| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/Binder.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   [**NEXT CLASS**](http://docs.google.com/javax/xml/bind/DatatypeConverter.html) | [**FRAMES**](http://docs.google.com/index.html?javax/xml/bind/Binder.html)    [**NO FRAMES**](http://docs.google.com/Binder.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | [CONSTR](#3znysh7) | [METHOD](#2et92p0) | DETAIL: FIELD | [CONSTR](#3dy6vkm) | [METHOD](#4d34og8) |

## **javax.xml.bind**

Class Binder<XmlNode>

[java.lang.Object](http://docs.google.com/java/lang/Object.html)  
 **javax.xml.bind.Binder<XmlNode>**

public abstract class **Binder<XmlNode>**extends [Object](http://docs.google.com/java/lang/Object.html)

Enable synchronization between XML infoset nodes and JAXB objects representing same XML document.

An instance of this class maintains the association between XML nodes of an infoset preserving view and a JAXB representation of an XML document. Navigation between the two views is provided by the methods [getXMLNode(Object)](http://docs.google.com/javax/xml/bind/Binder.html#getXMLNode(java.lang.Object)) and [getJAXBNode(Object)](http://docs.google.com/javax/xml/bind/Binder.html#getJAXBNode(XmlNode)).

Modifications can be made to either the infoset preserving view or the JAXB representation of the document while the other view remains unmodified. The binder is able to synchronize the changes made in the modified view back into the other view using the appropriate Binder update methods, [updateXML(Object, Object)](http://docs.google.com/javax/xml/bind/Binder.html#updateXML(java.lang.Object,%20XmlNode)) or [updateJAXB(Object)](http://docs.google.com/javax/xml/bind/Binder.html#updateJAXB(XmlNode)).

A typical usage scenario is the following:

* load XML document into an XML infoset representation
* [unmarshal(Object)](http://docs.google.com/javax/xml/bind/Binder.html#unmarshal(XmlNode)) XML infoset view to JAXB view. (Note to conserve resources, it is possible to only unmarshal a subtree of the XML infoset view to the JAXB view.)
* application access/updates JAXB view of XML document.
* [updateXML(Object)](http://docs.google.com/javax/xml/bind/Binder.html#updateXML(java.lang.Object)) synchronizes modifications to JAXB view back into the XML infoset view. Update operation preserves as much of original XML infoset as possible (i.e. comments, PI, ...)

A Binder instance is created using the factory method [JAXBContext.createBinder()](http://docs.google.com/javax/xml/bind/JAXBContext.html#createBinder()) or [JAXBContext.createBinder(Class)](http://docs.google.com/javax/xml/bind/JAXBContext.html#createBinder(java.lang.Class)).

The template parameter, XmlNode, is the root interface/class for the XML infoset preserving representation. A Binder implementation is required to minimally support an XmlNode value of org.w3c.dom.Node.class. A Binder implementation can support alternative XML infoset preserving representations.

**Since:** JAXB 2.0

| **Constructor Summary** | |
| --- | --- |
| [**Binder**](http://docs.google.com/javax/xml/bind/Binder.html#Binder())() |

| **Method Summary** | |
| --- | --- |
| abstract  [ValidationEventHandler](http://docs.google.com/javax/xml/bind/ValidationEventHandler.html) | [**getEventHandler**](http://docs.google.com/javax/xml/bind/Binder.html#getEventHandler())()            Return the current event handler or the default event handler if one hasn't been set. |
| abstract  [Object](http://docs.google.com/java/lang/Object.html) | [**getJAXBNode**](http://docs.google.com/javax/xml/bind/Binder.html#getJAXBNode(XmlNode))([XmlNode](http://docs.google.com/javax/xml/bind/Binder.html) xmlNode)            Gets the JAXB object associated with the given XML element. |
| abstract  [Object](http://docs.google.com/java/lang/Object.html) | [**getProperty**](http://docs.google.com/javax/xml/bind/Binder.html#getProperty(java.lang.String))([String](http://docs.google.com/java/lang/String.html) name)            Get the particular property in the underlying implementation of Binder. |
| abstract  [Schema](http://docs.google.com/javax/xml/validation/Schema.html) | [**getSchema**](http://docs.google.com/javax/xml/bind/Binder.html#getSchema())()            Gets the last [Schema](http://docs.google.com/javax/xml/validation/Schema.html) object (including null) set by the [setSchema(Schema)](http://docs.google.com/javax/xml/bind/Binder.html#setSchema(javax.xml.validation.Schema)) method. |
| abstract  [XmlNode](http://docs.google.com/javax/xml/bind/Binder.html) | [**getXMLNode**](http://docs.google.com/javax/xml/bind/Binder.html#getXMLNode(java.lang.Object))([Object](http://docs.google.com/java/lang/Object.html) jaxbObject)            Gets the XML element associated with the given JAXB object. |
| abstract  void | [**marshal**](http://docs.google.com/javax/xml/bind/Binder.html#marshal(java.lang.Object,%20XmlNode))([Object](http://docs.google.com/java/lang/Object.html) jaxbObject, [XmlNode](http://docs.google.com/javax/xml/bind/Binder.html) xmlNode)            Marshal a JAXB object tree to a new XML document. |
| abstract  void | [**setEventHandler**](http://docs.google.com/javax/xml/bind/Binder.html#setEventHandler(javax.xml.bind.ValidationEventHandler))([ValidationEventHandler](http://docs.google.com/javax/xml/bind/ValidationEventHandler.html) handler)            Allow an application to register a ValidationEventHandler. |
| abstract  void | [**setProperty**](http://docs.google.com/javax/xml/bind/Binder.html#setProperty(java.lang.String,%20java.lang.Object))([String](http://docs.google.com/java/lang/String.html) name, [Object](http://docs.google.com/java/lang/Object.html) value)            Set the particular property in the underlying implementation of Binder. |
| abstract  void | [**setSchema**](http://docs.google.com/javax/xml/bind/Binder.html#setSchema(javax.xml.validation.Schema))([Schema](http://docs.google.com/javax/xml/validation/Schema.html) schema)            Specifies whether marshal, unmarshal and update methods performs validation on their XML content. |
| abstract  [Object](http://docs.google.com/java/lang/Object.html) | [**unmarshal**](http://docs.google.com/javax/xml/bind/Binder.html#unmarshal(XmlNode))([XmlNode](http://docs.google.com/javax/xml/bind/Binder.html) xmlNode)            Unmarshal XML infoset view to a JAXB object tree. |
| abstract   | <T> [JAXBElement](http://docs.google.com/javax/xml/bind/JAXBElement.html)<T> | | --- | | [**unmarshal**](http://docs.google.com/javax/xml/bind/Binder.html#unmarshal(XmlNode,%20java.lang.Class))([XmlNode](http://docs.google.com/javax/xml/bind/Binder.html) xmlNode, [Class](http://docs.google.com/java/lang/Class.html)<T> declaredType)            Unmarshal XML root element by provided declaredType to a JAXB object tree. |
| abstract  [Object](http://docs.google.com/java/lang/Object.html) | [**updateJAXB**](http://docs.google.com/javax/xml/bind/Binder.html#updateJAXB(XmlNode))([XmlNode](http://docs.google.com/javax/xml/bind/Binder.html) xmlNode)            Takes an XML node and updates its associated JAXB object and its descendants. |
| abstract  [XmlNode](http://docs.google.com/javax/xml/bind/Binder.html) | [**updateXML**](http://docs.google.com/javax/xml/bind/Binder.html#updateXML(java.lang.Object))([Object](http://docs.google.com/java/lang/Object.html) jaxbObject)            Takes an JAXB object and updates its associated XML node and its descendants. |
| abstract  [XmlNode](http://docs.google.com/javax/xml/bind/Binder.html) | [**updateXML**](http://docs.google.com/javax/xml/bind/Binder.html#updateXML(java.lang.Object,%20XmlNode))([Object](http://docs.google.com/java/lang/Object.html) jaxbObject, [XmlNode](http://docs.google.com/javax/xml/bind/Binder.html) xmlNode)            Changes in JAXB object tree are updated in its associated XML parse tree. |

| **Methods inherited from class java.lang.**[**Object**](http://docs.google.com/java/lang/Object.html) |
| --- |
| [clone](http://docs.google.com/java/lang/Object.html#clone()), [equals](http://docs.google.com/java/lang/Object.html#equals(java.lang.Object)), [finalize](http://docs.google.com/java/lang/Object.html#finalize()), [getClass](http://docs.google.com/java/lang/Object.html#getClass()), [hashCode](http://docs.google.com/java/lang/Object.html#hashCode()), [notify](http://docs.google.com/java/lang/Object.html#notify()), [notifyAll](http://docs.google.com/java/lang/Object.html#notifyAll()), [toString](http://docs.google.com/java/lang/Object.html#toString()), [wait](http://docs.google.com/java/lang/Object.html#wait()), [wait](http://docs.google.com/java/lang/Object.html#wait(long)), [wait](http://docs.google.com/java/lang/Object.html#wait(long,%20int)) |

| **Constructor Detail** |
| --- |

### Binder

public **Binder**()

| **Method Detail** |
| --- |

### unmarshal

public abstract [Object](http://docs.google.com/java/lang/Object.html) **unmarshal**([XmlNode](http://docs.google.com/javax/xml/bind/Binder.html) xmlNode)  
 throws [JAXBException](http://docs.google.com/javax/xml/bind/JAXBException.html)

Unmarshal XML infoset view to a JAXB object tree.

This method is similar to [Unmarshaller.unmarshal(Node)](http://docs.google.com/javax/xml/bind/Unmarshaller.html#unmarshal(org.w3c.dom.Node)) with the addition of maintaining the association between XML nodes and the produced JAXB objects, enabling future update operations, [updateXML(Object, Object)](http://docs.google.com/javax/xml/bind/Binder.html#updateXML(java.lang.Object,%20XmlNode)) or [updateJAXB(Object)](http://docs.google.com/javax/xml/bind/Binder.html#updateJAXB(XmlNode)).

When [getSchema()](http://docs.google.com/javax/xml/bind/Binder.html#getSchema()) is non-null, xmlNode and its descendants is validated during this operation.

This method throws [UnmarshalException](http://docs.google.com/javax/xml/bind/UnmarshalException.html) when the Binder's [JAXBContext](http://docs.google.com/javax/xml/bind/JAXBContext.html) does not have a mapping for the XML element name or the type, specifiable via @xsi:type, of xmlNode to a JAXB mapped class. The method [unmarshal(Object, Class)](http://docs.google.com/javax/xml/bind/Binder.html#unmarshal(XmlNode,%20java.lang.Class)) enables an application to specify the JAXB mapped class that the xmlNode should be mapped to.

**Parameters:**xmlNode - the document/element to unmarshal XML data from. **Returns:**the newly created root object of the JAXB object tree. **Throws:** [JAXBException](http://docs.google.com/javax/xml/bind/JAXBException.html) - If any unexpected errors occur while unmarshalling [UnmarshalException](http://docs.google.com/javax/xml/bind/UnmarshalException.html) - If the [ValidationEventHandler](http://docs.google.com/javax/xml/bind/ValidationEventHandler.html) returns false from its handleEvent method or the Binder is unable to perform the XML to Java binding. [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - If the node parameter is null

### unmarshal

public abstract <T> [JAXBElement](http://docs.google.com/javax/xml/bind/JAXBElement.html)<T> **unmarshal**([XmlNode](http://docs.google.com/javax/xml/bind/Binder.html) xmlNode,  
 [Class](http://docs.google.com/java/lang/Class.html)<T> declaredType)  
 throws [JAXBException](http://docs.google.com/javax/xml/bind/JAXBException.html)

Unmarshal XML root element by provided declaredType to a JAXB object tree.

Implements [Unmarshal by Declared Type](http://docs.google.com/Unmarshaller.html#unmarshalByDeclaredType)

This method is similar to [Unmarshaller.unmarshal(Node, Class)](http://docs.google.com/javax/xml/bind/Unmarshaller.html#unmarshal(org.w3c.dom.Node,%20java.lang.Class)) with the addition of maintaining the association between XML nodes and the produced JAXB objects, enabling future update operations, [updateXML(Object, Object)](http://docs.google.com/javax/xml/bind/Binder.html#updateXML(java.lang.Object,%20XmlNode)) or [updateJAXB(Object)](http://docs.google.com/javax/xml/bind/Binder.html#updateJAXB(XmlNode)).

When [getSchema()](http://docs.google.com/javax/xml/bind/Binder.html#getSchema()) is non-null, xmlNode and its descendants is validated during this operation.

**Parameters:**xmlNode - the document/element to unmarshal XML data from.declaredType - appropriate JAXB mapped class to hold node's XML data. **Returns:**[JAXB Element](#3o7alnk) representation of node **Throws:** [JAXBException](http://docs.google.com/javax/xml/bind/JAXBException.html) - If any unexpected errors occur while unmarshalling [UnmarshalException](http://docs.google.com/javax/xml/bind/UnmarshalException.html) - If the [ValidationEventHandler](http://docs.google.com/javax/xml/bind/ValidationEventHandler.html) returns false from its handleEvent method or the Binder is unable to perform the XML to Java binding. [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - If any of the input parameters are null**Since:** JAXB2.0

### marshal

public abstract void **marshal**([Object](http://docs.google.com/java/lang/Object.html) jaxbObject,  
 [XmlNode](http://docs.google.com/javax/xml/bind/Binder.html) xmlNode)  
 throws [JAXBException](http://docs.google.com/javax/xml/bind/JAXBException.html)

Marshal a JAXB object tree to a new XML document.

This method is similar to [Marshaller.marshal(Object, Node)](http://docs.google.com/javax/xml/bind/Marshaller.html#marshal(java.lang.Object,%20org.w3c.dom.Node)) with the addition of maintaining the association between JAXB objects and the produced XML nodes, enabling future update operations such as [updateXML(Object, Object)](http://docs.google.com/javax/xml/bind/Binder.html#updateXML(java.lang.Object,%20XmlNode)) or [updateJAXB(Object)](http://docs.google.com/javax/xml/bind/Binder.html#updateJAXB(XmlNode)).

When [getSchema()](http://docs.google.com/javax/xml/bind/Binder.html#getSchema()) is non-null, the marshalled xml content is validated during this operation.

**Parameters:**jaxbObject - The content tree to be marshalled.xmlNode - The parameter must be a Node that accepts children. **Throws:** [JAXBException](http://docs.google.com/javax/xml/bind/JAXBException.html) - If any unexpected problem occurs during the marshalling. [MarshalException](http://docs.google.com/javax/xml/bind/MarshalException.html) - If the [ValidationEventHandler](http://docs.google.com/javax/xml/bind/ValidationEventHandler.html) returns false from its handleEvent method or the Binder is unable to marshal jaxbObject (or any object reachable from jaxbObject). [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - If any of the method parameters are null

### getXMLNode

public abstract [XmlNode](http://docs.google.com/javax/xml/bind/Binder.html) **getXMLNode**([Object](http://docs.google.com/java/lang/Object.html) jaxbObject)

Gets the XML element associated with the given JAXB object.

Once a JAXB object tree is associated with an XML fragment, this method enables navigation between the two trees.

An association between an XML element and a JAXB object is established by the bind methods and the update methods. Note that this association is partial; not all XML elements have associated JAXB objects, and not all JAXB objects have associated XML elements.

**Parameters:**jaxbObject - An instance that is reachable from a prior call to a bind or update method that returned a JAXB object tree. **Returns:**null if the specified JAXB object is not known to this [Binder](http://docs.google.com/javax/xml/bind/Binder.html), or if it is not associated with an XML element. **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - If the jaxbObject parameter is null

### getJAXBNode

public abstract [Object](http://docs.google.com/java/lang/Object.html) **getJAXBNode**([XmlNode](http://docs.google.com/javax/xml/bind/Binder.html) xmlNode)

Gets the JAXB object associated with the given XML element.

Once a JAXB object tree is associated with an XML fragment, this method enables navigation between the two trees.

An association between an XML element and a JAXB object is established by the unmarshal, marshal and update methods. Note that this association is partial; not all XML elements have associated JAXB objects, and not all JAXB objects have associated XML elements.

**Returns:**null if the specified XML node is not known to this [Binder](http://docs.google.com/javax/xml/bind/Binder.html), or if it is not associated with a JAXB object. **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - If the node parameter is null

### updateXML

public abstract [XmlNode](http://docs.google.com/javax/xml/bind/Binder.html) **updateXML**([Object](http://docs.google.com/java/lang/Object.html) jaxbObject)  
 throws [JAXBException](http://docs.google.com/javax/xml/bind/JAXBException.html)

Takes an JAXB object and updates its associated XML node and its descendants.

This is a convenience method of:

updateXML( jaxbObject, getXMLNode(jaxbObject));

**Throws:** [JAXBException](http://docs.google.com/javax/xml/bind/JAXBException.html) - If any unexpected problem occurs updating corresponding XML content. [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - If the jaxbObject parameter is null

### updateXML

public abstract [XmlNode](http://docs.google.com/javax/xml/bind/Binder.html) **updateXML**([Object](http://docs.google.com/java/lang/Object.html) jaxbObject,  
 [XmlNode](http://docs.google.com/javax/xml/bind/Binder.html) xmlNode)  
 throws [JAXBException](http://docs.google.com/javax/xml/bind/JAXBException.html)

Changes in JAXB object tree are updated in its associated XML parse tree.

This operation can be thought of as an "in-place" marshalling. The difference is that instead of creating a whole new XML tree, this operation updates an existing tree while trying to preserve the XML as much as possible.

For example, unknown elements/attributes in XML that were not bound to JAXB will be left untouched (whereas a marshalling operation would create a new tree that doesn't contain any of those.)

As a side-effect, this operation updates the association between XML nodes and JAXB objects.

**Parameters:**jaxbObject - root of potentially modified JAXB object treexmlNode - root of update target XML parse tree **Returns:**Returns the updated XML node. Typically, this is the same node you passed in as *xmlNode*, but it maybe a different object, for example when the tag name of the object has changed. **Throws:** [JAXBException](http://docs.google.com/javax/xml/bind/JAXBException.html) - If any unexpected problem occurs updating corresponding XML content. [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - If any of the input parameters are null

### updateJAXB

public abstract [Object](http://docs.google.com/java/lang/Object.html) **updateJAXB**([XmlNode](http://docs.google.com/javax/xml/bind/Binder.html) xmlNode)  
 throws [JAXBException](http://docs.google.com/javax/xml/bind/JAXBException.html)

Takes an XML node and updates its associated JAXB object and its descendants.

This operation can be thought of as an "in-place" unmarshalling. The difference is that instead of creating a whole new JAXB tree, this operation updates an existing tree, reusing as much JAXB objects as possible.

As a side-effect, this operation updates the association between XML nodes and JAXB objects.

**Returns:**Returns the updated JAXB object. Typically, this is the same object that was returned from earlier [marshal(Object,Object)](http://docs.google.com/javax/xml/bind/Binder.html#marshal(java.lang.Object,%20XmlNode)) or [updateJAXB(Object)](http://docs.google.com/javax/xml/bind/Binder.html#updateJAXB(XmlNode)) method invocation, but it maybe a different object, for example when the name of the XML element has changed. **Throws:** [JAXBException](http://docs.google.com/javax/xml/bind/JAXBException.html) - If any unexpected problem occurs updating corresponding JAXB mapped content. [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - If node parameter is null

### setSchema

public abstract void **setSchema**([Schema](http://docs.google.com/javax/xml/validation/Schema.html) schema)

Specifies whether marshal, unmarshal and update methods performs validation on their XML content.

**Parameters:**schema - set to null to disable validation.**See Also:**[Unmarshaller.setSchema(Schema)](http://docs.google.com/javax/xml/bind/Unmarshaller.html#setSchema(javax.xml.validation.Schema))

### getSchema

public abstract [Schema](http://docs.google.com/javax/xml/validation/Schema.html) **getSchema**()

Gets the last [Schema](http://docs.google.com/javax/xml/validation/Schema.html) object (including null) set by the [setSchema(Schema)](http://docs.google.com/javax/xml/bind/Binder.html#setSchema(javax.xml.validation.Schema)) method.

**Returns:**the Schema object for validation or null if not present

### setEventHandler

public abstract void **setEventHandler**([ValidationEventHandler](http://docs.google.com/javax/xml/bind/ValidationEventHandler.html) handler)  
 throws [JAXBException](http://docs.google.com/javax/xml/bind/JAXBException.html)

Allow an application to register a ValidationEventHandler.

The ValidationEventHandler will be called by the JAXB Provider if any validation errors are encountered during calls to any of the Binder unmarshal, marshal and update methods.

Calling this method with a null parameter will cause the Binder to revert back to the default default event handler.

**Parameters:**handler - the validation event handler **Throws:** [JAXBException](http://docs.google.com/javax/xml/bind/JAXBException.html) - if an error was encountered while setting the event handler

### getEventHandler

public abstract [ValidationEventHandler](http://docs.google.com/javax/xml/bind/ValidationEventHandler.html) **getEventHandler**()  
 throws [JAXBException](http://docs.google.com/javax/xml/bind/JAXBException.html)

Return the current event handler or the default event handler if one hasn't been set.

**Returns:**the current ValidationEventHandler or the default event handler if it hasn't been set **Throws:** [JAXBException](http://docs.google.com/javax/xml/bind/JAXBException.html) - if an error was encountered while getting the current event handler

### setProperty

public abstract void **setProperty**([String](http://docs.google.com/java/lang/String.html) name,  
 [Object](http://docs.google.com/java/lang/Object.html) value)  
 throws [PropertyException](http://docs.google.com/javax/xml/bind/PropertyException.html)

Set the particular property in the underlying implementation of Binder. This method can only be used to set one of the standard JAXB defined unmarshal/marshal properties or a provider specific property for binder, unmarshal or marshal. Attempting to set an undefined property will result in a PropertyException being thrown. See [Supported Unmarshal Properties](http://docs.google.com/Unmarshaller.html#supportedProps) and [Supported Marshal Properties](http://docs.google.com/Marshaller.html#supportedProps).

**Parameters:**name - the name of the property to be set. This value can either be specified using one of the constant fields or a user supplied string.value - the value of the property to be set **Throws:** [PropertyException](http://docs.google.com/javax/xml/bind/PropertyException.html) - when there is an error processing the given property or value [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - If the name parameter is null

### getProperty

public abstract [Object](http://docs.google.com/java/lang/Object.html) **getProperty**([String](http://docs.google.com/java/lang/String.html) name)  
 throws [PropertyException](http://docs.google.com/javax/xml/bind/PropertyException.html)

Get the particular property in the underlying implementation of Binder. This method can only be used to get one of the standard JAXB defined unmarshal/marshal properties or a provider specific property for binder, unmarshal or marshal. Attempting to get an undefined property will result in a PropertyException being thrown. See [Supported Unmarshal Properties](http://docs.google.com/Unmarshaller.html#supportedProps) and [Supported Marshal Properties](http://docs.google.com/Marshaller.html#supportedProps).

**Parameters:**name - the name of the property to retrieve **Returns:**the value of the requested property **Throws:** [PropertyException](http://docs.google.com/javax/xml/bind/PropertyException.html) - when there is an error retrieving the given property or value property name [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - If the name parameter is null

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/Binder.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   [**NEXT CLASS**](http://docs.google.com/javax/xml/bind/DatatypeConverter.html) | [**FRAMES**](http://docs.google.com/index.html?javax/xml/bind/Binder.html)    [**NO FRAMES**](http://docs.google.com/Binder.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | [CONSTR](#3znysh7) | [METHOD](#2et92p0) | DETAIL: FIELD | [CONSTR](#3dy6vkm) | [METHOD](#4d34og8) |

[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).