| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/XmlAttribute.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/XmlAttachmentRef.html)   [**NEXT CLASS**](http://docs.google.com/javax/xml/bind/annotation/XmlElement.html) | [**FRAMES**](http://docs.google.com/index.html?javax/xml/bind/annotation/XmlAttribute.html)    [**NO FRAMES**](http://docs.google.com/XmlAttribute.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: REQUIRED | [OPTIONAL](#3znysh7) | DETAIL: [ELEMENT](#17dp8vu) |

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

Annotation Type XmlAttribute

[@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())={[FIELD](http://docs.google.com/java/lang/annotation/ElementType.html#FIELD),[METHOD](http://docs.google.com/java/lang/annotation/ElementType.html#METHOD)})  
public @interface **XmlAttribute**

Maps a JavaBean property to a XML attribute.

**Usage**

The @XmlAttribute annotation can be used with the following program elements:

* JavaBean property
* field

A static final field is mapped to a XML fixed attribute.

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

The usage is subject to the following constraints:

* If type of the field or the property is a collection type, then the collection item type must be mapped to schema simple type.  
   // Examples  
   @XmlAttribute List<Integer> items; //legal  
   @XmlAttribute List<Bar> foo; // illegal if Bar does not map to a schema simple type
* If the type of the field or the property is a non collection type, then the type of the property or field must map to a simple schema type.  
   // Examples  
   @XmlAttribute int foo; // legal  
   @XmlAttribute Foo foo; // illegal if Foo does not map to a schema simple type
* This annotation can be used with the following annotations: [XmlID](http://docs.google.com/javax/xml/bind/annotation/XmlID.html), [XmlIDREF](http://docs.google.com/javax/xml/bind/annotation/XmlIDREF.html), [XmlList](http://docs.google.com/javax/xml/bind/annotation/XmlList.html), [XmlSchemaType](http://docs.google.com/javax/xml/bind/annotation/XmlSchemaType.html), [XmlValue](http://docs.google.com/javax/xml/bind/annotation/XmlValue.html), [XmlAttachmentRef](http://docs.google.com/javax/xml/bind/annotation/XmlAttachmentRef.html), [XmlMimeType](http://docs.google.com/javax/xml/bind/annotation/XmlMimeType.html), [XmlInlineBinaryData](http://docs.google.com/javax/xml/bind/annotation/XmlInlineBinaryData.html), [XmlJavaTypeAdapter](http://docs.google.com/javax/xml/bind/annotation/adapters/XmlJavaTypeAdapter.html)

.

**Example 1:** Map a JavaBean property to an XML attribute.

//Example: Code fragment  
 public class USPrice {   
 @XmlAttribute  
 public java.math.BigDecimal getPrice() {...} ;  
 public void setPrice(java.math.BigDecimal ) {...};  
 }  
  
 <!-- Example: XML Schema fragment -->  
 <xs:complexType name="USPrice">  
 <xs:sequence>  
 </xs:sequence>  
 <xs:attribute name="price" type="xs:decimal"/>  
 </xs:complexType>

**Example 2:** Map a JavaBean property to an XML attribute with anonymous type.

See Example 7 in @[XmlType](http://docs.google.com/javax/xml/bind/annotation/XmlType.html).

**Example 3:** Map a JavaBean collection property to an XML attribute.

// Example: Code fragment  
 class Foo {  
 ...  
 @XmlAttribute List<Integer> items;  
 }   
  
 <!-- Example: XML Schema fragment -->  
 <xs:complexType name="foo">  
 ...  
 <xs:attribute name="items">  
 <xs:simpleType>  
 <xs:list itemType="xs:int"/>  
 </xs:simpleType>  
 </xs:complexType>

**Since:** JAXB2.0 **See Also:**[XmlType](http://docs.google.com/javax/xml/bind/annotation/XmlType.html)

| **Optional Element Summary** | |
| --- | --- |
| [String](http://docs.google.com/java/lang/String.html) | [**name**](http://docs.google.com/javax/xml/bind/annotation/XmlAttribute.html#name())            Name of the XML Schema attribute. |
| [String](http://docs.google.com/java/lang/String.html) | [**namespace**](http://docs.google.com/javax/xml/bind/annotation/XmlAttribute.html#namespace())            Specifies the XML target namespace of the XML Schema attribute. |
| boolean | [**required**](http://docs.google.com/javax/xml/bind/annotation/XmlAttribute.html#required())            Specifies if the XML Schema attribute is optional or required. |

### name

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

Name of the XML Schema attribute. By default, the XML Schema attribute name is derived from the JavaBean property name.

**Default:**"##default"

### required

public abstract boolean **required**

Specifies if the XML Schema attribute is optional or required. If true, then the JavaBean property is mapped to a XML Schema attribute that is required. Otherwise it is mapped to a XML Schema attribute that is optional.

**Default:**false

### namespace

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

Specifies the XML target namespace of the XML Schema attribute.

**Default:**"##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/XmlAttribute.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/XmlAttachmentRef.html)   [**NEXT CLASS**](http://docs.google.com/javax/xml/bind/annotation/XmlElement.html) | [**FRAMES**](http://docs.google.com/index.html?javax/xml/bind/annotation/XmlAttribute.html)    [**NO FRAMES**](http://docs.google.com/XmlAttribute.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: REQUIRED | [OPTIONAL](#3znysh7) | DETAIL: [ELEMENT](#17dp8vu) |

[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).