXParserFactory;

import org.xml.sax.Attributes;

import org.xml.sax.InputSource;

import org.xml.sax.SAXException;

import org.xml.sax.helpers.DefaultHandler;

public class ReadXMLUTF8FileSAX

{

public static void main( String[] args )

{

try {

SAXParserFactory factory = SAXParserFactory.newInstance();

SAXParser saxParser = factory.newSAXParser();

DefaultHandler handler = new DefaultHandler() {

boolean bfname = false;

boolean blname = false;

boolean bnname = false;

boolean bsalary = false;

public void startElement(String uri, String localName,

String qName, Attributes attributes)

throws SAXException {

System.out.println("Start Element :" + qName);

if (qName.equalsIgnoreCase("FIRSTNAME")) {

bfname = true;

}

if (qName.equalsIgnoreCase("LASTNAME")) {

blname = true;

}

if (qName.equalsIgnoreCase("NICKNAME")) {

bnname = true;

}

if (qName.equalsIgnoreCase("SALARY")) {

bsalary = true;

}

}

public void endElement(String uri, String localName,

String qName)

throws SAXException {

System.out.println("End Element :" + qName);

}

public void characters(char ch[], int start, int length)

throws SAXException {

System.out.println(new String(ch, start, length));

if (bfname) {

System.out.println("First Name : "

+ new String(ch, start, length));

bfname = false;

}

if (blname) {

System.out.println("Last Name : "

+ new String(ch, start, length));

blname = false;

}

if (bnname) {

System.out.println("Nick Name : "

+ new String(ch, start, length));

bnname = false;

}

if (bsalary) {

System.out.println("Salary : "

+ new String(ch, start, length));

bsalary = false;

}

}

};

File file = new File("c:\\file.xml");

InputStream inputStream= new FileInputStream(file);

Reader reader = new InputStreamReader(inputStream,"UTF-8");

InputSource is = new InputSource(reader);

is.setEncoding("UTF-8");

saxParser.parse(is, handler);

} catch (Exception e) {

e.printStackTrace();

}

}

}

Copy

# JAXB hello world example

By [mkyong](http://www.mkyong.com/author/mkyong/" \o "mkyong class=) | August 4, 2011 | Updated : August 29, 2012 | Viewed : 1,258,431 | +2,469 pv/w

**JAXB**, stands for **Java Architecture for XML Binding**, using JAXB annotation to convert Java object to / from XML file. In this tutorial, we show you how to use JAXB to do following stuffs :

1. Marshalling – Convert a Java object into a XML file.
2. Unmarshalling – Convert XML content into a Java Object.

Technologies used in this article

1. JDK 1.6
2. JAXB 2.0

Working with JAXB is easy, just annotate object with JAXB annotation, later use jaxbMarshaller.marshal() or jaxbMarshaller.unmarshal() to do the object / XML conversion.

## 1. JAXB Dependency

No extra jaxb libraries are required if you are using JDK1.6 or above, because [JAXB is bundled in JDK 1.6](http://jaxb.java.net/guide/Which_JAXB_RI_is_included_in_which_JDK_.html).

Note  
For JDK < 1.6, download JAXB from [here](http://jaxb.java.net/), and puts “**jaxb-api.jar**” and “**jaxb-impl.jar**” on your project classpath.

## 2. JAXB Annotation

For object that need to convert to / from XML file, it have to annotate with JAXB annotation. The annotation are pretty self-explanatory, you can refer to this [JAXB guide](http://jaxb.java.net/tutorial/) for detail explanation.

package com.mkyong.core;

import javax.xml.bind.annotation.XmlAttribute;

import javax.xml.bind.annotation.XmlElement;

import javax.xml.bind.annotation.XmlRootElement;

@XmlRootElement

public class Customer {

String name;

int age;

int id;

public String getName() {

return name;

}

@XmlElement

public void setName(String name) {

this.name = name;

}

public int getAge() {

return age;

}

@XmlElement

public void setAge(int age) {

this.age = age;

}

public int getId() {

return id;

}

@XmlAttribute

public void setId(int id) {

this.id = id;

}

}

Copy

## 3. Convert Object to XML

JAXB marshalling example, convert customer object into a XML file. The jaxbMarshaller.marshal() contains a lot of overloaded methods, find one that suit your output.

package com.mkyong.core;

import java.io.File;

import javax.xml.bind.JAXBContext;

import javax.xml.bind.JAXBException;

import javax.xml.bind.Marshaller;

public class JAXBExample {

public static void main(String[] args) {

Customer customer = new Customer();

customer.setId(100);

customer.setName("mkyong");

customer.setAge(29);

try {

File file = new File("C:\\file.xml");

JAXBContext jaxbContext = JAXBContext.newInstance(Customer.class);

Marshaller jaxbMarshaller = jaxbContext.createMarshaller();

// output pretty printed

jaxbMarshaller.setProperty(Marshaller.JAXB\_FORMATTED\_OUTPUT, true);

jaxbMarshaller.marshal(customer, file);

jaxbMarshaller.marshal(customer, System.out);

} catch (JAXBException e) {

e.printStackTrace();

}

}

}

Copy

Output

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>

<customer id="100">

<age>29</age>

<name>mkyong</name>

</customer>

Copy

## 4. Convert XML to Object

JAXB unmarshalling example, convert a XML file content into a customer object. The jaxbMarshaller.unmarshal() contains a lot of overloaded methods, find one that suit yours.

package com.mkyong.core;

import java.io.File;

import javax.xml.bind.JAXBContext;

import javax.xml.bind.JAXBException;

import javax.xml.bind.Unmarshaller;

public class JAXBExample {

public static void main(String[] args) {

try {

File file = new File("C:\\file.xml");

JAXBContext jaxbContext = JAXBContext.newInstance(Customer.class);

Unmarshaller jaxbUnmarshaller = jaxbContext.createUnmarshaller();

Customer customer = (Customer) jaxbUnmarshaller.unmarshal(file);

System.out.println(customer);

} catch (JAXBException e) {

e.printStackTrace();

}

}

}

Copy

Output

Customer [name=mkyong, age=29, id=100]

# How to convert properties file into XML file – Java

By [mkyong](http://www.mkyong.com/author/mkyong/" \o "mkyong class=) | August 23, 2010 | Updated : August 4, 2011 | Viewed : 40,014 | +43 pv/w

Many developers may not aware of this function, actually, the **java.util.Properties** class come with a storeToXML() method to convert existing properties data into a XML file.

**Note**  
Please refer to this [Properties JavaDoc](http://download.oracle.com/javase/1.5.0/docs/api/java/util/Properties.html) for detail explanation.

import java.io.FileOutputStream;

import java.io.IOException;

import java.io.OutputStream;

import java.util.Properties;

public class PropertiesXMLExample

{

public static void main(String[] args) throws IOException

{

Properties props = new Properties();

props.setProperty("email.support", "donot-spam-me@nospam.com");

//where to store?

OutputStream os = new FileOutputStream("c:/email-configuration.xml");

//store the properties detail into a pre-defined XML file

props.storeToXML(os, "Support Email","UTF-8");

System.out.println("Done");

}

}

Copy

The above example will write the properties detail into a XML file “c:/email-configuration.xml“.

<?xml version="1.0" encoding="UTF-8" standalone="no"?>

<!DOCTYPE properties SYSTEM "http://java.sun.com/dtd/properties.dtd">

<properties>

<comment>Support Email</comment>

<entry key="email.support">donot-spam-me@nospam.com</entry>

</properties>

Copy

# How to convert XML file into properties file – Java

By [mkyong](http://www.mkyong.com/author/mkyong/" \o "mkyong class=) | August 23, 2010 | Updated : August 4, 2011 | Viewed : 42,714 | +61 pv/w

In last article, we show you how to [convert properties file into XML file](http://www.mkyong.com/java/how-to-store-properties-into-xml-file/). See following XML file :

<?xml version="1.0" encoding="UTF-8" standalone="no"?>

<!DOCTYPE properties SYSTEM "http://java.sun.com/dtd/properties.dtd">

<properties>

<comment>Support Email</comment>

<entry key="email.support">donot-spam-me@nospam.com</entry>

</properties>

Copy

In this example, we show you how to use **loadFromXML()** method to load above XML file into a properties object, and get the key “email.support” value via getProperty() method.

package com.mkyong;

import java.io.FileInputStream;

import java.io.IOException;

import java.io.InputStream;

import java.util.Properties;

public class PropertiesXMLExample

{

public static void main(String[] args) throws IOException

{

Properties props = new Properties();

InputStream is = new FileInputStream("c:/email-configuration.xml");

//load the xml file into properties format

props.loadFromXML(is);

String email = props.getProperty("email.support");

System.out.println(email);

}

}

Copy

**Output**  
The above example will print out the value of properties key : “**email.support**” :

donot-spam-me@nospam.com

Copy