| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/JAXBContext.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/Element.html)   [**NEXT CLASS**](http://docs.google.com/javax/xml/bind/JAXBElement.html) | [**FRAMES**](http://docs.google.com/index.html?javax/xml/bind/JAXBContext.html)    [**NO FRAMES**](http://docs.google.com/JAXBContext.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#2et92p0) | [CONSTR](#tyjcwt) | [METHOD](#3dy6vkm) | DETAIL: [FIELD](#4d34og8) | [CONSTR](#17dp8vu) | [METHOD](#26in1rg) |

## **javax.xml.bind**

Class JAXBContext

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

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

The JAXBContext class provides the client's entry point to the JAXB API. It provides an abstraction for managing the XML/Java binding information necessary to implement the JAXB binding framework operations: unmarshal, marshal and validate.

A client application normally obtains new instances of this class using one of these two styles for newInstance methods, although there are other specialized forms of the method available:

* [JAXBContext.newInstance( "com.acme.foo:com.acme.bar" )](http://docs.google.com/javax/xml/bind/JAXBContext.html#newInstance(java.lang.String,%20java.lang.ClassLoader))  
  The JAXBContext instance is initialized from a list of colon separated Java package names. Each java package contains JAXB mapped classes, schema-derived classes and/or user annotated classes. Additionally, the java package may contain JAXB package annotations that must be processed. (see JLS 3rd Edition, Section 7.4.1. Package Annotations).
* [JAXBContext.newInstance( com.acme.foo.Foo.class )](http://docs.google.com/javax/xml/bind/JAXBContext.html#newInstance(java.lang.Class...))  
  The JAXBContext instance is intialized with class(es) passed as parameter(s) and classes that are statically reachable from these class(es). See [newInstance(Class...)](http://docs.google.com/javax/xml/bind/JAXBContext.html#newInstance(java.lang.Class...)) for details.

***SPEC REQUIREMENT:*** *the provider must supply an implementation class containing the following method signatures:*

public static JAXBContext createContext( String contextPath, ClassLoader classLoader, Map properties ) throws JAXBException  
 public static JAXBContext createContext( Class[] classes, Map properties ) throws JAXBException

*The following JAXB 1.0 requirement is only required for schema to java interface/implementation binding. It does not apply to JAXB annotated classes. JAXB Providers must generate a jaxb.properties file in each package containing schema derived classes. The property file must contain a property named javax.xml.bind.context.factory whose value is the name of the class that implements the createContext APIs.*

*The class supplied by the provider does not have to be assignable to javax.xml.bind.JAXBContext, it simply has to provide a class that implements the createContext APIs.*

*In addition, the provider must call the* [*DatatypeConverter.setDatatypeConverter*](http://docs.google.com/javax/xml/bind/DatatypeConverter.html#setDatatypeConverter(javax.xml.bind.DatatypeConverterInterface)) *api prior to any client invocations of the marshal and unmarshal methods. This is necessary to configure the datatype converter that will be used during these operations.*

**Unmarshalling**

The [Unmarshaller](http://docs.google.com/javax/xml/bind/Unmarshaller.html) class provides the client application the ability to convert XML data into a tree of Java content objects. The unmarshal method allows for any global XML element declared in the schema to be unmarshalled as the root of an instance document. Additionally, the unmarshal method allows for an unrecognized root element that has an xsi:type attribute's value that references a type definition declared in the schema to be unmarshalled as the root of an instance document. The JAXBContext object allows the merging of global elements and type definitions across a set of schemas (listed in the contextPath). Since each schema in the schema set can belong to distinct namespaces, the unification of schemas to an unmarshalling context should be namespace independent. This means that a client application is able to unmarshal XML documents that are instances of any of the schemas listed in the contextPath. For example:

JAXBContext jc = JAXBContext.newInstance( "com.acme.foo:com.acme.bar" );  
 Unmarshaller u = jc.createUnmarshaller();  
 FooObject fooObj = (FooObject)u.unmarshal( new File( "foo.xml" ) ); // ok  
 BarObject barObj = (BarObject)u.unmarshal( new File( "bar.xml" ) ); // ok  
 BazObject bazObj = (BazObject)u.unmarshal( new File( "baz.xml" ) ); // error, "com.acme.baz" not in contextPath

The client application may also generate Java content trees explicitly rather than unmarshalling existing XML data. For all JAXB-annotated value classes, an application can create content using constructors. For schema-derived interface/implementation classes and for the creation of elements that are not bound to a JAXB-annotated class, an application needs to have access and knowledge about each of the schema derived ObjectFactory classes that exist in each of java packages contained in the contextPath. For each schema derived java class, there is a static factory method that produces objects of that type. For example, assume that after compiling a schema, you have a package com.acme.foo that contains a schema derived interface named PurchaseOrder. In order to create objects of that type, the client application would use the factory method like this:

com.acme.foo.PurchaseOrder po =   
 com.acme.foo.ObjectFactory.createPurchaseOrder();

Once the client application has an instance of the the schema derived object, it can use the mutator methods to set content on it.

For more information on the generated ObjectFactory classes, see Section 4.2 *Java Package* of the specification.

***SPEC REQUIREMENT:*** *the provider must generate a class in each package that contains all of the necessary object factory methods for that package named ObjectFactory as well as the static newInstance( javaContentInterface ) method*

**Marshalling**

The [Marshaller](http://docs.google.com/javax/xml/bind/Marshaller.html) class provides the client application the ability to convert a Java content tree back into XML data. There is no difference between marshalling a content tree that is created manually using the factory methods and marshalling a content tree that is the result an unmarshal operation. Clients can marshal a java content tree back to XML data to a java.io.OutputStream or a java.io.Writer. The marshalling process can alternatively produce SAX2 event streams to a registered ContentHandler or produce a DOM Node object. Client applications have control over the output encoding as well as whether or not to marshal the XML data as a complete document or as a fragment.

Here is a simple example that unmarshals an XML document and then marshals it back out:

JAXBContext jc = JAXBContext.newInstance( "com.acme.foo" );  
  
 // unmarshal from foo.xml  
 Unmarshaller u = jc.createUnmarshaller();  
 FooObject fooObj = (FooObject)u.unmarshal( new File( "foo.xml" ) );  
  
 // marshal to System.out  
 Marshaller m = jc.createMarshaller();  
 m.marshal( fooObj, System.out );

**Validation**

Validation has been changed significantly since JAXB 1.0. The [Validator](http://docs.google.com/javax/xml/bind/Validator.html) class has been deprecated and made optional. This means that you are advised not to use this class and, in fact, it may not even be available depending on your JAXB provider. JAXB 1.0 client applications that rely on Validator will still work properly when deployed with the JAXB 1.0 runtime system. In JAXB 2.0, the [Unmarshaller](http://docs.google.com/javax/xml/bind/Unmarshaller.html) has included convenince methods that expose the JAXP 1.3 [javax.xml.validation](http://docs.google.com/javax/xml/validation/package-summary.html) framework. Please refer to the [Unmarshaller.setSchema(javax.xml.validation.Schema)](http://docs.google.com/javax/xml/bind/Unmarshaller.html#setSchema(javax.xml.validation.Schema)) API for more information.

**JAXB Runtime Binding Framework Compatibility**

The following JAXB 1.0 restriction only applies to binding schema to interfaces/implementation classes. Since this binding does not require a common runtime system, a JAXB client application must not attempt to mix runtime objects (JAXBContext, Marshaller, etc. ) from different providers. This does not mean that the client application isn't portable, it simply means that a client has to use a runtime system provided by the same provider that was used to compile the schema.

**Since:** JAXB1.0 **See Also:**[Marshaller](http://docs.google.com/javax/xml/bind/Marshaller.html), [Unmarshaller](http://docs.google.com/javax/xml/bind/Unmarshaller.html), [S 7.4.1.1 "Package Annotations" in Java Language Specification, 3rd Edition](http://java.sun.com/docs/books/jls)

| **Field Summary** | |
| --- | --- |
| static [String](http://docs.google.com/java/lang/String.html) | [**JAXB\_CONTEXT\_FACTORY**](http://docs.google.com/javax/xml/bind/JAXBContext.html#JAXB_CONTEXT_FACTORY)            The name of the property that contains the name of the class capable of creating new JAXBContext objects. |

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

| **Method Summary** | |
| --- | --- |
| [Binder](http://docs.google.com/javax/xml/bind/Binder.html)<[Node](http://docs.google.com/org/w3c/dom/Node.html)> | [**createBinder**](http://docs.google.com/javax/xml/bind/JAXBContext.html#createBinder())()            Creates a Binder for W3C DOM. |
| | <T> [Binder](http://docs.google.com/javax/xml/bind/Binder.html)<T> | | --- | | [**createBinder**](http://docs.google.com/javax/xml/bind/JAXBContext.html#createBinder(java.lang.Class))([Class](http://docs.google.com/java/lang/Class.html)<T> domType)            Creates a Binder object that can be used for associative/in-place unmarshalling/marshalling. |
| [JAXBIntrospector](http://docs.google.com/javax/xml/bind/JAXBIntrospector.html) | [**createJAXBIntrospector**](http://docs.google.com/javax/xml/bind/JAXBContext.html#createJAXBIntrospector())()            Creates a JAXBIntrospector object that can be used to introspect JAXB objects. |
| abstract  [Marshaller](http://docs.google.com/javax/xml/bind/Marshaller.html) | [**createMarshaller**](http://docs.google.com/javax/xml/bind/JAXBContext.html#createMarshaller())()            Create a Marshaller object that can be used to convert a java content tree into XML data. |
| abstract  [Unmarshaller](http://docs.google.com/javax/xml/bind/Unmarshaller.html) | [**createUnmarshaller**](http://docs.google.com/javax/xml/bind/JAXBContext.html#createUnmarshaller())()            Create an Unmarshaller object that can be used to convert XML data into a java content tree. |
| abstract  [Validator](http://docs.google.com/javax/xml/bind/Validator.html) | [**createValidator**](http://docs.google.com/javax/xml/bind/JAXBContext.html#createValidator())()  **Deprecated.** *since JAXB2.0* |
| void | [**generateSchema**](http://docs.google.com/javax/xml/bind/JAXBContext.html#generateSchema(javax.xml.bind.SchemaOutputResolver))([SchemaOutputResolver](http://docs.google.com/javax/xml/bind/SchemaOutputResolver.html) outputResolver)            Generates the schema documents for this context. |
| static [JAXBContext](http://docs.google.com/javax/xml/bind/JAXBContext.html) | [**newInstance**](http://docs.google.com/javax/xml/bind/JAXBContext.html#newInstance(java.lang.Class...))([Class](http://docs.google.com/java/lang/Class.html)... classesToBeBound)             Obtain a new instance of a JAXBContext class. |
| static [JAXBContext](http://docs.google.com/javax/xml/bind/JAXBContext.html) | [**newInstance**](http://docs.google.com/javax/xml/bind/JAXBContext.html#newInstance(java.lang.Class%5B%5D,%20java.util.Map))([Class](http://docs.google.com/java/lang/Class.html)[] classesToBeBound, [Map](http://docs.google.com/java/util/Map.html)<[String](http://docs.google.com/java/lang/String.html),?> properties)             Obtain a new instance of a JAXBContext class. |
| static [JAXBContext](http://docs.google.com/javax/xml/bind/JAXBContext.html) | [**newInstance**](http://docs.google.com/javax/xml/bind/JAXBContext.html#newInstance(java.lang.String))([String](http://docs.google.com/java/lang/String.html) contextPath)             Obtain a new instance of a JAXBContext class. |
| static [JAXBContext](http://docs.google.com/javax/xml/bind/JAXBContext.html) | [**newInstance**](http://docs.google.com/javax/xml/bind/JAXBContext.html#newInstance(java.lang.String,%20java.lang.ClassLoader))([String](http://docs.google.com/java/lang/String.html) contextPath, [ClassLoader](http://docs.google.com/java/lang/ClassLoader.html) classLoader)             Obtain a new instance of a JAXBContext class. |
| static [JAXBContext](http://docs.google.com/javax/xml/bind/JAXBContext.html) | [**newInstance**](http://docs.google.com/javax/xml/bind/JAXBContext.html#newInstance(java.lang.String,%20java.lang.ClassLoader,%20java.util.Map))([String](http://docs.google.com/java/lang/String.html) contextPath, [ClassLoader](http://docs.google.com/java/lang/ClassLoader.html) classLoader, [Map](http://docs.google.com/java/util/Map.html)<[String](http://docs.google.com/java/lang/String.html),?> properties)             Obtain a new instance of a JAXBContext class. |

| **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)) |

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

### JAXB\_CONTEXT\_FACTORY

public static final [String](http://docs.google.com/java/lang/String.html) **JAXB\_CONTEXT\_FACTORY**

The name of the property that contains the name of the class capable of creating new JAXBContext objects.

**See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#javax.xml.bind.JAXBContext.JAXB_CONTEXT_FACTORY)

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

### JAXBContext

protected **JAXBContext**()

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

### newInstance

public static [JAXBContext](http://docs.google.com/javax/xml/bind/JAXBContext.html) **newInstance**([String](http://docs.google.com/java/lang/String.html) contextPath)  
 throws [JAXBException](http://docs.google.com/javax/xml/bind/JAXBException.html)

Obtain a new instance of a JAXBContext class.

This is a convenience method for the [newInstance](http://docs.google.com/javax/xml/bind/JAXBContext.html#newInstance(java.lang.String,%20java.lang.ClassLoader)) method. It uses the context class loader of the current thread. To specify the use of a different class loader, either set it via the Thread.setContextClassLoader() api or use the [newInstance](http://docs.google.com/javax/xml/bind/JAXBContext.html#newInstance(java.lang.String,%20java.lang.ClassLoader)) method.

**Throws:** [JAXBException](http://docs.google.com/javax/xml/bind/JAXBException.html) - if an error was encountered while creating the JAXBContext such as

1. failure to locate either ObjectFactory.class or jaxb.index in the packages
2. an ambiguity among global elements contained in the contextPath
3. failure to locate a value for the context factory provider property
4. mixing schema derived packages from different providers on the same contextPath

### newInstance

public static [JAXBContext](http://docs.google.com/javax/xml/bind/JAXBContext.html) **newInstance**([String](http://docs.google.com/java/lang/String.html) contextPath,  
 [ClassLoader](http://docs.google.com/java/lang/ClassLoader.html) classLoader)  
 throws [JAXBException](http://docs.google.com/javax/xml/bind/JAXBException.html)

Obtain a new instance of a JAXBContext class.

The client application must supply a context path which is a list of colon (':', \u003A) separated java package names that contain schema-derived classes and/or fully qualified JAXB-annotated classes. Schema-derived code is registered with the JAXBContext by the ObjectFactory.class generated per package. Alternatively than being listed in the context path, programmer annotated JAXB mapped classes can be listed in a jaxb.index resource file, format described below. Note that a java package can contain both schema-derived classes and user annotated JAXB classes. Additionally, the java package may contain JAXB package annotations that must be processed. (see JLS 3rd Edition, Section 7.4.1. "Package Annotations").

Every package listed on the contextPath must meet **one or both** of the following conditions otherwise a JAXBException will be thrown:

1. it must contain ObjectFactory.class
2. it must contain jaxb.index

**Format for jaxb.index**

The file contains a newline-separated list of class names. Space and tab characters, as well as blank lines, are ignored. The comment character is '#' (0x23); on each line all characters following the first comment character are ignored. The file must be encoded in UTF-8. Classes that are reachable, as defined in [newInstance(Class...)](http://docs.google.com/javax/xml/bind/JAXBContext.html#newInstance(java.lang.Class...)), from the listed classes are also registered with JAXBContext.

Constraints on class name occuring in a jaxb.index file are:

* Must not end with ".class".
* Class names are resolved relative to package containing jaxb.index file. Only classes occuring directly in package containing jaxb.index file are allowed.
* Fully qualified class names are not allowed. A qualified class name,relative to current package, is only allowed to specify a nested or inner class.

To maintain compatibility with JAXB 1.0 schema to java interface/implementation binding, enabled by schema customization , the JAXB provider will ensure that each package on the context path has a jaxb.properties file which contains a value for the javax.xml.bind.context.factory property and that all values resolve to the same provider. This requirement does not apply to JAXB annotated classes.

If there are any global XML element name collisions across the various packages listed on the contextPath, a JAXBException will be thrown.

Mixing generated interface/impl bindings from multiple JAXB Providers in the same context path may result in a JAXBException being thrown.

**Parameters:**contextPath - list of java package names that contain schema derived class and/or java to schema (JAXB-annotated) mapped classesclassLoader - This class loader will be used to locate the implementation classes. **Returns:**a new instance of a JAXBContext **Throws:** [JAXBException](http://docs.google.com/javax/xml/bind/JAXBException.html) - if an error was encountered while creating the JAXBContext such as

1. failure to locate either ObjectFactory.class or jaxb.index in the packages
2. an ambiguity among global elements contained in the contextPath
3. failure to locate a value for the context factory provider property
4. mixing schema derived packages from different providers on the same contextPath

### newInstance

public static [JAXBContext](http://docs.google.com/javax/xml/bind/JAXBContext.html) **newInstance**([String](http://docs.google.com/java/lang/String.html) contextPath,  
 [ClassLoader](http://docs.google.com/java/lang/ClassLoader.html) classLoader,  
 [Map](http://docs.google.com/java/util/Map.html)<[String](http://docs.google.com/java/lang/String.html),?> properties)  
 throws [JAXBException](http://docs.google.com/javax/xml/bind/JAXBException.html)

Obtain a new instance of a JAXBContext class.

This is mostly the same as [newInstance(String, ClassLoader)](http://docs.google.com/javax/xml/bind/JAXBContext.html#newInstance(java.lang.String,%20java.lang.ClassLoader)), but this version allows you to pass in provider-specific properties to configure the instanciation of [JAXBContext](http://docs.google.com/javax/xml/bind/JAXBContext.html).

The interpretation of properties is up to implementations.

**Parameters:**contextPath - list of java package names that contain schema derived classesclassLoader - This class loader will be used to locate the implementation classes.properties - provider-specific properties **Returns:**a new instance of a JAXBContext **Throws:** [JAXBException](http://docs.google.com/javax/xml/bind/JAXBException.html) - if an error was encountered while creating the JAXBContext such as

1. failure to locate either ObjectFactory.class or jaxb.index in the packages
2. an ambiguity among global elements contained in the contextPath
3. failure to locate a value for the context factory provider property
4. mixing schema derived packages from different providers on the same contextPath

**Since:** JAXB2.0

### newInstance

public static [JAXBContext](http://docs.google.com/javax/xml/bind/JAXBContext.html) **newInstance**([Class](http://docs.google.com/java/lang/Class.html)... classesToBeBound)  
 throws [JAXBException](http://docs.google.com/javax/xml/bind/JAXBException.html)

Obtain a new instance of a JAXBContext class.

The client application must supply a list of classes that the new context object needs to recognize. Not only the new context will recognize all the classes specified, but it will also recognize any classes that are directly/indirectly referenced statically from the specified classes. Subclasses of referenced classes nor @XmlTransient referenced classes are not registered with JAXBContext. For example, in the following Java code, if you do newInstance(Foo.class), the newly created [JAXBContext](http://docs.google.com/javax/xml/bind/JAXBContext.html) will recognize both Foo and Bar, but not Zot or FooBar:

class Foo {  
 @XmlTransient FooBar c;  
 Bar b;  
 }  
 class Bar { int x; }  
 class Zot extends Bar { int y; }  
 class FooBar { }

Therefore, a typical client application only needs to specify the top-level classes, but it needs to be careful.

Note that for each java package registered with JAXBContext, when the optional package annotations exist, they must be processed. (see JLS 3rd Edition, Section 7.4.1. "Package Annotations").

**Parameters:**classesToBeBound - list of java classes to be recognized by the new [JAXBContext](http://docs.google.com/javax/xml/bind/JAXBContext.html). Can be empty, in which case a [JAXBContext](http://docs.google.com/javax/xml/bind/JAXBContext.html) that only knows about spec-defined classes will be returned. **Returns:**A new instance of a JAXBContext. Always non-null valid object. **Throws:** [JAXBException](http://docs.google.com/javax/xml/bind/JAXBException.html) - if an error was encountered while creating the JAXBContext, such as (but not limited to):

1. No JAXB implementation was discovered
2. Classes use JAXB annotations incorrectly
3. Classes have colliding annotations (i.e., two classes with the same type name)
4. The JAXB implementation was unable to locate provider-specific out-of-band information (such as additional files generated at the development time.)

[IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if the parameter contains null (i.e., newInstance(null);)**Since:** JAXB2.0

### newInstance

public static [JAXBContext](http://docs.google.com/javax/xml/bind/JAXBContext.html) **newInstance**([Class](http://docs.google.com/java/lang/Class.html)[] classesToBeBound,  
 [Map](http://docs.google.com/java/util/Map.html)<[String](http://docs.google.com/java/lang/String.html),?> properties)  
 throws [JAXBException](http://docs.google.com/javax/xml/bind/JAXBException.html)

Obtain a new instance of a JAXBContext class.

An overloading of [newInstance(Class...)](http://docs.google.com/javax/xml/bind/JAXBContext.html#newInstance(java.lang.Class...)) to configure 'properties' for this instantiation of [JAXBContext](http://docs.google.com/javax/xml/bind/JAXBContext.html).

The interpretation of properties is implementation specific.

**Parameters:**classesToBeBound - list of java classes to be recognized by the new [JAXBContext](http://docs.google.com/javax/xml/bind/JAXBContext.html). Can be empty, in which case a [JAXBContext](http://docs.google.com/javax/xml/bind/JAXBContext.html) that only knows about spec-defined classes will be returned. **Returns:**A new instance of a JAXBContext. Always non-null valid object. **Throws:** [JAXBException](http://docs.google.com/javax/xml/bind/JAXBException.html) - if an error was encountered while creating the JAXBContext, such as (but not limited to):

1. No JAXB implementation was discovered
2. Classes use JAXB annotations incorrectly
3. Classes have colliding annotations (i.e., two classes with the same type name)
4. The JAXB implementation was unable to locate provider-specific out-of-band information (such as additional files generated at the development time.)

[IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if the parameter contains null (i.e., newInstance(null);)**Since:** JAXB2.0

### createUnmarshaller

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

Create an Unmarshaller object that can be used to convert XML data into a java content tree.

**Returns:**an Unmarshaller object **Throws:** [JAXBException](http://docs.google.com/javax/xml/bind/JAXBException.html) - if an error was encountered while creating the Unmarshaller object

### createMarshaller

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

Create a Marshaller object that can be used to convert a java content tree into XML data.

**Returns:**a Marshaller object **Throws:** [JAXBException](http://docs.google.com/javax/xml/bind/JAXBException.html) - if an error was encountered while creating the Marshaller object

### createValidator

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

**Deprecated.** *since JAXB2.0*

[Validator](http://docs.google.com/javax/xml/bind/Validator.html) has been made optional and deprecated in JAXB 2.0. Please refer to the javadoc for [Validator](http://docs.google.com/javax/xml/bind/Validator.html) for more detail.

Create a Validator object that can be used to validate a java content tree against its source schema.

**Returns:**a Validator object **Throws:** [JAXBException](http://docs.google.com/javax/xml/bind/JAXBException.html) - if an error was encountered while creating the Validator object

### createBinder

public <T> [Binder](http://docs.google.com/javax/xml/bind/Binder.html)<T> **createBinder**([Class](http://docs.google.com/java/lang/Class.html)<T> domType)

Creates a Binder object that can be used for associative/in-place unmarshalling/marshalling.

**Parameters:**domType - select the DOM API to use by passing in its DOM Node class. **Returns:**always a new valid Binder object. **Throws:** [UnsupportedOperationException](http://docs.google.com/java/lang/UnsupportedOperationException.html) - if DOM API corresponding to domType is not supported by the implementation.**Since:** JAXB2.0

### createBinder

public [Binder](http://docs.google.com/javax/xml/bind/Binder.html)<[Node](http://docs.google.com/org/w3c/dom/Node.html)> **createBinder**()

Creates a Binder for W3C DOM.

**Returns:**always a new valid Binder object.**Since:** JAXB2.0

### createJAXBIntrospector

public [JAXBIntrospector](http://docs.google.com/javax/xml/bind/JAXBIntrospector.html) **createJAXBIntrospector**()

Creates a JAXBIntrospector object that can be used to introspect JAXB objects.

**Returns:**always return a non-null valid JAXBIntrospector object. **Throws:** [UnsupportedOperationException](http://docs.google.com/java/lang/UnsupportedOperationException.html) - Calling this method on JAXB 1.0 implementations will throw an UnsupportedOperationException.**Since:** JAXB2.0

### generateSchema

public void **generateSchema**([SchemaOutputResolver](http://docs.google.com/javax/xml/bind/SchemaOutputResolver.html) outputResolver)  
 throws [IOException](http://docs.google.com/java/io/IOException.html)

Generates the schema documents for this context.

**Parameters:**outputResolver - this object controls the output to which schemas will be sent. **Throws:** [IOException](http://docs.google.com/java/io/IOException.html) - if [SchemaOutputResolver](http://docs.google.com/javax/xml/bind/SchemaOutputResolver.html) throws an [IOException](http://docs.google.com/java/io/IOException.html). [UnsupportedOperationException](http://docs.google.com/java/lang/UnsupportedOperationException.html) - Calling this method on JAXB 1.0 implementations will throw an UnsupportedOperationException.**Since:** JAXB 2.0

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/JAXBContext.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/Element.html)   [**NEXT CLASS**](http://docs.google.com/javax/xml/bind/JAXBElement.html) | [**FRAMES**](http://docs.google.com/index.html?javax/xml/bind/JAXBContext.html)    [**NO FRAMES**](http://docs.google.com/JAXBContext.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#2et92p0) | [CONSTR](#tyjcwt) | [METHOD](#3dy6vkm) | DETAIL: [FIELD](#4d34og8) | [CONSTR](#17dp8vu) | [METHOD](#26in1rg) |

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

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

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