| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/AbstractUnmarshallerImpl.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/helpers/AbstractMarshallerImpl.html)   [**NEXT CLASS**](http://docs.google.com/javax/xml/bind/helpers/DefaultValidationEventHandler.html) | [**FRAMES**](http://docs.google.com/index.html?javax/xml/bind/helpers/AbstractUnmarshallerImpl.html)    [**NO FRAMES**](http://docs.google.com/AbstractUnmarshallerImpl.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#tyjcwt) | [CONSTR](#3dy6vkm) | [METHOD](#1t3h5sf) | DETAIL: [FIELD](#17dp8vu) | [CONSTR](#26in1rg) | [METHOD](#35nkun2) |

## **javax.xml.bind.helpers**

Class AbstractUnmarshallerImpl

[java.lang.Object](http://docs.google.com/java/lang/Object.html)  
 **javax.xml.bind.helpers.AbstractUnmarshallerImpl**

**All Implemented Interfaces:** [Unmarshaller](http://docs.google.com/javax/xml/bind/Unmarshaller.html)

public abstract class **AbstractUnmarshallerImpl**extends [Object](http://docs.google.com/java/lang/Object.html)implements [Unmarshaller](http://docs.google.com/javax/xml/bind/Unmarshaller.html)

Partial default Unmarshaller implementation.

This class provides a partial default implementation for the [Unmarshaller](http://docs.google.com/javax/xml/bind/Unmarshaller.html)interface.

A JAXB Provider has to implement five methods (getUnmarshallerHandler, unmarshal(Node), unmarshal(XMLReader,InputSource), unmarshal(XMLStreamReader), and unmarshal(XMLEventReader).

**Since:** JAXB1.0 **See Also:**[Unmarshaller](http://docs.google.com/javax/xml/bind/Unmarshaller.html)

| **Nested Class Summary** | |
| --- | --- |

| **Nested classes/interfaces inherited from interface javax.xml.bind.**[**Unmarshaller**](http://docs.google.com/javax/xml/bind/Unmarshaller.html) |
| --- |
| [Unmarshaller.Listener](http://docs.google.com/javax/xml/bind/Unmarshaller.Listener.html) |

| **Field Summary** | |
| --- | --- |
| protected  boolean | [**validating**](http://docs.google.com/javax/xml/bind/helpers/AbstractUnmarshallerImpl.html#validating)            whether or not the unmarshaller will validate |

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

| **Method Summary** | |
| --- | --- |
| protected  [UnmarshalException](http://docs.google.com/javax/xml/bind/UnmarshalException.html) | [**createUnmarshalException**](http://docs.google.com/javax/xml/bind/helpers/AbstractUnmarshallerImpl.html#createUnmarshalException(org.xml.sax.SAXException))([SAXException](http://docs.google.com/org/xml/sax/SAXException.html) e)            Creates an UnmarshalException from a SAXException. |
| | <A extends [XmlAdapter](http://docs.google.com/javax/xml/bind/annotation/adapters/XmlAdapter.html)>  A | | --- | | [**getAdapter**](http://docs.google.com/javax/xml/bind/helpers/AbstractUnmarshallerImpl.html#getAdapter(java.lang.Class))([Class](http://docs.google.com/java/lang/Class.html)<A> type)            Gets the adapter associated with the specified type. |
| [AttachmentUnmarshaller](http://docs.google.com/javax/xml/bind/attachment/AttachmentUnmarshaller.html) | [**getAttachmentUnmarshaller**](http://docs.google.com/javax/xml/bind/helpers/AbstractUnmarshallerImpl.html#getAttachmentUnmarshaller())() |
| [ValidationEventHandler](http://docs.google.com/javax/xml/bind/ValidationEventHandler.html) | [**getEventHandler**](http://docs.google.com/javax/xml/bind/helpers/AbstractUnmarshallerImpl.html#getEventHandler())()            Return the current event handler or the default event handler if one hasn't been set. |
| [Unmarshaller.Listener](http://docs.google.com/javax/xml/bind/Unmarshaller.Listener.html) | [**getListener**](http://docs.google.com/javax/xml/bind/helpers/AbstractUnmarshallerImpl.html#getListener())()            Return [Unmarshaller.Listener](http://docs.google.com/javax/xml/bind/Unmarshaller.Listener.html) registered with this [Unmarshaller](http://docs.google.com/javax/xml/bind/Unmarshaller.html). |
| [Object](http://docs.google.com/java/lang/Object.html) | [**getProperty**](http://docs.google.com/javax/xml/bind/helpers/AbstractUnmarshallerImpl.html#getProperty(java.lang.String))([String](http://docs.google.com/java/lang/String.html) name)            Default implementation of the getProperty method always throws PropertyException since there are no required properties. |
| [Schema](http://docs.google.com/javax/xml/validation/Schema.html) | [**getSchema**](http://docs.google.com/javax/xml/bind/helpers/AbstractUnmarshallerImpl.html#getSchema())()            Get the JAXP 1.3 [Schema](http://docs.google.com/javax/xml/validation/Schema.html) object being used to perform unmarshal-time validation. |
| protected  [XMLReader](http://docs.google.com/org/xml/sax/XMLReader.html) | [**getXMLReader**](http://docs.google.com/javax/xml/bind/helpers/AbstractUnmarshallerImpl.html#getXMLReader())()            Obtains a configured XMLReader. |
| boolean | [**isValidating**](http://docs.google.com/javax/xml/bind/helpers/AbstractUnmarshallerImpl.html#isValidating())()            Indicates whether or not the Unmarshaller is configured to validate during unmarshal operations. |
| | <A extends [XmlAdapter](http://docs.google.com/javax/xml/bind/annotation/adapters/XmlAdapter.html)>  void | | --- | | [**setAdapter**](http://docs.google.com/javax/xml/bind/helpers/AbstractUnmarshallerImpl.html#setAdapter(java.lang.Class,%20A))([Class](http://docs.google.com/java/lang/Class.html)<A> type, A adapter)            Associates a configured instance of [XmlAdapter](http://docs.google.com/javax/xml/bind/annotation/adapters/XmlAdapter.html) with this unmarshaller. |
| void | [**setAdapter**](http://docs.google.com/javax/xml/bind/helpers/AbstractUnmarshallerImpl.html#setAdapter(javax.xml.bind.annotation.adapters.XmlAdapter))([XmlAdapter](http://docs.google.com/javax/xml/bind/annotation/adapters/XmlAdapter.html) adapter)            Associates a configured instance of [XmlAdapter](http://docs.google.com/javax/xml/bind/annotation/adapters/XmlAdapter.html) with this unmarshaller. |
| void | [**setAttachmentUnmarshaller**](http://docs.google.com/javax/xml/bind/helpers/AbstractUnmarshallerImpl.html#setAttachmentUnmarshaller(javax.xml.bind.attachment.AttachmentUnmarshaller))([AttachmentUnmarshaller](http://docs.google.com/javax/xml/bind/attachment/AttachmentUnmarshaller.html) au)            Associate a context that resolves cid's, content-id URIs, to binary data passed as attachments. |
| void | [**setEventHandler**](http://docs.google.com/javax/xml/bind/helpers/AbstractUnmarshallerImpl.html#setEventHandler(javax.xml.bind.ValidationEventHandler))([ValidationEventHandler](http://docs.google.com/javax/xml/bind/ValidationEventHandler.html) handler)            Allow an application to register a validation event handler. |
| void | [**setListener**](http://docs.google.com/javax/xml/bind/helpers/AbstractUnmarshallerImpl.html#setListener(javax.xml.bind.Unmarshaller.Listener))([Unmarshaller.Listener](http://docs.google.com/javax/xml/bind/Unmarshaller.Listener.html) listener)             Register unmarshal event callback [Unmarshaller.Listener](http://docs.google.com/javax/xml/bind/Unmarshaller.Listener.html) with this [Unmarshaller](http://docs.google.com/javax/xml/bind/Unmarshaller.html). |
| void | [**setProperty**](http://docs.google.com/javax/xml/bind/helpers/AbstractUnmarshallerImpl.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)            Default implementation of the setProperty method always throws PropertyException since there are no required properties. |
| void | [**setSchema**](http://docs.google.com/javax/xml/bind/helpers/AbstractUnmarshallerImpl.html#setSchema(javax.xml.validation.Schema))([Schema](http://docs.google.com/javax/xml/validation/Schema.html) schema)            Specify the JAXP 1.3 [Schema](http://docs.google.com/javax/xml/validation/Schema.html) object that should be used to validate subsequent unmarshal operations against. |
| void | [**setValidating**](http://docs.google.com/javax/xml/bind/helpers/AbstractUnmarshallerImpl.html#setValidating(boolean))(boolean validating)            Specifies whether or not the Unmarshaller should validate during unmarshal operations. |
| [Object](http://docs.google.com/java/lang/Object.html) | [**unmarshal**](http://docs.google.com/javax/xml/bind/helpers/AbstractUnmarshallerImpl.html#unmarshal(java.io.File))([File](http://docs.google.com/java/io/File.html) f)            Unmarshal XML data from the specified file and return the resulting content tree. |
| [Object](http://docs.google.com/java/lang/Object.html) | [**unmarshal**](http://docs.google.com/javax/xml/bind/helpers/AbstractUnmarshallerImpl.html#unmarshal(org.xml.sax.InputSource))([InputSource](http://docs.google.com/org/xml/sax/InputSource.html) source)            Unmarshal XML data from the specified SAX InputSource and return the resulting content tree. |
| [Object](http://docs.google.com/java/lang/Object.html) | [**unmarshal**](http://docs.google.com/javax/xml/bind/helpers/AbstractUnmarshallerImpl.html#unmarshal(java.io.InputStream))([InputStream](http://docs.google.com/java/io/InputStream.html) is)            Unmarshal XML data from the specified InputStream and return the resulting content tree. |
| | <T> [JAXBElement](http://docs.google.com/javax/xml/bind/JAXBElement.html)<T> | | --- | | [**unmarshal**](http://docs.google.com/javax/xml/bind/helpers/AbstractUnmarshallerImpl.html#unmarshal(org.w3c.dom.Node,%20java.lang.Class))([Node](http://docs.google.com/org/w3c/dom/Node.html) node, [Class](http://docs.google.com/java/lang/Class.html)<T> expectedType)            Unmarshal XML data by JAXB mapped declaredType and return the resulting content tree. |
| [Object](http://docs.google.com/java/lang/Object.html) | [**unmarshal**](http://docs.google.com/javax/xml/bind/helpers/AbstractUnmarshallerImpl.html#unmarshal(java.io.Reader))([Reader](http://docs.google.com/java/io/Reader.html) reader)            Unmarshal XML data from the specified Reader and return the resulting content tree. |
| [Object](http://docs.google.com/java/lang/Object.html) | [**unmarshal**](http://docs.google.com/javax/xml/bind/helpers/AbstractUnmarshallerImpl.html#unmarshal(javax.xml.transform.Source))([Source](http://docs.google.com/javax/xml/transform/Source.html) source)            Unmarshal XML data from the specified XML Source and return the resulting content tree. |
| | <T> [JAXBElement](http://docs.google.com/javax/xml/bind/JAXBElement.html)<T> | | --- | | [**unmarshal**](http://docs.google.com/javax/xml/bind/helpers/AbstractUnmarshallerImpl.html#unmarshal(javax.xml.transform.Source,%20java.lang.Class))([Source](http://docs.google.com/javax/xml/transform/Source.html) source, [Class](http://docs.google.com/java/lang/Class.html)<T> expectedType)            Unmarshal XML data from the specified XML Source by declaredType and return the resulting content tree. |
| [Object](http://docs.google.com/java/lang/Object.html) | [**unmarshal**](http://docs.google.com/javax/xml/bind/helpers/AbstractUnmarshallerImpl.html#unmarshal(java.net.URL))([URL](http://docs.google.com/java/net/URL.html) url)            Unmarshal XML data from the specified URL and return the resulting content tree. |
| [Object](http://docs.google.com/java/lang/Object.html) | [**unmarshal**](http://docs.google.com/javax/xml/bind/helpers/AbstractUnmarshallerImpl.html#unmarshal(javax.xml.stream.XMLEventReader))([XMLEventReader](http://docs.google.com/javax/xml/stream/XMLEventReader.html) reader)            Unmarshal XML data from the specified pull parser and return the resulting content tree. |
| | <T> [JAXBElement](http://docs.google.com/javax/xml/bind/JAXBElement.html)<T> | | --- | | [**unmarshal**](http://docs.google.com/javax/xml/bind/helpers/AbstractUnmarshallerImpl.html#unmarshal(javax.xml.stream.XMLEventReader,%20java.lang.Class))([XMLEventReader](http://docs.google.com/javax/xml/stream/XMLEventReader.html) reader, [Class](http://docs.google.com/java/lang/Class.html)<T> expectedType)            Unmarshal root element to JAXB mapped declaredType and return the resulting content tree. |
| protected abstract  [Object](http://docs.google.com/java/lang/Object.html) | [**unmarshal**](http://docs.google.com/javax/xml/bind/helpers/AbstractUnmarshallerImpl.html#unmarshal(org.xml.sax.XMLReader,%20org.xml.sax.InputSource))([XMLReader](http://docs.google.com/org/xml/sax/XMLReader.html) reader, [InputSource](http://docs.google.com/org/xml/sax/InputSource.html) source)            Unmarshals an object by using the specified XMLReader and the InputSource. |
| [Object](http://docs.google.com/java/lang/Object.html) | [**unmarshal**](http://docs.google.com/javax/xml/bind/helpers/AbstractUnmarshallerImpl.html#unmarshal(javax.xml.stream.XMLStreamReader))([XMLStreamReader](http://docs.google.com/javax/xml/stream/XMLStreamReader.html) reader)            Unmarshal XML data from the specified pull parser and return the resulting content tree. |
| | <T> [JAXBElement](http://docs.google.com/javax/xml/bind/JAXBElement.html)<T> | | --- | | [**unmarshal**](http://docs.google.com/javax/xml/bind/helpers/AbstractUnmarshallerImpl.html#unmarshal(javax.xml.stream.XMLStreamReader,%20java.lang.Class))([XMLStreamReader](http://docs.google.com/javax/xml/stream/XMLStreamReader.html) reader, [Class](http://docs.google.com/java/lang/Class.html)<T> expectedType)            Unmarshal root element to JAXB mapped declaredType and return the resulting content 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)) |

| **Methods inherited from interface javax.xml.bind.**[**Unmarshaller**](http://docs.google.com/javax/xml/bind/Unmarshaller.html) |
| --- |
| [getUnmarshallerHandler](http://docs.google.com/javax/xml/bind/Unmarshaller.html#getUnmarshallerHandler()), [unmarshal](http://docs.google.com/javax/xml/bind/Unmarshaller.html#unmarshal(org.w3c.dom.Node)) |

| **Field Detail** |
| --- |

### validating

protected boolean **validating**

whether or not the unmarshaller will validate

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

### AbstractUnmarshallerImpl

public **AbstractUnmarshallerImpl**()

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

### getXMLReader

protected [XMLReader](http://docs.google.com/org/xml/sax/XMLReader.html) **getXMLReader**()  
 throws [JAXBException](http://docs.google.com/javax/xml/bind/JAXBException.html)

Obtains a configured XMLReader. This method is used when the client-specified [SAXSource](http://docs.google.com/javax/xml/transform/sax/SAXSource.html) object doesn't have XMLReader. [Unmarshaller](http://docs.google.com/javax/xml/bind/Unmarshaller.html) is not re-entrant, so we will only use one instance of XMLReader.

**Throws:** [JAXBException](http://docs.google.com/javax/xml/bind/JAXBException.html)

### unmarshal

public [Object](http://docs.google.com/java/lang/Object.html) **unmarshal**([Source](http://docs.google.com/javax/xml/transform/Source.html) source)  
 throws [JAXBException](http://docs.google.com/javax/xml/bind/JAXBException.html)

**Description copied from interface:** [**Unmarshaller**](http://docs.google.com/javax/xml/bind/Unmarshaller.html#unmarshal(javax.xml.transform.Source)) Unmarshal XML data from the specified XML Source and return the resulting content tree.

Implements [Unmarshal Global Root Element](#46r0co2).

**SAX 2.0 Parser Pluggability**

A client application can choose not to use the default parser mechanism supplied with their JAXB provider. Any SAX 2.0 compliant parser can be substituted for the JAXB provider's default mechanism. To do so, the client application must properly configure a SAXSource containing an XMLReader implemented by the SAX 2.0 parser provider. If the XMLReader has an org.xml.sax.ErrorHandler registered on it, it will be replaced by the JAXB Provider so that validation errors can be reported via the ValidationEventHandler mechanism of JAXB. If the SAXSource does not contain an XMLReader, then the JAXB provider's default parser mechanism will be used.

This parser replacement mechanism can also be used to replace the JAXB provider's unmarshal-time validation engine. The client application must properly configure their SAX 2.0 compliant parser to perform validation (as shown in the example above). Any SAXParserExceptions encountered by the parser during the unmarshal operation will be processed by the JAXB provider and converted into JAXB ValidationEvent objects which will be reported back to the client via the ValidationEventHandler registered with the Unmarshaller. *Note:* specifying a substitute validating SAX 2.0 parser for unmarshalling does not necessarily replace the validation engine used by the JAXB provider for performing on-demand validation.

The only way for a client application to specify an alternate parser mechanism to be used during unmarshal is via the unmarshal(SAXSource) API. All other forms of the unmarshal method (File, URL, Node, etc) will use the JAXB provider's default parser and validator mechanisms.

**Specified by:**[unmarshal](http://docs.google.com/javax/xml/bind/Unmarshaller.html#unmarshal(javax.xml.transform.Source)) in interface [Unmarshaller](http://docs.google.com/javax/xml/bind/Unmarshaller.html) **Parameters:**source - the XML Source to unmarshal XML data from (providers are only required to support SAXSource, DOMSource, and StreamSource) **Returns:**the newly created root object of the java content 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 Unmarshaller is unable to perform the XML to Java binding. See [Unmarshalling XML Data](#2lwamvv)**See Also:**[Unmarshaller.unmarshal(javax.xml.transform.Source, Class)](http://docs.google.com/javax/xml/bind/Unmarshaller.html#unmarshal(javax.xml.transform.Source,%20java.lang.Class))

### unmarshal

protected abstract [Object](http://docs.google.com/java/lang/Object.html) **unmarshal**([XMLReader](http://docs.google.com/org/xml/sax/XMLReader.html) reader,  
 [InputSource](http://docs.google.com/org/xml/sax/InputSource.html) source)  
 throws [JAXBException](http://docs.google.com/javax/xml/bind/JAXBException.html)

Unmarshals an object by using the specified XMLReader and the InputSource. The callee should call the setErrorHandler method of the XMLReader so that errors are passed to the client-specified ValidationEventHandler.

**Throws:** [JAXBException](http://docs.google.com/javax/xml/bind/JAXBException.html)

### unmarshal

public final [Object](http://docs.google.com/java/lang/Object.html) **unmarshal**([InputSource](http://docs.google.com/org/xml/sax/InputSource.html) source)  
 throws [JAXBException](http://docs.google.com/javax/xml/bind/JAXBException.html)

**Description copied from interface:** [**Unmarshaller**](http://docs.google.com/javax/xml/bind/Unmarshaller.html#unmarshal(org.xml.sax.InputSource)) Unmarshal XML data from the specified SAX InputSource and return the resulting content tree.

Implements [Unmarshal Global Root Element](#46r0co2).

**Specified by:**[unmarshal](http://docs.google.com/javax/xml/bind/Unmarshaller.html#unmarshal(org.xml.sax.InputSource)) in interface [Unmarshaller](http://docs.google.com/javax/xml/bind/Unmarshaller.html) **Parameters:**source - the input source to unmarshal XML data from **Returns:**the newly created root object of the java content 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 Unmarshaller is unable to perform the XML to Java binding. See [Unmarshalling XML Data](#2lwamvv)

### unmarshal

public final [Object](http://docs.google.com/java/lang/Object.html) **unmarshal**([URL](http://docs.google.com/java/net/URL.html) url)  
 throws [JAXBException](http://docs.google.com/javax/xml/bind/JAXBException.html)

**Description copied from interface:** [**Unmarshaller**](http://docs.google.com/javax/xml/bind/Unmarshaller.html#unmarshal(java.net.URL)) Unmarshal XML data from the specified URL and return the resulting content tree.

Implements [Unmarshal Global Root Element](#46r0co2).

**Specified by:**[unmarshal](http://docs.google.com/javax/xml/bind/Unmarshaller.html#unmarshal(java.net.URL)) in interface [Unmarshaller](http://docs.google.com/javax/xml/bind/Unmarshaller.html) **Parameters:**url - the url to unmarshal XML data from **Returns:**the newly created root object of the java content 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 Unmarshaller is unable to perform the XML to Java binding. See [Unmarshalling XML Data](#2lwamvv)

### unmarshal

public final [Object](http://docs.google.com/java/lang/Object.html) **unmarshal**([File](http://docs.google.com/java/io/File.html) f)  
 throws [JAXBException](http://docs.google.com/javax/xml/bind/JAXBException.html)

**Description copied from interface:** [**Unmarshaller**](http://docs.google.com/javax/xml/bind/Unmarshaller.html#unmarshal(java.io.File)) Unmarshal XML data from the specified file and return the resulting content tree.

Implements [Unmarshal Global Root Element](#46r0co2).

**Specified by:**[unmarshal](http://docs.google.com/javax/xml/bind/Unmarshaller.html#unmarshal(java.io.File)) in interface [Unmarshaller](http://docs.google.com/javax/xml/bind/Unmarshaller.html) **Parameters:**f - the file to unmarshal XML data from **Returns:**the newly created root object of the java content 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 Unmarshaller is unable to perform the XML to Java binding. See [Unmarshalling XML Data](#2lwamvv)

### unmarshal

public final [Object](http://docs.google.com/java/lang/Object.html) **unmarshal**([InputStream](http://docs.google.com/java/io/InputStream.html) is)  
 throws [JAXBException](http://docs.google.com/javax/xml/bind/JAXBException.html)

**Description copied from interface:** [**Unmarshaller**](http://docs.google.com/javax/xml/bind/Unmarshaller.html#unmarshal(java.io.InputStream)) Unmarshal XML data from the specified InputStream and return the resulting content tree. Validation event location information may be incomplete when using this form of the unmarshal API.

Implements [Unmarshal Global Root Element](#46r0co2).

**Specified by:**[unmarshal](http://docs.google.com/javax/xml/bind/Unmarshaller.html#unmarshal(java.io.InputStream)) in interface [Unmarshaller](http://docs.google.com/javax/xml/bind/Unmarshaller.html) **Parameters:**is - the InputStream to unmarshal XML data from **Returns:**the newly created root object of the java content 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 Unmarshaller is unable to perform the XML to Java binding. See [Unmarshalling XML Data](#2lwamvv)

### unmarshal

public final [Object](http://docs.google.com/java/lang/Object.html) **unmarshal**([Reader](http://docs.google.com/java/io/Reader.html) reader)  
 throws [JAXBException](http://docs.google.com/javax/xml/bind/JAXBException.html)

**Description copied from interface:** [**Unmarshaller**](http://docs.google.com/javax/xml/bind/Unmarshaller.html#unmarshal(java.io.Reader)) Unmarshal XML data from the specified Reader and return the resulting content tree. Validation event location information may be incomplete when using this form of the unmarshal API, because a Reader does not provide the system ID.

Implements [Unmarshal Global Root Element](#46r0co2).

**Specified by:**[unmarshal](http://docs.google.com/javax/xml/bind/Unmarshaller.html#unmarshal(java.io.Reader)) in interface [Unmarshaller](http://docs.google.com/javax/xml/bind/Unmarshaller.html) **Parameters:**reader - the Reader to unmarshal XML data from **Returns:**the newly created root object of the java content 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 Unmarshaller is unable to perform the XML to Java binding. See [Unmarshalling XML Data](#2lwamvv)

### isValidating

public boolean **isValidating**()  
 throws [JAXBException](http://docs.google.com/javax/xml/bind/JAXBException.html)

Indicates whether or not the Unmarshaller is configured to validate during unmarshal operations.

***Note:*** *I named this method isValidating() to stay in-line with JAXP, as opposed to naming it getValidating().*

**Specified by:**[isValidating](http://docs.google.com/javax/xml/bind/Unmarshaller.html#isValidating()) in interface [Unmarshaller](http://docs.google.com/javax/xml/bind/Unmarshaller.html) **Returns:**true if the Unmarshaller is configured to validate during unmarshal operations, false otherwise **Throws:** [JAXBException](http://docs.google.com/javax/xml/bind/JAXBException.html) - if an error occurs while retrieving the validating flag

### setEventHandler

public 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 validation event handler.

The validation event handler will be called by the JAXB Provider if any validation errors are encountered during calls to any of the unmarshal methods. If the client application does not register a validation event handler before invoking the unmarshal methods, then all validation events will be silently ignored and may result in unexpected behaviour.

**Specified by:**[setEventHandler](http://docs.google.com/javax/xml/bind/Unmarshaller.html#setEventHandler(javax.xml.bind.ValidationEventHandler)) in interface [Unmarshaller](http://docs.google.com/javax/xml/bind/Unmarshaller.html) **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

### setValidating

public void **setValidating**(boolean validating)  
 throws [JAXBException](http://docs.google.com/javax/xml/bind/JAXBException.html)

Specifies whether or not the Unmarshaller should validate during unmarshal operations. By default, the Unmarshaller does not validate.

This method may only be invoked before or after calling one of the unmarshal methods.

**Specified by:**[setValidating](http://docs.google.com/javax/xml/bind/Unmarshaller.html#setValidating(boolean)) in interface [Unmarshaller](http://docs.google.com/javax/xml/bind/Unmarshaller.html) **Parameters:**validating - true if the Unmarshaller should validate during unmarshal, false otherwise **Throws:** [JAXBException](http://docs.google.com/javax/xml/bind/JAXBException.html) - if an error occurred while enabling or disabling validation at unmarshal time

### getEventHandler

public [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.

**Specified by:**[getEventHandler](http://docs.google.com/javax/xml/bind/Unmarshaller.html#getEventHandler()) in interface [Unmarshaller](http://docs.google.com/javax/xml/bind/Unmarshaller.html) **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

### createUnmarshalException

protected [UnmarshalException](http://docs.google.com/javax/xml/bind/UnmarshalException.html) **createUnmarshalException**([SAXException](http://docs.google.com/org/xml/sax/SAXException.html) e)

Creates an UnmarshalException from a SAXException. This is an utility method provided for the derived classes.

When a provider-implemented ContentHandler wants to throw a JAXBException, it needs to wrap the exception by a SAXException. If the unmarshaller implementation blindly wrap SAXException by JAXBException, such an exception will be a JAXBException wrapped by a SAXException wrapped by another JAXBException. This is silly.

This method checks the nested exception of SAXException and reduce those excessive wrapping.

**Returns:**the resulting UnmarshalException

### setProperty

public 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)

Default implementation of the setProperty method always throws PropertyException since there are no required properties. If a provider needs to handle additional properties, it should override this method in a derived class.

**Specified by:**[setProperty](http://docs.google.com/javax/xml/bind/Unmarshaller.html#setProperty(java.lang.String,%20java.lang.Object)) in interface [Unmarshaller](http://docs.google.com/javax/xml/bind/Unmarshaller.html) **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

### getProperty

public [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)

Default implementation of the getProperty method always throws PropertyException since there are no required properties. If a provider needs to handle additional properties, it should override this method in a derived class.

**Specified by:**[getProperty](http://docs.google.com/javax/xml/bind/Unmarshaller.html#getProperty(java.lang.String)) in interface [Unmarshaller](http://docs.google.com/javax/xml/bind/Unmarshaller.html) **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

### unmarshal

public [Object](http://docs.google.com/java/lang/Object.html) **unmarshal**([XMLEventReader](http://docs.google.com/javax/xml/stream/XMLEventReader.html) reader)  
 throws [JAXBException](http://docs.google.com/javax/xml/bind/JAXBException.html)

**Description copied from interface:** [**Unmarshaller**](http://docs.google.com/javax/xml/bind/Unmarshaller.html#unmarshal(javax.xml.stream.XMLEventReader)) Unmarshal XML data from the specified pull parser and return the resulting content tree.

This method is an [Unmarshal Global Root method](#46r0co2).

This method assumes that the parser is on a START\_DOCUMENT or START\_ELEMENT event. Unmarshalling will be done from this start event to the corresponding end event. If this method returns successfully, the reader will be pointing at the token right after the end event.

**Specified by:**[unmarshal](http://docs.google.com/javax/xml/bind/Unmarshaller.html#unmarshal(javax.xml.stream.XMLEventReader)) in interface [Unmarshaller](http://docs.google.com/javax/xml/bind/Unmarshaller.html) **Parameters:**reader - The parser to be read. **Returns:**the newly created root object of the java content 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 Unmarshaller is unable to perform the XML to Java binding. See [Unmarshalling XML Data](#2lwamvv)**See Also:**[Unmarshaller.unmarshal(javax.xml.stream.XMLEventReader, Class)](http://docs.google.com/javax/xml/bind/Unmarshaller.html#unmarshal(javax.xml.stream.XMLEventReader,%20java.lang.Class))

### unmarshal

public [Object](http://docs.google.com/java/lang/Object.html) **unmarshal**([XMLStreamReader](http://docs.google.com/javax/xml/stream/XMLStreamReader.html) reader)  
 throws [JAXBException](http://docs.google.com/javax/xml/bind/JAXBException.html)

**Description copied from interface:** [**Unmarshaller**](http://docs.google.com/javax/xml/bind/Unmarshaller.html#unmarshal(javax.xml.stream.XMLStreamReader)) Unmarshal XML data from the specified pull parser and return the resulting content tree.

Implements [Unmarshal Global Root Element](#46r0co2).

This method assumes that the parser is on a START\_DOCUMENT or START\_ELEMENT event. Unmarshalling will be done from this start event to the corresponding end event. If this method returns successfully, the reader will be pointing at the token right after the end event.

**Specified by:**[unmarshal](http://docs.google.com/javax/xml/bind/Unmarshaller.html#unmarshal(javax.xml.stream.XMLStreamReader)) in interface [Unmarshaller](http://docs.google.com/javax/xml/bind/Unmarshaller.html) **Parameters:**reader - The parser to be read. **Returns:**the newly created root object of the java content 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 Unmarshaller is unable to perform the XML to Java binding. See [Unmarshalling XML Data](#2lwamvv)**See Also:**[Unmarshaller.unmarshal(javax.xml.stream.XMLStreamReader, Class)](http://docs.google.com/javax/xml/bind/Unmarshaller.html#unmarshal(javax.xml.stream.XMLStreamReader,%20java.lang.Class))

### unmarshal

public <T> [JAXBElement](http://docs.google.com/javax/xml/bind/JAXBElement.html)<T> **unmarshal**([Node](http://docs.google.com/org/w3c/dom/Node.html) node,  
 [Class](http://docs.google.com/java/lang/Class.html)<T> expectedType)  
 throws [JAXBException](http://docs.google.com/javax/xml/bind/JAXBException.html)

**Description copied from interface:** [**Unmarshaller**](http://docs.google.com/javax/xml/bind/Unmarshaller.html#unmarshal(org.w3c.dom.Node,%20java.lang.Class)) Unmarshal XML data by JAXB mapped declaredType and return the resulting content tree.

Implements [Unmarshal by Declared Type](#111kx3o)

**Specified by:**[unmarshal](http://docs.google.com/javax/xml/bind/Unmarshaller.html#unmarshal(org.w3c.dom.Node,%20java.lang.Class)) in interface [Unmarshaller](http://docs.google.com/javax/xml/bind/Unmarshaller.html) **Parameters:**node - the document/element to unmarshal XML data from. The caller must support at least Document and Element.expectedType - appropriate JAXB mapped class to hold node's XML data. **Returns:**[JAXB Element](#3l18frh) 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 Unmarshaller is unable to perform the XML to Java binding. See [Unmarshalling XML Data](#2lwamvv)

### unmarshal

public <T> [JAXBElement](http://docs.google.com/javax/xml/bind/JAXBElement.html)<T> **unmarshal**([Source](http://docs.google.com/javax/xml/transform/Source.html) source,  
 [Class](http://docs.google.com/java/lang/Class.html)<T> expectedType)  
 throws [JAXBException](http://docs.google.com/javax/xml/bind/JAXBException.html)

**Description copied from interface:** [**Unmarshaller**](http://docs.google.com/javax/xml/bind/Unmarshaller.html#unmarshal(javax.xml.transform.Source,%20java.lang.Class)) Unmarshal XML data from the specified XML Source by declaredType and return the resulting content tree.

Implements [Unmarshal by Declared Type](#111kx3o)

See [SAX 2.0 Parser Pluggability](#2jxsxqh)

**Specified by:**[unmarshal](http://docs.google.com/javax/xml/bind/Unmarshaller.html#unmarshal(javax.xml.transform.Source,%20java.lang.Class)) in interface [Unmarshaller](http://docs.google.com/javax/xml/bind/Unmarshaller.html) **Parameters:**source - the XML Source to unmarshal XML data from (providers are only required to support SAXSource, DOMSource, and StreamSource)expectedType - appropriate JAXB mapped class to hold source's xml root element **Returns:**Java content rooted by [JAXB Element](#3l18frh) **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 Unmarshaller is unable to perform the XML to Java binding. See [Unmarshalling XML Data](#2lwamvv)

### unmarshal

public <T> [JAXBElement](http://docs.google.com/javax/xml/bind/JAXBElement.html)<T> **unmarshal**([XMLStreamReader](http://docs.google.com/javax/xml/stream/XMLStreamReader.html) reader,  
 [Class](http://docs.google.com/java/lang/Class.html)<T> expectedType)  
 throws [JAXBException](http://docs.google.com/javax/xml/bind/JAXBException.html)

**Description copied from interface:** [**Unmarshaller**](http://docs.google.com/javax/xml/bind/Unmarshaller.html#unmarshal(javax.xml.stream.XMLStreamReader,%20java.lang.Class)) Unmarshal root element to JAXB mapped declaredType and return the resulting content tree.

This method implements [unmarshal by declaredType](#111kx3o).

This method assumes that the parser is on a START\_DOCUMENT or START\_ELEMENT event. Unmarshalling will be done from this start event to the corresponding end event. If this method returns successfully, the reader will be pointing at the token right after the end event.

**Specified by:**[unmarshal](http://docs.google.com/javax/xml/bind/Unmarshaller.html#unmarshal(javax.xml.stream.XMLStreamReader,%20java.lang.Class)) in interface [Unmarshaller](http://docs.google.com/javax/xml/bind/Unmarshaller.html) **Parameters:**reader - The parser to be read.expectedType - appropriate JAXB mapped class to hold reader's START\_ELEMENT XML data. **Returns:**content tree rooted by [JAXB Element representation](#3l18frh) **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 Unmarshaller is unable to perform the XML to Java binding. See [Unmarshalling XML Data](#2lwamvv)

### unmarshal

public <T> [JAXBElement](http://docs.google.com/javax/xml/bind/JAXBElement.html)<T> **unmarshal**([XMLEventReader](http://docs.google.com/javax/xml/stream/XMLEventReader.html) reader,  
 [Class](http://docs.google.com/java/lang/Class.html)<T> expectedType)  
 throws [JAXBException](http://docs.google.com/javax/xml/bind/JAXBException.html)

**Description copied from interface:** [**Unmarshaller**](http://docs.google.com/javax/xml/bind/Unmarshaller.html#unmarshal(javax.xml.stream.XMLEventReader,%20java.lang.Class)) Unmarshal root element to JAXB mapped declaredType and return the resulting content tree.

This method implements [unmarshal by declaredType](#111kx3o).

This method assumes that the parser is on a START\_DOCUMENT or START\_ELEMENT event. Unmarshalling will be done from this start event to the corresponding end event. If this method returns successfully, the reader will be pointing at the token right after the end event.

**Specified by:**[unmarshal](http://docs.google.com/javax/xml/bind/Unmarshaller.html#unmarshal(javax.xml.stream.XMLEventReader,%20java.lang.Class)) in interface [Unmarshaller](http://docs.google.com/javax/xml/bind/Unmarshaller.html) **Parameters:**reader - The parser to be read.expectedType - appropriate JAXB mapped class to hold reader's START\_ELEMENT XML data. **Returns:**content tree rooted by [JAXB Element representation](#3l18frh) **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 Unmarshaller is unable to perform the XML to Java binding. See [Unmarshalling XML Data](#2lwamvv)

### setSchema

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

**Description copied from interface:** [**Unmarshaller**](http://docs.google.com/javax/xml/bind/Unmarshaller.html#setSchema(javax.xml.validation.Schema)) Specify the JAXP 1.3 [Schema](http://docs.google.com/javax/xml/validation/Schema.html) object that should be used to validate subsequent unmarshal operations against. Passing null into this method will disable validation.

This method replaces the deprecated [setValidating(boolean)](http://docs.google.com/javax/xml/bind/Unmarshaller.html#setValidating(boolean)) API.

Initially this property is set to null.

**Specified by:**[setSchema](http://docs.google.com/javax/xml/bind/Unmarshaller.html#setSchema(javax.xml.validation.Schema)) in interface [Unmarshaller](http://docs.google.com/javax/xml/bind/Unmarshaller.html) **Parameters:**schema - Schema object to validate unmarshal operations against or null to disable validation

### getSchema

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

**Description copied from interface:** [**Unmarshaller**](http://docs.google.com/javax/xml/bind/Unmarshaller.html#getSchema()) Get the JAXP 1.3 [Schema](http://docs.google.com/javax/xml/validation/Schema.html) object being used to perform unmarshal-time validation. If there is no Schema set on the unmarshaller, then this method will return null indicating that unmarshal-time validation will not be performed.

This method provides replacement functionality for the deprecated [Unmarshaller.isValidating()](http://docs.google.com/javax/xml/bind/Unmarshaller.html#isValidating()) API as well as access to the Schema object. To determine if the Unmarshaller has validation enabled, simply test the return type for null:

boolean isValidating = u.getSchema()!=null;

**Specified by:**[getSchema](http://docs.google.com/javax/xml/bind/Unmarshaller.html#getSchema()) in interface [Unmarshaller](http://docs.google.com/javax/xml/bind/Unmarshaller.html) **Returns:**the Schema object being used to perform unmarshal-time validation or null if not present

### setAdapter

public void **setAdapter**([XmlAdapter](http://docs.google.com/javax/xml/bind/annotation/adapters/XmlAdapter.html) adapter)

**Description copied from interface:** [**Unmarshaller**](http://docs.google.com/javax/xml/bind/Unmarshaller.html#setAdapter(javax.xml.bind.annotation.adapters.XmlAdapter)) Associates a configured instance of [XmlAdapter](http://docs.google.com/javax/xml/bind/annotation/adapters/XmlAdapter.html) with this unmarshaller.

This is a convenience method that invokes setAdapter(adapter.getClass(),adapter);.

**Specified by:**[setAdapter](http://docs.google.com/javax/xml/bind/Unmarshaller.html#setAdapter(javax.xml.bind.annotation.adapters.XmlAdapter)) in interface [Unmarshaller](http://docs.google.com/javax/xml/bind/Unmarshaller.html) **See Also:**[Unmarshaller.setAdapter(Class,XmlAdapter)](http://docs.google.com/javax/xml/bind/Unmarshaller.html#setAdapter(java.lang.Class,%20A))

### setAdapter

public <A extends [XmlAdapter](http://docs.google.com/javax/xml/bind/annotation/adapters/XmlAdapter.html)> void **setAdapter**([Class](http://docs.google.com/java/lang/Class.html)<A> type,  
 A adapter)

**Description copied from interface:** [**Unmarshaller**](http://docs.google.com/javax/xml/bind/Unmarshaller.html#setAdapter(java.lang.Class,%20A)) Associates a configured instance of [XmlAdapter](http://docs.google.com/javax/xml/bind/annotation/adapters/XmlAdapter.html) with this unmarshaller.

Every unmarshaller internally maintains a [Map](http://docs.google.com/java/util/Map.html)<[Class](http://docs.google.com/java/lang/Class.html),[XmlAdapter](http://docs.google.com/javax/xml/bind/annotation/adapters/XmlAdapter.html)>, which it uses for unmarshalling classes whose fields/methods are annotated with [XmlJavaTypeAdapter](http://docs.google.com/javax/xml/bind/annotation/adapters/XmlJavaTypeAdapter.html).

This method allows applications to use a configured instance of [XmlAdapter](http://docs.google.com/javax/xml/bind/annotation/adapters/XmlAdapter.html). When an instance of an adapter is not given, an unmarshaller will create one by invoking its default constructor.

**Specified by:**[setAdapter](http://docs.google.com/javax/xml/bind/Unmarshaller.html#setAdapter(java.lang.Class,%20A)) in interface [Unmarshaller](http://docs.google.com/javax/xml/bind/Unmarshaller.html) **Parameters:**type - The type of the adapter. The specified instance will be used when [XmlJavaTypeAdapter.value()](http://docs.google.com/javax/xml/bind/annotation/adapters/XmlJavaTypeAdapter.html#value()) refers to this type.adapter - The instance of the adapter to be used. If null, it will un-register the current adapter set for this type.

### getAdapter

public <A extends [XmlAdapter](http://docs.google.com/javax/xml/bind/annotation/adapters/XmlAdapter.html)> A **getAdapter**([Class](http://docs.google.com/java/lang/Class.html)<A> type)

**Description copied from interface:** [**Unmarshaller**](http://docs.google.com/javax/xml/bind/Unmarshaller.html#getAdapter(java.lang.Class)) Gets the adapter associated with the specified type. This is the reverse operation of the [Unmarshaller.setAdapter(javax.xml.bind.annotation.adapters.XmlAdapter)](http://docs.google.com/javax/xml/bind/Unmarshaller.html#setAdapter(javax.xml.bind.annotation.adapters.XmlAdapter)) method.

**Specified by:**[getAdapter](http://docs.google.com/javax/xml/bind/Unmarshaller.html#getAdapter(java.lang.Class)) in interface [Unmarshaller](http://docs.google.com/javax/xml/bind/Unmarshaller.html)

### setAttachmentUnmarshaller

public void **setAttachmentUnmarshaller**([AttachmentUnmarshaller](http://docs.google.com/javax/xml/bind/attachment/AttachmentUnmarshaller.html) au)

**Description copied from interface:** [**Unmarshaller**](http://docs.google.com/javax/xml/bind/Unmarshaller.html#setAttachmentUnmarshaller(javax.xml.bind.attachment.AttachmentUnmarshaller))

Associate a context that resolves cid's, content-id URIs, to binary data passed as attachments.

Unmarshal time validation, enabled via [Unmarshaller.setSchema(Schema)](http://docs.google.com/javax/xml/bind/Unmarshaller.html#setSchema(javax.xml.validation.Schema)), must be supported even when unmarshaller is performing XOP processing.

**Specified by:**[setAttachmentUnmarshaller](http://docs.google.com/javax/xml/bind/Unmarshaller.html#setAttachmentUnmarshaller(javax.xml.bind.attachment.AttachmentUnmarshaller)) in interface [Unmarshaller](http://docs.google.com/javax/xml/bind/Unmarshaller.html)

### getAttachmentUnmarshaller

public [AttachmentUnmarshaller](http://docs.google.com/javax/xml/bind/attachment/AttachmentUnmarshaller.html) **getAttachmentUnmarshaller**()

**Specified by:**[getAttachmentUnmarshaller](http://docs.google.com/javax/xml/bind/Unmarshaller.html#getAttachmentUnmarshaller()) in interface [Unmarshaller](http://docs.google.com/javax/xml/bind/Unmarshaller.html)

### setListener

public void **setListener**([Unmarshaller.Listener](http://docs.google.com/javax/xml/bind/Unmarshaller.Listener.html) listener)

**Description copied from interface:** [**Unmarshaller**](http://docs.google.com/javax/xml/bind/Unmarshaller.html#setListener(javax.xml.bind.Unmarshaller.Listener))

Register unmarshal event callback [Unmarshaller.Listener](http://docs.google.com/javax/xml/bind/Unmarshaller.Listener.html) with this [Unmarshaller](http://docs.google.com/javax/xml/bind/Unmarshaller.html).

There is only one Listener per Unmarshaller. Setting a Listener replaces the previous set Listener. One can unregister current Listener by setting listener to null.

**Specified by:**[setListener](http://docs.google.com/javax/xml/bind/Unmarshaller.html#setListener(javax.xml.bind.Unmarshaller.Listener)) in interface [Unmarshaller](http://docs.google.com/javax/xml/bind/Unmarshaller.html) **Parameters:**listener - provides unmarshal event callbacks for this [Unmarshaller](http://docs.google.com/javax/xml/bind/Unmarshaller.html)

### getListener

public [Unmarshaller.Listener](http://docs.google.com/javax/xml/bind/Unmarshaller.Listener.html) **getListener**()

**Description copied from interface:** [**Unmarshaller**](http://docs.google.com/javax/xml/bind/Unmarshaller.html#getListener())

Return [Unmarshaller.Listener](http://docs.google.com/javax/xml/bind/Unmarshaller.Listener.html) registered with this [Unmarshaller](http://docs.google.com/javax/xml/bind/Unmarshaller.html).

**Specified by:**[getListener](http://docs.google.com/javax/xml/bind/Unmarshaller.html#getListener()) in interface [Unmarshaller](http://docs.google.com/javax/xml/bind/Unmarshaller.html) **Returns:**registered [Unmarshaller.Listener](http://docs.google.com/javax/xml/bind/Unmarshaller.Listener.html) or null if no Listener is registered with this Unmarshaller.

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/AbstractUnmarshallerImpl.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/helpers/AbstractMarshallerImpl.html)   [**NEXT CLASS**](http://docs.google.com/javax/xml/bind/helpers/DefaultValidationEventHandler.html) | [**FRAMES**](http://docs.google.com/index.html?javax/xml/bind/helpers/AbstractUnmarshallerImpl.html)    [**NO FRAMES**](http://docs.google.com/AbstractUnmarshallerImpl.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#tyjcwt) | [CONSTR](#3dy6vkm) | [METHOD](#1t3h5sf) | DETAIL: [FIELD](#17dp8vu) | [CONSTR](#26in1rg) | [METHOD](#35nkun2) |

[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).