| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/XmlAnyElement.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/XmlAnyAttribute.html)   [**NEXT CLASS**](http://docs.google.com/javax/xml/bind/annotation/XmlAttachmentRef.html) | [**FRAMES**](http://docs.google.com/index.html?javax/xml/bind/annotation/XmlAnyElement.html)    [**NO FRAMES**](http://docs.google.com/XmlAnyElement.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: REQUIRED | [OPTIONAL](#3znysh7) | DETAIL: [ELEMENT](#2s8eyo1) |

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

Annotation Type XmlAnyElement

[@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 **XmlAnyElement**

Maps a JavaBean property to XML infoset representation and/or JAXB element.

This annotation serves as a "catch-all" property while unmarshalling xml content into a instance of a JAXB annotated class. It typically annotates a multi-valued JavaBean property, but it can occur on single value JavaBean property. During unmarshalling, each xml element that does not match a static @XmlElement or @XmlElementRef annotation for the other JavaBean properties on the class, is added to this "catch-all" property.

## Usages:

@XmlAnyElement  
 public [Element](http://docs.google.com/org/w3c/dom/Element.html)[] others;  
   
 // Collection of [Element](http://docs.google.com/org/w3c/dom/Element.html) or JAXB elements.  
 @XmlAnyElement(lax="true")  
 public [Object](http://docs.google.com/java/lang/Object.html)[] others;  
  
 @XmlAnyElement  
 private List<[Element](http://docs.google.com/org/w3c/dom/Element.html)> nodes;  
  
 @XmlAnyElement  
 private [Element](http://docs.google.com/org/w3c/dom/Element.html) node;

## Restriction usage constraints

This annotation is mutually exclusive with [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), [XmlElements](http://docs.google.com/javax/xml/bind/annotation/XmlElements.html), [XmlID](http://docs.google.com/javax/xml/bind/annotation/XmlID.html), and [XmlIDREF](http://docs.google.com/javax/xml/bind/annotation/XmlIDREF.html).

There can be only one [XmlAnyElement](http://docs.google.com/javax/xml/bind/annotation/XmlAnyElement.html) annotated JavaBean property in a class and its super classes.

## Relationship to other annotations

This annotation can be used with [XmlJavaTypeAdapter](http://docs.google.com/javax/xml/bind/annotation/adapters/XmlJavaTypeAdapter.html), so that users can map their own data structure to DOM, which in turn can be composed into XML.

This annotation can be used with [XmlMixed](http://docs.google.com/javax/xml/bind/annotation/XmlMixed.html) like this:

// List of java.lang.String or DOM nodes.  
 @XmlAnyElement @XmlMixed  
 List<Object> others;

## Schema To Java example

The following schema would produce the following Java class:

<xs:complexType name="foo">  
 <xs:sequence>  
 <xs:element name="a" type="xs:int"></xs:element>  
 <xs:element name="b" type="xs:int"></xs:element>  
 <xs:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"></xs:any>  
 </xs:sequence>  
 </xs:complexType>

class Foo {  
 int a;  
 int b;  
 @[XmlAnyElement](http://docs.google.com/javax/xml/bind/annotation/XmlAnyElement.html)  
 List<Element> any;  
 }

It can unmarshal instances like

<foo xmlns:e="extra">  
 <a>1</a>  
 <e:other></e:other> // this will be bound to DOM, because unmarshalling is orderless  
 <b>3</b>  
 <e:other></e:other>  
 <c>5</c> // this will be bound to DOM, because the annotation doesn't remember namespaces.  
 </foo>

The following schema would produce the following Java class:

<xs:complexType name="bar">  
 <xs:complexContent>  
 <xs:extension base="foo">  
 <xs:sequence>  
 <xs:element name="c" type="xs:int"></xs:element>  
 <xs:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"></xs:any>  
 </xs:sequence>  
 </xs:extension>  
 </xs:complexType>

class Bar extends Foo {  
 int c;  
 // Foo.getAny() also represents wildcard content for type definition bar.  
 }

It can unmarshal instances like

<bar xmlns:e="extra">  
 <a>1</a>  
 <e:other></e:other> // this will be bound to DOM, because unmarshalling is orderless  
 <b>3</b>  
 <e:other></e:other>  
 <c>5</c> // this now goes to Bar.c  
 <e:other></e:other> // this will go to Foo.any  
 </bar>

## Using [**XmlAnyElement**](http://docs.google.com/javax/xml/bind/annotation/XmlAnyElement.html) with [**XmlElementRef**](http://docs.google.com/javax/xml/bind/annotation/XmlElementRef.html)

The [XmlAnyElement](http://docs.google.com/javax/xml/bind/annotation/XmlAnyElement.html) annotation can be used with [XmlElementRef](http://docs.google.com/javax/xml/bind/annotation/XmlElementRef.html)s to designate additional elements that can participate in the content tree.

The following schema would produce the following Java class:

<xs:complexType name="foo">  
 <xs:choice maxOccurs="unbounded" minOccurs="0">  
 <xs:element name="a" type="xs:int"></xs:element>  
 <xs:element name="b" type="xs:int"></xs:element>  
 <xs:any namespace="##other" processContents="lax"></xs:any>  
 </xs:choice>  
 </xs:complexType>

class Foo {  
 @[XmlAnyElement](http://docs.google.com/javax/xml/bind/annotation/XmlAnyElement.html)(lax="true")  
 @[XmlElementRefs](http://docs.google.com/javax/xml/bind/annotation/XmlElementRefs.html)({  
 @[XmlElementRef](http://docs.google.com/javax/xml/bind/annotation/XmlElementRef.html)(name="a", type="JAXBElement.class")  
 @[XmlElementRef](http://docs.google.com/javax/xml/bind/annotation/XmlElementRef.html)(name="b", type="JAXBElement.class")  
 })  
 [List](http://docs.google.com/java/util/List.html)<[Object](http://docs.google.com/java/lang/Object.html)> others;  
 }  
  
 @XmlRegistry  
 class ObjectFactory {  
 ...  
 @XmlElementDecl(name = "a", namespace = "", scope = Foo.class)  
 [JAXBElement](http://docs.google.com/javax/xml/bind/JAXBElement.html)<Integer> createFooA( Integer i ) { ... }  
  
 @XmlElementDecl(name = "b", namespace = "", scope = Foo.class)  
 [JAXBElement](http://docs.google.com/javax/xml/bind/JAXBElement.html)<Integer> createFooB( Integer i ) { ... }

It can unmarshal instances like

<foo xmlns:e="extra">  
 <a>1</a> // this will unmarshal to a <A HREF="../../../../javax/xml/bind/JAXBElement.html" title="class in javax.xml.bind"><CODE>JAXBElement</CODE></A> instance whose value is 1.  
 <e:other></e:other> // this will unmarshal to a DOM <A HREF="../../../../org/w3c/dom/Element.html" title="interface in org.w3c.dom"><CODE>Element</CODE></A>.  
 <b>3</b> // this will unmarshal to a <A HREF="../../../../javax/xml/bind/JAXBElement.html" title="class in javax.xml.bind"><CODE>JAXBElement</CODE></A> instance whose value is 1.  
 </foo>

## W3C XML Schema "lax" wildcard emulation

The lax element of the annotation enables the emulation of the "lax" wildcard semantics. For example, when the Java source code is annotated like this:

@[XmlRootElement](http://docs.google.com/javax/xml/bind/annotation/XmlRootElement.html)  
 class Foo {  
 @XmlAnyElement(lax=true)  
 public [Object](http://docs.google.com/java/lang/Object.html)[] others;  
 }

then the following document will unmarshal like this:

<foo>  
 <unknown></unknown>  
 <foo></foo>  
 </foo>  
  
 Foo foo = unmarshal();  
 // 1 for 'unknown', another for 'foo'  
 assert foo.others.length==2;  
 // 'unknown' unmarshals to a DOM element  
 assert foo.others[0] instanceof Element;  
 // because of lax=true, the 'foo' element eagerly  
 // unmarshals to a Foo object.  
 assert foo.others[1] instanceof Foo;

**Since:** JAXB2.0

| **Optional Element Summary** | |
| --- | --- |
| boolean | [**lax**](http://docs.google.com/javax/xml/bind/annotation/XmlAnyElement.html#lax())            Controls the unmarshaller behavior when it sees elements known to the current [JAXBContext](http://docs.google.com/javax/xml/bind/JAXBContext.html). |
| [Class](http://docs.google.com/java/lang/Class.html)<? extends [DomHandler](http://docs.google.com/javax/xml/bind/annotation/DomHandler.html)> | [**value**](http://docs.google.com/javax/xml/bind/annotation/XmlAnyElement.html#value())            Specifies the [DomHandler](http://docs.google.com/javax/xml/bind/annotation/DomHandler.html) which is responsible for actually converting XML from/to a DOM-like data structure. |

### lax

public abstract boolean **lax**

Controls the unmarshaller behavior when it sees elements known to the current [JAXBContext](http://docs.google.com/javax/xml/bind/JAXBContext.html).

### When false

If false, all the elements that match the property will be unmarshalled to DOM, and the property will only contain DOM elements.

### When true

If true, when an element matches a property marked with [XmlAnyElement](http://docs.google.com/javax/xml/bind/annotation/XmlAnyElement.html) is known to [JAXBContext](http://docs.google.com/javax/xml/bind/JAXBContext.html) (for example, there's a class with [XmlRootElement](http://docs.google.com/javax/xml/bind/annotation/XmlRootElement.html) that has the same tag name, or there's [XmlElementDecl](http://docs.google.com/javax/xml/bind/annotation/XmlElementDecl.html) that has the same tag name), the unmarshaller will eagerly unmarshal this element to the JAXB object, instead of unmarshalling it to DOM. Additionally, if the element is unknown but it has a known xsi:type, the unmarshaller eagerly unmarshals the element to a [JAXBElement](http://docs.google.com/javax/xml/bind/JAXBElement.html), with the unknown element name and the JAXBElement value is set to an instance of the JAXB mapping of the known xsi:type.

As a result, after the unmarshalling, the property can become heterogeneous; it can have both DOM nodes and some JAXB objects at the same time.

This can be used to emulate the "lax" wildcard semantics of the W3C XML Schema.

**Default:**false

### value

public abstract [Class](http://docs.google.com/java/lang/Class.html)<? extends [DomHandler](http://docs.google.com/javax/xml/bind/annotation/DomHandler.html)> **value**

Specifies the [DomHandler](http://docs.google.com/javax/xml/bind/annotation/DomHandler.html) which is responsible for actually converting XML from/to a DOM-like data structure.

**Default:**javax.xml.bind.annotation.W3CDomHandler.class

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/XmlAnyElement.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/XmlAnyAttribute.html)   [**NEXT CLASS**](http://docs.google.com/javax/xml/bind/annotation/XmlAttachmentRef.html) | [**FRAMES**](http://docs.google.com/index.html?javax/xml/bind/annotation/XmlAnyElement.html)    [**NO FRAMES**](http://docs.google.com/XmlAnyElement.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: REQUIRED | [OPTIONAL](#3znysh7) | DETAIL: [ELEMENT](#2s8eyo1) |

[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).