| | [**Overview**](http://docs.google.com/overview-summary.html) | **Package** | Class | [**Use**](http://docs.google.com/package-use.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 PACKAGE**](http://docs.google.com/javax/xml/transform/stream/package-summary.html)   [**NEXT PACKAGE**](http://docs.google.com/javax/xml/ws/package-summary.html) | [**FRAMES**](http://docs.google.com/index.html?javax/xml/validation/package-summary.html)    [**NO FRAMES**](http://docs.google.com/package-summary.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |

## Package javax.xml.validation

This package provides an API for validation of XML documents.

**See:**

[**Description**](#3znysh7)

| **Class Summary** | |
| --- | --- |
| [**Schema**](http://docs.google.com/javax/xml/validation/Schema.html) | Immutable in-memory representation of grammar. |
| [**SchemaFactory**](http://docs.google.com/javax/xml/validation/SchemaFactory.html) | Factory that creates [Schema](http://docs.google.com/javax/xml/validation/Schema.html) objects. Entry-point to the validation API. |
| [**SchemaFactoryLoader**](http://docs.google.com/javax/xml/validation/SchemaFactoryLoader.html) | Factory that creates [SchemaFactory](http://docs.google.com/javax/xml/validation/SchemaFactory.html). |
| [**TypeInfoProvider**](http://docs.google.com/javax/xml/validation/TypeInfoProvider.html) | This class provides access to the type information determined by [ValidatorHandler](http://docs.google.com/javax/xml/validation/ValidatorHandler.html). |
| [**Validator**](http://docs.google.com/javax/xml/validation/Validator.html) | A processor that checks an XML document against [Schema](http://docs.google.com/javax/xml/validation/Schema.html). |
| [**ValidatorHandler**](http://docs.google.com/javax/xml/validation/ValidatorHandler.html) | Streaming validator that works on SAX stream. |

## Package javax.xml.validation Description

This package provides an API for validation of XML documents. *Validation* is the process of verifying that an XML document is an instance of a specified XML *schema*. An XML schema defines the content model (also called a *grammar* or *vocabulary*) that its instance documents will represent.

There are a number of popular technologies available for creating an XML schema. Some of the most popular include:

* **Document Type Definition (DTD)** - XML's built-in schema language.
* [**W3C XML Schema (WXS)**](http://www.w3.org/XML/Schema) - an object-oriented XML schema language. WXS also provides a type system for constraining the character data of an XML document. WXS is maintained by the [World Wide Web Consortium (W3C)](http://www.w3.org) and is a W3C Recommendation (that is, a ratified W3C standard specification).
* [**RELAX NG (RNG)**](http://www.relaxng.org) - a pattern-based, user-friendly XML schema language. RNG schemas may also use types to constrain XML character data. RNG is maintained by the [Organization for the Advancement of Structured Information Standards (OASIS)](http://www.oasis-open.org) and is both an OASIS and an [ISO (International Organization for Standardization)](http://www.iso.org) standard.
* [**Schematron**](http://www.schematron.com/) - a rules-based XML schema language. Whereas DTD, WXS, and RNG are designed to express the structure of a content model, Schematron is designed to enforce individual rules that are difficult or impossible to express with other schema languages. Schematron is intended to supplement a schema written in structural schema language such as the aforementioned. Schematron is in the process of becoming an ISO standard.

Previous versions of JAXP supported validation as a feature of an XML parser, represented by either a [SAXParser](http://docs.google.com/javax/xml/parsers/SAXParser.html) or [DocumentBuilder](http://docs.google.com/javax/xml/parsers/DocumentBuilder.html) instance.

The JAXP validation API decouples the validation of an instance document from the parsing of an XML document. This is advantageous for several reasons, some of which are:

* **Support for additional schema langauges.** As of JDK 1.5, the two most popular JAXP parser implementations, Crimson and Xerces, only support a subset of the available XML schema languages. The Validation API provides a standard mechanism through which applications may take of advantage of specialization validation libraries which support additional schema languages.
* **Easy runtime coupling of an XML instance and schema.** Specifying the location of a schema to use for validation with JAXP parsers can be confusing. The Validation API makes this process simple (see [example](#2et92p0) below).

**Usage example**. The following example demonstrates validating an XML document with the Validation API (for readability, some exception handling is not shown):

// parse an XML document into a DOM tree  
 DocumentBuilder parser = DocumentBuilderFactory.newInstance().newDocumentBuilder();  
 Document document = parser.parse(new File("instance.xml"));  
  
 // create a SchemaFactory capable of understanding WXS schemas  
 SchemaFactory factory = SchemaFactory.newInstance(XMLConstants.W3C\_XML\_SCHEMA\_NS\_URI);  
  
 // load a WXS schema, represented by a Schema instance  
 Source schemaFile = new StreamSource(new File("mySchema.xsd"));  
 Schema schema = factory.newSchema(schemaFile);  
  
 // create a Validator instance, which can be used to validate an instance document  
 Validator validator = schema.newValidator();  
  
 // validate the DOM tree  
 try {  
 validator.validate(new DOMSource(document));  
 } catch (SAXException e) {  
 // instance document is invalid!  
 }

The JAXP parsing API has been integrated with the Validation API. Applications may create a [Schema](http://docs.google.com/javax/xml/validation/Schema.html) with the validation API and associate it with a [DocumentBuilderFactory](http://docs.google.com/javax/xml/parsers/DocumentBuilderFactory.html) or a [SAXParserFactory](http://docs.google.com/javax/xml/parsers/SAXParserFactory.html) instance by using the [DocumentBuilderFactory.setSchema(Schema)](http://docs.google.com/javax/xml/parsers/DocumentBuilderFactory.html#setSchema(javax.xml.validation.Schema)) and [SAXParserFactory.setSchema(Schema)](http://docs.google.com/javax/xml/parsers/SAXParserFactory.html#setSchema(javax.xml.validation.Schema)) methods. **You should not** both set a schema and call setValidating(true) on a parser factory. The former technique will cause parsers to use the new validation API; the latter will cause parsers to use their own internal validation facilities. **Turning on both of these options simultaneously will cause either redundant behavior or error conditions.**

| | [**Overview**](http://docs.google.com/overview-summary.html) | **Package** | Class | [**Use**](http://docs.google.com/package-use.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 PACKAGE**](http://docs.google.com/javax/xml/transform/stream/package-summary.html)   [**NEXT PACKAGE**](http://docs.google.com/javax/xml/ws/package-summary.html) | [**FRAMES**](http://docs.google.com/index.html?javax/xml/validation/package-summary.html)    [**NO FRAMES**](http://docs.google.com/package-summary.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |

[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).