



ADVERTISEMENT

Successfully experiment and integrate product features in your software applications

OP

ANDROID ▾ CORE JAVA ▾ DESKTOP JAVA ▾ ENTERPRISE JAVA ▾ JAVA BASICS ▾ JVM LANGUAGES ▾ PYTHON SOFTWARE DEVELOPMENT ▾ DEVOPS ▾

Home » Core Java » xml » bind » JAXB: Generate Classes from XSD

ABOUT SAURABH ARORA



Saurabh graduated with an engineering degree in Information Technology from YMCA Institute of Engineering, India. He is SCJP, OCWCD certified and currently working as Technical Lead with one of the biggest service based firms and is involved in projects extensively using Java and JEE technologies. He has worked in E-Commerce, Banking and Telecom domain.



JAXB: Generate Classes from XSD

Posted by: Saurabh Arora in bind March 2nd, 2016 2 Comments 18777 Views

ADVERTISEMENT



In this tutorial, we shall learn generating classes from XML Schema Design (XSD) using JAXB. This can be achieved using JAXB binding compiler

XJC

command.

XJC

is included in the bin directory in the JDK starting with Java SE 6.

1. Requirements

To see this example in action, following is the minimum requirement:

1. JDK 6 (Java SE 6) or later

2. JAXB 2.1 API

2. XJC command

The JAXB

XJC

schema binding compiler transforms/binds, a source XML schema (XSD) to a set of JAXB content classes in the Java programming language.

To see the usage of

XJC

command, just type in the same in command prompt/shell:

NEWSLETTE

128,13 updates and
Join the
access to
well as insig
other relate

Email a

Your em:

☒ Receive Area

Sign up

O'Reilly fe:
experim
Experiment at sc
Measure immedi

split + O'REILLY

Download



JOIN US



unique and
check out o
be a **guest**
your writing

```

saurabh-aroras-macbook:~ saurabharora123$ xjc
grammar is not specified

Usage: xjc [-options ...] <schema file/URL/dir/jar> ... [-b <bindinfo>] ...
If dir is specified, all schema files in it will be compiled.
If jar is specified, /META-INF/sun-jaxb.episode binding file will be compiled.
Options:
  -nv                : do not perform strict validation of the input schema(s)
  -extension         : allow vendor extensions - do not strictly follow the
                      Compatibility Rules and App E.2 from the JAXB Spec
  -b <file/dir>      : specify external bindings files (each <file> must have its own -b)
                      If a directory is given, **/*.xjb is searched
  -d <dir>           : generated files will go into this directory
  -p <pkg>           : specifies the target package
  -httpproxy <proxy> : set HTTP/HTTPS proxy. Format is [user[:password]@]proxyHost:proxyPort
  -httpproxyfile <f> : Works like -httpproxy but takes the argument in a file to protect password
  -classpath <arg>   : specify where to find user class files
  -catalog <file>    : specify catalog files to resolve external entity references
                      support TR9401, XCatalog, and OASIS XML Catalog format.
  -readOnly          : generated files will be in read-only mode
  -npa               : suppress generation of package level annotations (**/package-info.java)
  -no-header         : suppress generation of a file header with timestamp
  -target (2.0|2.1)  : behave like XJC 2.0 or 2.1 and generate code that doesn't use any 2.2 features.
  -encoding <encoding> : specify character encoding for generated source files
  -enableIntrospection : enable correct generation of Boolean getters/setters to enable Bean Introspection apis
  -disableXmlSecurity : disables XML security features when parsing XML documents
  -contentForWildcard : generates content property for types with multiple xs:any derived elements
  -xmlschema         : treat input as W3C XML Schema (default)
  -relaxng            : treat input as RELAX NG (experimental, unsupported)
  -relaxng-compact    : treat input as RELAX NG compact syntax (experimental, unsupported)
  -dtd                : treat input as XML DTD (experimental, unsupported)
  -wsdl              : treat input as WSDL and compile schemas inside it (experimental, unsupported)
  -verbose            : be extra verbose
  -quiet             : suppress compiler output
  -help              : display this help message
  -version            : display version information
  -fullversion        : display full version information

Extensions:
  -Xinject-code       : inject specified Java code fragments into the generated code
  -Xlocator           : enable source location support for generated code
  -Xsync-methods      : generate accessor methods with the 'synchronized' keyword
  -mark-generated     : mark the generated code as @javax.annotation.Generated
  -episode <FILE>     : generate the episode file for separate compilation
  -Xpropertyaccessors : Use XmlAccessType PROPERTY instead of FIELD for generated classes
saurabh-aroras-macbook:~ saurabharora123$

```

XJC Command Usage

3. XJC in Action: Generating classes from XSD

To see the command

```
XJC
```

in action, we will need an XSD file. We will be using following XSD file in our example.

Employee.xsd

```

01 <xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified" xmlns:xs="http://www.w3.org/200
02 <xs:element name="employee">
03   <xs:complexType>
04     <xs:sequence>
05       <xs:element type="xs:byte" name="id"/>
06       <xs:element type="xs:string" name="name"/>
07       <xs:element name="address">
08         <xs:complexType>
09           <xs:sequence>
10             <xs:element type="xs:string" name="addressLine1"/>
11             <xs:element type="xs:string" name="addressLine2"/>
12             <xs:element type="xs:string" name="country"/>
13             <xs:element type="xs:string" name="state"/>
14             <xs:element type="xs:short" name="zip"/>
15           </xs:sequence>
16         </xs:complexType>
17       </xs:element>
18       <xs:element type="xs:string" name="assetsAllocated" maxOccurs="unbounded" minOccurs="0"/>
19     </xs:sequence>
20   </xs:complexType>
21 </xs:element>
22 </xs:schema>

```

To run the command, we shall browse to the directory having the xsd file, and then we shall execute following command:

```
xjc -d src -p com.javacodegeeks.examples.xjc Employee.xsd
```

Here

```
-d
```

specifies to which folder generated classes shall go. In this case it shall be src directory, make sure that the target directory exists.

```
-p
```

access to
well as insig
other relate

Email a

Your em:

☒ Receive
Area

Sign up

JOIN US



unique and
check out o
be a **guest**
your writing

specifies the target package structure. In this case it would be com.javacodegeeks.examples.xjc

Following shall be the output of above command:

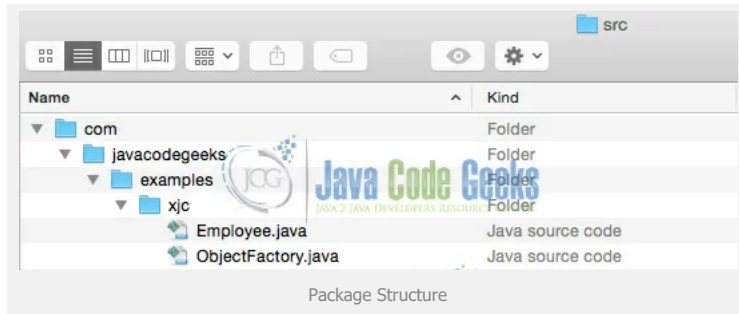
```

saurabh-aroras-macbook:~ saurabh-aroras123$ xjc -d src -p com.javacodegeeks.examples.xjc Employee.xsd
parsing a schema...
compiling a schema...
com/javacodegeeks/examples/xjc/Employee.java
com/javacodegeeks/examples/xjc/ObjectFactory.java
saurabh-aroras-macbook:~ saurabh-aroras123$

```

XJC Execution

And we can see the package structure is created in the desired manner:



Now let us see the java files created.

Employee.java

```

001 //
002 // This file was generated by the JavaTM Architecture for XML Binding(JAXB) Reference Implementation, v2.2.
003 // See <a href="http://java.sun.com/xml/jaxb">http://java.sun.com/xml/jaxb</a>
004 // Any modifications to this file will be lost upon recompilation of the source schema.
005 // Generated on: 2016.02.28 at 03:27:10 PM IST
006 //
007
008 package com;
009
010
011 import java.util.ArrayList;
012 import java.util.List;
013 import javax.xml.bind.annotation.XmlAccessType;
014 import javax.xml.bind.annotation.XmlAccessorType;
015 import javax.xml.bind.annotation.XmlElement;
016 import javax.xml.bind.annotation.XmlRootElement;
017 import javax.xml.bind.annotation.XmlType;
018
019 /**
020  * <p>Java class for anonymous complex type.
021  *
022  * <p>The following schema fragment specifies the expected content contained within this class.
023  *
024  * <complexType>
025  *   <complexContent>
026  *     <restriction base="{http://www.w3.org/2001/XMLSchema}anyType">
027  *       <sequence>
028  *         <element name="name" type="{http://www.w3.org/2001/XMLSchema}string"/>
029  *         <element name="address">
030  *           <complexType>
031  *             <complexContent>
032  *               <restriction base="{http://www.w3.org/2001/XMLSchema}anyType">
033  *                 <sequence>
034  *                   <element name="addressLine1" type="{http://www.w3.org/2001/XMLSchema}string"/>
035  *                   <element name="addressLine2" type="{http://www.w3.org/2001/XMLSchema}string"/>
036  *                   <element name="country" type="{http://www.w3.org/2001/XMLSchema}string"/>
037  *                   <element name="state" type="{http://www.w3.org/2001/XMLSchema}string"/>
038  *                   <element name="zip" type="{http://www.w3.org/2001/XMLSchema}short"/>
039  *                 </sequence>
040  *               </restriction>
041  *             </complexContent>
042  *           </complexType>
043  *         </element>
044  *         <element name="assetsAllocated" type="{http://www.w3.org/2001/XMLSchema}string" maxOccurs="unbo
045  *         <element name="id" type="{http://www.w3.org/2001/XMLSchema}byte"/>
046  *       </sequence>
047  *     </restriction>
048  *   </complexContent>
049  * </complexType>
050  *
051  *
052  */
053 @XmlAccessorType(XmlAccessType.FIELD)
054 @XmlType(name = "", propOrder = {
055     "name",
056     "address",
057     "assetsAllocated",
058     "id"
059 })
060 @XmlRootElement(name = "employee")
061 public class Employee {
062
063

```

access to
well as insig
other relate

Email a

Your em:

☒ Receive
Area

Sign up

JOIN US



unique and
check out o
be a **guest**
your writing

```

064 @XmlElement(required = true)
065 protected String name;
066 @XmlElement(required = true)
067 protected Employee.Address address;
068 protected List assestsAllocated;
069 protected byte id;
070
071 /**
072  * Gets the value of the name property.
073  *
074  * @return
075  *     possible object is
076  *     {@link String }
077  */
078
079 public String getName() {
080     return name;
081 }
082
083 /**
084  * Sets the value of the name property.
085  *
086  * @param value
087  *     allowed object is
088  *     {@link String }
089  */
090
091 public void setName(String value) {
092     this.name = value;
093 }
094
095 /**
096  * Gets the value of the address property.
097  *
098  * @return
099  *     possible object is
100  *     {@link Employee.Address }
101  */
102
103 public Employee.Address getAddress() {
104     return address;
105 }
106
107 /**
108  * Sets the value of the address property.
109  *
110  * @param value
111  *     allowed object is
112  *     {@link Employee.Address }
113  */
114
115 public void setAddress(Employee.Address value) {
116     this.address = value;
117 }
118
119 /**
120  * Gets the value of the assestsAllocated property.
121  *
122  * <p>
123  * This accessor method returns a reference to the live list,
124  * not a snapshot. Therefore any modification you make to the
125  * returned list will be present inside the JAXB object.
126  * This is why there is not a <code>set</code> method for the assestsAllocated property.
127  *
128  * <p>
129  * For example, to add a new item, do as follows:
130  *     getAssestsAllocated().add(newItem);
131  *
132  *
133  * <p>
134  * Objects of the following type(s) are allowed in the list
135  * {@link String }
136  */
137
138
139 public List getAssestsAllocated() {
140     if (assestsAllocated == null) {
141         assestsAllocated = new ArrayList();
142     }
143     return this.assestsAllocated;
144 }
145
146 /**
147  * Gets the value of the id property.
148  *
149  */
150
151 public byte getId() {
152     return id;
153 }
154
155 /**
156  * Sets the value of the id property.
157  *
158  */
159
160 public void setId(byte value) {
161     this.id = value;
162 }
163
164 /**
165  * <p>Java class for anonymous complex type.
166  *
167  * <p>The following schema fragment specifies the expected content contained within this class.
168  *
169  * <complexType>

```

access to
well as insig
other relate

Email a

Your em:

☒ Receive
Area

Sign up

JOIN US



unique and
check out o
be a **guest**
your writing



```

169 * <complexContent>
170 * <restriction base="{http://www.w3.org/2001/XMLSchema}anyType">
171 * <sequence>
172 * <element name="addressLine1" type="{http://www.w3.org/2001/XMLSchema}string"/>
173 * <element name="addressLine2" type="{http://www.w3.org/2001/XMLSchema}string"/>
174 * <element name="country" type="{http://www.w3.org/2001/XMLSchema}string"/>
175 * <element name="state" type="{http://www.w3.org/2001/XMLSchema}string"/>
176 * <element name="zip" type="{http://www.w3.org/2001/XMLSchema}short"/>
177 * </sequence>
178 * </restriction>
179 * </complexContent>
180 * </complexType>
181 *
182 *
183 */
184 @XmlAccessorType(XmlAccessType.FIELD)
185 @XmlType(name = "", propOrder = {
186     "addressLine1",
187     "addressLine2",
188     "country",
189     "state",
190     "zip"
191 })
192 public static class Address {
193
194     @XmlElement(required = true)
195     protected String addressLine1;
196     @XmlElement(required = true)
197     protected String addressLine2;
198     @XmlElement(required = true)
199     protected String country;
200     @XmlElement(required = true)
201     protected String state;
202     protected short zip;
203
204     /**
205      * Gets the value of the addressLine1 property.
206      *
207      * @return
208      *     possible object is
209      *     {@link String }
210      */
211     public String getAddressLine1() {
212         return addressLine1;
213     }
214
215     /**
216      * Sets the value of the addressLine1 property.
217      *
218      * @param value
219      *     allowed object is
220      *     {@link String }
221      */
222     public void setAddressLine1(String value) {
223         this.addressLine1 = value;
224     }
225
226     /**
227      * Gets the value of the addressLine2 property.
228      *
229      * @return
230      *     possible object is
231      *     {@link String }
232      */
233     public String getAddressLine2() {
234         return addressLine2;
235     }
236
237     /**
238      * Sets the value of the addressLine2 property.
239      *
240      * @param value
241      *     allowed object is
242      *     {@link String }
243      */
244     public void setAddressLine2(String value) {
245         this.addressLine2 = value;
246     }
247
248     /**
249      * Gets the value of the country property.
250      *
251      * @return
252      *     possible object is
253      *     {@link String }
254      */
255     public String getCountry() {
256         return country;
257     }
258
259     /**
260      * Sets the value of the country property.
261      *
262      * @param value
263      *     allowed object is
264      *     {@link String }
265      */
266     public void setCountry(String value) {
267         this.country = value;
268     }
269
270
271
272
273

```

access to
well as insig
other relate

Email ac

Your em:

☒ Receive
Area

Sign up

JOIN US



unique and
check out o
be a **guest**
your writing



```

274     }
275
276     /**
277     * Gets the value of the state property.
278     *
279     * @return
280     *     possible object is
281     *     {@link String }
282     */
283
284     public String getState() {
285         return state;
286     }
287
288     /**
289     * Sets the value of the state property.
290     *
291     * @param value
292     *     allowed object is
293     *     {@link String }
294     */
295
296     public void setState(String value) {
297         this.state = value;
298     }
299
300     /**
301     * Gets the value of the zip property.
302     *
303     */
304     public short getZip() {
305         return zip;
306     }
307
308     /**
309     * Sets the value of the zip property.
310     *
311     */
312     public void setZip(short value) {
313         this.zip = value;
314     }
315
316 }
317
318 }

```

We can see that Employee.java also has static class

Address

as it was desired.

Also we can see that an unexpected ObjectFactory.java is also created. This contains factory methods to create objects of classes created. This can come into use when creating JAXBElement representation of objects.

ObjectFactory.java

```

01 //
02 // This file was generated by the JavaTM Architecture for XML Binding(JAXB) Reference Implementation, v2.2.8
03 // See <a href="http://java.sun.com/xml/jaxb">http://java.sun.com/xml/jaxb</a>
04 // Any modifications to this file will be lost upon recompilation of the source schema.
05 // Generated on: 2016.02.27 at 09:09:47 PM IST
06 //
07
08
09 package com.javacodegeeks.examples.xjc;
10
11 import javax.xml.bind.annotation.XmlRegistry;
12
13
14 /**
15 * This object contains factory methods for each
16 * Java content interface and Java element interface
17 * generated in the com.javacodegeeks.examples.xjc package.
18 * <p>An ObjectFactory allows you to programatically
19 * construct new instances of the Java representation
20 * for XML content. The Java representation of XML
21 * content can consist of schema derived interfaces
22 * and classes representing the binding of schema
23 * type definitions, element declarations and model
24 * groups. Factory methods for each of these are
25 * provided in this class.
26 */
27
28 @XmlRegistry
29 public class ObjectFactory {
30
31
32     /**
33     * Create a new ObjectFactory that can be used to create new instances of schema derived classes for pac
34     */
35
36     public ObjectFactory() {
37     }
38
39     /**
40     * Create an instance of {@link Employee }
41     */
42
43     public Employee createEmployee() {
44         return new Employee();
45     }
46
47 }

```

access to
well as insig
other relate

Email a

Your em:

☒ Receive
Area

Sign up

JOIN US



unique and
check out o
be a **guest**
your writing




```

45     }
46
47     /**
48     * Create an instance of {@link Employee.Address }
49     */
50     public Employee.Address createEmployeeAddress() {
51         return new Employee.Address();
52     }
53 }
54
55 }

```

4. Conclusion

In this example, we learnt what is

XJC

, how to generate binding Java classes from an XSD, and the sample code generated. The code generated was the desired class type and the object factory that can be used to create objects of the generated class.

Tagged with:

GENERATE CLASSES

JAX-B

XJC

XSD



Free For School Districts

ManagedMethods [Learn More >](#)

Do you want to know how to develop your skillset to become a **Java Rockstar**?



Subscribe to our newsletter to start Rocking right now!

To get you started we give you our best selling eBooks for **FREE!**

1. JPA Mini Book
2. JVM Troubleshooting Guide
3. JUnit Tutorial for Unit Testing
4. Java Annotations Tutorial
5. Java Interview Questions
6. Spring Interview Questions
7. Android UI Design

and many more

Email address:

Your email address

☒ Receive Java & Developer job alerts in your Area

[Sign up](#)

LIKE THIS ARTICLE? READ MORE FROM JAVA CODE GEEKS

access to
well as insig
other relate

Email ad

Your em:

☒ Receive
Area

[Sign up](#)

JOIN US



unique and
check out o
be a **guest**
your writing



Download Free
O'Reilly eBook

Ad Split Software

JAXB Tutorial for
Java XML Binding –
The ULTIMATE Guide
(PDF Download)

javacodegeeks.com

Programming
Homework Help -
Online Java Tutors

Ad programminghomeworkhelp.com

Using JAXB to
generate XML from
XSD | Java Code
Geeks

javacodegeeks.com

Embed Self-Service
Dashboards

Ad Reveal

JAXB Schema
Validation Example |
Examples Java Code
Geeks

javacodegeeks.com

MOXy's Object
Graphs & Dynamic
JAXB

javacodegeeks.com

JAX-WS JAX
Example | Ex
Java Code G

javacodegeeks.com

✉ Subscribe ▾



Join the discussion

B **I** **U** **S** **☰** **☷** **”** **</>** **🔗** **{ }** **[+]**



This site uses Akismet to reduce spam. [Learn how your comment data is processed.](#)

2 COMMENTS



Oldest ▾



David ⌚ 2 years ago

Did you try on Java 8? For me it works on 6, and 7, but not 8 or greater.

+ -2 - ➡ Reply



Ansari ⌚ 1 year ago

🗨 Reply to [David](#)

Even for me its not working on Java 8, I tired its say Xjc is not recognoized.

+ 0 - ➡ Reply

access to
well as insig
other relate

Email ad

Your em:

☒ Receive
Area

Sign up

JOIN US



unique and
check out o
be a **guest**
your writing

KNOWLEDGE BASE

[Courses](#)

[Minibooks](#)

[News](#)

[Resources](#)

[Tutorials](#)

HALL OF FAME

[Android Alert Dialog Example](#)

[Android OnClickListener Example](#)

[How to convert Character to String and a
String to Character Array in Java](#)

[Java Inheritance example](#)

[Java write to File Example](#)

ABOUT JAVA CODE GEEKS

JCGs (Java Code Geeks) is an independent online community focused on creating the ultimate Java to Java developers resource center; targeted at the technical and non-technical team lead (senior developer), project manager and junior developer. JCGs serve the Java, SOA, Agile and Telecom communities with daily news, webinars, domain experts, articles, tutorials, reviews, announcements, code snippets and open source projects.

DISCLAIMER

