Table of Contents

Java in a Nutshell, 5th Edition By David Flanagan ISBN: 0596007736 Publisher: O'Reilly
Print Publication Date: 2005/03/01
Prepared for Ronald Fischer, Safari ID: ronald.fischer@fusshuhn.de
User number: 628024 Copyright 2008, Safari Books Online, LLC.
This PDF is exclusively for your use in accordance with the Safari Terms of Service. No part of it may be reproduced or transmitted in any form by any means without the prior written permission for reprints and excerpts from the publisher. Redistribution or other use that violates the fair use priviledge under U.S. copyright laws (see 17 USC107) or that otherwise violates the Safari Terms of Service is strictly prohibited.

Java in a Nutshell, 5th Edition By David Flanagan ISBN: 0596007736 Publisher: O'Reilly
Print Publication Date: 2005/03/01
Prepared for Ronald Fischer, Safari ID: ronald.fischer@fusshuhn.de
User number: 628024 Copyright 2008, Safari Books Online, LLC.
This PDF is exclusively for your use in accordance with the Safari Terms of Service. No part of it may be reproduced or transmitted in any form by any means without the prior written permission for reprints and excerpts from the publisher. Redistribution or other use that violates the fair use priviledge under U.S. copyright laws (see 17 USC107) or that otherwise violates the Safari Terms of Service is strictly prohibited.

Chapter 20. javax.xml and Subpackages

This chapter documents `javax.xml` and its subpackages:

`java.xml`

This simple package simply defines constants for use by its subpackages. Added in Java 5.0.

Copyright Safari Books Online #628024

`javax.xml.datatype`

This package contains Java types corresponding to types defined by XML standards such as W3C XML Schema, XQuery, and XPath.

`javax.xml.namespace`

This package defines types for working with XML namespaces.

`javax.xml.parsers`

This package defines parser classes that serve as a wrapper around underlying DOM and SAX XML parsers, and also defines factory classes that are used to obtain instances of those parser classes.

`javax.xml.transform`

This package defines classes and interfaces for transforming the representation and content of an XML document with XSLT. It defines `Source` and `Result` interfaces to represent a source document and a result document. subpackages provide implementations of these classes that represent documents in different ways.

`javax.xml.transform.dom`

This package implements the `Source` and `Result` interfaces that represent documents as DOM document trees.

Java in a Nutshell, 5th Edition By David Flanagan ISBN: 0596007736 Publisher: O'Reilly
Print Publication Date: 2005/03/01
Prepared for Ronald Fischer, Safari ID: ronald.fischer@fusshuhn.de
User number: 628024 Copyright 2008, Safari Books Online, LLC.
This PDF is exclusively for your use in accordance with the Safari Terms of Service. No part of it may be reproduced or transmitted in any form by any means without the prior written permission for reprints and excerpts from the publisher. Redistribution or other use that violates the fair use priviledge under U.S. copyright laws (see 17 USC107) or that otherwise violates the Safari Terms of Service is strictly prohibited.

`javax.xml.transform.sax`

This package implements the `Source` and `Result` interfaces to represent documents as sequences of SAX parser events. It also defines other SAX-related transformation classes.

`javax.xml.transform.stream`

This package implements the `Source` and `Result` interfaces that represent documents as streams of text.

`javax.xml.validation`

This package contains classes for validating XML documents against a schema.

`javax.xml.xpath`

This package defines types for the evaluation of XPath expressions in the context of an XML document.

Package javax.xml

Java 5.0

This package has many important subpackages but defines only a single class `XMLConstants`, which, as its name implies, provides symbolic names for constants defined by various XML specifications.

Classes

```
public final class XMLConstants;
```

XMLConstants javax.xml

Java 5.0

This class is a repository for constants defined by various XML standards. Most are URIs that identify XML namespaces.

Java in a Nutshell, 5th Edition By David Flanagan ISBN: 0596007736 Publisher: O'Reilly
Print Publication Date: 2005/03/01
Prepared for Ronald Fischer, Safari ID: ronald.fischer@fusshuhn.de
User number: 628024 Copyright 2008, Safari Books Online, LLC.
This PDF is exclusively for your use in accordance with the Safari Terms of Service. No part of it may be reproduced or transmitted in any form by any means without the prior written permission for reprints and excerpts from the publisher. Redistribution or other use that violates the fair use priviledge under U.S. copyright laws (see 17 USC107) or that otherwise violates the Safari Terms of Service is strictly prohibited.

```
public final class XMLConstants {
// No Constructor
// Public Constants
     public static final String DEFAULT_NS_PREFIX; =" "
     public static final String FEATURE_SECURE_PROCESSING;
      ="http://javax.xml.XMLConstants/feature/secure-processing"
     public static final String NULL_NS_URI;    =" "
     public static final String RELAXNG_NS_URI; ="http://relaxng.org/ns/structure/1.0"
     public static final String W3C_XML_SCHEMA_INSTANCE_NS_URI;
     ="http://www.w3.org/2001/XMLSchema-instance"
     public static final String W3C_XML_SCHEMA_NS_URI; ="http://www.w3.org/2001/XMLSchema"
     public static final String W3C_XPATH_DATATYPE_NS_URI;
      ="http://www.w3.org/2003/11/xpath-datatypes"
     public static final String XML_DTD_NS_URI;      ="http://www.w3.org/TR/REC-xml"
     public static final String XML_NS_PREFIX;       ="xml"
     public static final String XML_NS_URI;          ="http://www.w3.org/XML/1998/namespace"
     public static final String XMLNS_ATTRIBUTE;     ="xmlns"
     public static final String XMLNS_ATTRIBUTE_NS_URI; ="http://www.w3.org/2000/xmlns/"
}
```

Package javax.xml.datatype

Java 5.0

This package defines Java data types that correspond to certain time, date, and duration data types required by the W3C XML Schema, XQuery, and XPath standards. This package is of primary interest to those implementing schema validators and XPath evaluators and should not be required by applications that use schemas or XPath expressions.

Classes

```
public final class DatatypeConstants;
public static final class DatatypeConstants.Field;
public abstract class DatatypeFactory;
public abstract class Duration;
public abstract class XMLGregorianCalendar implements Cloneable;
```

Exceptions

```
public class DatatypeConfigurationException extends Exception;
```

DatatypeConfigurationException javax.xml.datatype

Java 5.0 ***serializable checked***

An exception of this type is thrown by `DatatypeFactory.newInstance( )` to indicate a factory configuration error.

Java in a Nutshell, 5th Edition By David Flanagan ISBN: 0596007736 Publisher: O'Reilly
Print Publication Date: 2005/03/01
Prepared for Ronald Fischer, Safari ID: ronald.fischer@fusshuhn.de
User number: 628024 Copyright 2008, Safari Books Online, LLC.
This PDF is exclusively for your use in accordance with the Safari Terms of Service. No part of it may be reproduced or transmitted in any form by any means without the prior written permission for reprints and excerpts from the publisher. Redistribution or other use that violates the fair use priviledge under U.S. copyright laws (see 17 USC107) or that otherwise violates the Safari Terms of Service is strictly prohibited.

Figure 20-1. javax.xml.datatype.DatatypeConfigurationException


```
public class DatatypeConfigurationException extends Exception {
// Public Constructors
     public DatatypeConfigurationException( );
     public DatatypeConfigurationException(Throwable cause);
     public DatatypeConfigurationException(String message);
     public DatatypeConfigurationException(String message, Throwable cause);
}
```

Thrown By

`DatatypeFactory.newInstance( )`

DatatypeConstants javax.xml.datatype

Java 5.0

This class defines constants used in this package. Most of the constants are `int` values, but some are qualified names and some are instances of the `DatatypeConstants.Field` type.

```
public final class DatatypeConstants {
// No Constructor
// Public Constants
     public static final int APRIL;                                    =4
     public static final int AUGUST;                                   =8
     public static final javax.xml.namespace.QName DATE;
     public static final javax.xml.namespace.QName DATETIME;
     public static final DatatypeConstants.Field DAYS;
     public static final int DECEMBER;                                 =12
     public static final javax.xml.namespace.QName DURATION;
     public static final javax.xml.namespace.QName DURATION_DAYTIME;
     public static final javax.xml.namespace.QName DURATION_YEARMONTH;
     public static final int EQUAL;                                    =0
     public static final int FEBRUARY;                                 =2
     public static final int FIELD_UNDEFINED;                          =-2147483648
     public static final javax.xml.namespace.QName GDAY;
     public static final javax.xml.namespace.QName GMONTH;
     public static final javax.xml.namespace.QName GMONTHDAY;
     public static final int GREATER;                                  =1
     public static final javax.xml.namespace.QName GYEAR;
     public static final javax.xml.namespace.QName GYEARMONTH;
     public static final DatatypeConstants.Field HOURS;
     public static final int INDETERMINATE;                            =2
     public static final int JANUARY;                                  =1
     public static final int JULY;                                     =7
     public static final int JUNE;                                     =6
     public static final int LESSER;                                   =-1
     public static final int MARCH;                                    =3
     public static final int MAX_TIMEZONE_OFFSET;                      =-840
     public static final int MAY;                                      =5
     public static final int MIN_TIMEZONE_OFFSET;                      =840
     public static final DatatypeConstants.Field MINUTES;
     public static final DatatypeConstants.Field MONTHS;
     public static final int NOVEMBER;                                 =11
     public static final int OCTOBER;                                  =10
```


Java in a Nutshell, 5th Edition By David Flanagan ISBN: 0596007736 Publisher: O'Reilly
Print Publication Date: 2005/03/01
Prepared for Ronald Fischer, Safari ID: ronald.fischer@fusshuhn.de
User number: 628024 Copyright 2008, Safari Books Online, LLC.
This PDF is exclusively for your use in accordance with the Safari Terms of Service. No part of it may be reproduced or transmitted in any form by any means without the prior written permission for reprints and excerpts from the publisher. Redistribution or other use that violates the fair use priviledge under U.S. copyright laws (see 17 USC107) or that otherwise violates the Safari Terms of Service is strictly prohibited.

```
     public static final DatatypeConstants.Field SECONDS;
     public static final int SEPTEMBER;                                =9
     public static final javax.xml.namespace.QName TIME;
     public static final DatatypeConstants.Field YEARS;
// Nested Types
     public static final class Field;
}
```

DatatypeConstants.Field javax.xml.datatype

Java 5.0

This class defines a typesafe enumeration for some of the constants in `DatatypeConstants`. Note that it is a class, not a Java 5.0 `enum` type.

```
public static final class DatatypeConstants.Field {
// No Constructor
// Public Instance Methods
     public int getId( );
// Public Methods Overriding Object
     public String toString( );
}
```

Passed To

`Duration.{getField( ),isSet( )}`

Type Of

`DatatypeConstants.{DAYS,HOURS,MINUTES,MONTHS,SECONDS,YEARS}`

DatatypeFactory javax.xml.datatype

Java 5.0

This class defines factory methods for creating `Duration` and `XMLGregorianCalendar` objects.

```
public abstract class DatatypeFactory {
// Protected Constructors
     protected DatatypeFactory( );
// Public Constants
     public static final String DATATYPEFACTORY_IMPLEMENTATION_CLASS;
      ="com.sun.org.apache.xerces.internal.jaxp.datatype.DatatypeFactoryImpl"
     public static final String DATATYPEFACTORY_PROPERTY;
      ="javax.xml.datatype.DatatypeFactory"
// Public Class Methods
     public static DatatypeFactory newInstance( ) throws DatatypeConfigurationException;
// Public Instance Methods
     public abstract Duration newDuration(String lexicalRepresentation);
     public abstract Duration newDuration(long durationInMilliSeconds);
     public Duration newDuration(boolean isPositive, int years, int months,
        int days, int hours,
```


Java in a Nutshell, 5th Edition By David Flanagan ISBN: 0596007736 Publisher: O'Reilly
Print Publication Date: 2005/03/01
Prepared for Ronald Fischer, Safari ID: ronald.fischer@fusshuhn.de
User number: 628024 Copyright 2008, Safari Books Online, LLC.
This PDF is exclusively for your use in accordance with the Safari Terms of Service. No part of it may be reproduced or transmitted in any form by any means without the prior written permission for reprints and excerpts from the publisher. Redistribution or other use that violates the fair use priviledge under U.S. copyright laws (see 17 USC107) or that otherwise violates the Safari Terms of Service is strictly prohibited.

```
        int minutes, int seconds);
    public abstract Duration newDuration(boolean isPositive,
        java.math.BigInteger years, java.math.BigInteger months,
        java.math.BigInteger days, java.math.BigInteger hours,
        java.math.BigInteger minutes, java.math.BigDecimal seconds);
    public Duration newDurationDayTime(long durationInMilliseconds);
    public Duration newDurationDayTime(String lexicalRepresentation);
    public Duration newDurationDayTime(boolean isPositive, int day, int hour,
        int minute, int second);
    public Duration newDurationDayTime(boolean isPositive,
        java.math.BigInteger day, java.math.BigInteger hour,
        java.math.BigInteger minute, java.math.BigInteger second);
    public Duration newDurationYearMonth(long durationInMilliseconds);
    public Duration newDurationYearMonth(String lexicalRepresentation);
    public Duration newDurationYearMonth(boolean isPositive, int year, int month);
    public Duration newDurationYearMonth(boolean isPositive,
        java.math.BigInteger year, java.math.BigInteger month);
    public abstract XMLGregorianCalendar newXMLGregorianCalendar( );
    public abstract XMLGregorianCalendar newXMLGregorianCalendar
        (java.util.GregorianCalendar cal);
    public abstract XMLGregorianCalendar newXMLGregorianCalendar(String lexicalRepresentation);
    public XMLGregorianCalendar newXMLGregorianCalendar(int year, int month,
        int day, int hour,
        int minute, int second,
        int millisecond, int timezone);
    public abstract XMLGregorianCalendar newXMLGregorianCalendar
        (java.math.BigInteger year, int month,
        int day, int hour, int minute,
        int second,
        java.math.BigDecimal fractionalSecond,
        int timezone);
    public XMLGregorianCalendar newXMLGregorianCalendarDate(int year, int month,
        int day, int timezone);
    public XMLGregorianCalendar newXMLGregorianCalendarTime(int hours, int minutes,
        int seconds, int timezone);
    public XMLGregorianCalendar newXMLGregorianCalendarTime(int hours, int minutes,
        int seconds, int milliseconds,
        int timezone);
    public XMLGregorianCalendar newXMLGregorianCalendarTime(int hours, int minutes,
        int seconds, java.math.BigDecimal fractionalSecond,
        int timezone);
}
```

Duration javax.xml.datatype

Java 5.0

An instance of this class represents a length of time. Create `Duration` objects with `DatatypeFactory`.

```
public abstract class Duration {
// Public Constructors
    public Duration( );
// Public Instance Methods
    public abstract Duration add(Duration rhs);
    public abstract void addTo(java.util.Calendar calendar);
    public void addTo(java.util.Date date);
    public abstract int compare(Duration duration);
    public int getDays( );
    public abstract Number getField(DatatypeConstants.Field field);
    public int getHours( );
    public int getMinutes( );
```


Java in a Nutshell, 5th Edition By David Flanagan ISBN: 0596007736 Publisher: O'Reilly
Print Publication Date: 2005/03/01
Prepared for Ronald Fischer, Safari ID: ronald.fischer@fusshuhn.de
User number: 628024 Copyright 2008, Safari Books Online, LLC.
This PDF is exclusively for your use in accordance with the Safari Terms of Service. No part of it may be reproduced or transmitted in any form by any means without the prior written permission for reprints and excerpts from the publisher. Redistribution or other use that violates the fair use priviledge under U.S. copyright laws (see 17 USC107) or that otherwise violates the Safari Terms of Service is strictly prohibited.

```
        public int getMonths( );
        public int getSeconds( );
        public abstract int getSign( );
        public long getTimeInMillis(java.util.Date startInstant);
        public long getTimeInMillis(java.util.Calendar startInstant);
        public javax.xml.namespace.QName getXMLSchemaType( );
        public int getYears( );
        public boolean isLongerThan(Duration duration);
        public abstract boolean isSet(DatatypeConstants.Field field);
        public boolean isShorterThan(Duration duration);
        public Duration multiply(int factor);
        public abstract Duration multiply(java.math.BigDecimal factor);
        public abstract Duration negate( );
        public abstract Duration normalizeWith(java.util.Calendar startTimeInstant);
        public Duration subtract(Duration rhs);
    // Public Methods Overriding Object
        public boolean equals(Object duration);
        public abstract int hashCode( );
        public String toString( );
    }
```

Passed To

`XMLGregorianCalendar.add( )`

Returned By

`DatatypeFactory.{newDuration( ),newDurationDayTime( ), newDurationYearMonth( )}`

XMLGregorianCalendar javax.xml.datatype

Java 5.0 ***cloneable***

Instances of this class represent a date or time. Create `XMLGregorianCalendar` objects with a `DatatypeFactory`.

Figure 20-2. javax.xml.datatype.XMLGregorianCalendar


```
public abstract class XMLGregorianCalendar implements Cloneable {
// Public Constructors
     public XMLGregorianCalendar( );
// Public Instance Methods
     public abstract void add(Duration duration);
     public abstract void clear( );
     public abstract int compare(XMLGregorianCalendar xmlGregorianCalendar);
     public abstract int getDay( );
     public abstract java.math.BigInteger getEon( );
     public abstract java.math.BigInteger getEonAndYear( );
     public abstract java.math.BigDecimal getFractionalSecond( );
     public abstract int getHour( );
     public int getMillisecond( );
     public abstract int getMinute( );
     public abstract int getMonth( );
     public abstract int getSecond( );
     public abstract int getTimezone( );
     public abstract java.util.TimeZone getTimeZone(int defaultZoneoffset);
     public abstract javax.xml.namespace.QName getXMLSchemaType( );
```


Java in a Nutshell, 5th Edition By David Flanagan ISBN: 0596007736 Publisher: O'Reilly
Print Publication Date: 2005/03/01
Prepared for Ronald Fischer, Safari ID: ronald.fischer@fusshuhn.de
User number: 628024 Copyright 2008, Safari Books Online, LLC.
This PDF is exclusively for your use in accordance with the Safari Terms of Service. No part of it may be reproduced or transmitted in any form by any means without the prior written permission for reprints and excerpts from the publisher. Redistribution or other use that violates the fair use priviledge under U.S. copyright laws (see 17 USC107) or that otherwise violates the Safari Terms of Service is strictly prohibited.

```
        public abstract int getYear( );
        public abstract boolean isValid( );
        public abstract XMLGregorianCalendar normalize( );
        public abstract void reset( );
        public abstract void setDay(int day);
        public abstract void setFractionalSecond(java.math.BigDecimal fractional);
        public abstract void setHour(int hour);
        public abstract void setMillisecond(int millisecond);
        public abstract void setMinute(int minute);
        public abstract void setMonth(int month);
        public abstract void setSecond(int second);
        public void setTime(int hour, int minute, int second);
        public void setTime(int hour, int minute, int second, int millisecond);
        public void setTime(int hour, int minute, int second,
           java.math.BigDecimal fractional);
        public abstract void setTimezone(int offset);
        public abstract void setYear(int year);
        public abstract void setYear(java.math.BigInteger year);
        public abstract java.util.GregorianCalendar toGregorianCalendar( );
        public abstract java.util.GregorianCalendar toGregorianCalendar
           (java.util.TimeZone timezone, java.util.Locale aLocale,
           XMLGregorianCalendar defaults);
        public abstract String toXMLFormat( );
    // Public Methods Overriding Object
        public abstract Object clone( );
        public boolean equals(Object obj);
        public int hashCode( );
        public String toString( );
    }
```

Returned By

`DatatypeFactory.{newXMLGregorianCalendar( ), newXMLGregorianCalendarDate( ),newXMLGregorianCalendarTime( )}`

Package javax.xml.namespace

Java 5.0

This small package defines types for working with XML namespaces. `NamespaceContext` represents a mapping between namespace URIs and namespace prefixes. `QName` represents a qualified name consisting of a local part and a namespace.

Interfaces

```
public interface NamespaceContext;
```

Classes

```
public class QName implements Serializable;
```

NamespaceContext javax.xml.namespace

Java in a Nutshell, 5th Edition By David Flanagan ISBN: 0596007736 Publisher: O'Reilly
Print Publication Date: 2005/03/01
Prepared for Ronald Fischer, Safari ID: ronald.fischer@fusshuhn.de
User number: 628024 Copyright 2008, Safari Books Online, LLC.
This PDF is exclusively for your use in accordance with the Safari Terms of Service. No part of it may be reproduced or transmitted in any form by any means without the prior written permission for reprints and excerpts from the publisher. Redistribution or other use that violates the fair use priviledge under U.S. copyright laws (see 17 USC107) or that otherwise violates the Safari Terms of Service is strictly prohibited.

Java 5.0

This interface represents a mapping between namespace URIs and the local prefixes that are bound to them. Use `getNamepaceURI( )` to obtain the URI that a prefix is bound to. Use `getPrefix( )` to do the reverse. More than one prefix can be bound to the same URI, and the `getPrefixes( )` method returns an `Iterator` that you can use to loop through all prefixes that have been associated with a given URI.

```
public interface NamespaceContext {
// Public Instance Methods
     String getNamespaceURI(String prefix);
     String getPrefix(String namespaceURI);
     java.util.Iterator getPrefixes(String namespaceURI);
}
```

Passed To

`javax.xml.xpath.XPath.setNamespaceContext( )`

Returned By

`javax.xml.xpath.XPath.getNamespaceContext( )`

QName — javax.xml.namespace

Java 5.0 — ***serializable***

A `QName` represents an XML "qualified name," such as an XML element name that has both a local name and a namespace. `getLocalPart( )` returns the unqualified local part of the name. `getNamespaceURI( )` returns the canonical URI that formally identifies the namespace. `getPrefix( )` returns the locally declared namespace prefix. Note that a `QName` does not always have a prefix and that the prefix, if it exists, is ignored for the purposes of the `equals( )`, `hashCode( )`, and `toString( )` methods. The static `valueOf( )` method parses a `QName` from a string in the format of `toString( )`:

```
{namespaceURI}localPart
```

javax.xml.namespace.QName


```
public class QName implements Serializable {
// Public Constructors
     public QName(String localPart);
     public QName(String namespaceURI, String localPart);
     public QName(String namespaceURI, String localPart, String prefix);
// Public Class Methods
     public static QName valueOf(String qNameAsString);
```


Java in a Nutshell, 5th Edition By David Flanagan ISBN: 0596007736 Publisher: O'Reilly
Print Publication Date: 2005/03/01
Prepared for Ronald Fischer, Safari ID: ronald.fischer@fusshuhn.de
User number: 628024 Copyright 2008, Safari Books Online, LLC.
This PDF is exclusively for your use in accordance with the Safari Terms of Service. No part of it may be reproduced or transmitted in any form by any means without the prior written permission for reprints and excerpts from the publisher. Redistribution or other use that violates the fair use priviledge under U.S. copyright laws (see 17 USC107) or that otherwise violates the Safari Terms of Service is strictly prohibited.

```
// Public Instance Methods
     public String getLocalPart( );
     public String getNamespaceURI( );
     public String getPrefix( );
// Public Methods Overriding Object
     public final boolean equals(Object objectToTest);
     public final int hashCode( );
     public String toString( );
}
```

Passed To

`javax.xml.xpath.XPath.evaluate( )`,
`javax.xml.xpath.XPathExpression.evaluate( )`,
`javax.xml.xpath.XPathFunctionResolver.resolveFunction( )`,
`javax.xml.xpath.XPathVariableResolver.resolveVariable( )`

Returned By

`javax.xml.datatype.Duration.getXMLSchemaType( )`,
`javax.xml.datatype.XMLGregorianCalendar.getXMLSchemaType( )`

Type Of

Too many fields to list.

Package javax.xml.parsers

Java 1.4

This package defines classes that represent XML parsers and factory classes for obtaining instances of those parser classes. `DocumentBuilder` is a DOM-based XML parser created from a `DocumentBuilderFactory`. `SAXParser` is a SAX-based XML parser created from a `SAXParserFactory`. In Java 5.0, you can configure either of the factory classes to create parsers that validate against a W3C XML Schema specified with a `javax.xml.validation.Schema` object. Note that this package does not include parser implementations. Instead, it is an implementation-independent layer that supports "pluggable" XML parsers. Furthermore, this package does not define a DOM or SAX API for working with XML documents. The DOM API is defined in `org.w3c.dom`, and the SAX API is defined in `org.xml.sax` and its subpackages.

Classes

```
public abstract class DocumentBuilder;
public abstract class DocumentBuilderFactory;
public abstract class SAXParser;
public abstract class SAXParserFactory;
```

Exceptions

```
public class ParserConfigurationException extends Exception;
```


Java in a Nutshell, 5th Edition By David Flanagan ISBN: 0596007736 Publisher: O'Reilly
Print Publication Date: 2005/03/01
Prepared for Ronald Fischer, Safari ID: ronald.fischer@fusshuhn.de
User number: 628024 Copyright 2008, Safari Books Online, LLC.
This PDF is exclusively for your use in accordance with the Safari Terms of Service. No part of it may be reproduced or transmitted in any form by any means without the prior written permission for reprints and excerpts from the publisher. Redistribution or other use that violates the fair use priviledge under U.S. copyright laws (see 17 USC107) or that otherwise violates the Safari Terms of Service is strictly prohibited.

Errors

```
public class FactoryConfigurationError extends Error;
```

DocumentBuilder javax.xml.parsers

Java 1.4

This class defines a high-level API to an underlying DOM parser implementation. Obtain a `DocumentBuilder` from a `DocumentBuilderFactory`. After obtaining a `DocumentBuilder`, you can provide `org.xml.sax.ErrorHandler` and `org.xml.sax.EntityResolver` objects, if desired. (These classes are defined by the SAX API but are useful for DOM parsers as well.) You may also want to call `isNamespaceAware( )`, `isXIncludeAware( )` and `isValidating( )` to ensure that the parser is configured with the features your application requires. Finally, use one of the `parse( )` methods to read an XML document from a stream, file, URL, or `org.xml.sax.InputSource` object, parse that document, and convert it into a `org.w3c.dom.Document` tree. Note that `DocumentBuilder` objects are not typically threadsafe. In Java 5.0, you can call `reset( )` to restore the parser to its original state for reuse. Another Java 5.0 method, `getSchema( )` returns the `Schema` object, if any, registered with the `DocumentBuilderFactory` that created this parser.

If you want to obtain an empty `Document` object (so that you can build the document tree from scratch, for example) call `newDocument( )`. Or use `getDOMImplementation( )` to obtain a the `org.w3c.dom.DOMImplementation` object of the underlying DOM implementation from which you can also create an empty `Document`.

See the `org.w3c.dom` package for information on what you can do with a `Document` object once you have used a `DocumentBuilder` to create it.

```
public abstract class DocumentBuilder {
// Protected Constructors
     protected DocumentBuilder( );
// Public Instance Methods
     public abstract org.w3c.dom.DOMImplementation getDOMImplementation( );
5.0  public javax.xml.validation.Schema getSchema( );
     public abstract boolean isNamespaceAware( );
     public abstract boolean isValidating( );
5.0  public boolean isXIncludeAware( );
     public abstract org.w3c.dom.Document newDocument( );
     public org.w3c.dom.Document parse(java.io.InputStream is)
        throws org.xml.sax.SAXException, java.io.IOException;
     public org.w3c.dom.Document parse(String uri)
        throws org.xml.sax.SAXException, java.io.IOException;
     public abstract org.w3c.dom.Document parse(org.xml.sax.InputSource is)
throws org.xml.sax.SAXException, java.io.IOException;
     public org.w3c.dom.Document parse(java.io.File f)
        throws org.xml.sax.SAXException, java.io.IOException;
```


Java in a Nutshell, 5th Edition By David Flanagan ISBN: 0596007736 Publisher: O'Reilly
Print Publication Date: 2005/03/01
Prepared for Ronald Fischer, Safari ID: ronald.fischer@fusshuhn.de
User number: 628024 Copyright 2008, Safari Books Online, LLC.
This PDF is exclusively for your use in accordance with the Safari Terms of Service. No part of it may be reproduced or transmitted in any form by any means without the prior written permission for reprints and excerpts from the publisher. Redistribution or other use that violates the fair use priviledge under U.S. copyright laws (see 17 USC107) or that otherwise violates the Safari Terms of Service is strictly prohibited.

```
    public org.w3c.dom.Document parse(java.io.InputStream is, String systemId)
throws org.xml.sax.SAXException, java.io.IOException;
5.0 public void reset( );
    public abstract void setEntityResolver(org.xml.sax.EntityResolver er);
    public abstract void setErrorHandler(org.xml.sax.ErrorHandler eh);
}
```

Returned By

`DocumentBuilderFactory.newDocumentBuilder( )`

DocumentBuilderFactory javax.xml.parsers

Java 1.4

A `DocumentBuilderFactory` is a factory class for creating `DocumentBuilder` objects. You can obtain a `DocumentBuilderFactory` by instantiating an implementation-specific subclass provided by a parser vendor, but it is much more common to simply call `newInstance( )` to obtain an instance of the factory that has been configured as the default for the system. Once you have obtained a factory object, you can use the various `set` methods to configure the properties of the `DocumentBuilder` objects it will create. These methods allow you to specify whether the parsers created by the factory will:

In Java 5.0, you can use `setSchema( )` to specify the `javax.xml.vaidation.Schema` object against which parsers should validate their documents. And you can use `setXIncludeAware( )` to indicate that parsers should process XInclude markup.

In addition to the various implementation-independent `set` methods, you can also use `setAttribute( )` pass an implementation-dependent named attribute to the underlying parser implementation. Once you have configured the factory object as desired, simply call `newDocumentBuilder( )` to create a `DocumentBuilder` object with the all of the attributes you have specified. Note that `DocumentBuilderFactory` objects are not typically threadsafe.

The `javax.xml.parsers` package allows parser implementations to be "plugged in." This pluggability is provided by the `getInstance( )` method, which follows the following steps to determine which `DocumentBuilderFactory` implementation to use:

```
public abstract class DocumentBuilderFactory {
// Protected Constructors
    protected DocumentBuilderFactory( );
// Public Class Methods
    public static DocumentBuilderFactory newInstance( );
// Public Instance Methods
    public abstract Object getAttribute(String name)
       throws IllegalArgumentException;
```


Java in a Nutshell, 5th Edition By David Flanagan ISBN: 0596007736 Publisher: O'Reilly
Print Publication Date: 2005/03/01
Prepared for Ronald Fischer, Safari ID: ronald.fischer@fusshuhn.de
User number: 628024 Copyright 2008, Safari Books Online, LLC.
This PDF is exclusively for your use in accordance with the Safari Terms of Service. No part of it may be reproduced or transmitted in any form by any means without the prior written permission for reprints and excerpts from the publisher. Redistribution or other use that violates the fair use priviledge under U.S. copyright laws (see 17 USC107) or that otherwise violates the Safari Terms of Service is strictly prohibited.

```
5.0  public abstract boolean getFeature(String name)
        throws ParserConfigurationException;
5.0  public javax.xml.validation.Schema getSchema( );
     public boolean isCoalescing( );
     public boolean isExpandEntityReferences( );
     public boolean isIgnoringComments( );
     public boolean isIgnoringElementContentWhitespace( );
     public boolean isNamespaceAware( );
     public boolean isValidating( );
5.0  public boolean isXIncludeAware( );
     public abstract DocumentBuilder newDocumentBuilder( )
        throws ParserConfigurationException;
     public abstract void setAttribute(String name, Object value)
        throws IllegalArgumentException;
     public void setCoalescing(boolean coalescing);
     public void setExpandEntityReferences(boolean expandEntityRef);
5.0  public abstract void setFeature(String name, boolean value)
        throws ParserConfigurationException;
     public void setIgnoringComments(boolean ignoreComments);
     public void setIgnoringElementContentWhitespace(boolean whitespace);
     public void setNamespaceAware(boolean awareness);
5.0  public void setSchema(javax.xml.validation.Schema schema);
     public void setValidating(boolean validating);
5.0  public void setXIncludeAware(boolean state);
}
```

FactoryConfigurationError javax.xml.parsers

Java 1.4 ***serializable error***

Signals a nonrecoverable problem instantiating a parser factory. This usually means that a pluggable parser implementation has been incorrectly plugged in and the `getInstance( )` method cannot locate the specified factory implementation class.

Figure 20-3. javax.xml.parsers.FactoryConfigurationError


```
public class FactoryConfigurationError extends Error {
// Public Constructors
     public FactoryConfigurationError( );
     public FactoryConfigurationError(Exception e);
     public FactoryConfigurationError(String msg);
     public FactoryConfigurationError(Exception e, String msg);
// Public Instance Methods
     public Exception getException( );                                    default:null
// Public Methods Overriding Throwable
     public String getMessage( );                                         default:null
}
```


Java in a Nutshell, 5th Edition By David Flanagan ISBN: 0596007736 Publisher: O'Reilly
Print Publication Date: 2005/03/01
Prepared for Ronald Fischer, Safari ID: ronald.fischer@fusshuhn.de
User number: 628024 Copyright 2008, Safari Books Online, LLC.
This PDF is exclusively for your use in accordance with the Safari Terms of Service. No part of it may be reproduced or transmitted in any form by any means without the prior written permission for reprints and excerpts from the publisher. Redistribution or other use that violates the fair use priviledge under U.S. copyright laws (see 17 USC107) or that otherwise violates the Safari Terms of Service is strictly prohibited.

ParserConfigurationException javax.xml.parsers

Java 1.4 ***serializable checked***

Signals a parser configuration problem that prevents a parser factory object from creating a parser object.

Figure 20-4. javax.xml.parsers.ParserConfigurationException


```
public class ParserConfigurationException extends Exception {
// Public Constructors
     public ParserConfigurationException( );
     public ParserConfigurationException(String msg);
}
```

Thrown By

`DocumentBuilderFactory.{getFeature( ),newDocumentBuilder( ), setFeature( )},SAXParserFactory.{getFeature( ),newSAXParser( ), setFeature( )}`

SAXParser javax.xml.parsers

Java 1.4

The `SAXParser` class is a wrapper around an `org.xml.sax.XMLReader` class and is used to parse XML documents using the SAX version 2 API. Obtain a `SAXParser` from a `SAXParserFactory`. Call `setProperty( )` if desired to set a property on the underlying parser. (See www.saxproject.org for a description of standard SAX properties and their values. Finally, call one of the `parse( )` methods to parse an XML document from a stream, file, URL, or `org.xml.sax.InputSource`. The SAX API is an event-driven one. A SAX parser does not build a document tree to describe an XML document like a DOM parser does. Instead, it describes the XML document to your application by invoking methods on an object the application provides. This is the purpose of the `org.xml.sax.helpers.DefaultHandler` object that is passed to the `parse( )` method: you subclass this class to implement the methods you care about, and the parser will invoke those methods at appropriate times. For example, when the parser encounters an XML tag in a document, it parses the tag, and calls the `startElement( )` method to

Java in a Nutshell, 5th Edition By David Flanagan ISBN: 0596007736 Publisher: O'Reilly
Print Publication Date: 2005/03/01
Prepared for Ronald Fischer, Safari ID: ronald.fischer@fusshuhn.de
User number: 628024 Copyright 2008, Safari Books Online, LLC.
This PDF is exclusively for your use in accordance with the Safari Terms of Service. No part of it may be reproduced or transmitted in any form by any means without the prior written permission for reprints and excerpts from the publisher. Redistribution or other use that violates the fair use priviledge under U.S. copyright laws (see 17 USC107) or that otherwise violates the Safari Terms of Service is strictly prohibited.

tell you about it. And when it finds a run of plain text, it passes that text to the `characters( )` method. In Java 5.0, the `reset( )` method restores a `SAXParser` to its original state so that it can be reused.

Instead of using one of the `parse( )` methods of this class, you can also call `getXMLReader( )` to obtain the underlying `XMLReader` object and work with it directly to parse the desired document. `SAXParser` objects are not typically threadsafe.

Note that the `getParser( )` method as well as the `parse( )` methods that take an `org.xml.sax.HandlerBase` object are based on the SAX version 1 API, and should be avoided.

```
public abstract class SAXParser {
// Protected Constructors
     protected SAXParser( );
// Public Instance Methods
     public abstract org.xml.sax.Parser getParser( ) throws org.xml.sax.SAXException;
     public abstract Object getProperty(String name)
        throws org.xml.sax.SAXNotRecognizedException,
        org.xml.sax.SAXNotSupportedException;
5.0  public javax.xml.validation.Schema getSchema( );
     public abstract org.xml.sax.XMLReader getXMLReader( ) throws org.xml.sax.SAXException;
     public abstract boolean isNamespaceAware( );
     public abstract boolean isValidating( );
5.0  public boolean isXIncludeAware( );
     public void parse(org.xml.sax.InputSource is, org.xml.sax.HandlerBase hb)
        throws org.xml.sax.SAXException, java.io.IOException;
     public void parse(org.xml.sax.InputSource is, org.xml.sax.helpers.DefaultHandler dh)
        throws org.xml.sax.SAXException, java.io.IOException;
     public void parse(java.io.File f, org.xml.sax.helpers.DefaultHandler dh)
        throws org.xml.sax.SAXException, java.io.IOException;
     public void parse(java.io.InputStream is, org.xml.sax.helpers.DefaultHandler dh)
        throws org.xml.sax.SAXException, java.io.IOException;
     public void parse(java.io.InputStream is, org.xml.sax.HandlerBase hb)
        throws org.xml.sax.SAXException, java.io.IOException;
     public void parse(String uri, org.xml.sax.HandlerBase hb) throws org.xml.sax.SAXException, java.io.IOException;
     public void parse(String uri, org.xml.sax.helpers.DefaultHandler dh)
        throws org.xml.sax.SAXException, java.io.IOException;
     public void parse(java.io.File f, org.xml.sax.HandlerBase hb) throws org.xml.sax.SAXException, java.io.IOException;
     public void parse(java.io.InputStream is, org.xml.sax.HandlerBase hb, String systemId)
        throws org.xml.sax.SAXException, java.io.IOException;
     public void parse(java.io.InputStream is, org.xml.sax.helpers.DefaultHandler dh, String systemId) throws org.xml.sax.SAX
5.0  public void reset( );
     public abstract void setProperty(String name, Object value)
        throws org.xml.sax.SAXNotRecognizedException,
        org.xml.sax.SAXNotSupportedException;
}
```

Returned By

`SAXParserFactory.newSAXParser( )`

SAXParserFactory **javax.xml.parsers**

Java 1.4

Java in a Nutshell, 5th Edition By David Flanagan ISBN: 0596007736 Publisher: O'Reilly
Print Publication Date: 2005/03/01
Prepared for Ronald Fischer, Safari ID: ronald.fischer@fusshuhn.de
User number: 628024 Copyright 2008, Safari Books Online, LLC.
This PDF is exclusively for your use in accordance with the Safari Terms of Service. No part of it may be reproduced or transmitted in any form by any means without the prior written permission for reprints and excerpts from the publisher. Redistribution or other use that violates the fair use priviledge under U.S. copyright laws (see 17 USC107) or that otherwise violates the Safari Terms of Service is strictly prohibited.

This class is a factory for `SAXParser` objects. Obtain a `SAXParserFactory` by calling the `newInstance( )` method which instantiates the default `SAXParserFactory` subclass provided with your Java implementation, or instantiates some other `SAXParserFactory` that has been "plugged in".

Once you have a `SAXParserFactory` object, you can use `setValidating( )` and `setNamespaceAware( )` to specify whether the parsers it creates will be validating parsers or not and whether they will know how to handle XML namespaces. You may also call `setFeature( )` to set a feature of the underlying parser implementation. See http://www.saxproject.org for the names of standard parser features that can be enabled and disabled with this method. In Java 5.0, call `setXIncludeAware( )` to specify that created parsers will recognize XInclude markup. Use `setSchema( )` to specify a W3C XML Schema against which parsers should validate the document.

Once you have created and configured your factory object, simply call `newSAXParser( )` to create a `SAXParser` object. Note that `SAXParserFactory` implementations are not typically threadsafe.

The `javax.xml.parsers` package allows parser implementations to be "plugged in". This pluggability is provided by the `getInstance( )` method, which follows the following steps to determine which `SAXBuilderFactory` subclass to use:

```
public abstract class SAXParserFactory {
// Protected Constructors
     protected SAXParserFactory( );
// Public Class Methods
     public static SAXParserFactory newInstance( );
// Public Instance Methods
     public abstract boolean getFeature(String name)
        throws ParserConfigurationException,
        org.xml.sax.SAXNotRecognizedException,
        org.xml.sax.SAXNotSupportedException;
5.0  public javax.xml.validation.Schema getSchema( );
     public boolean isNamespaceAware( );
     public boolean isValidating( );
5.0  public boolean isXIncludeAware( );
     public abstract SAXParser newSAXParser( )
        throws ParserConfigurationException,
        org.xml.sax.SAXException;
     public abstract void setFeature(String name, boolean value)
        throws ParserConfigurationException,
        org.xml.sax.SAXNotRecognizedException,
        org.xml.sax.SAXNotSupportedException;
     public void setNamespaceAware(boolean awareness);
5.0  public void setSchema(javax.xml.validation.Schema schema);
     public void setValidating(boolean validating);
5.0  public void setXIncludeAware(boolean state);
}
```

Package javax.xml.transform

Java in a Nutshell, 5th Edition By David Flanagan ISBN: 0596007736 Publisher: O'Reilly
Print Publication Date: 2005/03/01
Prepared for Ronald Fischer, Safari ID: ronald.fischer@fusshuhn.de
User number: 628024 Copyright 2008, Safari Books Online, LLC.
This PDF is exclusively for your use in accordance with the Safari Terms of Service. No part of it may be reproduced or transmitted in any form by any means without the prior written permission for reprints and excerpts from the publisher. Redistribution or other use that violates the fair use priviledge under U.S. copyright laws (see 17 USC107) or that otherwise violates the Safari Terms of Service is strictly prohibited.

Java 1.4

This package defines an high-level implementation-independent API for using an XSLT engine or other document transformation system for transforming XML document content, and also for transforming XML documents from one form (such as a stream of text in a file) to anther form (such as a tree of DOM nodes). The `Source` interface is a very generic description of a document source. Three concrete implementations that represent documents in text form, as DOM trees, and as sequences of SAX parser events are defined in the three subpackages of this package. The `Result` interface is a similarly high-level description of what form the source document should be transformed into. The three subpackages define three `Result` implementations that represent XML documents as streams or files, as DOM trees, and as sequnces of SAX parser events.

The `TransformerFactory` class represents the document transformation engine. The implementation provides a default factory that represents an XSLT engine. A `TransformerFactory` can be used to produce `Templates` objects that represent compiled XSL stylesheets (or other implementation-dependent forms of transformation instructions). Documents are actually transfomed from `Soruce` to `Result` with a `Transformer` object, which is obtained from a `Templates` object, or directly from a `TransformerFactory`.

Interfaces

```
public interface ErrorListener;
public interface Result;
public interface Source;
public interface SourceLocator;
public interface Templates;
public interface URIResolver;
```

Classes

```
public class OutputKeys;
public abstract class Transformer;
public abstract class TransformerFactory;
```

Exceptions

```
public class TransformerException extends Exception;
   public class TransformerConfigurationException extends TransformerException;
```

Errors

```
public class TransformerFactoryConfigurationError extends Error;
```

ErrorListener javax.xml.transform

Java 1.4

Java in a Nutshell, 5th Edition By David Flanagan ISBN: 0596007736 Publisher: O'Reilly
Print Publication Date: 2005/03/01
Prepared for Ronald Fischer, Safari ID: ronald.fischer@fusshuhn.de
User number: 628024 Copyright 2008, Safari Books Online, LLC.
This PDF is exclusively for your use in accordance with the Safari Terms of Service. No part of it may be reproduced or transmitted in any form by any means without the prior written permission for reprints and excerpts from the publisher. Redistribution or other use that violates the fair use priviledge under U.S. copyright laws (see 17 USC107) or that otherwise violates the Safari Terms of Service is strictly prohibited.

This interface defines methods that `Transformer` and `TransformerFactory` use for reporting warnings, errors, and fatal errors to an application. To use an `ErrorListener`, an application must implement this interface and pass an implementing object to the `setErrorListener( )` method of `Transformer` or `TransformerFactory`. The argument to each method of this interface is a `TransformerException` object, and the implementation of these methods can throw that exception if it chooses, or it can simply log the warning or error in some way and return. A `Transformer` or `TransformerFactory` is not required to continue processing after reporting a nonrecoverable error with an invocation of the `fatalError( )` method.

If you are familiar with the SAX API for parsing XML documents, you'll recognize that this interface is very similar to `org.xml.sax.ErrorHandler`.

```
public interface ErrorListener {
// Public Instance Methods
     void error(TransformerException exception) throws TransformerException;
     void fatalError(TransformerException exception) throws TransformerException;
     void warning(TransformerException exception) throws TransformerException;
}
```

Passed To

`Transformer.setErrorListener( )`, `TransformerFactory.setErrorListener( )`

Returned By

`Transformer.getErrorListener( )`, `TransformerFactory.getErrorListener( )`

OutputKeys javax.xml.transform

Java 1.4

This class defines string constants that hold the names of the attributes of an `<xsl:output>` tag in an XSLT stylesheet. These are also legal key values for the `Properties` object returned by `Templates.getOutputProperties( )` and passed to `Transformer.setOutputProperties( )`.

```
public class OutputKeys {
// No Constructor
// Public Constants
     public static final String CDATA_SECTION_ELEMENTS;      ="cdata-section-elements"
     public static final String DOCTYPE_PUBLIC;              ="doctype-public"
     public static final String DOCTYPE_SYSTEM;              ="doctype-system"
     public static final String ENCODING;                    ="encoding"
     public static final String INDENT;                      ="indent"
     public static final String MEDIA_TYPE;                  ="media-type"
     public static final String METHOD;                      ="method"
     public static final String OMIT_XML_DECLARATION; ="omit-xml-declaration"
     public static final String STANDALONE; ="standalone"
```


Java in a Nutshell, 5th Edition By David Flanagan ISBN: 0596007736 Publisher: O'Reilly
Print Publication Date: 2005/03/01
Prepared for Ronald Fischer, Safari ID: ronald.fischer@fusshuhn.de
User number: 628024 Copyright 2008, Safari Books Online, LLC.
This PDF is exclusively for your use in accordance with the Safari Terms of Service. No part of it may be reproduced or transmitted in any form by any means without the prior written permission for reprints and excerpts from the publisher. Redistribution or other use that violates the fair use priviledge under U.S. copyright laws (see 17 USC107) or that otherwise violates the Safari Terms of Service is strictly prohibited.

```
     public static final String VERSION;                    ="version"
}
```

Result javax.xml.transform

Java 1.4

This interface represents, in a very general way, the result of an XML transformation. `setSystemId( )` specifies a the system identifier of the result as a URL. This is useful when the result is to be written as a file, but it can also be useful for error reporting and for resolution of relative URLs even when the `Result` object does not represent a file. All other methods related to the result are the responsibility of the concrete implementation of this interface. See the `DOMResult`, `SAXResult` and `StreamResult` implementations in the three subpackages of this package.

```
public interface Result {
// Public Constants
     public static final String PI_DISABLE_OUTPUT_ESCAPING;
         ="javax.xml.transform.disable-output-escaping"
     public static final String PI_ENABLE_OUTPUT_ESCAPING;
         ="javax.xml.transform.enable-output-escaping"
// Public Instance Methods
     String getSystemId( );
     void setSystemId(String systemId);
}
```

Implementations

`javax.xml.transform.dom.DOMResult`,
`javax.xml.transform.sax.SAXResult`,
`javax.xml.transform.stream.StreamResult`

Passed To

`Transformer.transform( )`,
`javax.xml.transform.sax.TransformerHandler.setResult( )`,
`javax.xml.validation.Validator.validate( )`

Source javax.xml.transform

Java 1.4

This interface represents, in a very general way, the source of an XML document. `setSystemId( )` specifies a the system identifier of the document in the form of a URL. This is useful for resolving relative URLs and for error reporting even when the document

Java in a Nutshell, 5th Edition By David Flanagan ISBN: 0596007736 Publisher: O'Reilly
Print Publication Date: 2005/03/01
Prepared for Ronald Fischer, Safari ID: ronald.fischer@fusshuhn.de
User number: 628024 Copyright 2008, Safari Books Online, LLC.
This PDF is exclusively for your use in accordance with the Safari Terms of Service. No part of it may be reproduced or transmitted in any form by any means without the prior written permission for reprints and excerpts from the publisher. Redistribution or other use that violates the fair use priviledge under U.S. copyright laws (see 17 USC107) or that otherwise violates the Safari Terms of Service is strictly prohibited.

is not read directly from a URL. All other methods related to the document source are the responsibility of the concrete implementation of this interface. See the `DOMSource`, `SAXSource` and `StreamSource` implementations in the three subpackages of this package.

```
public interface Source {
// Public Instance Methods
     String getSystemId( );
     void setSystemId(String systemId);
}
```

Implementations

`javax.xml.transform.dom.DOMSource`,
`javax.xml.transform.sax.SAXSource`,
`javax.xml.transform.stream.StreamSource`

Passed To

`Transformer.transform( )`, `TransformerFactory.{getAssociatedStylesheet( ), newTemplates( ), newTransformer( )}`, `javax.xml.transform.sax.SAXSource.sourceToInputSource( )`, `javax.xml.transform.sax.SAXTransformerFactory.{newTransformerHandler( ), newXMLFilter( )}`, `javax.xml.validation.SchemaFactory.newSchema( )`, `javax.xml.validation.Validator.validate( )`

Returned By

`TransformerFactory.getAssociatedStylesheet( )`, `URIResolver.resolve( )`

SourceLocator javax.xml.transform

Java 1.4

This interface defines methods that return the system and public identifiers of an XML document, and return a line number and column number within that document. `SourceLocator` objects are used with `TransformerException` and `TransformerConfigurationException` objects to specify the location in an XML file at which the exception occurred. Note, however that system and public identifiers are not always available for a document, and so `getSystemId( )` and `getPublicId( )` may return `null`. Also, a `Tranformer` is not required to track line and column numbers precisely, or at all, so `getLineNumber( )` and `getColumnNumber( )` may return -1 to indicate that line and column number information is not available. If they return a value other than -1, it should be considered an approximation to the actual value. Note that lines and columns within a document are numbered starting with 1, not with 0.

Java in a Nutshell, 5th Edition By David Flanagan ISBN: 0596007736 Publisher: O'Reilly
Print Publication Date: 2005/03/01
Prepared for Ronald Fischer, Safari ID: ronald.fischer@fusshuhn.de
User number: 628024 Copyright 2008, Safari Books Online, LLC.
This PDF is exclusively for your use in accordance with the Safari Terms of Service. No part of it may be reproduced or transmitted in any form by any means without the prior written permission for reprints and excerpts from the publisher. Redistribution or other use that violates the fair use priviledge under U.S. copyright laws (see 17 USC107) or that otherwise violates the Safari Terms of Service is strictly prohibited.

If you are familiar with the SAX API for parsing XML, you'll recognize this interface as a renamed version of `org.xml.sax.Locator`.

```
public interface SourceLocator {
// Public Instance Methods
     int getColumnNumber( );
     int getLineNumber( );
     String getPublicId( );
     String getSystemId( );
}
```

Implementations

`javax.xml.transform.dom.DOMLocator`

Passed To

`TransformerConfigurationException.TransformerConfigurationException( ), TransformerException.{setLocator( ), TransformerException( )}`

Returned By

`TransformerException.getLocator( )`

Templates javax.xml.transform

Java 1.4

This interface represents a set of transformation instructions for transforming a `Source` document into a `Result` document. The `javax.xml.transform` package is nominally independent of type of transformation, but in practice, an object of this type always represents the compiled form of an XSLT stylesheet. Obtain a `Templates` object from a `TransformerFactory` object, or with a `javax.xml.transform.sax.TemplatesHandler`. Once you have a `Templates` object, you can use the `newTransformer( )` method to create a `Transformer` object for applying the templates to a `Source` to produce a `Result` document.

`getOutputProperties( )` returns a `java.util.Properties` object that defines name/value pairs specifying details about how a textual version of the `Result` document should be produced. These properties are specified in an XSLT stylesheet with the `<xsl:output>` element. The constants defined by the `OutputKeys` are legal output property names. The returned `Properties` object contains explicitly properties directly, and contains default values in a parent `Properties` object. This means that if you query a property value with `getProperty( )`, you'll get an explicitly specified value of a default value. On the other hand, if you query a property with the `get( )` method (inherited by `Properties` from its superclass) you'll get a property value if it was explictly specified in the stylesheet, or `null` if it was not specified. The returned `Properties` object is a clone

Java in a Nutshell, 5th Edition By David Flanagan ISBN: 0596007736 Publisher: O'Reilly
Print Publication Date: 2005/03/01
Prepared for Ronald Fischer, Safari ID: ronald.fischer@fusshuhn.de
User number: 628024 Copyright 2008, Safari Books Online, LLC.
This PDF is exclusively for your use in accordance with the Safari Terms of Service. No part of it may be reproduced or transmitted in any form by any means without the prior written permission for reprints and excerpts from the publisher. Redistribution or other use that violates the fair use priviledge under U.S. copyright laws (see 17 USC107) or that otherwise violates the Safari Terms of Service is strictly prohibited.

of the internal value, so you can modify it (before passing it to the `setOutputProperties( )` method of a `Transformer` object, for example) without affecting the `Templates` object.

`Templates` implementations are required to be threadsafe. A `Templates` object can be used to create any number of `Transformer` objects.

```
public interface Templates {
// Public Instance Methods
     java.util.Properties getOutputProperties( );
     Transformer newTransformer( ) throws TransformerConfigurationException;
}
```

Passed To

`javax.xml.transform.sax.SAXTransformerFactory.{newTransformerHandler( ),newXMLFilter( )}`

Returned By

`TransformerFactory.newTemplates( )`, `javax.xml.transform.sax.TemplatesHandler.getTemplates( )`

Transformer javax.xml.transform

Java 1.4

Objects of this type are used to transform a `Source` document into a `Result` document. Obtain a `Transformer` object from a `TransformerFactory` object, from a `Templates` object created by a `TransformerFactory`, or from a `TransformerHandler` object created by a `SAXTransformerFactory` (these last two types are from the `javax.xml.transform.sax` package).

Once you have a `Transformer` object, you may need to configure it before using it to transform documents. `setErrorListener( )` and `setURIResolver( )` allow you to specify `ErrorListener` and `URLResolver` object that the `Transformer` can use. `setOutputProperty( )` and `setOutputProperties( )` allow you to specify name/value pairs that affect the text formatting of the `Result` document (if that document is written out in text format). `OutputKeys` defines constants that represent the set of standard output property names. The output properties you specify with these methods override any output properties specified (with an `<xsl:output>` tag) in the `Templates` object. Use `setParameter( )` to supply values for any top-level parameters defined (with `<xsl:param>` tags) in the stylesheet. Note that if the name of any such parameter is a qualified name, then it appears in the stylesheet with a namespace prefix. You can't use the prefix with the `setParameter( )` method, however, and you must

Java in a Nutshell, 5th Edition By David Flanagan ISBN: 0596007736 Publisher: O'Reilly
Print Publication Date: 2005/03/01
Prepared for Ronald Fischer, Safari ID: ronald.fischer@fusshuhn.de
User number: 628024 Copyright 2008, Safari Books Online, LLC.
This PDF is exclusively for your use in accordance with the Safari Terms of Service. No part of it may be reproduced or transmitted in any form by any means without the prior written permission for reprints and excerpts from the publisher. Redistribution or other use that violates the fair use priviledge under U.S. copyright laws (see 17 USC107) or that otherwise violates the Safari Terms of Service is strictly prohibited.

instead specify the parameter name using the URI of the namespace within curly braces followed by the local name. If no namespace is involved, then you can just use the simple name of the parameter with no curly braces or URIs.

Once you have created and configured a `Transformer` object, use the `transform( )` method to perform a document transformation. This method transforms the specified `Source` document and creates the transformed document specified by the `Result` object. In Java 5.0, you can `reset( )` a `Transformer` to restore it to its original state and prepare it for reuse.

`Transformer` implementations are not typically threadsafe. You can reuse a `Transformer` object and call `transform( )` any number of times (just not concurrently). The output properties and parameters you specify are not changed by calling the `transform( )` method, and can be reused.

```
public abstract class Transformer {
// Protected Constructors
     protected Transformer( );
// Public Instance Methods
     public abstract void clearParameters( );
     public abstract ErrorListener getErrorListener( );
     public abstract java.util.Properties getOutputProperties( );
     public abstract String getOutputProperty(String name)
        throws IllegalArgumentException;
     public abstract Object getParameter(String name);
     public abstract URIResolver getURIResolver( );
5.0  public void reset( );
     public abstract void setErrorListener(ErrorListener listener)
        throws IllegalArgumentException;
     public abstract void setOutputProperties(java.util.Properties oformat);
     public abstract void setOutputProperty(String name, String value)
        throws IllegalArgumentException;
     public abstract void setParameter(String name, Object value);
     public abstract void setURIResolver(URIResolver resolver);
     public abstract void transform(Source xmlSource, Result outputTarget)
        throws TransformerException;
}
```

Returned By

`Templates.newTransformer( )`, `TransformerFactory.newTransformer( )`, `javax.xml.transform.sax.TransformerHandler.getTransformer( )`

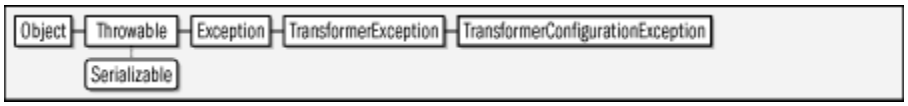
TransformerConfigurationException — javax.xml.transform

Java 1.4 — ***serializable checked***

Signals a problem creating a `Transformer` object. This may occur, for exmaple, if there is a syntax error in the XSL stylesheet that contains the transformation instructions. Use the inherited `getLocator( )` method to obtain a `SourceLocator` that describes the document location at which the exception occurred.

Java in a Nutshell, 5th Edition By David Flanagan ISBN: 0596007736 Publisher: O'Reilly
Print Publication Date: 2005/03/01
Prepared for Ronald Fischer, Safari ID: ronald.fischer@fusshuhn.de
User number: 628024 Copyright 2008, Safari Books Online, LLC.
This PDF is exclusively for your use in accordance with the Safari Terms of Service. No part of it may be reproduced or transmitted in any form by any means without the prior written permission for reprints and excerpts from the publisher. Redistribution or other use that violates the fair use priviledge under U.S. copyright laws (see 17 USC107) or that otherwise violates the Safari Terms of Service is strictly prohibited.

Figure 20-5. javax.xml.transform.TransformerConfigurationException




```
public class TransformerConfigurationException extends TransformerException {
// Public Constructors
     public TransformerConfigurationException( );
     public TransformerConfigurationException(Throwable e);
     public TransformerConfigurationException(String msg);
     public TransformerConfigurationException(String message, SourceLocator locator);
     public TransformerConfigurationException(String msg, Throwable e);
     public TransformerConfigurationException(String message, SourceLocator locator,
        Throwable e);
}
```

Thrown By

`Templates.newTransformer( ),TransformerFactory.{getAssociatedStylesheet( ),newTemplates( ),newTransformer( ),setFeature( )},javax.xml.transform.sax.SAXTransformerFactory.{newTemplatesHandler( ),newTransformerHandler( ),newXMLFilter( )}`

TransformerException — javax.xml.transform

Java 1.4 — ***serializable checked***

Signals a problem while reading or transforming a document. Call `getLocator( )` to obtain a `SourceLocator` object that describes the document location at which the exception occured.

Figure 20-6. javax.xml.transform.TransformerException


```
public class TransformerException extends Exception {
// Public Constructors
     public TransformerException(String message);
     public TransformerException(Throwable e);
     public TransformerException(String message, Throwable e);
     public TransformerException(String message, SourceLocator locator);
     public TransformerException(String message, SourceLocator locator, Throwable e);
// Public Instance Methods
     public Throwable getException( );
     public String getLocationAsString( );
     public SourceLocator getLocator( );
     public String getMessageAndLocation( );
     public void setLocator(SourceLocator location);
// Public Methods Overriding Throwable
     public Throwable getCause( );
```


Java in a Nutshell, 5th Edition By David Flanagan ISBN: 0596007736 Publisher: O'Reilly
Print Publication Date: 2005/03/01
Prepared for Ronald Fischer, Safari ID: ronald.fischer@fusshuhn.de
User number: 628024 Copyright 2008, Safari Books Online, LLC.
This PDF is exclusively for your use in accordance with the Safari Terms of Service. No part of it may be reproduced or transmitted in any form by any means without the prior written permission for reprints and excerpts from the publisher. Redistribution or other use that violates the fair use priviledge under U.S. copyright laws (see 17 USC107) or that otherwise violates the Safari Terms of Service is strictly prohibited.

```
        public Throwable initCause(Throwable cause);                synchronized
        public void printStackTrace( );
        public void printStackTrace(java.io.PrintStream s);
        public void printStackTrace(java.io.PrintWriter s);
    }
```

Subclasses

`TransformerConfigurationException`

Passed To

`ErrorListener.{error( ),fatalError( ),warning( )}`

Thrown By

`ErrorListener.{error( ),fatalError( ),warning( )}`, `Transformer.transform( )`, `URIResolver.resolve( )`

TransformerFactory javax.xml.transform

Java 1.4

An instance of this abstract class represents a document "transformation engine" such as an XSLT processor. A `TransformerFactory` is used to create `Transformer` objects that perform document transformations, and can also be used to process transformation instructions (such as XSLT stylesheets) into compiled `Templates` objects.

Obtain a `TransformerFactory` instance by calling the static `newInstance( )` method. `newInstance( )` returns an instance of the default implementation for your Java installation, or, if the system property `javax.xml.transform.TransformerFactory` is set, then it returns an instance of the implementation class named by that property. The default `TransformerFactory` implementation provided with the Java distribution transforms XML documents using XSL stylesheets.

You can configure a `TransformerFactory` instance by calling `setErrorListener( )` and `setURIResolver( )` to specify an `ErrorListener` object and a `URIResolver` object to be used by the factory when reading and parsing XSL stylesheets. The `setAttribute( )` and `getAttribute( )` methods can be used to set and query implementation-dependent attributes of the transformation engine. The default engine supplied by Sun does not define any attributes. The `getFeature( )` method is used to test whether the factory supports a given feature. For uniqueness, feature names are expressed as URIs, and each of the `Source` and `Result` implementations defined in the three subpackages of this package define a `FEATURE` constant that specifies a URL that you can use to test whether a `TransformerFactory` supports that particular `Source` or `Result` type.

Java in a Nutshell, 5th Edition By David Flanagan ISBN: 0596007736 Publisher: O'Reilly
Print Publication Date: 2005/03/01
Prepared for Ronald Fischer, Safari ID: ronald.fischer@fusshuhn.de
User number: 628024 Copyright 2008, Safari Books Online, LLC.
This PDF is exclusively for your use in accordance with the Safari Terms of Service. No part of it may be reproduced or transmitted in any form by any means without the prior written permission for reprints and excerpts from the publisher. Redistribution or other use that violates the fair use priviledge under U.S. copyright laws (see 17 USC107) or that otherwise violates the Safari Terms of Service is strictly prohibited.

Once you have obtained and configured your `TransformerFactory` object, you can use it in several ways. If you call the `newTransformer( )` method that takes no arguments, you'll obtain a `Transformer` object that transforms the format or representation of an XML document without transforming its content. For example, you could use a `Transformer` created in this way to transform a DOM tree (represented by a `javax.xml.transform.dom.DOMSource` object) to a stream of XML text stored in a file named by a `javax.xml.transform.stream.StreamResult`.

Another way to use a `TransformerFactory` is to call the `newTemplates( )` method, passing in a `Source` object that represents an XSL stylesheet. This produces a `Templates` object, which you can use to obtain a `Transformer` object that applies the stylesheet to transform document content. Alternatively, if you do not plan to create more than one `Transformer` object from the `Templates` object, you can combine the two steps and simply pass the `Source` object representing the stylesheet to the one-argument version of `newTransformer( )`.

XML documents may include references to XSL stylesheets in the form of an xml-stylesheet processing instruction. The `getAssociatedStylesheet( )` method reads the XML document represented by a `Source` object and returns a new `Source` object that represents the stylesheet (or the concatenation of all the stylesheets) contained in that document that match the media, title, and charset constraints defined by the other three parameters (which may be null). If you want to process an XML document using the stylesheet that it defines itself, use this method to obtain a `Source` object that you can pass to `newTransformer( )` to create the `Transformer` object that you can use to transform the document.

`TransformerFactory` implementations are not typically threadsafe.

```
public abstract class TransformerFactory {
// Protected Constructors
     protected TransformerFactory( );
// Public Class Methods
     public static TransformerFactory newInstance( )
        throws TransformerFactoryConfigurationError;
// Public Instance Methods
     public abstract Source getAssociatedStylesheet(Source source, String media,
        String title, String charset)
        throws TransformerConfigurationException;
     public abstract Object getAttribute(String name);
     public abstract ErrorListener getErrorListener( );
     public abstract boolean getFeature(String name);
     public abstract URIResolver getURIResolver( );
     public abstract Templates newTemplates(Source source)
        throws TransformerConfigurationException;
     public abstract Transformer newTransformer( ) throws TransformerConfigurationException;
     public abstract Transformer newTransformer(Source source)
        throws TransformerConfigurationException;
     public abstract void setAttribute(String name, Object value);
     public abstract void setErrorListener(ErrorListener listener);
5.0  public abstract void setFeature(String name, boolean value)
        throws TransformerConfigurationException;
```


Java in a Nutshell, 5th Edition By David Flanagan ISBN: 0596007736 Publisher: O'Reilly
Print Publication Date: 2005/03/01
Prepared for Ronald Fischer, Safari ID: ronald.fischer@fusshuhn.de
User number: 628024 Copyright 2008, Safari Books Online, LLC.
This PDF is exclusively for your use in accordance with the Safari Terms of Service. No part of it may be reproduced or transmitted in any form by any means without the prior written permission for reprints and excerpts from the publisher. Redistribution or other use that violates the fair use priviledge under U.S. copyright laws (see 17 USC107) or that otherwise violates the Safari Terms of Service is strictly prohibited.

```
    public abstract void setURIResolver(URIResolver resolver);
}
```

Subclasses

`javax.xml.transform.sax.SAXTransformerFactory`

TransformerFactoryConfigurationError javax.xml.transform

Java 1.4 ***serializable error***

This error class signals a fatal problem while creating a `TransformerFactory`. It usually signals a configuration problem, such as the system property `javax.xml.transform.TransformerFactory` has a value that is not a valid classname, or that the class path does not contain the specified factory implementation class.

Figure 20-7. javax.xml.transform.TransformerFactoryConfigurationError


```
public class TransformerFactoryConfigurationError extends Error {
// Public Constructors
     public TransformerFactoryConfigurationError( );
     public TransformerFactoryConfigurationError(String msg);
     public TransformerFactoryConfigurationError(Exception e);
     public TransformerFactoryConfigurationError(Exception e, String msg);
// Public Instance Methods
     public Exception getException( );                                    default:null
// Public Methods Overriding Throwable
     public String getMessage( );                                         default:null
}
```

Thrown By

`TransformerFactory.newInstance( )`

URIResolver javax.xml.transform

Java 1.4

This interface allows an application to tell a `Transformer` how to resolve the URIs that appear in an XSLT stylesheet. If you pass a `URIResolver` to the `setURIResolver( )` method of a `Transformer` or `TransformerFactory` then when the `Transformer` or `TransformerFactory` encounters a URI, it first passes that URI, along with the base

Java in a Nutshell, 5th Edition By David Flanagan ISBN: 0596007736 Publisher: O'Reilly
Print Publication Date: 2005/03/01
Prepared for Ronald Fischer, Safari ID: ronald.fischer@fusshuhn.de
User number: 628024 Copyright 2008, Safari Books Online, LLC.
This PDF is exclusively for your use in accordance with the Safari Terms of Service. No part of it may be reproduced or transmitted in any form by any means without the prior written permission for reprints and excerpts from the publisher. Redistribution or other use that violates the fair use priviledge under U.S. copyright laws (see 17 USC107) or that otherwise violates the Safari Terms of Service is strictly prohibited.

URI to the `resolve( )` method of the `URIResolver`. If `resolve( )` returns a `Source` object, then the `Transformer` will use that `Source`. If a `Transformer` or `TransformerFactory` has no URIResolver registered, or if the `resolve( )` method returns `null`, then the tranformer or factory will attempt to resolve the URI itself.

```
public interface URIResolver {
// Public Instance Methods
     Source resolve(String href, String base) throws TransformerException;
}
```

Passed To

`Transformer.setURIResolver( )`, `TransformerFactory.setURIResolver( )`

Returned By

`Transformer.getURIResolver( )`, `TransformerFactory.getURIResolver( )`

Package javax.xml.transform.dom

Java 1.4

This package contains `Source` and `Result` implementations that work with DOM document trees and subtrees.

Interfaces

```
public interface DOMLocator extends javax.xml.transform.SourceLocator;
```

Classes

```
public class DOMResult implements javax.xml.transform.Result;
public class DOMSource implements javax.xml.transform.Source;
```

DOMLocator javax.xml.transform.dom

Java 1.4

This class extends `SourceLocator` to define a method for retrieving a DOM `Node` object, which is typically used to indicate the source of an error in the transformation process. See `SourceLocator` and `TransformerException`.

Java in a Nutshell, 5th Edition By David Flanagan ISBN: 0596007736 Publisher: O'Reilly
Print Publication Date: 2005/03/01
Prepared for Ronald Fischer, Safari ID: ronald.fischer@fusshuhn.de
User number: 628024 Copyright 2008, Safari Books Online, LLC.
This PDF is exclusively for your use in accordance with the Safari Terms of Service. No part of it may be reproduced or transmitted in any form by any means without the prior written permission for reprints and excerpts from the publisher. Redistribution or other use that violates the fair use priviledge under U.S. copyright laws (see 17 USC107) or that otherwise violates the Safari Terms of Service is strictly prohibited.

Figure 20-8. javax.xml.transform.dom.DOMLocator


```
public interface DOMLocator extends javax.xml.transform.SourceLocator {
// Public Instance Methods
     org.w3c.dom.Node getOriginatingNode( );
}
```

DOMResult javax.xml.transform.dom

Java 1.4

This class is a `Result` implementation that writes XML content by generating a DOM tree to represent that content. If you pass an `org.w3c.dom.Node` to the constructor or to `setNode( )`, the `DOMResult` will create the result tree as a child of the specified node (which should typically be a `Document` or `Element` node). If you do not specify a node, the `DOMResult` will create a new `Document` node when it creates the result tree. You can retrieve this `Document` with `getNode( )`. In Java 5.0, you can also pass two `Node` objects to the constructor: these specify the parent node of the result tree and the child of that parent before which the result tree should be inserted. See also `setNextSibling( )`.

Figure 20-9. javax.xml.transform.dom.DOMResult


```
public class DOMResult implements javax.xml.transform.Result {
// Public Constructors
     public DOMResult( );
     public DOMResult(org.w3c.dom.Node node);
5.0  public DOMResult(org.w3c.dom.Node node, org.w3c.dom.Node nextSibling);
     public DOMResult(org.w3c.dom.Node node, String systemId);
5.0  public DOMResult(org.w3c.dom.Node node, org.w3c.dom.Node nextSibling,
       String systemId);
// Public Constants
     public static final String FEATURE;  ="http://javax.xml.transform.dom.DOMResult/feature"
// Public Instance Methods
5.0  public org.w3c.dom.Node getNextSibling( );           default:null
     public org.w3c.dom.Node getNode( );             default:null
5.0  public void setNextSibling(org.w3c.dom.Node nextSibling);
     public void setNode(org.w3c.dom.Node node);
// Methods Implementing Result
     public String getSystemId( );                      default:null
     public void setSystemId(String systemId);
}
```

DOMSource javax.xml.transform.dom

Java in a Nutshell, 5th Edition By David Flanagan ISBN: 0596007736 Publisher: O'Reilly
Print Publication Date: 2005/03/01
Prepared for Ronald Fischer, Safari ID: ronald.fischer@fusshuhn.de
User number: 628024 Copyright 2008, Safari Books Online, LLC.
This PDF is exclusively for your use in accordance with the Safari Terms of Service. No part of it may be reproduced or transmitted in any form by any means without the prior written permission for reprints and excerpts from the publisher. Redistribution or other use that violates the fair use priviledge under U.S. copyright laws (see 17 USC107) or that otherwise violates the Safari Terms of Service is strictly prohibited.

Java 1.4

This class is a `Source` implementation that reads an XML document from a DOM document tree or subtree. Pass the `org.w3c.dom.Node` object that represents the root of the tree or subtree to the constructor or to `setNode( )`. When possible, it is also useful to provide a system id (a filename or URL) for use in error messages and for resolving relative URLs contained in the document.

Figure 20-10. javax.xml.transform.dom.DOMSource


```
public class DOMSource implements javax.xml.transform.Source {
// Public Constructors
     public DOMSource( );
     public DOMSource(org.w3c.dom.Node n);
     public DOMSource(org.w3c.dom.Node node, String systemID);
// Public Constants
     public static final String FEATURE;    ="http://javax.xml.transform.dom.DOMSource/feature"
// Public Instance Methods
     public org.w3c.dom.Node getNode( );                default:null
     public void setNode(org.w3c.dom.Node node);
// Methods Implementing Source
     public String getSystemId( );                      default:null
     public void setSystemId(String systemID);
}
```

Package javax.xml.transform.sax

Java 1.4

This package defines `Source` and `Result` implementations that work with SAX events. In addition, it includes an extension to the `TransformerFactory` class that has additional methods for returning `TemplatesHandler` and `TransformerHandler` objects. These objects implement SAX handler interfaces and are able to work with a SAX parser object to turn a series of SAX parse events into a `Templates` object or into a `Result` document. `SAXSource` and `SAXResult` adapt the `org.xml.sax` framework for use in the `javax.xml.transform` framework. By contrast, `SAXTransformerFactory`, `TemplatesHandler`, and `TransfomerHandler` adapt the `javax.xml.transform` framework for use within the `org.xml.sax` parsing framework.

Java in a Nutshell, 5th Edition By David Flanagan ISBN: 0596007736 Publisher: O'Reilly
Print Publication Date: 2005/03/01
Prepared for Ronald Fischer, Safari ID: ronald.fischer@fusshuhn.de
User number: 628024 Copyright 2008, Safari Books Online, LLC.
This PDF is exclusively for your use in accordance with the Safari Terms of Service. No part of it may be reproduced or transmitted in any form by any means without the prior written permission for reprints and excerpts from the publisher. Redistribution or other use that violates the fair use priviledge under U.S. copyright laws (see 17 USC107) or that otherwise violates the Safari Terms of Service is strictly prohibited.

Interfaces

```
public interface TemplatesHandler extends org.xml.sax.ContentHandler;
public interface TransformerHandler extends org.xml.sax.ContentHandler,
   org.xml.sax.DTDHandler, org.xml.sax.ext.LexicalHandler;
```

Classes

```
public class SAXResult implements javax.xml.transform.Result;
public class SAXSource implements javax.xml.transform.Source;
public abstract class SAXTransformerFactory
   extends javax.xml.transform.TransformerFactory;
```

SAXResult javax.xml.transform.sax

Java 1.4

This class is a `Result` implementation that describes the content of a transformed document by triggering the methods of the specified `ContentHandler`. That is, a `SAXResult` acts like a `org.xml.sax.SAXReader` object, invoking the methods of the specified `org.xml.sax.ContentHandler` object as it parses the transformed document. You may also provide a `org.xml.sax.ext.LexicalHandler` object whose methods will be invoked by the `SAXResult` by calling `setLexicalHandler`, or by suppling a `ContentHandler` object that also implements the `LexicalHandler` interface.

Figure 20-11. javax.xml.transform.sax.SAXResult


```
public class SAXResult implements javax.xml.transform.Result {
// Public Constructors
     public SAXResult( );
     public SAXResult(org.xml.sax.ContentHandler handler);
// Public Constants
     public static final String FEATURE;  ="http://javax.xml.transform.sax.SAXResult/feature"
// Public Instance Methods
     public org.xml.sax.ContentHandler getHandler( );              default:null
     public org.xml.sax.ext.LexicalHandler getLexicalHandler( );   default:null
     public void setHandler(org.xml.sax.ContentHandler handler);
     public void setLexicalHandler(org.xml.sax.ext.LexicalHandler handler);
// Methods Implementing Result
     public String getSystemId( );                                 default:null
     public void setSystemId(String systemId);
}
```

SAXSource javax.xml.transform.sax

Java in a Nutshell, 5th Edition By David Flanagan ISBN: 0596007736 Publisher: O'Reilly
Print Publication Date: 2005/03/01
Prepared for Ronald Fischer, Safari ID: ronald.fischer@fusshuhn.de
User number: 628024 Copyright 2008, Safari Books Online, LLC.
This PDF is exclusively for your use in accordance with the Safari Terms of Service. No part of it may be reproduced or transmitted in any form by any means without the prior written permission for reprints and excerpts from the publisher. Redistribution or other use that violates the fair use priviledge under U.S. copyright laws (see 17 USC107) or that otherwise violates the Safari Terms of Service is strictly prohibited.

Java 1.4

This class is a `Source` implementation that describes a document represented as a series of SAX event method calls. A `SAXSource` requires an `org.xml.sax.InputSource` object that describes the stream to parse, and may optionally specify the `org.xml.sax.XMLReader` or `org.xml.sax.XMLFilter` that generates the SAX events. (If no `XMLReader` or `XMLFilter` is specified, then the `Transformer` object will a default `XMLReader`.) Note that since an `InputSource` is required, a `SAXSource` does not behave significantly differently than a `StreamSource` unless an `XMLFilter` is used.

`SAXSource` also has one static method, `sourceToInputSource( )` which returns a SAX `InputSource` method derived from the specified `Source` object, or `null` if the specified `Source` cannot be converted to an `InputSource`.

Figure 20-12. javax.xml.transform.sax.SAXSource


```
public class SAXSource implements javax.xml.transform.Source {
// Public Constructors
     public SAXSource( );
     public SAXSource(org.xml.sax.InputSource inputSource);
     public SAXSource(org.xml.sax.XMLReader reader, org.xml.sax.InputSource inputSource);
// Public Constants
     public static final String FEATURE;   ="http://javax.xml.transform.sax.SAXSource/feature"
// Public Class Methods
     public static org.xml.sax.InputSource sourceToInputSource(javax.xml.transform.Source source);
// Public Instance Methods
     public org.xml.sax.InputSource getInputSource( );        default:null
     public org.xml.sax.XMLReader getXMLReader( );            default:null
     public void setInputSource(org.xml.sax.InputSource inputSource);
     public void setXMLReader(org.xml.sax.XMLReader reader);
// Methods Implementing Source
     public String getSystemId( );                            default:null
     public void setSystemId(String systemId);
}
```

SAXTransformerFactory javax.xml.transform.sax

Java 1.4

This class extends `TransformerFactory` to define additional factory methods that are useful when working with documents that are represented as sequences of SAX events. Pass the `FEATURE` constant to the `getFeature( )` method of your `TransformerFactory` object to determine whether the `newTemplatesHandler( )`

Java in a Nutshell, 5th Edition By David Flanagan ISBN: 0596007736 Publisher: O'Reilly
Print Publication Date: 2005/03/01
Prepared for Ronald Fischer, Safari ID: ronald.fischer@fusshuhn.de
User number: 628024 Copyright 2008, Safari Books Online, LLC.
This PDF is exclusively for your use in accordance with the Safari Terms of Service. No part of it may be reproduced or transmitted in any form by any means without the prior written permission for reprints and excerpts from the publisher. Redistribution or other use that violates the fair use priviledge under U.S. copyright laws (see 17 USC107) or that otherwise violates the Safari Terms of Service is strictly prohibited.

and `newTransformerHandler( )` methods are supported and whether it is safe to cast your `TransformerFactory` object to a `SAXTransformerFactory`. Use the `FEATURE_XMLFILTER` constant with `getFeature( )` to determine if the `newXMLFilter( )` methods are also supported.

`newTemplatesHandler( )` returns a `TemplatesHandler` object that you can use as an `org.xml.sax.ContentHandler` object to receive SAX events generated by a SAX parser and transform those events into a `Templates` object.

The `newTransformerHandler( )` methods are similar: they return a `TransformerHandler` object that can receive SAX events and representing a source document and transform them into a `Result` document. The no-argument version of `newTransformerHandler( )` creates a `TransformerHandler` that simply modifies the form of the document without applying a stylesheet to its content. The other two versions of `newTransformerHandler( )` use a stylesheet specified either as a `Source` or `Templates` object.

The `newXMLFilter( )` methods, if supported, return an `org.xml.sax.XMLFilter` object that can acts as both a sink and a source of SAX events and filters those events by applying the transformation instructions specified by the `Templates` or `Source` objects.

Figure 20-13. javax.xml.transform.sax.SAXTransformerFactory


```
public abstract class SAXTransformerFactory extends javax.xml.transform.TransformerFactory {
// Protected Constructors
     protected SAXTransformerFactory( );
// Public Constants
     public static final String FEATURE;    ="http://javax.xml.transform.sax.SAXTransformerFactory/feature"
     public static final String FEATURE_XMLFILTER;
         ="http://javax.xml.transform.sax.SAXTransformerFactory/feature/xmlfilter"
// Public Instance Methods
     public abstract TemplatesHandler newTemplatesHandler( )
        throws javax.xml.transform.TransformerConfigurationException;
     public abstract TransformerHandler newTransformerHandler( )
        throws javax.xml.transform.TransformerConfigurationException;
     public abstract TransformerHandler newTransformerHandler
        (javax.xml.transform.Source src)
        throws javax.xml.transform.TransformerConfigurationException;
     public abstract TransformerHandler newTransformerHandler
        (javax.xml.transform.Templates templates)
        throws javax.xml.transform.TransformerConfigurationException;
     public abstract org.xml.sax.XMLFilter newXMLFilter
        (javax.xml.transform.Source src)
        throws javax.xml.transform.TransformerConfigurationException;
     public abstract org.xml.sax.XMLFilter newXMLFilter
        (javax.xml.transform.Templates templates)
        throws javax.xml.transform.TransformerConfigurationException;
}
```


Java in a Nutshell, 5th Edition By David Flanagan ISBN: 0596007736 Publisher: O'Reilly
Print Publication Date: 2005/03/01
Prepared for Ronald Fischer, Safari ID: ronald.fischer@fusshuhn.de
User number: 628024 Copyright 2008, Safari Books Online, LLC.
This PDF is exclusively for your use in accordance with the Safari Terms of Service. No part of it may be reproduced or transmitted in any form by any means without the prior written permission for reprints and excerpts from the publisher. Redistribution or other use that violates the fair use priviledge under U.S. copyright laws (see 17 USC107) or that otherwise violates the Safari Terms of Service is strictly prohibited.

TemplatesHandler javax.xml.transform.sax

Java 1.4

This interface extends `org.xml.sax.ContentHandler` and adds a `getTemplates( )` method. An object that implements this interface can be used to receive method calls from some source of SAX events and process those events (as a XSL stylesheet) into a `Templates` object. Obtain a `TemplatesHandler` from a `SAXTransformerFactory`. Register it with the `setContentHandler( )` method of an `org.xml.sax.XMLReader` and invoke the `parse( )` method of the reader. When `parse( )` returns, call the `getTemplates( )` method to obtain the `Templates` object.

Figure 20-14. javax.xml.transform.sax.TemplatesHandler


```
public interface TemplatesHandler extends org.xml.sax.ContentHandler {
// Public Instance Methods
     String getSystemId( );
     javax.xml.transform.Templates getTemplates( );
     void setSystemId(String systemID);
}
```

Returned By

`SAXTransformerFactory.newTemplatesHandler( )`

TransformerHandler javax.xml.transform.sax

Java 1.4

This interface extends `org.xml.sax.ContentHandler` and related interfaces so that it can consume SAX events generated by a `org.xml.sax.SAXReader` or `org.xml.sax.SAXFilter`. Create a `TransformerHandler` by calling one of the `newTransformerHandler( )` methods of a `SAXTransformerFactory`.

Next, call the `setResult( )` method to specify a `Result` object that describes the result document you'd like the transformation to produce. You may also call `getTransformer( )` to get the `Transformer` object associated with this `TransformerHandler` if you need to set output properties or parameter values for the transformation.

Java in a Nutshell, 5th Edition By David Flanagan ISBN: 0596007736 Publisher: O'Reilly
Print Publication Date: 2005/03/01
Prepared for Ronald Fischer, Safari ID: ronald.fischer@fusshuhn.de
User number: 628024 Copyright 2008, Safari Books Online, LLC.
This PDF is exclusively for your use in accordance with the Safari Terms of Service. No part of it may be reproduced or transmitted in any form by any means without the prior written permission for reprints and excerpts from the publisher. Redistribution or other use that violates the fair use priviledge under U.S. copyright laws (see 17 USC107) or that otherwise violates the Safari Terms of Service is strictly prohibited.

Now, register the `TransformerHandler` with the `SAXReader` or `SAXFilter` object by calling `setContentHandler( )`, `setDTDHandler( )`, and `setProperty( )`. `Then you use the property name "http://www.xml.org/sax/properties/lexical-handler" in the call to `setProperties( )` to register the `TransformerHandler` as a `org.xml.sax.ext.LexicalHandler` for the parser or filter.

Finally, invoke one of the `parse( )` methods on your `XMLReader` or `XMLFilter` object. This will cause the reader or filter to start parsing the source document and translating it into method calls on the `TransformerHandler`. The `TransformerHandler` will transform those calls as specified in the `Templates` or `Source` object (if any) that was passed to the original call to `newTransformerHandler( )` and generate a result document as directed by the `Result` object that was passed to `setResult( )`.

Figure 20-15. javax.xml.transform.sax.TransformerHandler


```
public interface TransformerHandler extends org.xml.sax.ContentHandler,
        org.xml.sax.DTDHandler, org.xml.sax.ext.LexicalHandler {
// Public Instance Methods
     String getSystemId( );
     javax.xml.transform.Transformer getTransformer( );
     void setResult(javax.xml.transform.Result result)
        throws IllegalArgumentException;
     void setSystemId(String systemID);
}
```

Returned By

`SAXTransformerFactory.newTransformerHandler( )`

Package javax.xml.transform.stream

Java 1.4

This package contains `Source` and `Result` implementations that work with files and streams.

Classes

```
public class StreamResult implements javax.xml.transform.Result;
public class StreamSource implements javax.xml.transform.Source;
```


Java in a Nutshell, 5th Edition By David Flanagan ISBN: 0596007736 Publisher: O'Reilly
Print Publication Date: 2005/03/01
Prepared for Ronald Fischer, Safari ID: ronald.fischer@fusshuhn.de
User number: 628024 Copyright 2008, Safari Books Online, LLC.
This PDF is exclusively for your use in accordance with the Safari Terms of Service. No part of it may be reproduced or transmitted in any form by any means without the prior written permission for reprints and excerpts from the publisher. Redistribution or other use that violates the fair use priviledge under U.S. copyright laws (see 17 USC107) or that otherwise violates the Safari Terms of Service is strictly prohibited.

StreamResult javax.xml.transform.stream

Java 1.4

This class is a `Result` implementation that writes a textual representation of a transformed document to stream or file. Because XML documents define their own encoding, it is usually preferable to construct a `StreamResult` using a `File` or `OutputStream` instead of a character-based `Writer` which may use a different encoding than that specified within the document.

Figure 20-16. javax.xml.transform.stream.StreamResult


```
public class StreamResult implements javax.xml.transform.Result {
// Public Constructors
     public StreamResult( );
     public StreamResult(java.io.File f);
     public StreamResult(String systemId);
     public StreamResult(java.io.Writer writer);
     public StreamResult(java.io.OutputStream outputStream);
// Public Constants
     public static final String FEATURE;
         ="http://javax.xml.transform.stream.StreamResult/feature"
// Public Instance Methods
     public java.io.OutputStream getOutputStream( );                    default:null
     public java.io.Writer getWriter( );                                default:null
     public void setOutputStream(java.io.OutputStream outputStream);
     public void setSystemId(java.io.File f);
     public void setWriter(java.io.Writer writer);
// Methods Implementing Result
     public String getSystemId( );                                      default:null
     public void setSystemId(String systemId);
}
```

StreamSource javax.xml.transform.stream

Java 1.4

This class is a `Source` implementation that reads the textual format of an XML document from a file, byte stream, or character stream. Because XML documents declare their own encoding, it is preferable to create a `StreamSource` object from an `InputStream` instead of from a `Reader`, so that the XML processor can correctly handle the declared encoding. When creating a `StreamSource` from a byte stream or character stream, you should provide the "system id" (i.e. the filename or URL) by using one of the two-argument

Java in a Nutshell, 5th Edition By David Flanagan ISBN: 0596007736 Publisher: O'Reilly
Print Publication Date: 2005/03/01
Prepared for Ronald Fischer, Safari ID: ronald.fischer@fusshuhn.de
User number: 628024 Copyright 2008, Safari Books Online, LLC.
This PDF is exclusively for your use in accordance with the Safari Terms of Service. No part of it may be reproduced or transmitted in any form by any means without the prior written permission for reprints and excerpts from the publisher. Redistribution or other use that violates the fair use priviledge under U.S. copyright laws (see 17 USC107) or that otherwise violates the Safari Terms of Service is strictly prohibited.

constructors or by caling `setSystemId( )`. The system id is required if the XML file to be processed includes relative URLs to be resolved.

Figure 20-17. javax.xml.transform.stream.StreamSource


```
public class StreamSource implements javax.xml.transform.Source {
// Public Constructors
     public StreamSource( );
     public StreamSource(java.io.InputStream inputStream);
     public StreamSource(java.io.Reader reader);
     public StreamSource(java.io.File f);
     public StreamSource(String systemId);
     public StreamSource(java.io.Reader reader, String systemId);
     public StreamSource(java.io.InputStream inputStream, String systemId);
// Public Constants
     public static final String FEATURE;
         ="http://javax.xml.transform.stream.StreamSource/feature"
// Public Instance Methods
     public java.io.InputStream getInputStream( );                        default:null
     public String getPublicId( );                                        default:null
     public java.io.Reader getReader( );                                  default:null
     public void setInputStream(java.io.InputStream inputStream);
     public void setPublicId(String publicId);
     public void setReader(java.io.Reader reader);
     public void setSystemId(java.io.File f);
// Methods Implementing Source
     public String getSystemId( );                                        default:null
     public void setSystemId(String systemId);
}
```

Package javax.xml.validation

Java 5.0

This package contains classes for validating XML documents against W3C XML Schema definitions. Implementations may also support additional schema types, such as RELAX NG. Typical usage begins with the `SchemaFactory` class, which parses schema specifications into immutable `Schema` objects. Next, the `Schema` object is used to create a `Validator` with which a document may be validated.

Classes

```
public abstract class Schema;
public abstract class SchemaFactory;
public abstract class SchemaFactoryLoader;
public abstract class TypeInfoProvider;
public abstract class Validator;
public abstract class ValidatorHandler implements org.xml.sax.ContentHandler;
```


Java in a Nutshell, 5th Edition By David Flanagan ISBN: 0596007736 Publisher: O'Reilly
Print Publication Date: 2005/03/01
Prepared for Ronald Fischer, Safari ID: ronald.fischer@fusshuhn.de
User number: 628024 Copyright 2008, Safari Books Online, LLC.
This PDF is exclusively for your use in accordance with the Safari Terms of Service. No part of it may be reproduced or transmitted in any form by any means without the prior written permission for reprints and excerpts from the publisher. Redistribution or other use that violates the fair use priviledge under U.S. copyright laws (see 17 USC107) or that otherwise violates the Safari Terms of Service is strictly prohibited.

Schema javax.xml.validation

Java 5.0

A `Schema` is an immutable opaque parsed representation of a schema. `Schema` objects don't perform validation themselves; instead, they are factories for `Validator` and `ValidatorHandler` objects that can be used to validate individual documents.

```
public abstract class Schema {
// Protected Constructors
     protected Schema( );
// Public Instance Methods
     public abstract Validator newValidator( );
     public abstract ValidatorHandler newValidatorHandler( );
}
```

Passed To

`javax.xml.parsers.DocumentBuilderFactory.setSchema( )`, `javax.xml.parsers.SAXParserFactory.setSchema( )`

Returned By

`javax.xml.parsers.DocumentBuilder.getSchema( )`, `javax.xml.parsers.DocumentBuilderFactory.getSchema( )`, `javax.xml.parsers.SAXParser.getSchema( )`, `javax.xml.parsers.SAXParserFactory.getSchema( )`, `SchemaFactory.newSchema( )`

SchemaFactory javax.xml.validation

Java 5.0

A `SchemaFactory` parses the textual representation of a schema into a `Schema` object. Obtain a `SchemaFactory` with the `newInstance( )` method, passing a string that identifies the type of schema you want to parse. All implementations are required to support the W3C XML Schema language, which is identified by `XMLConstants.W3C_XML_SCHEMA_NS_URL`. Other schema types may also be supported, such as RELAX NG schemas, identified by `XMLConstants.RELAXNG_NS_URL`.

To parse a schema, call the `newSchema( )` method, passing the `File` or `javax.xml.transform.Source` object that identifies the schema contents. For schemas in the W3C XML Schema language, you may also specify an array of `Source`

Java in a Nutshell, 5th Edition By David Flanagan ISBN: 0596007736 Publisher: O'Reilly
Print Publication Date: 2005/03/01
Prepared for Ronald Fischer, Safari ID: ronald.fischer@fusshuhn.de
User number: 628024 Copyright 2008, Safari Books Online, LLC.
This PDF is exclusively for your use in accordance with the Safari Terms of Service. No part of it may be reproduced or transmitted in any form by any means without the prior written permission for reprints and excerpts from the publisher. Redistribution or other use that violates the fair use priviledge under U.S. copyright laws (see 17 USC107) or that otherwise violates the Safari Terms of Service is strictly prohibited.

objects that contain the schema definition. If you call `newSchema( )` with no arguments, a special `Schema` object is returned that expects the document to specify the location of its own W3C XML Schema.

You can configure a `SchemaFactory` before calling `newSchema( )` with `setErrorHandler( )`, `setResourceResolver( )`, `setProperty( )`, and `setFeature( )`.

```
public abstract class SchemaFactory {
// Protected Constructors
     protected SchemaFactory( );
// Public Class Methods
     public static final SchemaFactory newInstance(String schemaLanguage);
// Public Instance Methods
     public abstract org.xml.sax.ErrorHandler getErrorHandler( );
     public boolean getFeature(String name)
        throws org.xml.sax.SAXNotRecognizedException,
        org.xml.sax.SAXNotSupportedException;
     public Object getProperty(String name)
        throws org.xml.sax.SAXNotRecognizedException,
        org.xml.sax.SAXNotSupportedException;
     public abstract org.w3c.dom.ls.LSResourceResolver getResourceResolver( );
     public abstract boolean isSchemaLanguageSupported(String schemaLanguage);
     public abstract Schema newSchema( ) throws org.xml.sax.SAXException;
     public Schema newSchema(javax.xml.transform.Source schema)
        throws org.xml.sax.SAXException;
     public Schema newSchema(java.io.File schema) throws org.xml.sax.SAXException;
     public abstract Schema newSchema(javax.xml.transform.Source[ ] schemas)
        throws org.xml.sax.SAXException;
     public Schema newSchema(java.net.URL schema) throws org.xml.sax.SAXException;
     public abstract void setErrorHandler(org.xml.sax.ErrorHandler errorHandler);
     public void setFeature(String name, boolean value)
throws org.xml.sax.SAXNotRecognizedException, org.xml.sax.SAXNotSupportedException;
     public void setProperty(String name, Object object)
throws org.xml.sax.SAXNotRecognizedException, org.xml.sax.SAXNotSupportedException;
     public abstract void setResourceResolver(org.w3c.dom.ls.LSResourceResolver
        resourceResolver);
}
```

Returned By

`SchemaFactoryLoader.newFactory( )`

SchemaFactoryLoader javax.xml.validation

Java 5.0

This class is used by implementations of the validation API to produce a `SchemaFactory` object for a specified schema type. Applications that use the `javax.xml.validation` package do not need to use this class.

```
public abstract class SchemaFactoryLoader {
// Protected Constructors
     protected SchemaFactoryLoader( );
// Public Instance Methods
     public abstract SchemaFactory newFactory(String schemaLanguage);
}
```


Java in a Nutshell, 5th Edition By David Flanagan ISBN: 0596007736 Publisher: O'Reilly
Print Publication Date: 2005/03/01
Prepared for Ronald Fischer, Safari ID: ronald.fischer@fusshuhn.de
User number: 628024 Copyright 2008, Safari Books Online, LLC.
This PDF is exclusively for your use in accordance with the Safari Terms of Service. No part of it may be reproduced or transmitted in any form by any means without the prior written permission for reprints and excerpts from the publisher. Redistribution or other use that violates the fair use priviledge under U.S. copyright laws (see 17 USC107) or that otherwise violates the Safari Terms of Service is strictly prohibited.

TypeInfoProvider javax.xml.validation

Java 5.0

A `TypeInfoProvider` provides information about the type of the element or attribute currently being processed by a `ValidatorHandler`. This type information is obtained by validating document content against a schema and may be useful to the `ContentHandler` to which the `ValidatorHandler` dispatches its method calls.

```
public abstract class TypeInfoProvider {
// Protected Constructors
     protected TypeInfoProvider( );
// Public Instance Methods
     public abstract org.w3c.dom.TypeInfo getAttributeTypeInfo(int index);
     public abstract org.w3c.dom.TypeInfo getElementTypeInfo( );
     public abstract boolean isIdAttribute(int index);
     public abstract boolean isSpecified(int index);
}
```

Returned By

`ValidatorHandler.getTypeInfoProvider( )`

Validator javax.xml.validation

Java 5.0

A `Validator` object validates an XML document against the `Schema` from which the `Validator` was created. The `validate( )` method performs validation. Specify the document to be validated with a `DOMSource` or `SAXSource` object (from the `javax.xml.transform.dom` or `javax.xml.transform.sax` packages). The `validate( )` method accepts any `javax.xml.transform.Source` object as an argument, but `SAXSource` and `DOMSource` are the only two supported implementations.

The document validation process can also be used to augment the source document by adding the default values of unspecified attributes. If you want to capture this augmented form of the document, pass a `Result` object to the two-argument version of `validate( )`. If the source is a `SAXSource`, the result must be a `SAXResult`, and if the source is a `DOMSource`, the result must be a `DOMResult` object.

Java in a Nutshell, 5th Edition By David Flanagan ISBN: 0596007736 Publisher: O'Reilly
Print Publication Date: 2005/03/01
Prepared for Ronald Fischer, Safari ID: ronald.fischer@fusshuhn.de
User number: 628024 Copyright 2008, Safari Books Online, LLC.
This PDF is exclusively for your use in accordance with the Safari Terms of Service. No part of it may be reproduced or transmitted in any form by any means without the prior written permission for reprints and excerpts from the publisher. Redistribution or other use that violates the fair use priviledge under U.S. copyright laws (see 17 USC107) or that otherwise violates the Safari Terms of Service is strictly prohibited.

If the document is valid, the `validate( )` method returns normally. If the document is not valid, `validate( )` throws an `org.xml.sax.SAXException`. You can alter this somewhat by passing a custom `org.xml.sax.ErrorHandler` to `setErrorHandler( )`. Validation exceptions are first passed to the error handler methods, which may throw the exception or handle them in some other way, such as printing a message. If the error handler does not throw an exception, the `validate( )` method attempts to continue validation. The default error handler ignores exceptions passed to its `warn( )` method but throws exceptions passed to its `error( )` and `fatalError( )` methods.

Before calling `validate( )`, a `Validator` may also be configured with `setResourceResolver( )`, `setFeature( )`, and `setProperty( )`.

```
public abstract class Validator {
// Protected Constructors
     protected Validator( );
// Public Instance Methods
     public abstract org.xml.sax.ErrorHandler getErrorHandler( );
     public boolean getFeature(String name)
        throws org.xml.sax.SAXNotRecognizedException,
        org.xml.sax.SAXNotSupportedException;
     public Object getProperty(String name)
        throws org.xml.sax.SAXNotRecognizedException,
        org.xml.sax.SAXNotSupportedException;
     public abstract org.w3c.dom.ls.LSResourceResolver getResourceResolver( );
     public abstract void reset( );
     public abstract void setErrorHandler(org.xml.sax.ErrorHandler errorHandler);
     public void setFeature(String name, boolean value)
        throws org.xml.sax.SAXNotRecognizedException,
        org.xml.sax.SAXNotSupportedException;
     public void setProperty(String name, Object object)
        throws org.xml.sax.SAXNotRecognizedException,
        org.xml.sax.SAXNotSupportedException;
     public abstract void setResourceResolver(org.w3c.dom.ls.LSResourceResolver
        resourceResolver);
     public void validate(javax.xml.transform.Source source)
        throws org.xml.sax.SAXException, java.io.IOException;
     public abstract void validate(javax.xml.transform.Source source,
        javax.xml.transform.Result result)
        throws org.xml.sax.SAXException, java.io.IOException;
}
```

Returned By

`Schema.newValidator( )`

ValidatorHandler javax.xml.validation

Java 5.0

A `ValidatorHandler` is an `org.xml.sax.ContentHandler` that uses the streaming SAX API to validate an XML document against the `Schema` from which the

Java in a Nutshell, 5th Edition By David Flanagan ISBN: 0596007736 Publisher: O'Reilly
Print Publication Date: 2005/03/01
Prepared for Ronald Fischer, Safari ID: ronald.fischer@fusshuhn.de
User number: 628024 Copyright 2008, Safari Books Online, LLC.
This PDF is exclusively for your use in accordance with the Safari Terms of Service. No part of it may be reproduced or transmitted in any form by any means without the prior written permission for reprints and excerpts from the publisher. Redistribution or other use that violates the fair use priviledge under U.S. copyright laws (see 17 USC107) or that otherwise violates the Safari Terms of Service is strictly prohibited.

`ValidatorHandler` was derived. The `Validator` class can be used to validate a `SAXSource`, but `ValidatorHandler` provides lower-level access to the SAX API.

If the document is not valid, one of the `ContentHandler` methods throws a `SAXException` that propagates up to your code. As with the `Validator` class, you can alter this by specifying a custom `org.xml.sax.ErrorHandler` class.

`ValidatorHandler` can be used as a filter for SAX parsing events. If you pass a `ContentHandler` to `setContentHandler( )`, the `ValidatorHandler` augments the source document with attribute defaults from the schema and invokes the appropriate callback methods on the `ContentHandler` you supply. If you are interested in attribute and element type information provided by the schema, your `ContentHandler` can use the `TypeInfoProvider` obtained from the `ValidatorHandler` `getTypeInfoProvider( )`.

Figure 20-18. javax.xml.validation.ValidatorHandler


```
public abstract class ValidatorHandler implements org.xml.sax.ContentHandler {
// Protected Constructors
     protected ValidatorHandler( );
// Public Instance Methods
     public abstract org.xml.sax.ContentHandler getContentHandler( );
     public abstract org.xml.sax.ErrorHandler getErrorHandler( );
     public boolean getFeature(String name)
throws org.xml.sax.SAXNotRecognizedException, org.xml.sax.SAXNotSupportedException;
     public Object getProperty(String name)
throws org.xml.sax.SAXNotRecognizedException, org.xml.sax.SAXNotSupportedException;
     public abstract org.w3c.dom.ls.LSResourceResolver getResourceResolver( );
     public abstract TypeInfoProvider getTypeInfoProvider( );
     public abstract void setContentHandler(org.xml.sax.ContentHandler receiver);
     public abstract void setErrorHandler(org.xml.sax.ErrorHandler errorHandler);
     public void setFeature(String name, boolean value)
throws org.xml.sax.SAXNotRecognizedException, org.xml.sax.SAXNotSupportedException;
     public void setProperty(String name, Object object)
throws org.xml.sax.SAXNotRecognizedException, org.xml.sax.SAXNotSupportedException;
     public abstract void setResourceResolver(org.w3c.dom.ls.LSResourceResolver
        resourceResolver);
}
```

Returned By

`Schema.newValidatorHandler( )`

Package javax.xml.xpath

Java 5.0

Java in a Nutshell, 5th Edition By David Flanagan ISBN: 0596007736 Publisher: O'Reilly
Print Publication Date: 2005/03/01
Prepared for Ronald Fischer, Safari ID: ronald.fischer@fusshuhn.de
User number: 628024 Copyright 2008, Safari Books Online, LLC.
This PDF is exclusively for your use in accordance with the Safari Terms of Service. No part of it may be reproduced or transmitted in any form by any means without the prior written permission for reprints and excerpts from the publisher. Redistribution or other use that violates the fair use priviledge under U.S. copyright laws (see 17 USC107) or that otherwise violates the Safari Terms of Service is strictly prohibited.

This package defines types for the evaluation of XPath expressions in the context of an XML document. XPath is a language for describing a "path" to a node or set of nodes within an XML document. Details of the XPath grammar are beyond the scope of this reference.

A typical use of this package begins with the `XPathFactory`, an instance of which is used to create an `XPath` object. After configuring the `XPath` object as desired, you can use it to evaluate XPath expressions directly or to compile XPath expressions into `XPathExpression` objects for later use.

Interfaces

```
public interface XPath;
public interface XPathExpression;
public interface XPathFunction;
public interface XPathFunctionResolver;
public interface XPathVariableResolver;
```

Classes

```
public class XPathConstants;
public abstract class XPathFactory;
```

Exceptions

```
public class XPathException extends Exception;
   public class XPathExpressionException extends XPathException;
      public class XPathFunctionException extends XPathExpressionException;
   public class XPathFactoryConfigurationException extends XPathException;
```

XPath javax.xml.xpath

Java 5.0

An `XPath` object is used to compile or evaluate an XPath expression. Create an `XPath` object through an `XPathFactory`. Configuration methods of `XPath` allow you to specify an `XPathVariableResolver` and an `XPathFunctionResolver` to resolve variable and function references in XPath expressions. You may also specify the `javax.xml.namespace.NamespaceContext` with which the `XPath` can resolve qualified names.

After creating and configuring an `XPath` object, you can use the `compile( )` method to compile an XPath expression for later evaluation, or you can use one of the `evaluate( )` methods to compile and evaluate an expression directly. There are four versions of `evaluate( )`. All expect a `String` containing an XPath expression as their first argument. The second argument is the document or portion of a document to evaluate the expression against. Two versions of `evaluate( )` expect an `org.xml.sax.InputSource` for this second argument. These versions of the method first parse the document and build a DOM (or other object model) tree. The other two

Java in a Nutshell, 5th Edition By David Flanagan ISBN: 0596007736 Publisher: O'Reilly
Print Publication Date: 2005/03/01
Prepared for Ronald Fischer, Safari ID: ronald.fischer@fusshuhn.de
User number: 628024 Copyright 2008, Safari Books Online, LLC.
This PDF is exclusively for your use in accordance with the Safari Terms of Service. No part of it may be reproduced or transmitted in any form by any means without the prior written permission for reprints and excerpts from the publisher. Redistribution or other use that violates the fair use priviledge under U.S. copyright laws (see 17 USC107) or that otherwise violates the Safari Terms of Service is strictly prohibited.

versions of `evaluate( )` expect an `Object` as the second argument. The object passed should be a DOM (or other object model) object representing the document or some portion of it. For the `org.w3c.dom` object model, this might be a `Document`, `DocumentFragment`, `Node`, or `NodeList` object.

The final difference between `evaluate( )` methods is the presence or absence of a third argument. The two-argument versions of `evaluate( )` return the result of the expression evaluation as a `String`. The three-argument versions expect a third argument that specifies the desired return type and return an `Object` of an appropriate type. The valid types are the `QName` objects defined in the `XPathConstants` class, such as `XPathConstants.NODE` and `XPathConstants.NODESET`. With the DOM object model, `evaluate( )` returns `org.w3c.dom.Node` and `org.w3c.dom.NodeList` objects for these types.

```
public interface XPath {
// Public Instance Methods
     XPathExpression compile(String expression) throws XPathExpressionException;
     String evaluate(String expression, Object item) throws XPathExpressionException;
     String evaluate(String expression, org.xml.sax.InputSource source)
        throws XPathExpressionException;
     Object evaluate(String expression, org.xml.sax.InputSource source,
        javax.xml.namespace.QName returnType)
        throws XPathExpressionException;
     Object evaluate(String expression, Object item, javax.xml.namespace.
        QName returnType) throws XPathExpressionException;
     javax.xml.namespace.NamespaceContext getNamespaceContext( );
     XPathFunctionResolver getXPathFunctionResolver( );
     XPathVariableResolver getXPathVariableResolver( );
     void reset( );
     void setNamespaceContext(javax.xml.namespace.NamespaceContext nsContext);
     void setXPathFunctionResolver(XPathFunctionResolver resolver);
     void setXPathVariableResolver(XPathVariableResolver resolver);
}
```

Returned By

`XPathFactory.newXPath( )`

XPathConstants javax.xml.xpath

Java 5.0

This class defines `javax.xml.namespace.QName` constants that represent the possible return types of the `evaluate( )` methods of `XPath` and `XPathExpression`. It also defines the `DOM_OBJECT_MODEL` constant that can be passed to `XPathFactory.newInstance( )` to specify that the resulting `XPathFactory` should be for the `org.w3c.dom` object model.

```
public class XPathConstants {
// No Constructor
```


Java in a Nutshell, 5th Edition By David Flanagan ISBN: 0596007736 Publisher: O'Reilly
Print Publication Date: 2005/03/01
Prepared for Ronald Fischer, Safari ID: ronald.fischer@fusshuhn.de
User number: 628024 Copyright 2008, Safari Books Online, LLC.
This PDF is exclusively for your use in accordance with the Safari Terms of Service. No part of it may be reproduced or transmitted in any form by any means without the prior written permission for reprints and excerpts from the publisher. Redistribution or other use that violates the fair use priviledge under U.S. copyright laws (see 17 USC107) or that otherwise violates the Safari Terms of Service is strictly prohibited.

```
// Public Constants
     public static final javax.xml.namespace.QName BOOLEAN;
     public static final String DOM_OBJECT_MODEL;
           ="http://java.sun.com/jaxp/xpath/dom"
     public static final javax.xml.namespace.QName NODE;
     public static final javax.xml.namespace.QName NODESET;
     public static final javax.xml.namespace.QName NUMBER;
     public static final javax.xml.namespace.QName STRING;
}
```

XPathException javax.xml.xpath

Java 5.0 ***serializable checked***

This is the common superclass of all XPath-related exception types.

Figure 20-19. javax.xml.xpath.XPathException


```
public class XPathException extends Exception {
// Public Constructors
     public XPathException(Throwable cause);
     public XPathException(String message);
// Public Methods Overriding Throwable
     public Throwable getCause( );
     public void printStackTrace( );
     public void printStackTrace(java.io.PrintWriter s);
     public void printStackTrace(java.io.PrintStream s);
}
```

Subclasses

`XPathExpressionException`, `XPathFactoryConfigurationException`

XPathExpression javax.xml.xpath

Java 5.0

If an XPath expression is to be evaluated more than once, it is not efficient to call the `XPath.evaluate( )` method repeatedly. Instead, compile the expression to an `XPathExpression` using the `XPath.compile( )` method and then evaluate it using one of the `evaluate( )` methods of `XPathExpression`. The `evaluate( )` methods of `XPathExpression` behave just like the corresponding methods of `XPath`. See `XPath` for details.

Java in a Nutshell, 5th Edition By David Flanagan ISBN: 0596007736 Publisher: O'Reilly
Print Publication Date: 2005/03/01
Prepared for Ronald Fischer, Safari ID: ronald.fischer@fusshuhn.de
User number: 628024 Copyright 2008, Safari Books Online, LLC.
This PDF is exclusively for your use in accordance with the Safari Terms of Service. No part of it may be reproduced or transmitted in any form by any means without the prior written permission for reprints and excerpts from the publisher. Redistribution or other use that violates the fair use priviledge under U.S. copyright laws (see 17 USC107) or that otherwise violates the Safari Terms of Service is strictly prohibited.

```
public interface XPathExpression {
// Public Instance Methods
     String evaluate(org.xml.sax.InputSource source)
        throws XPathExpressionException;
     String evaluate(Object item) throws XPathExpressionException;
     Object evaluate(Object item, javax.xml.namespace.QName returnType)
        throws XPathExpressionException;
     Object evaluate(org.xml.sax.InputSource source, javax.xml.namespace.
        QName returnType) throws XPathExpressionException;
}
```

Returned By

`XPath.compile( )`

XPathExpressionException javax.xml.xpath

Java 5.0 ***serializable checked***

Exceptions of this type indicate an error while compiling or evaluating an XPath expression. See the `compile( )` and `evaluate( )` methods of `XPath` and `XPathExpression`.

Figure 20-20. javax.xml.xpath.XPathExpressionException


```
public class XPathExpressionException extends XPathException {
// Public Constructors
     public XPathExpressionException(Throwable cause);
     public XPathExpressionException(String message);
}
```

Subclasses

`XPathFunctionException`

Thrown By

`XPath.{compile( ),evaluate( )},XPathExpression.evaluate( )`

XPathFactory javax.xml.xpath

Java 5.0

The `XPathFactory` class is a factory for creating `XPath` expression evaluators. Call the no-argument version of `newInstance( )` to obtain an `XPathFactory` object that creates `XPath` object to work with DOM documents. The `javax.xml.xpath` package is

Java in a Nutshell, 5th Edition By David Flanagan ISBN: 0596007736 Publisher: O'Reilly
Print Publication Date: 2005/03/01
Prepared for Ronald Fischer, Safari ID: ronald.fischer@fusshuhn.de
User number: 628024 Copyright 2008, Safari Books Online, LLC.
This PDF is exclusively for your use in accordance with the Safari Terms of Service. No part of it may be reproduced or transmitted in any form by any means without the prior written permission for reprints and excerpts from the publisher. Redistribution or other use that violates the fair use priviledge under U.S. copyright laws (see 17 USC107) or that otherwise violates the Safari Terms of Service is strictly prohibited.

nominally object-model independent, however, and you can specify the name of a different object model by calling the one-argument version of `newInstance( )`.

Once you have created an `XPathFactory` object, you can set default function and variable resolvers with `setXPathFunctionResolver( )` and `setXPathVariableResolver( )`. You can configure implementation-dependent features of an `XPathFactory` with `setFeature( )`. All implementations are required to support the `XMLConstants.FEATURE_SECURE_PROCESSING` feature. When this feature is set to true, external functions are not allowed in XPath expressions, and the `XPathFunctionResolver` is not used.

After creating and configuring an `XPathFactory` object, use the `newXPath( )` method to create one or more `XPath` objects for actually evaluating XPath expressions.

```
public abstract class XPathFactory {
// Protected Constructors
     protected XPathFactory( );
// Public Constants
     public static final String DEFAULT_OBJECT_MODEL_URI;   ="http://java.sun.com/jaxp/xpath/dom"
     public static final String DEFAULT_PROPERTY_NAME;      ="javax.xml.xpath.XPathFactory"
// Public Class Methods
     public static final XPathFactory newInstance( );
     public static final XPathFactory newInstance(String uri)
         throws XPathFactoryConfigurationException;
// Public Instance Methods
     public abstract boolean getFeature(String name)
        throws XPathFactoryConfigurationException;
     public abstract boolean isObjectModelSupported(String objectModel);
     public abstract XPath newXPath( );
     public abstract void setFeature(String name, boolean value)
        throws XPathFactoryConfigurationException;
     public abstract void setXPathFunctionResolver(XPathFunctionResolver resolver);
     public abstract void setXPathVariableResolver(XPathVariableResolver resolver);
}
```

XPathFactoryConfigurationException javax.xml.xpath

Java 5.0 ***serializable checked***

This exception is thrown by methods of `XPathFactory` to indicate that a specified object model or feature is not supported.

Figure 20-21. javax.xml.xpath.XPathFactoryConfigurationException


```
public class XPathFactoryConfigurationException extends XPathException {
// Public Constructors
```


Java in a Nutshell, 5th Edition By David Flanagan ISBN: 0596007736 Publisher: O'Reilly
Print Publication Date: 2005/03/01
Prepared for Ronald Fischer, Safari ID: ronald.fischer@fusshuhn.de
User number: 628024 Copyright 2008, Safari Books Online, LLC.
This PDF is exclusively for your use in accordance with the Safari Terms of Service. No part of it may be reproduced or transmitted in any form by any means without the prior written permission for reprints and excerpts from the publisher. Redistribution or other use that violates the fair use priviledge under U.S. copyright laws (see 17 USC107) or that otherwise violates the Safari Terms of Service is strictly prohibited.

```
    public XPathFactoryConfigurationException(Throwable cause);
    public XPathFactoryConfigurationException(String message);
}
```

Thrown By

`XPathFactory.{getFeature( ),newInstance( ),setFeature( )}`

XPathFunction javax.xml.xpath

Java 5.0

This interface defines the invocation API for user-defined XPath functions. Arguments are passed to the `evaluate( )` method as a `java.util.List` and the return value should be an `Object`. `evaluate( )` may throw an `XPathFunctionException`. See also `XPathFunctionResolver`.

```
public interface XPathFunction {
// Public Instance Methods
    Object evaluate(java.util.List args) throws XPathFunctionException;
}
```

Returned By

`XPathFunctionResolver.resolveFunction( )`

XPathFunctionException javax.xml.xpath

Java 5.0 ***serializable checked***

Exceptions of this type may be thrown by user-defined `XPathFunction` implementations. Note that this is a subclass of `XPathExpressionException`.

Figure 20-22. javax.xml.xpath.XPathFunctionException


```
public class XPathFunctionException extends XPathExpressionException {
// Public Constructors
    public XPathFunctionException(Throwable cause);
    public XPathFunctionException(String message);
}
```


Java in a Nutshell, 5th Edition By David Flanagan ISBN: 0596007736 Publisher: O'Reilly
Print Publication Date: 2005/03/01
Prepared for Ronald Fischer, Safari ID: ronald.fischer@fusshuhn.de
User number: 628024 Copyright 2008, Safari Books Online, LLC.
This PDF is exclusively for your use in accordance with the Safari Terms of Service. No part of it may be reproduced or transmitted in any form by any means without the prior written permission for reprints and excerpts from the publisher. Redistribution or other use that violates the fair use priviledge under U.S. copyright laws (see 17 USC107) or that otherwise violates the Safari Terms of Service is strictly prohibited.

Thrown By

`XPathFunction.evaluate( )`

XPathFunctionResolver javax.xml.xpath

Java 5.0

This interface defines a single method to return the `XPathFunction` with the specified qualified name and specified arity (number of arguments). Objects that implement this interface may be passed to the `setXPathFunctionResolver( )` methods of `XPath` or `XPathFactory`.

Note that the function resolvers are invoked only for functions defined in an external namespace, so they cannot be used to override the meaning of XPath's built-in functions or to add new core functions to the XPath language. Also, if the `XMLConstants.FEATURE_SECURE_PROCESSING` feature has been enabled on an `XPathFactory`, user-defined functions are not allowed in XPath expressions, and the `XPathFunctionResolver` is never called.

```
public interface XPathFunctionResolver {
// Public Instance Methods
     XPathFunction resolveFunction(javax.xml.namespace.QName functionName, int arity);
}
```

Passed To

`XPath.setXPathFunctionResolver( ),`
`XPathFactory.setXPathFunctionResolver( )`

Returned By

`XPath.getXPathFunctionResolver( )`

XPathVariableResolver javax.xml.xpath

Java 5.0

This interface defines a single method to return the `Object` value of a variable identified by a qualified name. The value of a named variable is allowed to change between XPath evaluations, but implementations of this interface must ensure that no variable changes *during* the evaluation of an expression. Objects that implement this interface may be passed to the `setXPathVariableResolver( )` methods of `XPath` or `XPathFactory`.

Java in a Nutshell, 5th Edition By David Flanagan ISBN: 0596007736 Publisher: O'Reilly
Print Publication Date: 2005/03/01
Prepared for Ronald Fischer, Safari ID: ronald.fischer@fusshuhn.de
User number: 628024 Copyright 2008, Safari Books Online, LLC.
This PDF is exclusively for your use in accordance with the Safari Terms of Service. No part of it may be reproduced or transmitted in any form by any means without the prior written permission for reprints and excerpts from the publisher. Redistribution or other use that violates the fair use priviledge under U.S. copyright laws (see 17 USC107) or that otherwise violates the Safari Terms of Service is strictly prohibited.

```
public interface XPathVariableResolver {
// Public Instance Methods
     Object resolveVariable(javax.xml.namespace.QName variableName);
}
```

Passed To

`XPath.setXPathVariableResolver( )`,
`XPathFactory.setXPathVariableResolver( )`

Returned By

`XPath.getXPathVariableResolver( )`

Java in a Nutshell, 5th Edition By David Flanagan ISBN: 0596007736 Publisher: O'Reilly
Print Publication Date: 2005/03/01
Prepared for Ronald Fischer, Safari ID: ronald.fischer@fusshuhn.de
User number: 628024 Copyright 2008, Safari Books Online, LLC.
This PDF is exclusively for your use in accordance with the Safari Terms of Service. No part of it may be reproduced or transmitted in any form by any means without the prior written permission for reprints and excerpts from the publisher. Redistribution or other use that violates the fair use priviledge under U.S. copyright laws (see 17 USC107) or that otherwise violates the Safari Terms of Service is strictly prohibited.