| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/Validator.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/ValidationException.html)   NEXT CLASS | [**FRAMES**](http://docs.google.com/index.html?javax/xml/bind/Validator.html)    [**NO FRAMES**](http://docs.google.com/Validator.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | CONSTR | [METHOD](#1t3h5sf) | DETAIL: FIELD | CONSTR | [METHOD](#4d34og8) |

## **javax.xml.bind**

Interface Validator

**Deprecated.** *since JAXB 2.0*

public interface **Validator**

As of JAXB 2.0, this class is deprecated and optional.

The Validator class is responsible for controlling the validation of content trees during runtime.

**Three Forms of Validation**

**Unmarshal-Time Validation** This form of validation enables a client application to receive information about validation errors and warnings detected while unmarshalling XML data into a Java content tree and is completely orthogonal to the other types of validation. To enable or disable it, see the javadoc for [Unmarshaller.setValidating](http://docs.google.com/javax/xml/bind/Unmarshaller.html#setValidating(boolean)). All JAXB 1.0 Providers are required to support this operation. **On-Demand Validation** This form of validation enables a client application to receive information about validation errors and warnings detected in the Java content tree. At any point, client applications can call the [Validator.validate](http://docs.google.com/javax/xml/bind/Validator.html#validate(java.lang.Object)) method on the Java content tree (or any sub-tree of it). All JAXB 1.0 Providers are required to support this operation. **Fail-Fast Validation** This form of validation enables a client application to receive immediate feedback about modifications to the Java content tree that violate type constraints on Java Properties as defined in the specification. JAXB Providers are not required support this type of validation. Of the JAXB Providers that do support this type of validation, some may require you to decide at schema compile time whether or not a client application will be allowed to request fail-fast validation at runtime.

The Validator class is responsible for managing On-Demand Validation. The Unmarshaller class is responsible for managing Unmarshal-Time Validation during the unmarshal operations. Although there is no formal method of enabling validation during the marshal operations, the Marshaller may detect errors, which will be reported to the ValidationEventHandler registered on it.

**Using the Default EventHandler**

If the client application does not set an event handler on their Validator, Unmarshaller, or Marshaller prior to calling the validate, unmarshal, or marshal methods, then a default event handler will receive notification of any errors or warnings encountered. The default event handler will cause the current operation to halt after encountering the first error or fatal error (but will attempt to continue after receiving warnings).

**Handling Validation Events**

There are three ways to handle events encountered during the unmarshal, validate, and marshal operations: Use the default event handler The default event handler will be used if you do not specify one via the setEventHandler API's on Validator, Unmarshaller, or Marshaller. Implement and register a custom event handler Client applications that require sophisticated event processing can implement the ValidationEventHandler interface and register it with the Unmarshaller and/or Validator. Use the [ValidationEventCollector](http://docs.google.com/javax/xml/bind/util/ValidationEventCollector.html) utility For convenience, a specialized event handler is provided that simply collects any ValidationEvent objects created during the unmarshal, validate, and marshal operations and returns them to the client application as a java.util.Collection.

**Validation and Well-Formedness**

Validation events are handled differently depending on how the client application is configured to process them as described in the previous section. However, there are certain cases where a JAXB Provider indicates that it is no longer able to reliably detect and report errors. In these cases, the JAXB Provider will set the severity of the ValidationEvent to FATAL\_ERROR to indicate that the unmarshal, validate, or marshal operations should be terminated. The default event handler and ValidationEventCollector utility class must terminate processing after being notified of a fatal error. Client applications that supply their own ValidationEventHandler should also terminate processing after being notified of a fatal error. If not, unexpected behaviour may occur.

**Supported Properties**

There currently are not any properties required to be supported by all JAXB Providers on Validator. However, some providers may support their own set of provider specific properties.

**Since:** JAXB1.0 **See Also:**[JAXBContext](http://docs.google.com/javax/xml/bind/JAXBContext.html), [Unmarshaller](http://docs.google.com/javax/xml/bind/Unmarshaller.html), [ValidationEventHandler](http://docs.google.com/javax/xml/bind/ValidationEventHandler.html), [ValidationEvent](http://docs.google.com/javax/xml/bind/ValidationEvent.html), [ValidationEventCollector](http://docs.google.com/javax/xml/bind/util/ValidationEventCollector.html)

| **Method Summary** | |
| --- | --- |
| [ValidationEventHandler](http://docs.google.com/javax/xml/bind/ValidationEventHandler.html) | [**getEventHandler**](http://docs.google.com/javax/xml/bind/Validator.html#getEventHandler())()  **Deprecated.** *since JAXB2.0* |
| [Object](http://docs.google.com/java/lang/Object.html) | [**getProperty**](http://docs.google.com/javax/xml/bind/Validator.html#getProperty(java.lang.String))([String](http://docs.google.com/java/lang/String.html) name)  **Deprecated.** *since JAXB2.0* |
| void | [**setEventHandler**](http://docs.google.com/javax/xml/bind/Validator.html#setEventHandler(javax.xml.bind.ValidationEventHandler))([ValidationEventHandler](http://docs.google.com/javax/xml/bind/ValidationEventHandler.html) handler)  **Deprecated.** *since JAXB2.0* |
| void | [**setProperty**](http://docs.google.com/javax/xml/bind/Validator.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)  **Deprecated.** *since JAXB2.0* |
| boolean | [**validate**](http://docs.google.com/javax/xml/bind/Validator.html#validate(java.lang.Object))([Object](http://docs.google.com/java/lang/Object.html) subrootObj)  **Deprecated.** *since JAXB2.0* |
| boolean | [**validateRoot**](http://docs.google.com/javax/xml/bind/Validator.html#validateRoot(java.lang.Object))([Object](http://docs.google.com/java/lang/Object.html) rootObj)  **Deprecated.** *since JAXB2.0* |

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

### setEventHandler

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

**Deprecated.** *since JAXB2.0*

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 [validate](http://docs.google.com/javax/xml/bind/Validator.html#validate(java.lang.Object)). If the client application does not register a validation event handler before invoking the validate method, then validation events will be handled by the default event handler which will terminate the validate operation after the first error or fatal error is encountered.

Calling this method with a null parameter will cause the Validator 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

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

**Deprecated.** *since JAXB2.0*

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

### validate

boolean **validate**([Object](http://docs.google.com/java/lang/Object.html) subrootObj)  
 throws [JAXBException](http://docs.google.com/javax/xml/bind/JAXBException.html)

**Deprecated.** *since JAXB2.0*

Validate the Java content tree starting at subrootObj.

Client applications can use this method to validate Java content trees on-demand at runtime. This method can be used to validate any arbitrary subtree of the Java content tree. Global constraint checking **will not**  be performed as part of this operation (i.e. ID/IDREF constraints).

**Parameters:**subrootObj - the obj to begin validation at **Returns:**true if the subtree rooted at subrootObj is valid, false otherwise **Throws:** [JAXBException](http://docs.google.com/javax/xml/bind/JAXBException.html) - if any unexpected problem occurs during validation [ValidationException](http://docs.google.com/javax/xml/bind/ValidationException.html) - If the [ValidationEventHandler](http://docs.google.com/javax/xml/bind/ValidationEventHandler.html) returns false from its handleEvent method or the Validator is unable to validate the content tree rooted at subrootObj [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - If the subrootObj parameter is null

### validateRoot

boolean **validateRoot**([Object](http://docs.google.com/java/lang/Object.html) rootObj)  
 throws [JAXBException](http://docs.google.com/javax/xml/bind/JAXBException.html)

**Deprecated.** *since JAXB2.0*

Validate the Java content tree rooted at rootObj.

Client applications can use this method to validate Java content trees on-demand at runtime. This method is used to validate an entire Java content tree. Global constraint checking **will** be performed as part of this operation (i.e. ID/IDREF constraints).

**Parameters:**rootObj - the root obj to begin validation at **Returns:**true if the tree rooted at rootObj is valid, false otherwise **Throws:** [JAXBException](http://docs.google.com/javax/xml/bind/JAXBException.html) - if any unexpected problem occurs during validation [ValidationException](http://docs.google.com/javax/xml/bind/ValidationException.html) - If the [ValidationEventHandler](http://docs.google.com/javax/xml/bind/ValidationEventHandler.html) returns false from its handleEvent method or the Validator is unable to validate the content tree rooted at rootObj [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - If the rootObj parameter is null

### setProperty

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)

**Deprecated.** *since JAXB2.0*

Set the particular property in the underlying implementation of Validator. This method can only be used to set one of the standard JAXB defined properties above or a provider specific property. Attempting to set an undefined property will result in a PropertyException being thrown. See  [Supported Properties](#3dy6vkm).

**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

[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)

**Deprecated.** *since JAXB2.0*

Get the particular property in the underlying implementation of Validator. This method can only be used to get one of the standard JAXB defined properties above or a provider specific property. Attempting to get an undefined property will result in a PropertyException being thrown. See  [Supported Properties](#3dy6vkm).

**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/Validator.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/ValidationException.html)   NEXT CLASS | [**FRAMES**](http://docs.google.com/index.html?javax/xml/bind/Validator.html)    [**NO FRAMES**](http://docs.google.com/Validator.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | CONSTR | [METHOD](#1t3h5sf) | DETAIL: FIELD | CONSTR | [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).