| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/XmlType.html) | [**Tree**](http://docs.google.com/package-tree.html) | [**Deprecated**](http://docs.google.com/deprecated-list.html) | [**Index**](http://docs.google.com/index-files/index-1.html) | [**Help**](http://docs.google.com/help-doc.html) | | --- | --- | --- | --- | --- | --- | --- | --- | | | ***Java™ Platform***  ***Standard Ed. 6*** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [**PREV CLASS**](http://docs.google.com/javax/xml/bind/annotation/XmlTransient.html)   [**NEXT CLASS**](http://docs.google.com/javax/xml/bind/annotation/XmlType.DEFAULT.html) | [**FRAMES**](http://docs.google.com/index.html?javax/xml/bind/annotation/XmlType.html)    [**NO FRAMES**](http://docs.google.com/XmlType.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: REQUIRED | [OPTIONAL](#3znysh7) | DETAIL: [ELEMENT](#26in1rg) |

## **javax.xml.bind.annotation**

Annotation Type XmlType

[@Retention](http://docs.google.com/java/lang/annotation/Retention.html)([value](http://docs.google.com/java/lang/annotation/Retention.html#value())=[RUNTIME](http://docs.google.com/java/lang/annotation/RetentionPolicy.html#RUNTIME))  
[@Target](http://docs.google.com/java/lang/annotation/Target.html)([value](http://docs.google.com/java/lang/annotation/Target.html#value())=[TYPE](http://docs.google.com/java/lang/annotation/ElementType.html#TYPE))  
public @interface **XmlType**

Maps a class or an enum type to a XML Schema type.

**Usage**

The @XmlType annnotation can be used with the following program elements:

* a top level class
* an enum type

See "Package Specification" in javax.xml.bind.package javadoc for additional common information.

### Mapping a Class

A class maps to a XML Schema type. A class is a data container for values represented by properties and fields. A schema type is a data container for values represented by schema components within a schema type's content model (e.g. model groups, attributes etc).

To be mapped, a class must either have a public no-arg constructor or a static no-arg factory method. The static factory method can be specified in factoryMethod() and factoryClass() annotation elements. The static factory method or the no-arg constructor is used during unmarshalling to create an instance of this class. If both are present, the static factory method overrides the no-arg constructor.

A class maps to either a XML Schema complex type or a XML Schema simple type. The XML Schema type is derived based on the mapping of JavaBean properties and fields contained within the class. The schema type to which the class is mapped can either be named or anonymous. A class can be mapped to an anonymous schema type by annotating the class with @XmlType(name="").

Either a global element, local element or a local attribute can be associated with an anonymous type as follows:

* **global element:**  A global element of an anonymous type can be derived by annotating the class with @[XmlRootElement](http://docs.google.com/javax/xml/bind/annotation/XmlRootElement.html). See Example 3 below.
* **local element:**  A JavaBean property that references a class annotated with @XmlType(name="") and is mapped to the element associated with the anonymous type. See Example 4 below.
* **attribute:**  A JavaBean property that references a class annotated with @XmlType(name="") and is mapped to the attribute associated with the anonymous type. See Example 5 below.

**Mapping to XML Schema Complex Type**

* If class is annotated with @XmlType(name="") , it is mapped to an anonymous type otherwise, the class name maps to a complex type name. The XmlName() annotation element can be used to customize the name.
* Properties and fields that are mapped to elements are mapped to a content model within a complex type. The annotation element propOrder() can be used to customize the content model to be xs:all or xs:sequence. It is used for specifying the order of XML elements in xs:sequence.
* Properties and fields can be mapped to attributes within the complex type.
* The targetnamespace of the XML Schema type can be customized using the annotation element namespace().

**Mapping class to XML Schema simple type**

A class can be mapped to a XML Schema simple type using the @XmlValue annotation. For additional details and examples, see @[XmlValue](http://docs.google.com/javax/xml/bind/annotation/XmlValue.html) annotation type.

The following table shows the mapping of the class to a XML Schema complex type or simple type. The notational symbols used in the table are:

* -> : represents a mapping
* [x]+ : one or more occurances of x
* [ @XmlValue property ]: JavaBean property annotated with @XmlValue
* X : don't care

| **Target** | **propOrder** | **ClassBody** | **ComplexType** | **SimpleType** |
| --- | --- | --- | --- | --- |
| Class | {} | [property]+ -> elements | complexcontent  xs:all |  |
| Class | non empty | [property]+ -> elements | complexcontent  xs:sequence |  |
| Class | X | no property -> element | complexcontent  empty sequence |  |
| Class | X | 1 [ @XmlValue property] &&  [property]+ ->attributes | simplecontent |  |
| Class | X | 1 [ @XmlValue property ]&&  no properties -> attribute |  | simpletype |  |

### Mapping an enum type

An enum type maps to a XML schema simple type with enumeration facets. The following annotation elements are ignored since they are not meaningful: propOrder() , factoryMethod() , factoryClass() .

### Usage with other annotations

This annotation can be used with the following annotations: [XmlRootElement](http://docs.google.com/javax/xml/bind/annotation/XmlRootElement.html), [XmlAccessorOrder](http://docs.google.com/javax/xml/bind/annotation/XmlAccessorOrder.html), [XmlAccessorType](http://docs.google.com/javax/xml/bind/annotation/XmlAccessorType.html), [XmlEnum](http://docs.google.com/javax/xml/bind/annotation/XmlEnum.html). However, [XmlAccessorOrder](http://docs.google.com/javax/xml/bind/annotation/XmlAccessorOrder.html) and [XmlAccessorType](http://docs.google.com/javax/xml/bind/annotation/XmlAccessorType.html) are ignored when this annotation is used on an enum type.

**Example 1:**  Map a class to a complex type with xs:sequence with a customized ordering of JavaBean properties.

@XmlType(propOrder={"street", "city" , "state", "zip", "name" })  
 public class USAddress {  
 String getName() {..};  
 void setName(String) {..};  
   
 String getStreet() {..};  
 void setStreet(String) {..};  
  
 String getCity() {..};   
 void setCity(String) {..};  
   
 String getState() {..};  
 void setState(String) {..};  
  
 java.math.BigDecimal getZip() {..};  
 void setZip(java.math.BigDecimal) {..};  
 }  
  
 <!-- XML Schema mapping for USAddress -->  
 <xs:complexType name="USAddress">  
 <xs:sequence>  
 <xs:element name="street" type="xs:string"/>  
 <xs:element name="city" type="xs:string"/>  
 <xs:element name="state" type="xs:string"/>  
 <xs:element name="zip" type="xs:decimal"/>  
 <xs:element name="name" type="xs:string"/>  
 </xs:all>  
 </xs:complexType>

**Example 2:**  Map a class to a complex type with xs:all

@XmlType(propOrder={})  
 public class USAddress { ...}  
   
 <!-- XML Schema mapping for USAddress -->  
 <xs:complexType name="USAddress">  
 <xs:all>  
 <xs:element name="name" type="xs:string"/>  
 <xs:element name="street" type="xs:string"/>  
 <xs:element name="city" type="xs:string"/>  
 <xs:element name="state" type="xs:string"/>  
 <xs:element name="zip" type="xs:decimal"/>  
 </xs:sequence>  
 </xs:complexType>

**Example 3:**  Map a class to a global element with an anonymous type.

@XmlRootElement  
 @XmlType(name="")  
 public class USAddress { ...}  
  
 <!-- XML Schema mapping for USAddress -->  
 <xs:element name="USAddress">  
 <xs:complexType>  
 <xs:sequence>  
 <xs:element name="name" type="xs:string"/>  
 <xs:element name="street" type="xs:string"/>  
 <xs:element name="city" type="xs:string"/>  
 <xs:element name="state" type="xs:string"/>  
 <xs:element name="zip" type="xs:decimal"/>  
 </xs:sequence>  
 </xs:complexType>  
 </xs:element>

**Example 4:**  Map a property to a local element with anonmyous type.

//Example: Code fragment  
 public class Invoice {  
 USAddress addr;  
 ...  
 }  
  
 @XmlType(name="")  
 public class USAddress { ... }  
 }   
  
 <!-- XML Schema mapping for USAddress -->  
 <xs:complexType name="Invoice">  
 <xs:sequence>  
 <xs:element name="addr">  
 <xs:complexType>  
 <xs:element name="name", type="xs:string"/>  
 <xs:element name="city", type="xs:string"/>  
 <xs:element name="city" type="xs:string"/>  
 <xs:element name="state" type="xs:string"/>  
 <xs:element name="zip" type="xs:decimal"/>  
 </xs:complexType>  
 ...  
 </xs:sequence>  
 </xs:complexType>

**Example 5:**  Map a property to an attribute with anonymous type.

//Example: Code fragment  
 public class Item {  
 public String name;  
 @XmlAttribute   
 public USPrice price;  
 }  
   
 // map class to anonymous simple type.   
 @XmlType(name="")  
 public class USPrice {   
 @XmlValue  
 public java.math.BigDecimal price;  
 }  
  
 <!-- Example: XML Schema fragment -->  
 <xs:complexType name="Item">  
 <xs:sequence>  
 <xs:element name="name" type="xs:string"/>  
 <xs:attribute name="price">  
 <xs:simpleType>  
 <xs:restriction base="xs:decimal"/>  
 </xs:simpleType>  
 </xs:attribute>  
 </xs:sequence>  
 </xs:complexType>

**Example 6:**  Define a factoryClass and factoryMethod

@XmlType(name="USAddressType", factoryClass=USAddressFactory.class,  
 factoryMethod="getUSAddress")  
 public class USAddress {  
  
 private String city;  
 private String name;  
 private String state;  
 private String street;  
 private int zip;  
  
 public USAddress(String name, String street, String city,   
 String state, int zip) {  
 this.name = name;  
 this.street = street;  
 this.city = city;  
 this.state = state;  
 this.zip = zip;  
 }  
 }  
  
 public class USAddressFactory {  
 public static USAddress getUSAddress(){  
 return new USAddress("Mark Baker", "23 Elm St",   
 "Dayton", "OH", 90952);  
 }

**Example 7:**  Define factoryMethod and use the default factoryClass

@XmlType(name="USAddressType", factoryMethod="getNewInstance")  
 public class USAddress {  
  
 private String city;  
 private String name;  
 private String state;  
 private String street;  
 private int zip;  
  
 private USAddress() {}  
  
 public static USAddress getNewInstance(){  
 return new USAddress();  
 }  
 }

**Since:** JAXB2.0 **See Also:**[XmlElement](http://docs.google.com/javax/xml/bind/annotation/XmlElement.html), [XmlAttribute](http://docs.google.com/javax/xml/bind/annotation/XmlAttribute.html), [XmlValue](http://docs.google.com/javax/xml/bind/annotation/XmlValue.html), [XmlSchema](http://docs.google.com/javax/xml/bind/annotation/XmlSchema.html)

| **Optional Element Summary** | |
| --- | --- |
| [Class](http://docs.google.com/java/lang/Class.html) | [**factoryClass**](http://docs.google.com/javax/xml/bind/annotation/XmlType.html#factoryClass())            Class containing a no-arg factory method for creating an instance of this class. |
| [String](http://docs.google.com/java/lang/String.html) | [**factoryMethod**](http://docs.google.com/javax/xml/bind/annotation/XmlType.html#factoryMethod())            Name of a no-arg factory method in the class specified in factoryClass factoryClass(). |
| [String](http://docs.google.com/java/lang/String.html) | [**name**](http://docs.google.com/javax/xml/bind/annotation/XmlType.html#name())            Name of the XML Schema type which the class is mapped. |
| [String](http://docs.google.com/java/lang/String.html) | [**namespace**](http://docs.google.com/javax/xml/bind/annotation/XmlType.html#namespace())            Name of the target namespace of the XML Schema type. |
| [String](http://docs.google.com/java/lang/String.html)[] | [**propOrder**](http://docs.google.com/javax/xml/bind/annotation/XmlType.html#propOrder())            Specifies the order for XML Schema elements when class is mapped to a XML Schema complex type. |

### name

public abstract [String](http://docs.google.com/java/lang/String.html) **name**

Name of the XML Schema type which the class is mapped.

**Default:**"##default"

### propOrder

public abstract [String](http://docs.google.com/java/lang/String.html)[] **propOrder**

Specifies the order for XML Schema elements when class is mapped to a XML Schema complex type.

Refer to the table for how the propOrder affects the mapping of class

The propOrder is a list of names of JavaBean properties in the class. Each name in the list is the name of a Java identifier of the JavaBean property. The order in which JavaBean properties are listed is the order of XML Schema elements to which the JavaBean properties are mapped.

All of the JavaBean properties being mapped to XML Schema elements must be listed.

A JavaBean property or field listed in propOrder must not be transient or annotated with @XmlTransient.

The default ordering of JavaBean properties is determined by @[XmlAccessorOrder](http://docs.google.com/javax/xml/bind/annotation/XmlAccessorOrder.html).

**Default:**""

### namespace

public abstract [String](http://docs.google.com/java/lang/String.html) **namespace**

Name of the target namespace of the XML Schema type. By default, this is the target namespace to which the package containing the class is mapped.

**Default:**"##default"

### factoryClass

public abstract [Class](http://docs.google.com/java/lang/Class.html) **factoryClass**

Class containing a no-arg factory method for creating an instance of this class. The default is this class.

If factoryClass is DEFAULT.class and factoryMethod is "", then there is no static factory method.

If factoryClass is DEFAULT.class and factoryMethod is not "", then factoryMethod is the name of a static factory method in this class.

If factoryClass is not DEFAULT.class, then factoryMethod must not be "" and must be the name of a static factory method specified in factoryClass.

**Default:**javax.xml.bind.annotation.XmlType.DEFAULT.class

### factoryMethod

public abstract [String](http://docs.google.com/java/lang/String.html) **factoryMethod**

Name of a no-arg factory method in the class specified in factoryClass factoryClass().

**Default:**""

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/XmlType.html) | [**Tree**](http://docs.google.com/package-tree.html) | [**Deprecated**](http://docs.google.com/deprecated-list.html) | [**Index**](http://docs.google.com/index-files/index-1.html) | [**Help**](http://docs.google.com/help-doc.html) | | --- | --- | --- | --- | --- | --- | --- | --- | | | ***Java™ Platform***  ***Standard Ed. 6*** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [**PREV CLASS**](http://docs.google.com/javax/xml/bind/annotation/XmlTransient.html)   [**NEXT CLASS**](http://docs.google.com/javax/xml/bind/annotation/XmlType.DEFAULT.html) | [**FRAMES**](http://docs.google.com/index.html?javax/xml/bind/annotation/XmlType.html)    [**NO FRAMES**](http://docs.google.com/XmlType.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: REQUIRED | [OPTIONAL](#3znysh7) | DETAIL: [ELEMENT](#26in1rg) |

[Submit a bug or feature](http://bugs.sun.com/services/bugreport/index.jsp)

For further API reference and developer documentation, see [Java SE Developer Documentation](http://docs.google.com/webnotes/devdocs-vs-specs.html). That documentation contains more detailed, developer-targeted descriptions, with conceptual overviews, definitions of terms, workarounds, and working code examples.

Copyright 2006 Sun Microsystems, Inc. All rights reserved. Use is subject to [license terms](http://docs.google.com/legal/license.html). Also see the [documentation redistribution policy](http://java.sun.com/docs/redist.html).