

Oracle9i

Supplied Java Packages Reference

Release 2 (9.2)

March 2002

Part No. A96609-01

Oracle9i Supplied Java Packages Reference, Release 2 (9.2)

Part No. A96609-01

Copyright © 1996, 2002 Oracle Corporation. All rights reserved.

Primary Author: Cathy Shea

Contributors: K Karun, Bhushan Khaladkar, Roza Leyderman, Vivek Maganty, Mehta Megna, Jack Melnick, Bhagat Nainani, Denis Raphaely, Jim Warner

The Programs (which include both the software and documentation) contain proprietary information of Oracle Corporation; they are provided under a license agreement containing restrictions on use and disclosure and are also protected by copyright, patent and other intellectual and industrial property laws. Reverse engineering, disassembly or decompilation of the Programs, except to the extent required to obtain interoperability with other independently created software or as specified by law, is prohibited.

The information contained in this document is subject to change without notice. If you find any problems in the documentation, please report them to us in writing. Oracle Corporation does not warrant that this document is error-free. Except as may be expressly permitted in your license agreement for these Programs, no part of these Programs may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without the express written permission of Oracle Corporation.

If the Programs are delivered to the U.S. Government or anyone licensing or using the programs on behalf of the U.S. Government, the following notice is applicable:

Restricted Rights Notice Programs delivered subject to the DOD FAR Supplement are "commercial computer software" and use, duplication, and disclosure of the Programs, including documentation, shall be subject to the licensing restrictions set forth in the applicable Oracle license agreement. Otherwise, Programs delivered subject to the Federal Acquisition Regulations are "restricted computer software" and use, duplication, and disclosure of the Programs shall be subject to the restrictions in FAR 52.227-19, Commercial Computer Software - Restricted Rights (June, 1987). Oracle Corporation, 500 Oracle Parkway, Redwood City, CA 94065.

The Programs are not intended for use in any nuclear, aviation, mass transit, medical, or other inherently dangerous applications. It shall be the licensee's responsibility to take all appropriate fail-safe, backup, redundancy, and other measures to ensure the safe use of such applications if the Programs are used for such purposes, and Oracle Corporation disclaims liability for any damages caused by such use of the Programs.

Oracle is a registered trademark, and JInitiator, Oracle8i, Oracle9i, Oracle Store, PL/SQL, and SQL*Plus are trademarks or registered trademarks of Oracle Corporation. Other names may be trademarks of their respective owners.

Contents

Send Us Your Comments	xv
Preface.....	xvii
About Oracle9i Supplied Java Packages Reference	xviii
Audience	xviii
Organization.....	xviii
Related Documentation	xix
Conventions.....	xx
Documentation Accessibility	xxiii
What's New in Supplied Java Packages?	xxv
New Features Introduced in Supplied Java Packages for Oracle9i Release 2 (9.2)	xxvi
Changes to Scripts that Update Java Classes for JServer	xxviii
New Features Introduced in Supplied Java Packages for <i>Oracle9i</i> Release 1 (9.0.1)	xxix
Part I Java Packages for Oracle9i RDBMS	
1 AppCtxManager in Package oracle.security.rdbms.appctx	
AppCtxManager Description	1-2
AppCtxManager Methods.....	1-3
AppCtxManager Example	1-5

2 Package oracle.AQ

Package oracle.AQ Description.....	2-2
Package oracle.AQ Summary.....	2-5
AQDriverManager.....	2-6
AQSession.....	2-8
AQConstants.....	2-12
AQAgent.....	2-13
AQQueueTableProperty.....	2-15
AQQueueProperty.....	2-20
AQQueueTable.....	2-23
AQQueueAdmin.....	2-27
AQQueue.....	2-35
AQEnqueueOption.....	2-39
AQDequeueOption.....	2-41
AQMessage.....	2-45
AQMessageProperty.....	2-47
AQRawPayload.....	2-51
AQObjectPayload.....	2-53
AQException.....	2-54
AQOracleSQLException.....	2-55

3 Package oracle.AQ.xml

Package oracle.AQ.xml Description.....	3-2
Package oracle.AQ.xml Summary.....	3-5
AQxmlCallback.....	3-6
AQxmlDataSource.....	3-8
AQxmlCallbackContext.....	3-11
AQxmlServlet.....	3-14
AQxmlServlet20.....	3-19
AQxmlDebug.....	3-24
AQxmlException.....	3-26

4 Package oracle.jms

Package oracle.jms Description	4-2
Package oracle.jms Summary	4-4
AdtMessage.....	4-8
AQjmsAdtMessage.....	4-11
AQjmsAgent	4-28
AQjmsBytesMessage	4-32
AQjmsConnection	4-47
AQjmsConnectionMetaData	4-55
AQjmsConstants	4-60
AQjmsConsumer	4-63
AQjmsDestination.....	4-71
AQjmsDestinationProperty	4-81
AQjmsException	4-85
AQjmsFactory	4-87
AQjmsInvalidDestinationException	4-93
AQjmsInvalidSelectorException	4-94
AQjmsMapMessage	4-95
AQjmsMessage	4-111
AQjmsMessageEOFException	4-134
AQjmsMessageFormatException.....	4-135
AQjmsMessageNotReadableException	4-136
AQjmsMessageNotWritableException.....	4-137
AQjmsObjectMessage	4-138
AQjmsOracleDebug.....	4-142
AQjmsProducer	4-144
AQjmsQueueBrowser.....	4-158
AQjmsQueueConnectionFactory.....	4-162
AQjmsQueueReceiver	4-165
AQjmsQueueSender	4-168
AQjmsSession	4-170
AQjmsStreamMessage.....	4-206
AQjmsTextMessage	4-220
AQjmsTopicBrowser	4-224
AQjmsIllegalStateException	4-228

AQjmsTopicConnectionFactory	4-229
AQjmsTopicPublisher.....	4-232
AQjmsTopicReceiver.....	4-236
AQjmsTopicSubscriber.....	4-239
TopicBrowser	4-242
TopicReceiver.....	4-243

5 Package oracle.ODCI

Package oracle.ODCI Description	5-2
Package oracle.ODCI Summary	5-3
ODCIArgDesc.....	5-4
ODCIArgDescList.....	5-7
ODCIArgDescRef	5-10
ODCIColInfo	5-12
ODCIColInfoList	5-15
ODCIColInfoRef.....	5-18
ODCICost	5-20
ODCICostRef.....	5-22
ODCIEnv	5-24
ODCIEnvRef	5-26
ODCIFuncInfo.....	5-28
ODCIFuncInfoRef	5-30
ODCIIndexCtx.....	5-32
ODCIIndexCtxRef	5-34
ODCIIndexInfo	5-36
ODCIIndexInfoRef.....	5-39
ODCIObject	5-41
ODCIObjectList	5-43
ODCIObjectRef.....	5-46
ODCIPartInfo	5-48
ODCIPartInfoRef.....	5-50
ODCIPredInfo	5-52
ODCIPredInfoRef.....	5-54
ODCIQueryInfo	5-56
ODCIQueryInfoRef.....	5-58

ODCIRidList.....	5-60
ODCISStatsOptions	5-63
ODCISStatsOptionsRef	5-65

Part II Java Packages for Oracle9i XDK for Java

6 Package oracle.xml.classgen

Package oracle.xml.classgen Description.....	6-2
Package oracle.xml.classgen Summary.....	6-3
CGDocument Class	6-4
CGNode Class	6-7
CGXSDElement Class.....	6-16
DTDClassGenerator Class	6-20
InvalidContentException Class.....	6-24
oracg Class.....	6-25
SchemaClassGenerator Class	6-26

7 Package oracle.XML.parser.schema

Package oracle.XML.parser.schema Description	7-2
Package oracle.XML.parser.schema Summary	7-3
XMLSchema Class	7-4
XMLSchemaNode	7-8
XSDAttribute Class	7-12
XSDBuilder Class	7-16
XSDComplexType Class.....	7-20
XSDConstants Interface	7-24
XSDConstrainingFacet Class.....	7-25
XSDDataValue Class	7-28
XSDElement Class	7-31
XSDException	7-38
XSDGroup Class	7-39
XSDIdentity Class	7-42
XSDNode Class	7-44
XSDSimpleType Class	7-47

	XSDTypeConstants Interface.....	7-54
	XSDValidator Class	7-59
8	Package oracle.xml.sql.dml	
	Package oracle.xml.sql.dml Description	8-2
	OracleXMLSave Class.....	8-3
9	Package oracle.xml.sql.query	
	Package oracle.xml.sql.query Description	9-2
	OracleXMLQuery Class	9-3
	OracleXMLSQLException Class	9-19
	OracleXMLSQLNoRowsException Class.....	9-22
10	Package oracle.xml.util	
	Package oracle.xml.util Description.....	10-2
	Package oracle.xml.util Summary.....	10-3
	NSName	10-4
	XMLError	10-6
	XMLException	10-18
11	Package oracle.xml.parser.v2	
	Package oracle.xml.parser.v2 Description	11-2
	Package oracle.xml.parser.v2 Summary.....	11-3
	NSResolver Interface.....	11-6
	PrintDriver Interface	11-7
	NSName	11-13
	AttrDecl.....	11-15
	DefaultXMLDocumentHandler	11-21
	DocumentBuilder.....	11-32
	DOMParser.....	11-49
	DTD	11-59
	ElementDecl	11-70
	NodeFactory	11-77
	oraxml.....	11-83

SAXAttrList	11-85
SAXParser	11-95
XMLAttr	11-103
XMLCDATA	11-113
XMLComment	11-116
XMLDeclPI	11-120
XMLDocument	11-126
XMLDocumentFragment	11-153
XMLDOMException	11-155
XMLDOMImplementation	11-156
XMLElement	11-159
XMLEntity	11-177
XMLEntityReference	11-182
XMLError	11-185
XMLNode	11-189
XMLNotation	11-213
XMLNSNode	11-218
XMLOutputStream	11-228
XMLParseException	11-233
XMLParser	11-237
XMLPI	11-249
XMLPrintDriver	11-253
XMLRangeException	11-260
XMLText	11-261
XMLToken Interface	11-267
XMLTokenizer	11-270
JXDocumentBuilder	11-275
JXDocumentBuilderFactory	11-278
JXSAXParser	11-282
JXSAXParserFactory	11-285
JXSAXTransformerFactory	11-288
JXTransformer	11-297
XSLT Processor Classes	11-305
oraxsl Class	11-306
XPathException Class	11-308

XSLException Class	11-310
XSLExtensionElement Class	11-311
XSLProcessor Class	11-314
XSLStylesheet Class	11-323
XSLTContext Class	11-326

Part III Java Packages for Oracle9i XDK for Java Beans

12 Package oracle.xml.async

Package oracle.xml.async Description	12-2
Package oracle.xml.async Summary	12-3
DOMBuilder	12-4
DOMBuilderBeanInfo	12-15
DOMBuilderErrorEvent	12-17
DOMBuilderErrorListener	12-19
DOMBuilderEvent	12-20
DOMBuilderListener	12-22
ResourceManager	12-24
XSLTransformer	12-26
XSLTransformerBeanInfo	12-31
XSLTransformerErrorEvent	12-33
XSLTransformerErrorListener	12-35
XSLTransformerEvent	12-36
XSLTransformerListener	12-38

13 Package oracle.xml.dbviewer

Package oracle.xml.dbviewer Description	13-2
Package oracle.xml.dbviewer Summary	13-3
DBViewer	13-4
DBViewerBeanInfo	13-19

14 Package oracle.xml.differ

Package oracle.xml.differ Description	14-2
XMLDiff Class	14-3

XMLDiffBeanInfo Class.....	14-13
15 Package oracle.xml.srcviewer	
Package oracle.xml.srcviewer Description.....	15-2
XMLSourceView Class	15-3
XMLSourceViewBeanInfo Class.....	15-15
16 Package oracle.xml.transviewer	
Package oracle.xml.transviewer Description.....	16-2
Package oracle.xml.transviewer Summary	16-3
DBAccess	16-4
DBAccessBeanInfo	16-10
XMLTransformPanel	16-11
XMLTransformPanelBeanInfo	16-12
XMLTransViewer	16-13
17 Package oracle.xml.treeviewer	
Package oracle.xml.treeviewer Description.....	17-2
XMLTreeView	17-3
XMLTreeViewBeanInfo	17-6

Part IV Java Packages for Oracle SOAP

18 Package oracle.soap.server	
Package oracle.soap.server Description	18-2
Package oracle.soap.server Summary	18-3
Interface Handler	18-5
Interface Provider	18-9
Interface ProviderManager	18-12
Interface ServiceManager.....	18-16
Class ContainerContext	18-19
Class Logger	18-22
Class ProviderDeploymentDescriptor.....	18-27
Class RequestContext	18-31

Class SOAPServerContext	18-38
Class ServiceDeploymentDescriptor	18-42
Class UserContext	18-51

19 Package oracle.soap.transport

Package oracle.soap.transport Description	19-2
Package oracle.soap.transport Summary	19-3
Interface OracleSOAPTransport	19-4

20 Package oracle.soap.transport.http

Package oracle.soap.transport.http Description	20-2
Package oracle.soap.transport.http Summary	20-3
Class OracleSOAPHTTPConnection	20-5

21 Package oracle.soap.util.xml

Package oracle.soap.util.xml Description	21-2
Package oracle.soap.util.xml Summary	21-3
Class XmlUtils	21-4

Part V Java Packages for Oracle XML DB

22 Package oracle.xdb.dom

Package oracle.xdb.dom Description	22-2
Package oracle.xdb.dom Class Summary	22-3
XDBAttribute Class	22-5
XDBCData Class	22-6
XDBCharData Class	22-7
XDBComment Class	22-8
XDBDocument Class	22-9
XBDDomImplementation Class	22-11
XDBElement Class	22-12
XDBNamedNodeMap Class	22-13
XDBNode Class	22-14
XDBNodeList Class	22-15

XDBProcInst Class	22-16
XDBText Class	22-17
XMLType Class	22-18

23 Package oracle.xdb.spi

Package oracle.xdb.spi Description	23-2
Package oracle.xdb.spi Class Summary	23-3
XDBContext Class	23-4
XDBContextFactory Class	23-5
XDBNameParser Class	23-6
XDBNamingEnumeration Class	23-7
XDBResource Class	23-8
XDBResourceContext Class	23-16

Index

Send Us Your Comments

Oracle9i Supplied Java Packages Reference, Release 2 (9.2)

Part No. A96609-01

Oracle Corporation welcomes your comments and suggestions on the quality and usefulness of this document. Your input is an important part of the information used for revision.

- Did you find any errors?
- Is the information clearly presented?
- Do you need more information? If so, where?
- Are the examples correct? Do you need more examples?
- What features did you like most?

If you find any errors or have any other suggestions for improvement, please indicate the document title and part number, and the chapter, section, and page number (if available). You can send comments to us in the following ways:

- Electronic mail: infodev_us@oracle.com
- FAX: (650) 506-7227 Attn: Server Technologies Documentation Manager
- Postal service:

Oracle Corporation
Server Technologies Documentation
500 Oracle Parkway, Mailstop 4op11
Redwood Shores, CA 94065
USA

If you would like a reply, please give your name, address, telephone number, and (optionally) electronic mail address.

If you have problems with the software, please contact your local Oracle Support Services.

Preface

This preface contains these topics:

- [About Oracle9i Supplied Java Packages Reference](#)
- [Audience](#)
- [Organization](#)
- [Related Documentation](#)
- [Conventions](#)
- [Documentation Accessibility](#)

About *Oracle9i Supplied Java Packages Reference*

Most of the information contained in this book is parsed from the Java source packages to ensure accuracy and completeness of the documentation for the Java APIs. As the book nears completion, updates may be parsed from individual classes (.java files) and inserted within the class hierarchy. If changes are minor, whole packages or individual classes are updated manually. Because the Java packages implement features and functionality that have a range of disparate requirements, the API documentation varies accordingly. Formatting is secondary to reliable information as intended by the developers who create the Java APIs. Therefore, you may notice some variation in the style of how the APIs are presented. For more information about automating documentation for Java APIs, refer to <http://java.sun.com/>.

Audience

Oracle9i Supplied Java Packages Reference is intended for Java programmers and others interested in developing database applications for Oracle9i Release 2 (9.2). This manual assumes a working knowledge of application programming for client/server enterprises and familiarity with Java and SQL to access and manipulate information in relational database systems. To take advantage of Java classes that implement XML features and Oracle XML DB, a familiarity with XML standards as set forth by the World Wide Web Consortium (W3C.org) is also important. Additionally, knowledge of Web development and object-relational database systems is helpful.

Organization

This reference is divided into five parts, each with related chapters, as follows:

Part I, "Java Packages for Oracle9i RDBMS"

This book part contains the chapters for Java packages that implement Java APIs for the Oracle RDBMS.

Part II, "Java Packages for Oracle9i XDK for Java"

This part describes Java packages contained in the Oracle XDK for Java.

Part III, "Java Packages for Oracle9i XDK for Java Beans"

This part describes the Java packages that comprise the Oracle XDK for Java Beans.

Part IV, "Java Packages for Oracle SOAP"

This part contains the chapters for Java packages that implement Oracle SOAP in the XDK for Java.

Part V, "Java Packages for Oracle XML DB"

This part contains the chapters for Java packages that implement features described in the *Oracle9i XML Database Developer's Guide - Oracle XML DB*. These Java APIs provide support for developing XML applications that run natively within Oracle9i in Oracle XML DB.

Related Documentation

For more information, see these Oracle resources:

- *Oracle9i JDBC Developer's Guide and Reference*
- *Oracle9i Application Developer's Guide - Fundamentals*
- *Oracle9i Application Developer's Guide - Advanced Queuing*
- *Oracle9i Data Cartridge Developer's Guide*
- *Oracle9i XML Developer's Kits Guide - XDK*
- *Oracle9i XML Database Developer's Guide - Oracle XML DB*
- *Oracle9i XML API Reference - XDK and Oracle XML DB*

Many of the examples in this book use the sample schemas of the seed database, which is installed by default when you install Oracle. Refer to *Oracle9i Sample Schemas* for information on how these schemas were created and how you can use them yourself.

In North America, printed documentation is available for sale in the Oracle Store at

<http://oraclestore.oracle.com/>

Customers in Europe, the Middle East, and Africa (EMEA) can purchase documentation from

<http://www.oraclebookshop.com/>

Other customers can contact their Oracle representative to purchase printed documentation.

To download free release notes, installation documentation, white papers, or other collateral, please visit the Oracle Technology Network (OTN). You must register online before using OTN; registration is free and can be done at

<http://otn.oracle.com/admin/account/membership.html>

If you already have a username and password for OTN, then you can go directly to the documentation section of the OTN Web site at

<http://otn.oracle.com/docs/index.htm>

To access the database documentation search engine directly, please visit

<http://tahiti.oracle.com>

For additional information, see:

- <http://www.w3c.org>
- <http://http://java.sun.com>

Conventions

This section describes the conventions used in the text and code examples of this documentation set. It describes:

- [Conventions in Text](#)
- [Conventions in Code Examples](#)

Conventions in Text

We use various conventions in text to help you more quickly identify special terms. The following table describes those conventions and provides examples of their use.

Convention	Meaning	Example
Bold	Bold typeface indicates terms that are defined in the text or terms that appear in a glossary, or both.	When you specify this clause, you create an index-organized table .
<i>Italics</i>	Italic typeface indicates book titles or emphasis.	<i>Oracle9i Database Concepts</i> Ensure that the recovery catalog and target database do <i>not</i> reside on the same disk.

Convention	Meaning	Example
UPPERCASE monospace (fixed-width) font	Uppercase monospace typeface indicates elements supplied by the system. Such elements include parameters, privileges, datatypes, RMAN keywords, SQL keywords, SQL*Plus or utility commands, packages and methods, as well as system-supplied column names, database objects and structures, usernames, and roles.	You can specify this clause only for a NUMBER column. You can back up the database by using the BACKUP command. Query the TABLE_NAME column in the USER_TABLES data dictionary view. Use the DBMS_STATS.GENERATE_STATS procedure.
lowercase monospace (fixed-width) font	Lowercase monospace typeface indicates executables, filenames, directory names, and sample user-supplied elements. Such elements include computer and database names, net service names, and connect identifiers, as well as user-supplied database objects and structures, column names, packages and classes, usernames and roles, program units, and parameter values. Note: Some programmatic elements use a mixture of UPPERCASE and lowercase. Enter these elements as shown.	Enter sqlplus to open SQL*Plus. The password is specified in the orapwd file. Back up the datafiles and control files in the /disk1/oracle/dbs directory. The department_id, department_name, and location_id columns are in the hr.departments table. Set the QUERY_REWRITE_ENABLED initialization parameter to true. Connect as oe user. The JRepuTil class implements these methods.
<i>lowercase italic monospace (fixed-width) font</i>	Lowercase italic monospace font represents placeholders or variables.	You can specify the <i>parallel_clause</i> . Run <i>Uold_release</i> .SQL where <i>old_release</i> refers to the release you installed prior to upgrading.

Conventions in Code Examples

Code examples illustrate SQL, PL/SQL, SQL*Plus, or other command-line statements. They are displayed in a monospace (fixed-width) font and separated from normal text as shown in this example:

```
SELECT username FROM dba_users WHERE username = 'MIGRATE';
```

The following table describes typographic conventions used in code examples and provides examples of their use.

Convention	Meaning	Example
[]	Brackets enclose one or more optional items. Do not enter the brackets.	DECIMAL (<i>digits</i> [, <i>precision</i>])
{ }	Braces enclose two or more items, one of which is required. Do not enter the braces.	{ENABLE DISABLE}
	A vertical bar represents a choice of two or more options within brackets or braces. Enter one of the options. Do not enter the vertical bar.	{ENABLE DISABLE} [COMPRESS NOCOMPRESS]
...	Horizontal ellipsis points indicate either: <ul style="list-style-type: none"> That we have omitted parts of the code that are not directly related to the example That you can repeat a portion of the code 	CREATE TABLE ... AS <i>subquery</i> ; SELECT <i>col1</i> , <i>col2</i> , ... , <i>coln</i> FROM <i>employees</i> ;
. . . .	Vertical ellipsis points indicate that we have omitted several lines of code not directly related to the example.	SQL> SELECT NAME FROM V\$DATAFILE; NAME ----- /fsl/dbs/tbs_01.dbf /fsl/dbs/tbs_02.dbf
Other notation	You must enter symbols other than brackets, braces, vertical bars, and ellipsis points as shown.	acctbal NUMBER(11,2); acct CONSTANT NUMBER(4) := 3;
<i>Italics</i>	Italicized text indicates placeholders or variables for which you must supply particular values.	CONNECT SYSTEM/ <i>system_password</i> DB_NAME = <i>database_name</i>
UPPERCASE	Uppercase typeface indicates elements supplied by the system. We show these terms in uppercase in order to distinguish them from terms you define. Unless terms appear in brackets, enter them in the order and with the spelling shown. However, because these terms are not case sensitive, you can enter them in lowercase.	SELECT last_name, employee_id FROM <i>employees</i> ; SELECT * FROM USER_TABLES; DROP TABLE hr.employees;

Convention	Meaning	Example
lowercase	<p>Lowercase typeface indicates programmatic elements that you supply. For example, lowercase indicates names of tables, columns, or files.</p> <p>Note: Some programmatic elements use a mixture of UPPERCASE and lowercase. Enter these elements as shown.</p>	<pre>SELECT last_name, employee_id FROM employees; sqlplus hr/hr CREATE USER mjones IDENTIFIED BY ty3MU9;</pre>

Documentation Accessibility

Our goal is to make Oracle products, services, and supporting documentation accessible, with good usability, to the disabled community. To that end, our documentation includes features that make information available to users of assistive technology. This documentation is available in HTML format, and contains markup to facilitate access by the disabled community. Standards will continue to evolve over time, and Oracle Corporation is actively engaged with other market-leading technology vendors to address technical obstacles so that our documentation can be accessible to all of our customers. For additional information, visit the Oracle Accessibility Program Web site at

<http://www.oracle.com/accessibility/>

Accessibility of Code Examples in Documentation JAWS, a Windows screen reader, may not always correctly read the code examples in this document. The conventions for writing code require that closing braces should appear on an otherwise empty line; however, JAWS may not always read a line of text that consists solely of a bracket or brace.

Accessibility of Links to External Web Sites in Documentation This documentation may contain links to Web sites of other companies or organizations that Oracle Corporation does not own or control. Oracle Corporation neither evaluates nor makes any representations regarding the accessibility of these Web sites.

What's New in Supplied Java Packages?

This chapter describes the new features introduced for Supplied Java Packages:

- [New Features Introduced in Supplied Java Packages for Oracle9i Release 2 \(9.2\)](#)
- [Changes to Scripts that Update Java Classes for JServer](#)
- [New Features Introduced in Supplied Java Packages for Oracle9i Release 1 \(9.0.1\)](#)

New Features Introduced in Supplied Java Packages for Oracle9i Release 2 (9.2)

This section lists the features introduced in the Supplied Java Packages for Oracle9i Release 2 (9.2).

New in Java Packages for Oracle RDBMS

Updates to the Java APIs for these features in the Oracle database:

- **Global Context Manager**
- **Advanced Queing**
- **Java Messaging Service**
- **Oracle Data Cartridges**

See: The chapters in *Part I, "Java Packages for Oracle9i RDBMS"*

New in Java Packages for XDK for Java

This section lists new features and additions for packages and classes for XDK for Java.

- **XML Schema Processor for Java**
 - Supports the latest World Wide Web Consortium (W3C.org) XML Schema recommendation.
- **XSQL Servlet**
 - New performance improvement option.
 - Now supports `XPath` attribute.
 - Simplified inclusion of XML from `CLOB` and `VARCHAR2` columns.
 - New action handler to include posted XML.
 - Support for PDF output using Apache FOP. XSQL pages can be combined with the Apache FOP processor to produce Adobe PDF output from any XML content. (FOP is Apache's print formatter that is driven by XSL formatting objects. It reads a formatting object tree and then renders the resulting pages to a specified output.)
 - Support for immediately read values set as cookies.
 - Support for setting multiple parameter values with a single SQL statement.

- **Class Generator for Java**

- New data binding feature added to the DTD Class Generator.
- An XML instance document can now be given as input to load the instance data to the generated classes.
- XSU support for SAX 2.0 and generating the XML schema of a SQL query.
- Support for DOM compression.
- Support for SAX2 Extension is added in the Java XML Parser.
- Support for XML Compression is added in the Java XML Parser.
- Support for JAXP 1.1.
- Oracle TransX Utility for loading data and text.
- XML Schema Processor for Java now supports both LAX mode and STRICT mode.

See: The chapters in *Part II, "Java Packages for Oracle9i XDK for Java"*

New in Java Packages for XDK for Java Beans

This section lists new features and support in packages for XDK for Java Beans.

- **New XMLDiff Bean.**
- **Internal DTD support for the SourceViewer Bean.**

See: The chapters in *Part III, "Java Packages for Oracle9i XDK for Java Beans"*

New in Oracle SOAP in XDK for Java

Updates and additions to Oracle SOAP have been added for this release.

- **New Oracle SOAP APIs.**
- **New support for SOAP services.**

See: The chapters in *Part IV, "Java Packages for Oracle SOAP"*

New Java Packages Introduced for New Oracle XML DB

XMLType Enhancements for Oracle XML DB

The `XMLType` datatype was first introduced in Oracle9i. This datatype is significantly enhanced and extended in Oracle9i Release 2 (9.2) to support the new Oracle XML DB.

- **XMLType Tables**
 - Datatype `XMLType` can now be used to create tables of `XMLType`.
- **XMLType Constructors**
 - Additional `XMLType` constructor methods have been added.
- **W3C XML Schema Support**
 - Extensive XML Schema support has been added in this release to Oracle XML DB.

New Oracle XML DB Repository

The new Oracle XML DB Repository provides a file system and Web access to all database data.

- **Oracle XML DB Resource API (JNDI)**
 - Uses JNDI (Java Naming and Directory Interface) to locate resources, and manage collections.
 - Supports JNDI Service Provider Interface (SPI). This interface works only inside the database server on the JServer platform.

See: The chapters in *Part V, "Java Packages for Oracle XML DB"*

Changes to Scripts that Update Java Classes for JServer

This section is intended for developers who use the Oracle scripts as a template or example for creating their own custom scripts. In this release, some new scripts have been added to consolidate the upgrade process. One new script is the `rdbms/admin/catjava.sql` script. The `catjava.sql` script is run automatically during an upgrade to Oracle9i Release 2 (9.2) if JServer is in the database.

The `cat java.sql` script runs the scripts listed here from `rdbms/admin`, calls individual scripts, and loads the associated Java classes as follows:

- **initapcx.sql**
 - oracle/security/rdbms/server/AppCtx/
- **initjms.sql**
 - javax/jms
 - * oracle/jms
 - * oracle/AQ
- **initsjty.sql**
 - oracle/aurora/sqljtype
- **initsoxx.sql**
 - oracle/CartridgeServices
 - * oracle/ODCI

For your information, the `cat java.sql` script also calls two additional scripts, which load Java classes that implement server functionality. (These are not related to the classes that are documented in Oracle9i Supplied Java Packages Reference.)

- **initcdc.sql**
 - oracle/CDC (Change Data Capture)
- **initqsm.sql**
 - oracle/qsm (Summary Advisor)

New Features Introduced in Supplied Java Packages for Oracle9i Release 1 (9.0.1)

This section lists the features introduced in the Supplied Java Packages for Oracle9i Release 1 (9.0.1).

XDK for Java

- XML Schema Processor for Java.
- XML Parser for Java with DOM 2.0 and SAX 2.0 support.
- Improved XSLT performance.

- Class Generator for Java, including XML Schema based class generator and a DTD based class generator.
- **XSQL Servlet and Pages**
 - Database Bind Variables. Both lexical substitution and true database bind variables are supported for improved performance.
 - PDF output using Apache FOP.
 - Trusted Host support for XSLT Stylesheets. Stylesheets cannot be executed from non-trusted hosts.
 - Full support for Non-Oracle JDBC Drivers. All query, insert, update, and delete operations support both Oracle and Non-Oracle JDBC drivers.
 - Dynamically constructed XSQL Pages. The XSQLRequest API processes programmatically constructed XSQL pages.
 - Custom connection manager. You can now implement your own Connection Manager to handle database connections in any way you like.
 - Inline XML Schema. Optionally can produce an inline XML Schema that describes the structure of XML query results.
 - Default Date Format for Queries. Can supply a date format mask to change the default way date data is formatted.
 - Custom serializers. Create and use custom serializers that control what the XSQL page processor will return to the client and how it will return results.
 - Dynamic stylesheet assignment. Assign stylesheets dynamically based on parameters or the result of a SQL query.
 - Update or delete posted XML. Insert XML, update, and delete.
 - Insert or update only targeted columns. Explicitly list what columns should be included in any insert or update request.
 - Page-request scoped objects. Action handlers can `get/set` objects in the page request context to share state between actions within a page.
 - Access to `ServletContext`. `ServletContext` can be accessed in addition to accessing the `HttpRequest` and `HttpResponse` objects.
- **XDK for Java Beans**
 - `DBViewer` bean. Displays database queries or any XML by applying XSL stylesheets and visualizing the resulting HTML in a scrollable swing panel.

- DBAccess bean. DBAccess bean maintains CLOB tables that hold multiple XML and text documents.

XML SQL Utility (XSU) Features

- Ability to generate XML Schema given an SQL Query.
- Support for XMLType and URI-ref.
- Ability to generate XML as a stream of SAX2 callbacks.
- XML attribute support when generating XML from the database. Provides an easy way of specifying that a particular column or group of columns should be mapped to an XML attribute instead of an XML element.

Part I

Java Packages for Oracle9i RDBMS

This book part contains reference information for Java packages that implement Java APIs for the Oracle RDBMS. The packages described in these chapters provide Oracle-specific extensions to public Java classes and standards.

This part contains these chapters:

- [Chapter 1, "AppCtxManager in Package oracle.security.rdbms.appctx"](#)
- [Chapter 2, "Package oracle.AQ"](#)
- [Chapter 3, "Package oracle.AQ.xml"](#)
- [Chapter 4, "Package oracle.jms"](#)
- [Chapter 5, "Package oracle.ODCI"](#)

AppCtxManager in Package `oracle.security.rdbms.appctx`

This chapter describes the public Java class `AppCtxManager`, which is exposed in package `oracle.security.rdbms.appctx`. The `AppCtxManager` and associated classes work only for `CONTEXT` that is created to be `ACCESSED GLOBALLY` and does not work for other types of `CONTEXT` like, for example, `INITIALIZED GLOBALLY` through LDAP (Lightweight Directory Access Protocol).

This API provides a centralized location to store the developer's application context, enabling applications to set up the user's contexts.

This chapter contains these sections:

- [AppCtxManager Description](#)
- [AppCtxManager Methods](#)
- [AppCtxManager Example](#)

AppCtxManager Description

`AppCtxManager` class manages the Application Context. All calls to methods in this class must be made by the Application-designated class that administers the globally accessed Application Context. The `AppCtxManager` class cannot be instantiated.

Specifically, `AppCtxManager` provides the Oracle Java API for handling the Application Context that can be accessed globally. This API specifies which user-defined Java classes are allowed to administer the globally accessible Application Context namespace. `AppCtxManager` supports Oracle Label Security labels. This feature enables the administrator to manage contexts for large numbers of users and databases in the enterprise.

The book *Oracle9i Application Developer's Guide - Fundamentals* provides detailed information about using globally accessed Application Context and about how it works.

See Also: For more information about this feature, please refer to *Oracle9i Application Developer's Guide - Fundamentals*. Also refer to the documentation for the associated PL/SQL supplied package `DBMS_APPCTX` in *Oracle9i Supplied PL/SQL Packages and Types Reference*.

Class Hierarchy

```
public class AppCtxManager extends java.lang.Object

java.lang.Object
|
+--oracle.security.rdbms.server.AppCtx.AppCtxManager
```

AppCtxManager Methods

Table 1–1 Summary of AppCtxManager Methods

Method	Description
<code>clearContext(AppCtxPermit, String, String, String)</code>	Checks the AppCtxPermit Object and lets the user do a clear Context
<code>createAppCtxPermit()</code>	Returns the AppCtxPermit Object.
<code>setContext(AppCtxPermit, String, String, String, String, String)</code>	Checks the AppCtxPermit Object and lets the user do a set Context.

clearContext(AppCtxPermit, String, String, String)

Description

This method checks the AppCtxPermit Object and lets the user do a clear Context

Syntax

```
public static void clearContext(AppCtxPermit permit, java.lang.String namespace,
java.lang.String client_id, java.lang.String attribute)
```

Parameters

`permit` - AppCtx object that stores information on the Class designed to administer the Application Context.

`namespace` - NameSpace

`client_id` - Client-identifier of the session

`attribute` - Attribute

`username` - Username of the user permitted to see the client

createAppCtxPermit()

Description

This method returns the AppCtxPermit Object.

A user can create a Globally Accessed Context as:

```
CREATE CONTEXT hr using HR.initclass ACCESSED GLOBALLY;
```

When a user intends to administer the HR application context using the Java API, the user is required to use an AppCtxPermit Object. The only Class that is authorized to create a valid AppCtxPermit Object is the HR.initclass Class in the HR Application schema as designated by the CREATE CONTEXT syntax shown above. The AppCtxPermit Object becomes the TRUST point for the administration of the HR Context.

Syntax

```
public static AppCtxPermit createAppCtxPermit()
```

Parameters

None

Returns

AppCtxPermitObject

setContext(AppCtxPermit, String, String, String, String, String)

Description

This method checks the AppCtxPermit Object and lets the user do a set Context.

Syntax

```
public static void setContext(AppCtxPermit permit, java.lang.String namespace,  
java.lang.String attribute, java.lang.String value, java.lang.String username,  
java.lang.String client_id)
```

Parameters

`permit` - AppCtx object that stores information on the Class designed to administer the Application Context.

`namespace` - NameSpace

`attribute` - Attribute

`value` - Value of the Attribute

`username` - Username of the user permitted to see the client

`client_id` - Client-identifier of the session

Returns

None

AppCtxManager Example

The example shown here provides a sample Java class that can be loaded into the database using the loadjava tool.

```
SQL> CREATE CONTEXT ctx1 using ctxj.employee ACCESSED GLOBALLY;

/* The java class */
import java.sql.*;
import oracle.sql.*;
import oracle.jdbc2.*;
import oracle.jdbc.driver.*;
import oracle.security.rdbms.server.AppCtx.*;
import java.util.ResourceBundle;

class Employee
{
    public static void setctx9() throws Exception
    {
        try
        {
            AppCtxPermit appCtxPermit = AppCtxManager.createAppCtxPermit() ;
            AppCtxManager.setContext(appCtxPermit, "Ctx1", "Attr1", "9", "ctxj", "10");
        }
        catch(Exception e)
        {
            e.printStackTrace() ;
            throw new Exception(e.toString());
        }
    }
    public static void clrctx4() throws Exception
    {
        try
        {
            AppCtxPermit appCtxPermit = AppCtxManager.createAppCtxPermit() ;
            AppCtxManager.clearContext(appCtxPermit, "Ctx1", "10", "Attr1") ;
        }
        catch(Exception e)
        {
            e.printStackTrace() ;
            throw new Exception(e.toString());
        }
    }
}
```

```
}  
  
/* load the java class into the database */  
loadjava -resolve -resolver "((* CTXJ) (* SYS))" -v -u ctxj/ctxj  
Employee.java
```

Package oracle.AQ

This chapter describes the Oracle Java interfaces and classes contained in package `oracle.AQ`. These are based on current PL/SQL interfaces for Oracle Advanced Queuing (AQ).

This chapter contains these sections:

- [Package oracle.AQ Description](#)
- [Package oracle.AQ Summary](#)

Package oracle.AQ Description

The Java AQ API supports both the administrative and operational features of Oracle Advanced Queuing. In developing Java programs for messaging applications, you use JDBC to open a connection to the database and then the interfaces in oracle.AQ, which contains the Java AQ API for message queuing. You need not use only PL/SQL interfaces.

Note: If the Java classes are not-preloaded, you can load them by connecting as SYS and loading the \$ORACLE_HOME/rdbms/admin/initjms.sql script.

Accessing Java AQ Classes

The Java AQ classes are located in \$ORACLE_HOME/rdbms/jlib/aqapi.jar. In Oracle9i Release 2 (9.2), rdbms/jlib/*.jar conforms to the JMS 1.0.2b standard published by Sun Microsystems. These classes can be used with any Oracle8i or Oracle9i JDBC driver.

For JDK 1.3 you must include the following classes in the CLASSPATH:

```
$ORACLE_HOME/rdbms/jlib/aqapi13.jar  
$ORACLE_HOME/lib/jndi.jar  
$ORACLE_HOME/jdbc/lib/classes12.zip
```

For JDK 1.2 you must include the following classes in the CLASSPATH:

```
$ORACLE_HOME/rdbms/jlib/aqapi12.jar  
$ORACLE_HOME/lib/jndi.jar  
$ORACLE_HOME/jdbc/lib/classes12.zip
```

For JDK 1.1 you must include the following classes in the CLASSPATH:

```
$ORACLE_HOME/rdbms/jlib/aqapi11.jar  
$ORACLE_HOME/lib/jndi.jar  
$ORACLE_HOME/jdbc/lib/classes111.zip
```

Oracle9i Application Developer's Guide - Advanced Queuing, Appendix A contains more examples in addition to those contained in this chapter.

Setup for oracle.AQ Examples

1. Create an oracle.AQ user named aqjava

An aqjava user is set up as follows:

```
CONNECT sys/change_on_install AS sysdba

DROP USER aqjava CASCADE;
GRANT CONNECT, RESOURCE, AQ_ADMINISTRATOR_ROLE TO aqjava
    IDENTIFIED BY aqjava;
GRANT EXECUTE ON SYS.DBMS_AQADM TO aqjava;
GRANT EXECUTE ON SYS.DBMS_AQ TO aqjava;
GRANT EXECUTE ON SYS.DBMS_AQIN TO aqjava;
CONNECT aqjava/aqjava
```

2. Set up main class

Next, set up the main class from which you will call subsequent examples and handle exceptions. The main class for the examples is named `test_aqjava`.

```
import java.sql.*;
import oracle.AQ.*;

public class test_aqjava
{
    public static void main(String args[])
    {
        AQSession aq_sess = null;

        try
        {
            aq_sess = createSession(args);

            /* now run the test: */
            runTest(aq_sess);
        }
        catch (Exception ex)
        {
            System.out.println("Exception-1: " + ex);
            ex.printStackTrace();
        }
    }
}
```

3. Create an AQ Session

Next, create an AQ Session for the aqjava user as shown in the previous step for AQDriverManager:

```
public static AQSession createSession(String args[])
{
    Connection db_conn;
    AQSession aq_sess = null;

    try
    {

        Class.forName("oracle.jdbc.driver.OracleDriver");
        /* your actual hostname, port number, and SID will
        vary from what follows. Here we use 'dlsun736,' '5521,'
        and 'test,' respectively: */

        db_conn =
            DriverManager.getConnection(
                "jdbc:oracle:thin:@dlsun736:5521:test",
                "aqjava", "aqjava");

        System.out.println("JDBC Connection opened ");
        db_conn.setAutoCommit(false);

        /* Load the Oracle9i AQ driver: */
        Class.forName("oracle.AQ.AQOracleDriver");
        /* Create an AQ Session: */
        aq_sess = AQDriverManager.createAQSession(db_conn);
        System.out.println("Successfully created AQSession ");
    }
    catch (Exception ex)
    {
        System.out.println("Exception: " + ex);
        ex.printStackTrace();
    }
    return aq_sess;
}
```

Package oracle.AQ Summary

Table 2–1 *Package oracle.AQ member summary*

Member	Description
Classes, Common	-
AQConstants	Constants used in AQ operations.
AQAgent	AQ Agent.
AQDriverManager	Driver Manager for various AQ drivers.
AQEnqueueOption	AQ Enqueue options.
AQDequeueOption	AQ Dequeue options.
AQMessageProperty	AQ Message properties.
AQQueueProperty	AQ Queue properties.
AQQueueTableProperty	AQ Queue Table properties.
Classes, Oracle8i (These classes are not described in this manual)	-
AQOracleSession	Oracle server implementation of AQSession.
AQOracleMessage	Oracle Server implementation of AQMessage.
AQOracleDriver	Oracle server implementation of AQDriver.
AQOracleQueue	Oracle server implementation of AQQueue.
AQOracleQueueTable	Oracle server implementation of AQQueueTable.
AQOracleRawPayload	Oracle server implementation of AQRawPayload.
AQOracleObjectPayload	Oracle server implementation of AQObjectPayload.
Exceptions	-
AQException	Raised when the user encounters any error while using the Java AQ API.
AQOracleSQLException	Raised for all errors that occur while performing SQL.

AQDriverManager

The various implementations of the Java AQ API are managed through a driver manager interface named `AQDriverManager`. Both Oracle Lite and Oracle9i have an `AQDriver` that is registered with the `AQDriverManager`. The driver manager is used to create an `AQSession` that can be used to perform messaging tasks.

When the `AQDriverManager.createAQSession()` method is invoked, it calls the appropriate `AQDriver` (amongst the registered drivers) depending on the parameter passed to the `createAQSession()` call.

The Oracle9i `AQDriver` expects a valid JDBC connection to be passed in as a parameter to create an `AQSession`. Users must have the `execute` privilege on the `DBMS_AQIN` package in order to use the AQ Java interfaces. Users can also acquire these rights through the `AQ_USER_ROLE` or the `AQ_ADMINISTRATOR_ROLE`. Users will also need the appropriate system and queue privileges for Oracle9i-style queue tables.

Methods

getDrivers

```
public static java.util.Vector getDrivers()
```

This method Returns the list of drivers registered with the driver manager. It Returns a Vector of strings containing the names of the registered drivers.

getAQSession

```
public static AQSession getAQSession (java.lang.Object conn)  
    throws AQException
```

This method creates an `AQSession`.

Parameter

`conn`

If the user is using the `AQOracleDriver`, then the object passed in must be a valid JDBC connection.

Multithreaded Program Support

Currently Java AQ objects are not thread safe. Therefore, methods on `AQSession`, `AQQueueTable`, `AQQueue` and other AQ objects should not be called concurrently

from different threads. You can pass these objects between threads, but the program must ensure that the methods on these AQ objects are not invoked concurrently.

We recommend that multithreaded programs create a different `AQSession` in each thread (using the same or a different JDBC connection) and get new queue table and queue handles using the `getQueueTable` and `getQueue` methods in `AQSession`.

Loading the Java AQ Driver

To create an `AQSession`, you must first open a JDBC connection. Then you must load the `AQDriver` that you need to use in the application. With Oracle9i, the driver is loaded using the `Class.forName("oracle.AQ.AQOracleDriver")` command.

Note that the driver needs to be loaded only once (before the first `createAQSession` call). Loading the driver multiple times will have no effect. For more information, see ["Setup for oracle.AQ Examples" on page 2-3](#).

Example

```

Connection db_conn;          /* JDBC connection */
AQSession  aq_sess;         /* AQSession */

/* JDBC setup and connection creation: */
Class.forName("oracle.jdbc.driver.OracleDriver");
db_conn = DriverManager.getConnection (
    "jdbc:oracle:oci8:@", "aquser", "aquser");
db_conn.setAutoCommit(false);

/* Load the Oracle9i AQ driver: */
Class.forName("oracle.AQ.AQOracleDriver");
/* Create an AQ Session: */
aq_sess = AQDriverManager.createAQSession(db_conn);

```

In general, use only the interfaces and classes that are common to both implementations. This will ensure that your applications are portable between Oracle9i and Oracle Lite AQ implementations.

Additionally, `oracle.AQ` classes should only be used when you need a method that is not available in the common interfaces. Note that since the `AQQueue` interface extends `AQQueueAdmin`, all queue administrative and operational functionality is available via `AQQueue`.

AQSession

Methods

createQueueTable

```
public AQQueueTable createQueueTable(java.lang.String owner,  
                                     java.lang.String name,  
                                     AQQueueTableProperty property) throws AQException
```

This method creates a new queue table in a particular user's schema according to the properties specified in the `AQQueueTableProperty` object passed in.

Parameter	Description
<code>owner</code>	schema (user) in which to create the queue table
<code>q_name</code>	name of the queue table
<code>property</code>	queue table properties

Returns

`AQQueueTable` object

getQueueTable

```
public AQQueueTable getQueueTable(java.lang.String owner,  
                                   java.lang.String name)
```

This method is used to get a handle to an existing queue table.

Parameter	Description
<code>owner</code>	schema (user) in which the queue table resides
<code>name</code>	name of the queue table

Returns

`AQQueueTable` object

createQueue

```
public AQQueue createQueue(AQQueueTable q_table,
                          java.lang.String q_name,
                          AQQueueProperty q_property) throws AQException
```

This method creates a queue in a queue_table with the specified queue properties. It uses the same schema name that was used to create the queue table.

Parameter	Description
q_table	queue table in which to create queue
name	name of the queue to be created
q_property	queue properties

Returns

AQQueue object

getQueue

```
public AQQueue getQueue(java.lang.String owner,
                       java.lang.String name)
```

This method can be used to get a handle to an existing queue.

Parameter	Description
owner	schema (user) in which the queue table resides
name	name of the queue

Returns

AQQueue object

getDB Connection

```
public java.sql.Connection getDBConnection()
```

This method can be used to get the underlying JDBC connection from an AQ session object

This method is available only in the Oracle server implementation of AQSession. Hence the AQSession object must be cast to AQOracleSession before calling this method.

Example

```
AQSession aq_sess;
Connection db_conn = ((AQOracleSession)aq_sess).getDBConnection();
```

listen

```
public AQAgent listen(AQAgent[] agent_list,
                    int wait_time)
```

This method can be used to listen to multiple queues for messages

Parameter	Description
agent_list	List of agents to listen for. * For single consumer queues, the name field of the AQAgent must be set to NULL and the address field must contain the [schema].[queue_name]. * For multi consumer queues, the name field of the AQAgent must be contain the consumer_name and the address field must have the [schema].[queue_name].
wait_time	time-out for the listen call (in seconds). To wait forever, this must be set to AQConstants.WAIT_FOREVER.

Returns

Agent with a message available for consumption

Throws

AQException if listen failed due to time-out (ORA-25254) or another error

Example

1. Create a queue table and a queue

With the runTest class, called from the AQDriverManager main class, create a queue table and queue for the aqjava user.

```
public static void runTest(AQSession aq_sess) throws AQException
{
    AQQueueTableProperty    qtable_prop;
    AQQueueProperty        queue_prop;
    AQQueueTable            q_table;
    AQQueue                 queue;

    /* Create a AQQueueTableProperty object (payload type - RAW): */
```

```
qtable_prop = new AQQueueTableProperty("RAW");

/* Create a queue table called aq_table1 in aqjava schema: */
q_table = aq_sess.createQueueTable ("aqjava", "aq_table1",
    qtable_prop);
System.out.println("Successfully created aq_table1 in aqjava
    schema");

/* Create a new AQQueueProperty object: */
queue_prop = new AQQueueProperty();

/* Create a queue called aq_queue1 in aq_table1: */
queue = aq_sess.createQueue (q_table, "aq_queue1", queue_prop);
System.out.println("Successfully created aq_queue1 in aq_table1");
}
```

2. Get a handle to an existing queue table and queue

```
public static void runTest(AQSession aq_sess) throws AQException
{
    AQQueueTable        q_table;
    AQQueue              queue;

    /* Get a handle to queue table - aq_table1 in aqjava schema: */
    q_table = aq_sess.getQueueTable ("aqjava", "aq_table1");
    System.out.println("Successful getQueueTable");

    /* Get a handle to a queue - aq_queue1 in aqjava schema: */
    queue = aq_sess.getQueue ("aqjava", "aq_queue1");
    System.out.println("Successful getQueue");
}
```

AQConstants

This class contains some constants used in the java AQ API.

Visibility constants

```
VISIBILITY_IMMEDIATE  
public static final int VISIBILITY_IMMEDIATE
```

```
VISIBILITY_ONCOMMIT  
public static final int VISIBILITY_ONCOMMIT
```

Payload type, Object

```
RAW_TYPE_PAYLOAD  
public static final int RAW_TYPE_PAYLOAD
```

Payload type, RAW

```
OBJECT_TYPE_PAYLOAD  
public static final int OBJECT_TYPE_PAYLOAD
```

AQAgent

This object specifies the producer or a consumer of a message.

Constructor

```
public AQAgent(java.lang.String name,  
               java.lang.String address,  
               double protocol)
```

```
public AQAgent(java.lang.String name,  
               java.lang.String address)
```

There are two implementations of the constructor, each of which allocates a new AQAgent with the specified parameters.

Parameter	Description
name	agent name
address	agent address
protocol	agent protocol (required only in the first constructor); default is 0

Methods

getName

```
public java.lang.String getName() throws AQException
```

This method gets the agent name.

setName

```
public void setName(java.lang.String name) throws AQException
```

This method sets the agent name.

Parameter	Description
name	Agent name

getAddress

`public java.lang.String getAddress() throws AQException`
This method gets the agent address.

setAddress

`public void setAddress(java.lang.String address) throws AQException`
This method sets the agent address.

Parameter	Description
address	queue at a specific destination

getProtocol

`public int getProtocol() throws AQException`
This method gets the agent protocol.

setProtocol

`public void setProtocol(int protocol) throws AQException`
This method sets the agent protocol.

Parameter	Description
protocol	Agent protocol

AQQueueTableProperty

This class represents queue table properties.

Constants

```
public static final int NONE
public static final int TRANSACTIONAL
```

Constructor

```
public AQQueueTableProperty(java.lang.String p_type)
```

This method creates an `AQQueueTableProperty` object with default property values and the specified payload type.

Parameter	Description
<code>p_type</code>	payload type: this is "RAW" for queue tables that will contain raw payloads or the object ADT type for queue tables that will contain structured payloads

Methods

getPayloadType

```
public java.lang.String getPayloadType() throws AQException
```

This method Returns "RAW" for raw payloads or the object type for object payloads.

setPayloadType

```
public void setPayloadType(java.lang.String p_type) throws AQException
```

This method is used to set the payload type.

Parameter	Description
<code>p_type</code>	payload type: this is "RAW" for queue tables that will contain raw payloads or the object (ADT) type for queue tables that will contain structured payloads

setStorageClause

`public void setStorageClause(java.lang.String s_clause) throws AQException`
This method is used to set the storage clause to be used to create the queue table.

Parameter	Description
<code>s_clauses</code>	storage parameter: this clause is used in the 'CREATE TABLE' statement when the queue table is created

getSortOrder

`public java.lang.String getSortOrder() throws AQException`
This method gets the sort order that is used.

Returns

The sort order used

setSortOrder

`public void setSortOrder(java.lang.String s_order) throws AQException`
This method sets the sort order to be used.

Parameter	Description
<code>s_order</code>	specifies the columns to be used as the <code>sort_key</code> in ascending order; the string has the format <code><sort_column1, sort_column2></code> ; the allowed columns name are <code>priority</code> and <code>enq_time</code> .

isMulticonsumerEnabled

`public boolean isMulticonsumerEnabled() throws AQException`
This method queries whether the queues created in the table can have multiple consumers per message or not.

Returns

TRUE if the queues created in the table can have multiple consumers per message.

FALSE if the queues created in the table can have only one consumer per message.

setMultiConsumer

`public void setMultiConsumer(boolean enable) throws AQException`

This method determines whether the queues created in the table can have multiple consumers per message or not.

Parameter	Description
enable	FALSE if the queues created in the table can have only one consumer per message TRUE if the queues created in the table can have multiple consumers per message

getMessageGrouping

`public int getMessageGrouping() throws AQException`

This method is used to get the message grouping behavior for the queues in this queue table.

Returns

NONE: each message is treated individually

TRANSACTIONAL: all messages enqueued as part of one transaction are considered part of the same group and can be dequeued as a group of related messages.

setMessageGrouping

`public void setMessageGrouping(int m_grouping) throws AQException`

This method is used to set the message grouping behavior for queues created in this queue table.

Parameter	Description
m_grouping	NONE or TRANSACTIONAL

getComment

`public java.lang.String getComment() throws AQException`

This method gets the queue table comment.

setComment

`public void setComment(java.lang.String qt_comment) throws AQException`
This method sets a comment.

Parameter	Description
<code>qt_comment</code>	comment

getCompatible

`public java.lang.String getCompatible() throws AQException`
This method gets the compatible property.

setCompatible

`public void setCompatible(java.lang.String qt_compatible)`
throws `AQException`
This method sets the compatible property.

Parameter	Description
<code>qt_compatible</code>	compatible property

getPrimaryInstance

`public int getPrimaryInstance() throws AQException`
This method gets the primary instance.

setPrimaryInstance

`public void setPrimaryInstance(int inst) throws AQException`
This method sets the primary instance.

Parameter	Description
<code>inst</code>	primary instance

getSecondaryInstance

`public int getSecondaryInstance() throws AQException`
This method gets the secondary instance.

setSecondaryInstance

public void setSecondaryInstance(int inst) throws AQException
 This method sets the secondary instance.

Parameter	Description
inst	secondary instance

Example

To run this example, first set up the `test_aqjava` class as described in ["Setup for oracle.AQ Examples" on page 2-3](#).

1. Create a queue table property object with raw payload type

```
public static void runTest(AQSession aq_sess) throws AQException
{
    AQueueTableProperty qtable_prop;

    /* Create AQueueTable Property object: */
    qtable_prop = new AQueueTableProperty("RAW");
    qtable_prop.setSortOrder("PRIORITY");
}
```

2. Create a queue table property object with raw payload type (for 8.1 style queues)

```
public static void runTest(AQSession aq_sess) throws AQException
{
    AQueueTableProperty qtable_prop;

    /* Create AQueueTable Property object: */
    qtable_prop = new AQueueTableProperty("RAW");
    qtable_prop.setComment("Qtable with raw payload");
    qtable_prop.setCompatible("8.1");
}
```

3. Create a queue table property object with "PERSON" payload type (ADT type):

```
public static void runTest(AQSession aq_sess) throws AQException
{
    AQueueTableProperty qtable_prop;
    qtable_prop = new AQueueTableProperty("PERSON");
    qtable_prop.setComment("Qtable with Person ADT payload");
    qtable_prop.setMessageGrouping(TRANSACTIONAL);
}
```

AQueueProperty

This class represents queue properties.

Constants

```
public static final int NORMAL_QUEUE
public static final int EXCEPTION_QUEUE
public static final int INFINITE /* infinite retention */
```

Constructor

```
public AQueueProperty()
```

This method creates a new AQueueProperty object with default property values.

Methods

getQueueType

```
public int getQueueType() throws AQueueException
```

This method gets the queue type.

Returns

NORMAL_QUEUE or EXCEPTION_QUEUE

setQueueType

```
public void setQueueType(int q_type) throws AQueueException
```

This method is used to set the queue type.

Parameter	Description
q_type	NORMAL_QUEUE or EXCEPTION_QUEUE

getMaxRetries

```
public int getMaxRetries() throws AQueueException
```

This method gets the maximum retries for dequeue with REMOVE mode.

setMaxRetries

```
public void setMaxRetries(int retries) throws AQException
public void setMaxRetries(Integer retries) throws AQException
```

This method sets the maximum retries for dequeue with REMOVE mode.

Parameter	Description
retries	maximum retries for dequeue with REMOVE mode; specifying NULL will use the default. The default applies to single consumer queues and 8.1. compatible multiconsumer queues. Max_retries is not supported for 8.0 compatible multiconsumer queues.

setRetryInterval

```
public void setRetryInterval(double interval) throws AQException
public void setRetryInterval(Double interval) throws AQException
```

This method sets the retry interval, that is the time before this message is scheduled for processing after an application rollback. Default is 0.

Parameter	Description
interval	retry interval; specifying NULL will use the default

getRetryInterval

```
public double getRetryInterval() throws AQException
```

This method gets the retry interval.

getRetentionTime

```
public double getRetentionTime() throws AQException
```

This method gets the retention time.

setRetentionTime

```
public void setRetentionTime(double r_time) throws AQException
public void setRetentionTime(Double r_time) throws AQException
```

This method gets the retention time.

Parameter

<code>r_time</code>	retention time; specifying <code>NULL</code> will use the default
---------------------	---

getComment

```
public java.lang.String getComment() throws AQException
```

This method gets the queue comment.

setComment

```
public void setComment(java.lang.String qt_comment) throws AQException
```

This method sets the queue comment.

Parameter

Description

<code>qt_comment</code>	queue comment
-------------------------	---------------

Example

To use this example, first set up the `test_aqjava` class as described in the [Setup for oracle.AQ Examples](#) section on [page 2-3](#).

Create a AQQueueProperty object

```
{
    AQQueueProperty    q_prop;
    q_prop = new AQQueueProperty();
    q_prop.setRetentionTime(15); /* set retention time */
    q_prop.setRetryInterval(30); /* set retry interval */
}
```

AQQueueTable

The AQQueueTable interface contains methods for queue table administration.

Methods

getOwner

public java.lang.String getOwner() throws AQException
This method gets the queue table owner.

getName

public java.lang.String getName() throws AQException
This method gets the queue table name.

getProperty

public AQQueueTableProperty getProperty() throws AQException
This method gets the queue table properties.

Returns

AQQueueTableProperty object

drop

public void drop(boolean force) throws AQException
This method drops the current queue table.

Parameter	Description
force	FALSE: this operation will not succeed if there are any queues in the queue table (the default) TRUE: all queues in the queue table are stopped and dropped automatically

alter

public void alter(java.lang.String comment,

```
        int primary_instance,  
        int secondary_instance) throws AQException  
public void alter(java.lang.String comment) throws AQException  
This method is used to alter queue table properties.
```

Parameter	Description
comment	new comment
primary_instance	new value for primary instance
secondary_instance	new value for secondary instance

createQueue

```
public AQQueue createQueue(java.lang.String queue_name,  
        AQQueueProperty q_property) throws AQException  
This method is used to create a queue in this queue table.
```

Parameter	Description
queue_name	name of the queue to be created
q_property	queue properties

Returns

AQQueue object

dropQueue

```
public void dropQueue(java.lang.String queue_name) throws AQException  
This method is used to drop a queue in this queue table.
```

Parameter	Description
queue_name	name of the queue to be dropped

Example

To run this example, first set up the `test_aqjava` class as described in the ["Setup for oracle.AQ Examples"](#) section on page 2-3.

1. Create a queue table and a queue

```
public static void runTest(AQSession aq_sess) throws AQException
{
    AQQueueTableProperty    qtable_prop;
    AQQueueProperty         queue_prop;
    AQQueueTable            q_table;
    AQQueue                 queue;

    /* Create a AQQueueTable property object (payload type - RAW): */
    qtable_prop = new AQQueueTableProperty("RAW");

    /* Create a queue table called aq_table2 in aquser schema: */
    qtable = aq_sess.createQueueTable ("aquser", "aq_table2", qtable_prop);
    System.out.println("Successfully createQueueTable");

    /* Create a new AQQueueProperty object: */
    queue_prop = new AQQueueProperty();

    /* Create a queue called aq_queue2 in aq_table2: */
    queue = qtable.createQueue ("aq_queue2", queue_prop);
    System.out.println("Successful createQueue");
}
```

2. Alter queue table, get properties and drop the queue table

```
{
    AQQueueTableProperty    qtable_prop;
    AQQueueTable            q_table;

    /*Get a handle to the queue table called aq_table2 in aquser schema: */
    q_table = aq_sess.getQueueTable ("aqjava", "aq_table2");
    System.out.println("Successful getQueueTable");
    /* Get queue table properties: */
    qtable_prop = q_table.getProperty();

    /* Alter the queue table: */
    q_table.alter("altered queue table");
}
```

```
    /* Drop the queue table (and automatically drop queues inside it): */  
    q_table.drop(true);  
    System.out.println("Successful drop");  
}
```

Note: Queues can be created via the `AQSession.createQueue` or the `AQQueueTable.createQueue` interfaces. The former expects an `AQQueueTable` object as a parameter in addition to the `queue_name` and queue properties.

AQQueueAdmin

Methods

start

```
public void start(boolean enqueue,  
                 boolean dequeue) throws AQException
```

This method is used to enable enqueue and dequeue on this queue.

Parameter	Description
enqueue	TRUE — enable enqueue on this queue FALSE — leave current setting unchanged
dequeue	TRUE — enable dequeue on this queue FALSE — leave current setting unchanged

startEnqueue

```
public void startEnqueue() throws AQException
```

This method is used to enable enqueue on this queue. This is equivalent to `start(TRUE, FALSE)`

startDequeue

```
public void startDequeue() throws AQException
```

This method is used to enable dequeue on this queue. This is equivalent to `start(FALSE, TRUE)`.

stop

```
public void stop(boolean enqueue,  
                boolean dequeue,  
                boolean wait) throws AQException
```

This method is used to disable enqueue/dequeue on this queue.

Parameter	Description
enqueue	TRUE — disable enqueue on this queue FALSE — leave current setting unchanged
dequeue	TRUE — disable dequeue on this queue FALSE — leave current setting unchanged
wait	TRUE — wait for outstanding transactions to complete FALSE — return immediately either with a success or an error

stopEnqueue

`public void stopEnqueue(boolean wait) throws AQException`

This method is used to disable enqueue on a queue. This is equivalent to `stop(TRUE, FALSE, wait)`.

Parameter	Description
wait	TRUE — wait for outstanding transactions to complete FALSE — return immediately either with a success or an error

stopDequeue

`public void stopDequeue(boolean wait) throws AQException`

This method is used to disable dequeue on a queue. This is equivalent to `stop(FALSE, TRUE, wait)`.

Parameter	Description
wait	TRUE — wait for outstanding transactions to complete FALSE — return immediately either with a success or an error

drop

`public void drop() throws AQException`

This method is used to drop a queue

alterQueue

`public void alterQueue(AQQueueProperty property) throws AQException`

This method is used to alter queue properties

Parameter	Description
property	AQueueProperty object with new property values. Note that only max_retries, retry_delay, retention_time and comment can be altered.

addSubscriber

```
public void addSubscriber(AQAgent subscriber,
    java.lang.String rule) throws AQException
```

This method is used to add a subscriber for this queue.

Parameter	Description
subscriber	the AQAgent on whose behalf the subscription is being defined
rule	a conditional expression based on message properties, and the message data properties

removeSubscriber

```
public void removeSubscriber(AQAgent subscriber) throws AQException
```

This method removes a subscriber from a queue.

Parameter	Description
subscriber	the AQAgent to be removed

alterSubscriber

```
public void alterSubscriber(AQAgent subscriber,
    java.lang.String rule) throws AQException
```

This method alters properties for a subscriber to a queue.

Parameter	Description
subscriber	the AQAgent whose subscription is being altered
rule	a conditional expression based on message properties, the message data properties

grantQueuePrivilege

```
public void grantQueuePrivilege(java.lang.String privilege,
                               java.lang.String grantee,
                               boolean grant_option) throws AQException
public void grantQueuePrivilege(java.lang.String privilege,
                               java.lang.String grantee) throws AQException
```

This method is used to grant queue privileges to users and roles. The method has been overloaded. The second implementation is equivalent to calling the first implementation with `grant_option = FALSE`.

Parameter	Description
<code>privilege</code>	specifies the privilege to be granted: <code>ENQUEUE</code> , <code>DEQUEUE</code> or <code>ALL</code>
<code>grantee</code>	specifies the grantee(s); the grantee(s) can be a user, a role or the <code>PUBLIC</code> roles
<code>grant_option</code>	<code>TRUE</code> — the grantee is allowed to use this method to grant access to others <code>FALSE</code> — default

revokeQueuePrivilege

```
public void revokeQueuePrivilege(java.lang.String privilege,
                                 java.lang.String grantee) throws AQException
```

This method is used to revoke a queue privilege.

Parameter	Description
<code>privilege</code>	specifies the privilege to be revoked: <code>ENQUEUE</code> , <code>DEQUEUE</code> or <code>ALL</code>
<code>grantee</code>	specifies the grantee(s); the grantee(s) can be a user, a role or the <code>PUBLIC</code> roles

schedulePropagation

```
public void schedulePropagation(java.lang.String destination,
                               java.util.Date start_time,
                               java.lang.Double duration,
                               java.lang.String next_time,
                               java.lang.Double latency) throws AQException
```

This method is used to schedule propagation from a queue to a destination identified by a database link.

Parameter	Description
destination	specifies the destination database link. Messages in the source queue for recipients at the destination will be propagated. NULL => destination is the local database and messages will be propagated to all other queues in the local database. Maximum length for this field is 128 bytes. If the name is not fully qualified, the default domain name is used.
start_time	specifies the initial start time for the propagation window for messages from this queue to the destination. NULL => start time is current time.
duration	specifies the duration of the propagation window in seconds. NULL => propagation window is forever or until propagation is unscheduled
next_time	date function to compute the start of the next propagation window from the end of the current window. (e.g use "SYSDATE+ 1 - duration/86400" to start the window at the same time everyday. NULL => propagation will be stopped at the end of the current window
latency	maximum wait, in seconds, in the propagation window for the message to be propagated after it is enqueued. NULL => use default value (60 seconds)

unschedulePropagation

```
public void unschedulePropagation(java.lang.String destination)
    throws AQException
```

This method is used to unschedule a previously scheduled propagation of messages from the current queue to a destination identified by a specific database link.

Parameter	Description
destination	specifies the destination database link. NULL => destination is the local database.

alterPropagationSchedule

```
public void alterPropagationSchedule(java.lang.String destination,
    java.lang.Double duration,
    java.lang.String next_time,
```

`java.lang.Double latency)` throws `AQException`

This method is used to alter a propagation schedule.

Parameter	Description
<code>destination</code>	specifies the destination database link. <code>NULL =></code> destination is the local database.
<code>duration</code>	specifies the duration of the propagation window in seconds. <code>NULL =></code> propagation window is forever or until propagation is unscheduled
<code>next_time</code>	date function to compute the start of the next propagation window from the end of the current window. (e.g use <code>"SYSDATE+ 1 - duration/86400"</code> to start the window at the same time everyday. <code>NULL =></code> propagation will be stopped at the end of the current window
<code>latency</code>	maximum wait, in seconds, in the propagation window for the message to be propagated after it is enqueued. <code>NULL =></code> use default value (60 seconds)

enablePropagationSchedule

```
public void enablePropagationSchedule(java.lang.String destination)
    throws AQException
```

This method is used to enable a propagation schedule.

Parameter	Description
<code>destination</code>	specifies the destination database link. <code>NULL =></code> destination is the local database.

disablePropagationSchedule

```
public void disablePropagationSchedule(java.lang.String destination)
    throws AQException
```

This method is used to disable a propagation schedule.

Parameter	Description
<code>destination</code>	specifies the destination database link. <code>NULL =></code> destination is the local database.

Examples

Set up the `test_aqjava` class. For more information, see ["Setup for oracle.AQ Examples" on page 2-3](#)

1. Create a queue and start enqueue/dequeue

```

{
    AQQueueTableProperty    qtable_prop;
    AQQueueProperty         queue_prop;
    AQQueueTable            q_table;
    AQQueue                 queue;

    /* Create a AQQueueTable property object (payload type - RAW): */
    qtable_prop = new AQQueueTableProperty("RAW");
    qtable_prop.setCompatible("8.1");

    /* Create a queue table called aq_table3 in aqjava schema: */
    q_table = aq_sess.createQueueTable ("aqjava","aq_table3", qtable_prop);
    System.out.println("Successful createQueueTable");

    /* Create a new AQQueueProperty object: */
    queue_prop = new AQQueueProperty();

    /* Create a queue called aq_queue3 in aq_table3: */
    queue = aq_sess.createQueue (q_table, "aq_queue3", queue_prop);
    System.out.println("Successful createQueue");

    /* Enable enqueue/dequeue on this queue: */
    queue.start();
    System.out.println("Successful start queue");

    /* Grant enqueue_any privilege on this queue to user scott: */
    queue.grantQueuePrivilege("ENQUEUE", "scott");
    System.out.println("Successful grantQueuePrivilege");
}

```

2. Create a multi-consumer queue and add subscribers

```

public static void runTest(AQSession aq_sess) throws AQException
{
    AQQueueTableProperty    qtable_prop;
    AQQueueProperty         queue_prop;
    AQQueueTable            q_table;
    AQQueue                 queue;
    AQAgent                 subs1, subs2;
}

```

```
    /* Create a AQQueueTable property object (payload type - RAW): */
    qtable_prop = new AQQueueTableProperty("RAW");
    System.out.println("Successful setCompatible");

    /* Set multiconsumer flag to true: */
    qtable_prop.setMultiConsumer(true);

    /* Create a queue table called aq_table4 in aqjava schema: */
    q_table = aq_sess.createQueueTable ("aqjava","aq_table4", qtable_prop);
    System.out.println("Successful createQueueTable");

    /* Create a new AQQueueProperty object: */
    queue_prop = new AQQueueProperty();

    /* Create a queue called aq_queue4 in aq_table4 */
    queue = aq_sess.createQueue (q_table, "aq_queue4", queue_prop);
    System.out.println("Successful createQueue");

    /* Enable enqueue/dequeue on this queue: */
    queue.start();
    System.out.println("Successful start queue");

    /* Add subscribers to this queue: */
    subs1 = new AQAgent("GREEN", null, 0);
    subs2 = new AQAgent("BLUE", null, 0);

    queue.addSubscriber(subs1, null); /* no rule */
    System.out.println("Successful addSubscriber 1");

    queue.addSubscriber(subs2, "priority < 2"); /* with rule */
    System.out.println("Successful addSubscriber 2");
}
```

AQQueue

This interface supports the operational interfaces of queues. `AQQueue` extends `AQQueueAdmin`. Hence, you can also use administrative functions through this interface.

Methods

`getOwner`

```
public java.lang.String getOwner() throws AQException
```

This method gets the queue owner.

`getName`

```
public java.lang.String getName() throws AQException
```

This method gets the queue name.

`getQueueTableName`

```
public java.lang.String getQueueTableName() throws AQException
```

This method gets the name of the queue table in which the queue resides.

`getProperty`

```
public AQQueueProperty getProperty() throws AQException
```

This method is used to get the queue properties.

Returns

`AQQueueProperty` object

`createMessage`

```
public AQMessage createMessage() throws AQException
```

This method is used to create a new `AQMessage` object that can be populated with data to be enqueued.

Returns

AQMessage object

enqueue

```
public byte[] enqueue(AQEnqueueOption enq_option,  
                    AQMessage message) throws AQException
```

This method is used to enqueue a message in a queue.

Parameter	Description
enq_option	AQEnqueueOption object
message	AQMessage to be enqueued

Returns

Message id of the enqueued message. The AQMessage object's messageId field is also populated after the completion of this call.

dequeue

```
public AQMessage dequeue(AQDequeueOption deq_option)  
    throws AQException
```

This method is used to dequeue a message from a queue.

Parameter	Description
deq_option	AQDequeueOption object

Returns

AQMessage, the dequeued message

dequeue (for queues with Oracle object type payloads - SQL data version)

```
public AQMessage dequeue(AQDequeueOption deq_option, java.lang.Class payload_  
class) throws AQException
```

This method is used to dequeue a message from a queue containing Oracle object payloads. This version must be used if your program uses the SQL Data interface for mapping java classes to Oracle object types.

Parameters

`deq_option` - `AQDequeueOption` object

`payload_class` - the payload dequeued is transformed as an object of this type. The class specified must implement the `SQLData` interface and correspond to the payload type defined for the queue.

Returns

`AQMessage`, the dequeued message

Users are also required to register all java classes that map to ADTs contained in the queue in the `typeMap` of the JDBC connection.

For more information on the `SQLData` interface and registering classes in the type map refer to the JDBC developer's guide.

dequeue (for queues with Oracle object type payloads - Custom Datum version)

```
public AQMessage dequeue(AQDequeueOption deq_option,  
oracle.sql.CustomDatumFactory payload_fact) throws AQException
```

This method is used to dequeue a message from a queue containing Oracle object payloads. This version must be used if your program uses the `CustomDatum` interface for mapping java classes to Oracle object types.

Parameters

`deq_option` - `AQDequeueOption` object

`payload_fact` - This is the `CustomDatum` factory for the class that maps to the SQL ADT type of the payload in the queue. For example, if `Person` is the java class that maps to `PERSON` ADT in the database, then the `CustomDatum` factory for this class can be obtained using `Person.getFactory()`

Returns

`AQMessage` - the dequeued message

For more information on the `CustomDatum` and `CustomDatumFactory` interface and registering classes in the type map refer to the JDBC developer's guide.

dequeue (for queues with Oracle object type payloads - ORADData version)

```
public AQMessage dequeue(AQDequeueOption deq_option,  
oracle.sql.ORADDataFactory payload_fact) throws AQException
```

This method is used to dequeue a message from a queue containing Oracle object payloads. This version must be used if your program uses the `ORADData` interface for mapping java classes to Oracle object types.

Parameters:

`deq_option` - `AQDequeueOption` object

`payload_fact` - This is the `ORADData` factory for the class that maps to the SQL ADT type of the payload in the queue. For example, if `Person` is the java class that maps to `PERSON` ADT in the database, then the `ORADData` factory for this class can be obtained using `Person.getORADDataFactory()`

Returns

`AQMessage` - the dequeued message

For more information on the `ORADData` and `ORADDataFactory` interface and registering classes in the type map refer to the JDBC developer's guide.

getSubscribers

`public AQAgent[] getSubscribers() throws AQException`

This method is used to get a subscriber list for the queue.

Returns

An array of `AQAgents`

AQEnqueueOption

This class is used to specify options available for the enqueue operation.

Constants

```
public static final int DEVIATION_NONE
public static final int DEVIATION_BEFORE
public static final int DEVIATION_TOP
public static final int VISIBILITY_ONCOMMIT
public static final int VISIBILITY_IMMEDIATE
```

Constructors

```
public AQEnqueueOption(int visibility,
                       byte[] relative_msgid,
                       int sequence_deviation)
public AQEnqueueOption()
```

There are two constructors available. The first creates an object with the specified options, the second creates an object with the default options.

Parameter	Description
visibility	VISIBILITY_IMMEDIATE or VISIBILITY_ONCOMMIT (default)
relative_msgid	when DEVIATION_BEFORE is used, this parameter identifies the message identifier of the message before which the current message is to be enqueued
sequence_deviation	DEVIATION_TOP— the message is enqueued ahead of any other messages DEVIATION_BEFORE — the message is enqueued ahead of the message specified by relative_msgid DEVIATION_NONE — default

getVisibility

```
public int getVisibility() throws AQException
```

This method gets the visibility.

Returns

VISIBILITY_IMMEDIATE or VISIBILITY_ONCOMMIT

setVisibility

```
public void setVisibility(int visibility) throws AQException
```

This method sets the visibility.

Parameter	Description
visibility	VISIBILITY_IMMEDIATE or VISIBILITY_ONCOMMIT

getRelMessageId

```
public byte[] getRelMessageId() throws AQException
```

This method gets the relative message id.

getSequenceDeviation

```
public int getSequenceDeviation() throws AQException
```

This method gets the sequence deviation.

setSequenceDeviation

```
public void setSequenceDeviation(int sequence_deviation,  
                                byte[] relative_msgid) throws AQException
```

This method specifies whether the message being enqueued should be dequeued before other message(s) already in the queue.

Parameter	Description
sequence_deviation	DEVIATION_TOP— the message is enqueued ahead of any other messages DEVIATION_BEFORE — the message is enqueued ahead of the message specified by relative_msgid DEVIATION_NONE — default
relative_msgid	when DEVIATION_BEFORE is used, this parameter identifies the message identifier of the message before which the current message is to be enqueued

AQDequeueOption

This class is used to specify the options available for the dequeue option.

Constants

```
public static final int NAVIGATION_FIRST_MESSAGE
public static final int NAVIGATION_NEXT_TRANSACTION
public static final int NAVIGATION_NEXT_MESSAGE
public static final int DEQUEUE_BROWSE
public static final int DEQUEUE_LOCKED
public static final int DEQUEUE_REMOVE
public static final int DEQUEUE_REMOVE_NODATA
public static final int WAIT_FOREVER
public static final int WAIT_NONE
public static final int VISIBILITY_ONCOMMIT
public static final int VISIBILITY_IMMEDIATE
```

Constructor

```
public AQDequeueOption()
```

This method creates an object with the default options.

Methods

getConsumerName

```
public java.lang.String getConsumerName() throws AQException
```

This method gets consumer name.

setConsumerName

```
public void setConsumerName(java.lang.String consumer_name)
    throws AQException
```

This method sets consumer name

Parameter	Description
consumer_name	Agent name

getDequeueMode

`public int getDequeueMode() throws AQException`
This method gets dequeue mode.

Returns

DEQUEUE_BROWSE, DEQUEUE_LOCKED, DEQUEUE_REMOVE or DEQUEUE_REMOVE_NODATA

setDequeueMode

`public void setDequeueMode(int dequeue_mode) throws AQException`
This method sets the dequeue mode.

Parameter	Description
<code>dequeue_mode</code>	DEQUEUE_BROWSE, DEQUEUE_LOCKED, DEQUEUE_REMOVE or DEQUEUE_REMOVE_NODATA

getNavigationMode

`public int getNavigationMode() throws AQException`
This method gets the navigation mode.

Returns

NAVIGATION_FIRST_MESSAGE or NAVIGATION_NEXT_MESSAGE or NAVIGATION_NEXT_TRANSACTION

setNavigationMode

`public void setNavigationMode(int navigation) throws AQException`
This method sets the navigation mode.

Parameter	Description
<code>navigation</code>	NAVIGATION_FIRST_MESSAGE or NAVIGATION_NEXT_MESSAGE or NAVIGATION_NEXT_TRANSACTION

getVisibility

`public int getVisibility() throws AQException`

This method gets the visibility.

Returns

VISIBILITY_IMMEDIATE or VISIBILITY_ONCOMMIT

setVisibility

public void setVisibility(int visibility) throws AQException

This method sets the visibility.

Parameter	Description
visibility	VISIBILITY_IMMEDIATE or VISIBILITY_ONCOMMIT

getWaitTime

public int getWaitTime() throws AQException

This method gets the wait time.

Returns

WAIT_FOREVER or WAIT_NONE or the actual time in seconds

setWaitTime

public void setWaitTime(int wait_time) throws AQException

This method sets the wait time.

Parameter	Description
wait_time	WAIT_FOREVER or WAIT_NONE or time in seconds

getMessageId

public byte[] getMessageId() throws AQException

This method gets the message id.

setMessageId

public void setMessageId(byte[] message_id) throws AQException

This method sets the message id.

Parameter	Description
message_id	message id

getCorrelation

public java.lang.String getCorrelation() throws AQException
This method gets the correlation id.

setCorrelation

public void setCorrelation(java.lang.String correlation)
throws AQException
This method sets the correlation id.

Parameter	Description
correlation	user-supplied information

AQMessage

This interface contains methods for AQ messages with raw or object payloads.

Methods

getMessageId

```
public byte[] getMessageId() throws AQException
```

This method gets the message id.

getRawPayload

```
public AQRawPayload getRawPayload() throws AQException
```

This method gets the raw payload

Returns

AQRawPayload object

setRawPayload

```
public void setRawPayload(AQRawPayload message_payload)  
    throws AQException
```

This method sets the raw payload. It throws AQException if this is called on messages created from object type queues.

Parameter	Description
message_payload	AQRawPayload object containing raw user data

getObjectPayload

```
public AQObjectPayload getObjectPayload() throws AQException
```

Get the object payload

Returns

AQObjectPayload object

setObjectPayload

```
public void setObjectPayload(AQObjectPayload message_payload)
    throws AQException
```

Set the object payload.

Parameter	Description
message_payload	AQObjectPayload object containing object user data. Throws AQException if this is called on Messages created from raw type queues.

getMessageProperty

```
public AQMessageProperty getMessageProperty() throws AQException
```

This method gets the message properties

Returns

AQMessageProperty object

setMessageProperty

```
public void setMessageProperty(AQMessageProperty property)
    throws AQException
```

This method sets the message properties.

Parameter	Description
property	AQMessageProperty object

AQMessageProperty

The AQMessageProperty class contains information that is used by AQ to manage individual messages. The properties are set at enqueue time and their values are returned at dequeue time.

Constants

```
public static final int DELAY_NONE
public static final int EXPIRATION_NEVER
public static final int STATE_READY
public static final int STATE_WAITING
public static final int STATE_PROCESSED
public static final int STATE_EXPIRED
```

Constructor

```
public AQMessageProperty()
```

This method creates the AQMessageProperty object with default property values.

Methods

getPriority

```
public int getPriority() throws AQException
```

This method gets the message priority.

setPriority

```
public void setPriority(int priority) throws AQException
```

This method sets the message priority.

Parameter	Description
priority	priority of the message; this can be any number, including negative number - a smaller number indicates a higher priority

getDelay

```
public int getDelay() throws AQException
```

This method gets the delay value.

setDelay

```
public void setDelay(int delay) throws AQException
```

This method sets delay value.

Parameter	Description
delay	the delay represents the number of seconds after which the message is available for dequeuing; with <code>NO_DELAY</code> the message is available for immediate dequeuing

getExpiration

```
public int getExpiration() throws AQException
```

This method gets expiration value.

setExpiration

```
public void setExpiration(int expiration) throws AQException
```

This method sets expiration value.

Parameter	Description
expiration	the duration the message is available for dequeuing; this parameter is an offset from the delay; if <code>NEVER</code> , the message will not expire

getCorrelation

```
public java.lang.String getCorrelation() throws AQException
```

This method gets correlation.

setCorrelation

```
public void setCorrelation(java.lang.String correlation)
```

```
throws AQException
```

This method sets correlation.

Parameter	Description
correlation	user-supplied information

getAttempts

`public int getAttempts() throws AQException`
 This method gets the number of attempts.

getRecipientList

`public java.util.Vector getRecipientList() throws AQException`
 This method gets the recipient list.

Returns

A vector of `AQAgents`. This parameter is not returned to a consumer at dequeue time.

setRecipientList

`public void setRecipientList(java.util.Vector r_list)`
 throws `AQException`
 This method sets the recipient list.

Parameter	Description
r_list	vector of <code>AQAgents</code> ; the default recipients are the queue subscribers

getOrigMessageId

`public byte[] getOrigMessageId() throws AQException`
 This method gets original message id.

getSender

`public AQAgent getSender() throws AQException`
 This method gets the sender of the message.

setSender

`public void setSender(AQAgent sender) throws AQException`
This method sets the sender of the message.

Parameter	Description
sender	AQAgent

getExceptionQueue

`public java.lang.String getExceptionQueue() throws AQException`
This method gets the exception queue name.

setExceptionQueue

`public void setExceptionQueue(java.lang.String queue)`
`throws AQException`
This method sets the exception queue name.

Parameter	Description
queue	exception queue name

getEnqueueTime

`public java.util.Date getEnqueueTime() throws AQException`
This method gets the enqueue time.

getState

`public int getState() throws AQException`
This method gets the message state.

Returns

STATE_READY or STATE_WAITING or STATE_PROCESSED or STATE_EXPIRED

AQRawPayload

This object represents the raw user data that is included in AQMessage.

Methods

getStream

```
public int getStream(byte[] value, int len) throws AQException
```

This method reads some portion of the raw payload data into the specified byte array.

Parameter	Description
value	byte array to hold the raw data
len	number of bytes to be read

Returns

The number of bytes read

getBytes

```
public byte[] getBytes() throws AQException
```

This method retrieves the entire raw payload data as a byte array.

Returns

byte - the raw payload as a byte array

setStream

```
public void setStream(byte[] value,  
                     int len) throws AQException
```

This method sets the value of the raw payload.

Parameter	Description
value	byte array containing the raw payload

(Cont.) Parameter	Description
len	number of bytes to be written to the raw stream

AQObjectPayload

This object represents the structured user data (for object queues) that is included in the AQMessage

Methods

setPayloadData

```
public void setPayloadData(java.lang.Object obj) throws AQException
```

This method is used to fill in the payload into the AQObjectPayload object

Parameter	Description
obj	User-data to be put. Depending on which AQ driver you use, there may be certain restrictions on the types of objects that can be passed in. The Oracle9i AQ driver accepts objects that implement the SQLData, ORADData, or CustomDatum interface inside the payload.

Please refer to the JDBC developer's guide for more information on SQLData, ORADData and CustomDatum interfaces

getPayloadData

```
public java.lang.Object getPayloadData() throws AQException
```

This method is used to retrieve the message payload from the AQObjectPayload object

Returns

Object payload in message - This will depend on the SQLData class, ORADDataFactory or CustomDatum Factory specified during dequeue.

AQException

```
public class AQException extends java.lang.RuntimeException
```

This exception is raised when the user encounters any error while using the Java AQ API.

This interface supports all methods supported by Java exceptions and some additional methods.

Methods

getMessage

This method gets the error message.

getErrorCode

This method gets the error number (Oracle error code).

getNextException

This method gets the next exception in the chain if any.

AQOracleSQLException

AQOracleSQLException extends AQException.

When using Oracle9i AQ driver, some errors may be raised from the client side and some from the RDBMS. The Oracle9i driver raises AQOracleSQLException for all errors that occur while performing SQL.

For sophisticated users interested in differentiating between the two types of exceptions, this interface might be useful. In general you will only use AQException.

Package oracle.AQ.xml

This chapter describes package oracle.AQ.xml, which contains the classes for Oracle9i Advanced Queuing (AQ) XML Servlet. This servlet is used to access Oracle9i AQ via open protocols like HTTP and SMTP using the Internet Data Access Presentation (iDAP).

The sections in this chapter are as follows:

- [Package oracle.AQ.xml Description](#)
- [Package oracle.AQ.xml Summary](#)

Package oracle.AQ.xml Description

Package oracle.AQ.xml contains classes required by the Oracle9i Advanced Queuing (AQ) XML Servlet.

Oracle9i Application Developer's Guide - Advanced Queuing describes how to develop Java applications for Oracle9i Advanced Queuing.

The common interfaces and classes are based on current PL/SQL interfaces.

- Common interfaces are prefixed with AQ.
- This document describes the common interfaces and their corresponding Oracle9i implementations, which are prefixed with AQOracle.

You use the AQ XML servlet to access Oracle9i AQ using open protocols like HTTP and SMTP and using an XML message format called Internet Data Access Presentation (iDAP).

Using the AQ servlet, a client can perform these actions:

- Send messages to single-consumer queues
- Publish messages to multi-consumer queues/topics
- Receive messages from queues
- Register to receive message notifications

Because the servlet uses JDBC OCI drivers to connect to the Oracle9i database server, the 9i Oracle Client libraries must be installed on the machine that hosts the servlet. The LD_LIBRARY_PATH must contain \$ORACLE_HOME/lib.

The servlet can be created by defining a Java™ class that extends the oracle.AQ.xml.AQxmlServlet or oracle.AQ.xml.AQxmlServlet20 class. These classes in turn extend the javax.servlet.http.HttpServlet class.

The servlet can be deployed in any Web server or Servlet Runner that implements Java™ Servlet 2.0, Java™ Servlet 2.2, or Java™ Servlet 2.3 interfaces as follows:

1. To deploy the AQ Servlet with a webserver that implements Javasoft's Servlet 2.0 interfaces, users must define a class that extends the oracle.AQ.xml.AQxmlServlet20 class.
2. To deploy the AQ Servlet with a webserver that implements Javasoft's Servlet2.2 interfaces, users must define a class that extends the oracle.AQ.xml.AQxmlServlet class.

3. The servlet can be compiled using JDK 1.3, JDK 1.2, or JDK 1.1 libraries as follows.

■ For JDK 1.3 the CLASSPATH must contain:

```
$ORACLE_HOME/jdbc/lib/classes13.zip  
$ORACLE_HOME/jdbc/lib/jta.zip  
$ORACLE_HOME/jdbc/lib/nls_charset13.zip  
$ORACLE_HOME/jdbc/lib/jndi.zip  
$ORACLE_HOME/lib/lclasses13.zip  
$ORACLE_HOME/lib/xmlparserv2.jar  
$ORACLE_HOME/lib/xschem.jar  
$ORACLE_HOME/rdbms/jlib/aqapi.jar  
$ORACLE_HOME/rdbms/jlib/jmscommon.jar  
$ORACLE_HOME/rdbms/jlib/aqxml.jar  
$ORACLE_HOME/rdbms/jlib/xsul3.jar  
$ORACLE_HOME/jis/lib/servlet.jar
```

■ For JDK 1.2.x the CLASSPATH must contain:

```
$ORACLE_HOME/jdbc/lib/classes12.zip  
$ORACLE_HOME/jdbc/lib/jta.zip  
$ORACLE_HOME/jdbc/lib/nls_charset12.zip  
$ORACLE_HOME/jdbc/lib/jndi.zip  
$ORACLE_HOME/lib/lclasses12.zip  
$ORACLE_HOME/lib/xmlparserv2.jar  
$ORACLE_HOME/lib/xschem.jar  
$ORACLE_HOME/rdbms/jlib/aqapi.jar  
$ORACLE_HOME/rdbms/jlib/jmscommon.jar  
$ORACLE_HOME/rdbms/jlib/aqxml.jar  
$ORACLE_HOME/rdbms/jlib/xsul2.jar  
$ORACLE_HOME/jis/lib/servlet.jar
```

■ For JDK 1.1.x the CLASSPATH must contain:

```
$ORACLE_HOME/jdbc/lib/classes111.zip  
$ORACLE_HOME/jdbc/lib/jta.zip  
$ORACLE_HOME/jdbc/lib/nls_charset11.zip  
$ORACLE_HOME/jdbc/lib/jndi.zip  
$ORACLE_HOME/lib/lclasses11.zip  
$ORACLE_HOME/lib/xmlparserv2.jar  
$ORACLE_HOME/lib/xschem.jar  
$ORACLE_HOME/rdbms/jlib/aqapi11.jar  
$ORACLE_HOME/rdbms/jlib/jmscommon.jar  
$ORACLE_HOME/rdbms/jlib/aqxml.jar  
$ORACLE_HOME/rdbms/jlib/xsul11.jar
```

`$(ORACLE_HOME)/jis/lib/servlet.jar`

Since the servlet uses JDBC OCI drivers to connect to the Oracle9i server, it is required that Oracle9i Client libraries be installed on the machine that hosts the servlet. The `LD_LIBRARY_PATH` must contain `$(ORACLE_HOME)/lib`.

For more information on Internet access to AQ, refer to *Oracle9i Application Developer's Guide - Advanced Queuing*.

Package oracle.AQ.xml Summary

A summary of package oracle.AQ.xml is provided here.

Class Summary	Description
Interfaces	-
AQxmlCallback	This interface is used to define callbacks to be invoked before/after the servlet performs AQ operations.
Classes	-
AQxmlDataSource	The AQ data source is used to specify the backend database to which the servlet connects to perform AQ operations.
AQxmlServlet	AQxmlServlet - this is the AQ xml servlet which handles HTTP POST requests from clients. To be used with Servlet 2.2 implementations
AQxmlServlet20	AQxmlServlet - this is the AQ xml servlet which handles HTTP POST requests from clients. To be used with Servlet 2.0 implementations
AQxmlCallbackContext	This is the context passed to the user before/after callback functions. This CallbackContext has methods to retrieve the parsed XML document, get a JDBC connection to the AQ database, override the response stream sent by the servlet and set the xml style sheet for the response.
AQxmlDebug	AQ xml Debug class
Exceptions	-
AQxmlException	AQ XML Exception

AQxmlCallback

Syntax

```
public interface AQxmlCallback
```

Description

This interface is used to define callbacks to be invoked before/after the servlet performs AQ operations. The callback must be defined in the init method of the servlet by using the setUserCallback method. The callback methods get the servlet request stream, the servlet response and the callback context. The CallbackContext has methods to retrieve the parsed XML document, get a JDBC connection to the AQ database, and override the response stream sent by the servlet.

Member Summary	Description
Methods	-
afterAQOperation(HttpServletRequest, HttpServletResponse, AQxmlCallbackContext)	Callback invoked after any AQ operations are performed by the servlet
beforeAQOperation(HttpServletRequest, HttpServletResponse, AQxmlCallbackContext)	Callback invoked before any AQ operations are performed by the servlet

Methods

afterAQOperation(HttpServletRequest, HttpServletResponse, AQxmlCallbackContext)

```
public void afterAQOperation(oracle.AQ.xml.HttpServletRequest request,
oracle.AQ.xml.HttpServletResponse response, AQxmlCallbackContext ctx)
```

Callback invoked after any AQ operations are performed by the servlet

Parameters

request - servlet request

response - servlet response

ctx - Callback context

beforeAQOperation(HttpServletRequest, HttpServletResponse, AQxmlCallbackContext)

```
public void beforeAQOperation(oracle.AQ.xml.HttpServletRequest request,  
oracle.AQ.xml.HttpServletResponse response, AQxmlCallbackContext ctx)
```

Callback invoked before any AQ operations are performed by the servlet

Parameters

request - servlet request

response - servlet response

ctx - Callback context

AQxmlDataSource

Syntax

```
public class AQxmlDataSource extends java.lang.Object

java.lang.Object
|
+--oracle.AQ.xml.AQxmlDataSource
```

Description

The AQ data source is used to specify the backend database to which the servlet connects to perform AQ operations. It contains the database SID, host name, listener port and the username/password of the AQ servlet super-user. The AQ servlet uses the JDBC-OCI driver to connect to the database. It creates a connection cache - the default size of the connection pool is 5.

Member Summary	Description
Constructors	-
AQxmlDataSource(OracleOCIConnectionPool pool_ds)	Creates an AQ data source
AQxmlDataSource(String, String, String, String, String)	Creates an AQ data source
Methods	-
getCacheSize()	Get the size of the connection cache
getDBDrv()	Get the JDBC driver used by the data source
getHost()	Get the host name
getPort()	Get the listener port
getSid()	Get the database SID
setCacheSize(int)	Set the size of the connection cache

Inherited Member Summary

Methods inherited from class java.lang.Object

(Cont.) Inherited Member Summary

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructors

AQxmlDataSource(OracleOCIConnectionPool pool_ds)

```
public AQxmlDataSource(OracleOCIConnectionPool pool_ds)
```

Creates an AQ data source given an OCI connection pool

Parameters

pool_ds - OCI connection pool

Throws

[AQxmlException](#) - if fails to create a data source

AQxmlDataSource(String, String, String, String, String)

```
public AQxmlDataSource(java.lang.String user, java.lang.String password,  
java.lang.String sid, java.lang.String host, java.lang.String port)
```

Creates an AQ data source

Parameters

user - username

password - user password

sid - database SID

port - listener port

Throws

[AQxmlException](#) - if fails to create a data source

Methods

getCacheSize()

```
public int getCacheSize()  
Get the size of the connection cache
```

getDBDrv()

```
public java.lang.String getDBDrv()  
Get the JDBC driver used by the data source
```

getHost()

```
public java.lang.String getHost()  
Get the host name
```

getPort()

```
public java.lang.String getPort()  
Get the listener port
```

getSid()

```
public java.lang.String getSid()  
Get the database SID
```

setCacheSize(int)

```
public void setCacheSize(int csize)  
Set the size of the connection cache
```

Parameters

csize - cache size

AQxmlCallbackContext

Syntax

```
public class AQxmlCallbackContext extends java.lang.Object
```

```
java.lang.Object
|
+--oracle.AQ.xml.AQxmlCallbackContext
```

Description

This is the context passed to the user before/after callback functions. This CallbackContext has methods to retrieve the parsed XML document, get a JDBC connection to the AQ database, override the response stream sent by the servlet and set the xml style sheet for the response.

Member Summary	Description
Methods	-
getDBConnection()	Get the JDBC connection that is used to perform this request. Users can perform SQL operations using this database connection.
getOverrideAQResponseFlag()	Get flag to override the response that will be sent back by the AQ servlet.
getServerResponseDoc()	Get the AQxmlDocument representing the response that will be sent back from the servlet.
getStyleSheetProcessingInstr()	Get the stylesheet processing instruction for the XML response.
parseRequestStream()	Parse the XML document in the servlet request.
setOverrideAQResponseFlag(boolean)	Set flag to override the response sent back by the AQ servlet.
setStyleSheet(String, String)	Set StyleSheet for the XML response.
setStyleSheetProcessingInstr(String)	Set StyleSheet processing instruction for the XML response.

Inherited Member Summary

Methods inherited from class `java.lang.Object`

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

Methods

getConnection()

```
public java.sql.Connection getConnection()
```

Get the JDBC connection that is used to perform this request. Users can perform SQL operations using this database connection. The operations performed will be part of the same transaction as the AQ operations. They will be committed or aborted when the AQ operation in the IDAP message is committed or aborted. Users cannot call commit/rollback on these connections. Commit/Rollback has to be done by sending an IDAP message to the servlet.

getOverrideAQResponseFlag()

```
public boolean getOverrideAQResponseFlag()
```

Get flag to override the response that will be sent back by the AQ servlet.

getServerResponseDoc()

```
public AQxmlDocument getServerResponseDoc()
```

Get the `AQxmlDocument` representing the response that will be sent back from the servlet. This is available only in the `afterAQOperation` callback.

getStyleSheetProcessingInstr()

```
public java.lang.String getStyleSheetProcessingInstr()
```

Get the stylesheet processing instruction for the XML response.

parseRequestStream()

```
public oracle.AQ.xml.Document parseRequestStream()
```

Parse the XML document in the servlet request

setOverrideAQResponseFlag(boolean)

```
public void setOverrideAQResponseFlag(boolean value)
```

Set flag to override the response sent back by the AQ servlet. The AQ servlet sends back an IDAP response to the requestor. User callbacks can set this flag if they want to write their own response instead of the one sent back by AQ

setStyleSheet(String, String)

```
public void setStyleSheet(java.lang.String type, java.lang.String href)
```

Set StyleSheet for the XML response. This can be used to set a xml-stylesheet processing instruction for the XML responses that will be sent fo this request

Parameters

`type` - stylesheet type (Example: "text/xml")

`href` - stylesheet href (Example: "http://www.aq.com/AQ/xslt.html")

Throws

[AQxmlException](#) - if invalid parameters specified

setStyleSheetProcessingInstr(String)

```
public void setStyleSheetProcessingInstr(java.lang.String proc_instr)
```

Set StyleSheet processing instruction for the XML response. This can be used to set a xml-stylesheet processing instruction for the XML responses that will be sent fo this request

Parameters

`proc_instr` - stylesheet processing instruction

(Example: "type=\"text/xsl\" href=\"http://www.oa.com/AQ/xslt23.html\"")

AQxmlServlet

Syntax

```
public class AQxmlServlet implements java.lang.Runnable
```

```
oracle.AQ.xml.AQxmlServlet
```

All Implemented Interfaces:

```
java.lang.Runnable
```

Description

AQxmlServlet - this is the AQ xml servlet which handles HTTP POST requests from clients. This servlet can be deployed in any servlet engine that implements Javasoft's Servlet2.2 standard. Users are required to extend this servlet and define a AQ data source (to connect to the database instance) before deploying it

Member Summary	Description
Methods	-
doGet(HttpServletRequest, HttpServletResponse)	this method handles HTTP GET requests.
doPost(HttpServletRequest, HttpServletResponse)	this method handles HTTP POST requests. This is the main entry point for the AQ xml servlet.
getAQDataSource()	get the AQ data source that will be used by this servlet to the database
getEmailServerAddr()	Get the IP address of the email server
getEmailServerHost()	Get the email server host name
getUserCallback()	get the callback registered by the user
setAQDataSource(AQxmlDataSource)	Subclasses must call this method in the init method of the servlet to specify the database connect parameters (username/password, sid, portno etc)
setAQSchemaLocation(String)	setAQxmlSchemaLocation - set the location of the AQ IDAP schema.
setEmailServerAddr(String)	Set the IP address of the Email server.

(Cont.) Member Summary	Description
setLdapContext(DirContext)	Set the LDAP context for the servlet.
setSessionMaxInactiveTime(int)	Set the maximum time a session can remain inactive.
setStyleSheet(String, String)	Set StyleSheet for responses.
setStyleSheetProcessingInstr(String)	Set StyleSheet processing instruction for responses.
setUserCallback(AQxmlCallback)	setUserCallback - set the user callback.

Methods

doGet(HttpServletRequest, HttpServletResponse)

```
protected void doGet(oracle.AQ.xml.HttpServletRequest request,
oracle.AQ.xml.HttpServletResponse response)
```

This method handles HTTP GET requests. This is just used to test whether the servlet has been deployed successfully. In general all AQ operations must be sent as HTTP POST requests.

doPost(HttpServletRequest, HttpServletResponse)

```
protected void doPost(oracle.AQ.xml.HttpServletRequest request,
oracle.AQ.xml.HttpServletResponse response)
```

This method handles HTTP POST requests. This is the main entry point for the AQ xml servlet. This routine expects the incoming stream to be of type text/xml which contains an XML message conforming to the IDAP schema

Parameters

`request` - the http post request

`response` - the response object. The output is written to this stream

Throws

`ServletException`, - `IOException`

getAQDataSource()

```
public synchronized AQxmlDataSource getAQDataSource()
get the AQ data source that will be used by this servlet to the database
```

getEmailServerAddr()

```
public java.lang.String getEmailServerAddr()
Get the IP address of the email server
```

getEmailServerHost()

```
public java.lang.String getEmailServerHost()
Get the email server host name
```

getUserCallback()

```
public final AQxmlCallback getUserCallback()
get the callback registered by the user
```

setAQDataSource(AQxmlDataSource)

```
public final synchronized void setAQDataSource(AQxmlDataSource data_source)
Subclasses must call this method in the init method of the servlet to specify the
database connect parameters (username/password, sid, portno etc)
```

Parameters

`data_source` - the AQ data source

setAQSchemaLocation(String)

```
public synchronized void setAQSchemaLocation(java.lang.String schema_location)
setAQxmlSchemaLocation - set the location of the AQ IDAP schema. By default we
pick up the schema from the envelope.xsd, aqxml.xsd file in the aqxml.jar file
```

setEmailServerAddr(String)

```
public synchronized void setEmailServerAddr(java.lang.String ip_address)
Set the IP address of the Email server.
```


Parameters

`ip_address` - IP address of email server

setLdapContext(DirContext)

```
public final synchronized void setLdapContext(oracle.AQ.xml.DirContext ctx)
```

Set the LDAP context for the servlet. This context must be set in the `init` method of the servlet, if the LDAP messages may contain queue/topic aliases that are to be looked up in an LDAP server.

Parameters

`ctx` - LDAP directory context

setSessionMaxInactiveTime(int)

```
protected synchronized void setSessionMaxInactiveTime(int secs)
```

Set the maximum time a session can remain inactive. If the session remains inactive for more than this time, the session is destroyed and all operations that have not been committed are rolled back. By default this is set to 120 seconds

Parameters

`secs` - time in seconds. This value cannot be set to less than 30secs

setStyleSheet(String, String)

```
public synchronized void setStyleSheet(java.lang.String type, java.lang.String href)
```

Set `StyleSheet` for responses. This can be called in the `init` method of the servlet to set a `xml-stylesheet` processing instruction for all XML responses sent by the servlet

Parameters

`type` - stylesheet type (e.g: "text/xml")

`href` - stylesheet href (e.g: "http://www.aq.com/AQ/xslt.html")

Throws

[AQxmlException](#) - if invalid parameters specified

setStyleSheetProcessingInstr(String)

```
public void setStyleSheetProcessingInstr(java.lang.String proc_instr)
```

Set StyleSheet processing instruction for responses. This can be called in the `init` method of the servlet to set a xml-stylesheet processing instruction for all XML responses sent by the servlet

Parameters

`proc_instr` - stylesheet processing instruction (e.g: "type=\"text/xsl\" href=\"http://www.oa.com/AQ/xslt23.html\"")

setUserCallback(AQxmlCallback)

```
public final void setUserCallback(AQxmlCallback callback)
```

`setUserCallback` - set the user callback. The callback methods are invoked before and after AQ operations

Parameters

`callback` - user callback

AQxmlServlet20

Syntax

```
public class AQxmlServlet20 implements java.lang.Runnable
```

```
oracle.AQ.xml.AQxmlServlet20
```

All Implemented Interfaces:

```
java.lang.Runnable
```

Description

AQxmlServlet - this is the AQ xml servlet which handles HTTP POST requests from clients. This servlet can be deployed in any servlet engine that implements Javasoft's Servlet2.0 standard. Users are required to extend this servlet and define a AQ data source (to connect to the database instance) before deploying it.

Member Summary	Description
Methods	-
doGet(HttpServletRequest, HttpServletResponse)	this method handles HTTP GET requests.
doPost(HttpServletRequest, HttpServletResponse)	this method handles HTTP POST requests. This is the main entry point for the AQ xml servlet.
getAQDataSource()	get the AQ data source that will be used by this servlet to the database
getEmailServerAddr()	Get the IP address of the email server
getEmailServerHost()	Get the email server host name
getUserCallback()	get the callback registered by the user
setAQDataSource(AQxmlDataSource)	subclasses must call this method in the init call to specify the database connect parameters (username/password, sid, portno etc)
setAQSchemaLocation(String)	setAQxmlSchemaLocation - set the location of the AQ IDAP schema.
setEmailServerAddr(String)	Set the IP address of the Email server.
setLdapContext(DirContext)	Set the LDAP context for the servlet.

(Cont.) Member Summary	Description
setManualInvalidation(boolean)	Set flag to turn on/off manual session invalidation For Servlet2.0 implementations we start a thread to automatically invalidate sessions that have stayed inactive beyond the max inactive time.
setSessionMaxInactiveTime(int)	Set the maximum time a session can remain inactive.
setStyleSheet(String, String)	Set StyleSheet for responses.
setStyleSheetProcessingInstr(String)	Set StyleSheet processing instruction for responses.
setUserCallback(AQxmlCallback)	setUserCallback - set the user callback.

Methods

doGet(HttpServletRequestRequest, HttpServletResponse)

```
protected void doGet(oracle.AQ.xml.HttpServletRequest request,
oracle.AQ.xml.HttpServletResponse response)
```

This method handles HTTP GET requests. This is just used to test whether the servlet has been deployed successfully. In general all AQ operations must be sent as HTTP POST requests.

doPost(HttpServletRequestRequest, HttpServletResponse)

```
protected void doPost(oracle.AQ.xml.HttpServletRequest request,
oracle.AQ.xml.HttpServletResponse response)
```

This method handles HTTP POST requests. This is the main entry point for the AQ xml servlet. This routine expects the incoming stream to be of type text/xml which contains an XML message conforming to the IDAP schema

Parameters

`request` - the http post request

`response` - the response object. The output is written to this stream

Throws

`ServletException`, `IOException`

getAQDataSource()

```
public synchronized AQxmlDataSource getAQDataSource()  
get the AQ data source that will be used by this servlet to the database
```

getEmailServerAddr()

```
public java.lang.String getEmailServerAddr()  
Get the IP address of the email server
```

getEmailServerHost()

```
public java.lang.String getEmailServerHost()  
Get the email server host name
```

getUserCallback()

```
public final AQxmlCallback getUserCallback()  
get the callback registered by the user
```

setAQDataSource(AQxmlDataSource)

```
public final synchronized void setAQDataSource(AQxmlDataSource data_source)  
Subclasses must call this method in the init method of the servlet to specify the  
database connect parameters (username/password, sid, portno etc)
```

Parameters

`data_source` - the AQ data source

setAQSchemaLocation(String)

```
public synchronized void setAQSchemaLocation(java.lang.String schema_location)  
setAQxmlSchemaLocation - set the location of the AQ IDAP schema. By default we  
pick up the schema from the envelope.xsd, aqxml.xsd file in the aqxml.jar  
file.
```

setEmailServerAddr(String)

```
public synchronized void setEmailServerAddr(java.lang.String ip_address)  
Set the IP address of the Email server.
```

Parameters

`ip_address` - IP address of email server

setLdapContext(DirContext)

```
public final synchronized void setLdapContext(oracle.AQ.xml.DirContext ctx)
```

Set the LDAP context for the servlet. This context must be set in the `init` method of the servlet, if the LDAP messages may contain queue/topic aliases that are to be looked up in an LDAP server.

Parameters

`ctx` - LDAP directory context

setManualInvalidation(boolean)

```
protected synchronized void setManualInvalidation(boolean flag)
```

Set flag to turn on/off manual session invalidation For Servlet2.0 implementations we start a thread to automatically invalidate sessions that have stayed inactive beyond the max inactive time. If your servlet runner does its own invalidation of sessions, you may set this flag to false.

Parameters

`flag` - true => indicates manual session invalidation is turned on false => indicates manual session invalidation is turned off

setSessionMaxInactiveTime(int)

```
protected synchronized void setSessionMaxInactiveTime(int secs)
```

Set the maximum time a session can remain inactive. If the session remains inactive for more than this time, the session is destroyed and all operations that have not been committed are rolled back. By default this is set to 120 seconds

Parameters

`secs` - time in seconds. This value cannot be set to less than 30secs

setStyleSheet(String, String)

```
public synchronized void setStyleSheet(java.lang.String type, java.lang.String href)
```

Set `StyleSheet` for responses. This can be called in the `init` method of the servlet to set a `xml-stylesheet` processing instruction for all XML responses sent by the servlet

Parameters

`type` - stylesheet type (e.g: "text/xml")

`href` - stylesheet href (e.g: "http://www.aq.com/AQ/xslt.html")

Throws

[AQxmlException](#) - if invalid parameters specified

setStyleSheetProcessingInstr(String)

```
public void setStyleSheetProcessingInstr(java.lang.String proc_instr)
```

Set `StyleSheet` processing instruction for responses. This can be called in the `init` method of the servlet to set a `xml-stylesheet` processing instruction for all XML responses sent by the servlet

Parameters

`proc_instr` - stylesheet processing instruction (e.g: "type=\"text/xsl\" "

`href=\"http://www.oa.com/AQ/xslt23.html\"")`

setUserCallback(AQxmlCallback)

```
public final void setUserCallback(AQxmlCallback callback)
```

`setUserCallback` - set the user callback. The callback methods are invoked before and after AQ operations

Parameters

`callback` - user callback

AQxmlDebug

Syntax

```
public class AQxmlDebug extends java.lang.Object
```

```
java.lang.Object  
|  
+--oracle.AQ.xml.AQxmlDebug
```

Description

This class has static methods to set trace levels for the AQ Servlet. Do not use unless instructed by Oracle Support

Member Summary	Description
Methods	-
getLogStream()	Get log stream to which trace information is written
getPrintWriter()	Get print stream
getTraceLevel()	Get trace level
setDebug(boolean)	Set debug flag
setLogStream(OutputStream)	Set log stream to which trace information is written
setTraceLevel(int)	Set trace level - AQ_ORA_TR1..5

Inherited Member Summary

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Methods

getLogStream()

```
public static java.io.OutputStream getLogStream()  
Get log stream to which trace information is written
```


getPrintWriter()

```
public static java.io.PrintWriter getPrintWriter()
Get print stream
```

getTraceLevel()

```
public static int getTraceLevel()
Get trace level
```

setDebug(boolean)

```
public static void setDebug(boolean val)
Set debug flag
```

setLogStream(OutputStream)

```
public static void setLogStream(java.io.OutputStream output_stream)
Set log stream to which trace information is written
```

Parameters

output - log stream

setTraceLevel(int)

```
public static void setTraceLevel(int level)
Set trace level
```

0 - no tracing (default)

1 - fatal errors

2 - other errors, imp messages

3 - exception trace, other trace info

4 - method entry/exit

5 - print stack traces, variables

AQxmlException

Syntax

```
public class AQxmlException extends java.lang.Exception
```

```
java.lang.Object
|
+--java.lang.Throwable
    |
    +--java.lang.Exception
        |
        +--oracle.AQ.xml.AQxmlException
```

All Implemented Interfaces:

```
java.io.Serializable
```

Description

AQ XML Exception

Member Summary	Description
Methods	-
getErrorCode()	Get the Oracle Error code for the exception
getNextException()	Get the exception linked to this one.
setNextException(Exception)	Set the linked exception

Inherited Member Summary

Methods inherited from class java.lang.Throwable

fillInStackTrace, getLocalizedMessage, getMessage, printStackTrace, printStackTrace, printStackTrace, toString

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

Methods

getErrorCode()

```
public int getErrorCode()  
Get the Oracle Error code for the exception
```

getNextException()

```
public java.lang.Exception getNextException()  
Get the exception linked to this one.
```

setNextException(Exception)

```
protected void setNextException(java.lang.Exception exc)  
Set the linked exception
```

Parameters

`exc` - linked exception

Package oracle.jms

This chapter describes the Oracle Java Messaging Service (OJMS) interfaces and classes that are contained in package `oracle.jms`. The Oracle JMS interfaces extend the standard JMS interfaces to support the Oracle9i Advanced Queuing (AQ) administrative operations and other AQ features that are not included in the public standard contained in the `javax.jms` package.

This chapter contains these sections:

- [Package oracle.jms Description](#)
- [Package oracle.jms Summary](#)
- [Accessing Standard JMS and Oracle JMS Packages](#)
- [Using the OCI9 or Thin JDBC Driver](#)
- [Using Oracle Server Driver in JServer](#)
- [Privileges Required](#)

Package oracle.jms Description

The Oracle package `oracle.jms` provides a set of interfaces and associated semantics based on the Java Messaging Service (JMS) standard. These interfaces define how a JMS client accesses the facilities of an enterprise messaging product like Oracle9i Advanced Queuing. Advanced Queuing (AQ) is the unique database-integrated message queuing feature of the Oracle9i database. Oracle supports the standard JMS interfaces and has extensions to support the AQ administrative operations and other AQ features that are not included in the public standard.

See Also: *Oracle9i Application Developer's Guide - Advanced Queuing*

Accessing Standard JMS and Oracle JMS Packages

Oracle JMS uses JDBC to connect to the database. Therefore, its applications can run as follows:

- Outside the database using the "OCI9" or "thin" JDBC driver.
- Inside Oracle8i or Oracle 9i JServer using the Oracle Server driver.

The standard JMS interfaces are contained in the `javax.jms` package (refer to Sun J2EE documentation for details). The Oracle JMS interfaces extend `javax.jms` and are contained in the `oracle.jms` package.

Using the OCI9 or Thin JDBC Driver

To use JMS interfaces with clients running outside the database, you must include the appropriate JDBC driver, JNDI jar files, and the following AQ jar files in your CLASSPATH. (The CLASSPATH is the operating system environmental variable that the JVM uses to find the classes it needs to run applications.)

For JDK 1.3 include:

```
$ORACLE_HOME/rdbms/jlib/jmscommon.jar
$ORACLE_HOME/rdbms/jlib/aqapi13.jar
$ORACLE_HOME/jdbc/lib/jndi.zip
$ORACLE_HOME/jdbc/lib/classes13.zip
```

For JDK 1.2 include:

```
$ORACLE_HOME/rdbms/jlib/jmscommon.jar
$ORACLE_HOME/rdbms/jlib/aqapi12.jar
$ORACLE_HOME/jdbc/lib/jndi.zip
$ORACLE_HOME/jdbc/lib/classes12.zip
```

For JDK 1.1 include:

```
$ORACLE_HOME/rdbms/jlib/jmscommon.jar  
$ORACLE_HOME/rdbms/jlib/aqapi11.jar  
$ORACLE_HOME/jdbc/lib/jndi.zip  
$ORACLE_HOME/jdbc/lib/classes11.zip
```

Using Oracle Server Driver in JServer

If your application is running inside the JServer, you should be able to access the Oracle JMS classes that have been automatically loaded when the JServer was installed. If these classes are not available, you may have to load `jmscommon.jar` followed by `aqapi.jar` using the `loadjava` utility.

Privileges Required

Users must have EXECUTE privilege on DBMS_AQIN and DBMS_AQJMS packages in order to use the Oracle JMS interfaces. Users can also acquire these rights through the AQ_USER_ROLE or the AQ_ADMINISTRATOR_ROLE.

Users will also need the appropriate system and Queue or Topic privileges to send or receive messages.

Package oracle.jms Summary

Table 4–1 Summary of interfaces for oracle.jms

Interface	Description
AdtMessage	This interface extends the Message interface and represents messages containing Oracle object type payloads - this is an AQ extension to JMS.
AQjmsQueueReceiver	This interface extends javax.jms.QueueReceiver and defines AQ extensions to JMS. A client uses a QueueReceiver for receiving messages that have been delivered to a Queue.
AQjmsQueueSender	This interface extends QueueSender and defines AQ extensions to JMS. A client uses a QueueSender to send messages to a Queue.
AQjmsTopicPublisher	This interface extends TopicPublisher and defines AQ extensions to JMS. A client uses a TopicPublisher for publishing messages to a Topic.
AQjmsTopicReceiver	This interface extends the TopicReceiver interface that defines AQ extensions for remote subscribers and explicitly specified recipients (in point-to-multipoint communication). A TopicReceiver is used to receive messages from a Topic.
AQjmsTopicSubscriber	This interface extends TopicSubscriber and defines AQ extensions to JMS. A client uses a TopicSubscriber to receive messages published on a Topic.
TopicBrowser	This interface extends MessageConsumer to allow remote subscribers to look at messages on a topic without removing them.
TopicReceiver	This interface extends MessageConsumer to allow remote subscribers and explicitly specified recipients (in point-to-multipoint communication) to receive messages.

Table 4–2 Summary of Classes for oracle.jms

Class	Description
AQjmsAdtMessage	This class implements the AdtMessage interface. An AdtMessage is used to send a message containing Oracle object type payloads.
AQjmsAgent	This class implements the Destination interface. It is used to define remote subscribers and ReplyTo Destinations

Table 4–2 (Cont.) Summary of Classes for oracle.jms

Class	Description
AQjmsBytesMessage	This class implements the BytesMessage interface. A BytesMessage is used to send a message containing a stream of uninterpreted bytes
AQjmsConnection	This class implements the Connection interface. This is an active connection to the JMS provider
AQjmsConnectionMetaData	class AQjmsConnectionMetaData represents the Meta Data information available for a JMS Connection.
AQjmsConstants	This class defines the constants used in the oracle.jms package
AQjmsConsumer	This class implements the MessageConsumer interface
AQjmsDestination	This class implements administered objects, Queue and Topic
AQjmsDestinationProperty	This class defines Destination properties
AQjmsFactory	This class is used for accessing administered ConnectionFactory objects in Oracle's implementation of JMS.
AQjmsMapMessage	This class implements the MapMessage interface. A MapMessage is used to send a set of name-value pairs where names are Strings and values are java primitive types
AQjmsMessage	This class implements the Message interface. This is the superclass of all JMS messages
AQjmsObjectMessage	This class implements the ObjectMessage interface. An ObjectMessage is used to send a message that contains a serializable java object
AQjmsOracleDebug	AQ Oracle Debug class - not to be used unless instructed by Oracle Support
AQjmsProducer	This class implements the MessageProducer interface. A MessageProducer is used to send messages to a Destination
AQjmsQueueBrowser	This class implements the QueueBrowser interface. A QueueBrowser is used to look at messages in a Queue without removing them.
AQjmsQueueConnectionFactory	This class implements the QueueConnectionFactory interface. A QueueConnectionFactory is used to create QueueConnections

Table 4–2 (Cont.) Summary of Classes for oracle.jms

Class	Description
AQjmsSession	This class implements the <code>javax.jms.Session</code> interface. A Session is a single threaded context for producing a consuming messages
AQjmsStreamMessage	This class implements the <code>StreamMessage</code> interface. A <code>StreamMessage</code> is used to send a stream of java primitives
AQjmsTextMessage	This class implements the <code>TextMessage</code> interface. A <code>TextMessage</code> is used to send a message containing a <code>java.lang.StringBuffer</code>
AQjmsTopicBrowser	This class implements the <code>TopicBrowser</code> interface. A <code>TopicBrowser</code> is used to look at messages in a <code>Topic</code> without removing them.
AQjmsTopicConnectionFactory	This class implements the <code>TopicConnectionFactory</code> interface. A <code>TopicConnectionFactory</code> is used to create <code>TopicConnections</code>

Table 4–3 Exceptions for oracle.jms

Exception	Description
AQjmsException	This exception extends <code>JMSEException</code> - adds Oracle error codes. This is the root of all JMS exceptions
AQjmsIllegalStateException	This exception extends <code>IllegalStateException</code> . It is thrown when when an OJMS method is invoked at an illegal or inappropriate time, or when OJMS is not in an appropriate state for the requested operation.
AQjmsInvalidDestinationException	This exception extends <code>InvalidDestinationException</code> . It is thrown when a <code>Destination</code> is not valid
AQjmsInvalidSelectorException	This exception extends <code>InvalidSelectorException</code> . It is thrown when the specified <code>MessageSelector</code> is not valid

Table 4–3 (Cont.) Exceptions for oracle.jms

Exception	Description
AQjmsMessageEOFException	This exception extends <code>MessageEOFException</code> . It is thrown when an unexpected end of stream has been reached when a <code>StreamMessage</code> or <code>BytesMessage</code> is being read
AQjmsMessageFormatException	This exception extends <code>MessageFormatException</code> . It is thrown when a client attempts to use a datatype not supported by a message or attempts to read data in the message as the wrong type
AQjmsMessageNotReadableException	This exception extends <code>MessageNotReadableException</code> . It is thrown when a client attempts to read a write-only message
AQjmsMessageNotWriteableException	This exception extends <code>MessageNotWriteableException</code> . It is thrown when a client attempts to write a read-only message

AdtMessage

Syntax

```
public interface AdtMessage extends javax.jms.Message
```

All Superinterfaces

`javax.jms.Message`

All Known Implementing Classes

`AQjmsAdtMessage`

Description

This interface extends the Message interface and represents messages containing Oracle object type payloads.

Member Summary	Description
Methods	-
<code>getAdtPayload()</code>	Get the CustomDatum object containing this Adt message's data.
<code>setAdtPayload(CustomDatum)</code>	Set the CustomDatum object containing this Adt message's data

Inherited Member Summary

Fields inherited from interface `javax.jms.Message`

`DEFAULT_DELIVERY_MODE`, `DEFAULT_PRIORITY`, `DEFAULT_TIME_TO_LIVE`

Methods inherited from interface `javax.jms.Message`

(Cont.) Inherited Member Summary

clearBody, clearProperties, getBooleanProperty, getByteProperty, getDoubleProperty, getFloatProperty, getIntProperty, getJMSCorrelationID, getJMSCorrelationIDAsBytes, getJMSDeliveryMode, getJMSDestination, getJMSExpiration, getJMSMessageID, getJMSPriority, getJMSRedelivered, getJMSReplyTo, getJMSTimestamp, getJMSType, getLongProperty, getObjectProperty, getPropertyNames, getShortProperty, getStringProperty, propertyExists, setBooleanProperty, setByteProperty, setDoubleProperty, setFloatProperty, setIntProperty, setJMSCorrelationID, setJMSCorrelationIDAsBytes, setJMSDeliveryMode, setJMSDestination, setJMSExpiration, setJMSMessageID, setJMSPriority, setJMSRedelivered, setJMSReplyTo, setJMSTimestamp, setJMSType, setLongProperty, setObjectProperty, setShortProperty, setStringProperty

Methods

getAdtPayload()

```
public oracle.sql.CustomDatum getAdtPayload()
```

Get the CustomDatum object containing this Adt message's data.

Returns

the object containing this message's data

Throws

JMSException - if JMS fails to get object due to some internal JMS error.

setAdtPayload(CustomDatum)

```
public void setAdtPayload(oracle.sql.CustomDatum payload)
```

set the CustomDatum object containing this ADT message's data.

ADT is for Asynchronous Data Transfer.

Parameters

payload - the message's data (the object must implement the CustomDatum interface). This payload must be a Java object that represents the ADT that is defined as the queue/topic payload type.

Throws

`JMSEException` - if JMS fails to set the ADT payload

`MessageNotWriteableException` - if message in read-only mode.

AQjmsAdtMessage

Syntax

```
public class AQjmsAdtMessage extends AQjmsMessage implements AdtMessage
```

```
java.lang.Object
|
+--AQjmsMessage
|
+--oracle.jms.AQjmsAdtMessage
```

All Implemented Interfaces

AdtMessage, javax.jms.Message

Description

This class implements the AdtMessage interface. An AdtMessage is used to send a message containing Oracle object type payloads

Member Summary	Description
Methods	-
clearBody()	Clear out the message body.
getAdtPayload()	Get the CustomDatum object containing this Adt message's data.
getBooleanProperty(String)	Return the boolean property value with the given name.
getByteProperty(String)	Return the byte property value with the given name.
getDoubleProperty(String)	Return the double property value with the given name.
getFloatProperty(String)	Return the float property value with the given name.
getIntProperty(String)	Return the integer property value with the given name.
getJMSReplyTo()	Get where a reply to this message should be sent.
getJMSType()	Get the message type.
getLongProperty(String)	Return the long property value with the given name.

(Cont.) Member Summary	Description
<code>getObjectProperty(String)</code>	Return the Java object property value with the given name.
<code>getPropertyNames()</code>	Return an Enumeration of all the property names.
<code>getShortProperty(String)</code>	Return the short property value with the given name.
<code>getStringProperty(String)</code>	Return the String property value with the given name.
<code>propertyExists(String)</code>	Check if a property value exists.
<code>setAdtPayload(CustomDatum)</code>	set the CustomDatum object containing this Adt message's data
<code>setBooleanProperty(String, boolean)</code>	Set a boolean property value with the given name, into the Message.
<code>setByteProperty(String, byte)</code>	Set a byte property value with the given name, into the Message.
<code>setDoubleProperty(String, double)</code>	Set a double property value with the given name, into the Message.
<code>setFloatProperty(String, float)</code>	Set a float property value with the given name, into the Message.
<code>setIntProperty(String, int)</code>	Set an integer property value with the given name, into the Message.
<code>setJMSReplyTo(Destination)</code>	Set where a reply to this message should be sent.
<code>setJMSType(String)</code>	Set the message type.
<code>setLongProperty(String, long)</code>	Set a long property value with the given name, into the Message.
<code>setObjectProperty(String, Object)</code>	Set a Java object property value with the given name, into the Message.
<code>setShortProperty(String, short)</code>	Set a short property value with the given name, into the Message.
<code>setStringProperty(String, String)</code>	Set a String property value with the given name, into the Message.

Inherited Member Summary

Fields inherited from interface `javax.jms.Message`

(Cont.) Inherited Member Summary

DEFAULT_DELIVERY_MODE, DEFAULT_PRIORITY, DEFAULT_TIME_TO_LIVE

Methods inherited from class AQjmsMessage

```
clearProperties(), getJMSCorrelationID(),
getJMSCorrelationIDAsBytes(), getJMSDeliveryMode(),
getJMSDestination(), getJMSExpiration(), getJMSMessageID(),
getJMSMessageIDAsBytes(), getJMSPriority(), getJMSRedelivered(),
getJMSTimestamp(), getSenderID(), setJMSCorrelationID(String),
setJMSCorrelationIDAsBytes(byte[]), setJMSDestination(Destination),
setJMSExpiration(long), setJMSMessageID(String),
setJMSPriority(int), setJMSRedelivered(boolean),
setJMSTimestamp(long), setSenderID(AQjmsAgent)
```

Methods inherited from class java.lang.Object

```
clone, equals, finalize, getClass, hashCode, notify, notifyAll,
toString, wait, wait, wait
```

Methods inherited from interface javax.jms.Message

```
clearProperties, getJMSCorrelationID, getJMSCorrelationIDAsBytes,
getJMSDeliveryMode, getJMSDestination, getJMSExpiration,
getJMSMessageID, getJMSPriority, getJMSRedelivered,
getJMSTimestamp, setJMSCorrelationID, setJMSCorrelationIDAsBytes,
setJMSDeliveryMode, setJMSDestination, setJMSExpiration,
setJMSMessageID, setJMSPriority, setJMSRedelivered, setJMSTimestamp
```

Methods

clearBody()

```
public void clearBody()
```

Clear out the message body. All other parts of the message are left untouched.

Specified By

javax.jms.Message.clearBody() in interface javax.jms.Message

Specified By

javax.jms.Message.clearBody() in interface javax.jms.Message

Overrides

clearBody() in class AQjmsMessage

Throws

`JMSEException` - if JMS fails to due to some internal JMS error.

getAdtPayload()

```
public oracle.sql.CustomDatum getAdtPayload()  
Get the CustomDatum object containing this Adt message's data.
```

Specified By

`getAdtPayload()` in interface `AdtMessage`

Returns

the object containing this message's data

Throws

`JMSEException` - if JMS fails to get object due to some internal JMS error.

getBooleanProperty(String)

```
public boolean getBooleanProperty(java.lang.String name)  
Return the boolean property value with the given name.
```

Specified By

`java.xml.messaging.jms.Message.getBooleanProperty(java.lang.String)` in interface `java.xml.messaging.jms.Message`

Overrides

`getBooleanProperty(String)` in class `AQjmsMessage`

Parameters

`name` - the name of the boolean property

Returns

the boolean property value with the given name.

Throws

`JMSEException` - if JMS fails to get Property due to some internal JMS error.

`MessageFormatException` - if this type conversion is invalid.

getBytesProperty(String)

```
public byte getBytesProperty(java.lang.String name)
```

Return the byte property value with the given name.

Specified By

`javax.jms.Message.getBytesProperty(java.lang.String)` in interface `javax.jms.Message`

Overrides

`getBytesProperty(String)` in class `AQjmsMessage`

Parameters

`name` - the name of the byte property

Returns

the byte property value with the given name.

Throws

`JMSEException` - if JMS fails to get Property due to some internal JMS error.

`MessageFormatException` - if this type conversion is invalid.

getDoubleProperty(String)

```
public double getDoubleProperty(java.lang.String name)
```

Return the double property value with the given name.

Specified By

`javax.jms.Message.getDoubleProperty(java.lang.String)` in interface `javax.jms.Message`

Overrides

`getDoubleProperty(String)` in class `AQjmsMessage`

Parameters

`name` - the name of the double property

Returns

the double property value with the given name.

Throws

`JMSException` - if JMS fails to get Property due to some internal JMS error.
`MessageFormatException` - if this type conversion is invalid.

getFloatProperty(String)

```
public float getFloatProperty(java.lang.String name)
```

Return the float property value with the given name.

Specified By

`javax.jms.Message.getFloatProperty(java.lang.String)` in interface `javax.jms.Message`

Overrides

`getFloatProperty(String)` in class `AQjmsMessage`

Parameters

`name` - the name of the float property

Returns

the float property value with the given name.

Throws

`JMSException` - if JMS fails to get Property due to some internal JMS error.
`MessageFormatException` - if this type conversion is invalid.

getIntProperty(String)

```
public int getIntProperty(java.lang.String name)
```

Return the integer property value with the given name.

Specified By

`javax.jms.Message.getIntProperty(java.lang.String)` in interface `javax.jms.Message`

Overrides

`getIntProperty(String)` in class `AQjmsMessage`

Parameters

`name` - the name of the integer property

Returns

the integer property value with the given name.

Throws

`JMSEException` - if JMS fails to get Property due to some internal JMS error.

`MessageFormatException` - if this type conversion is invalid.

getJMSReplyTo()

```
public javax.jms.Destination getJMSReplyTo()
```

Get where a reply to this message should be sent. This method is not supported for AdtMessages in this release

Specified By

`javax.jms.Message.getJMSReplyTo()` in interface `javax.jms.Message`

Overrides

`getJMSReplyTo()` in class `AQjmsMessage`

Throws

`JMSEException` - `NOT_SUPPORTED` for `AdtMessage`

getJMSType()

```
public java.lang.String getJMSType()
```

Get the message type. This method is not supported for AdtMessages in this release

Specified By

`javax.jms.Message.getJMSType()` in interface `javax.jms.Message`

Overrides

`getJMSType()` in class `AQjmsMessage`

Returns

the message type

Throws

`JMSEException` - `NOT_SUPPORTED` for `AdtMessage`

getLongProperty(String)

```
public long getLongProperty(java.lang.String name)
Return the long property value with the given name.
```

Specified By

`javax.jms.Message.getLongProperty(java.lang.String)` in interface `javax.jms.Message`

Overrides

`getLongProperty(String)` in class `AQjmsMessage`

Parameters

`name` - the name of the long property

Returns

the long property value with the given name.

Throws

`JMSException` - if JMS fails to get Property due to some internal JMS error.

`MessageFormatException` - if this type conversion is invalid.

getObjectProperty(String)

```
public java.lang.Object getObjectProperty(java.lang.String name)
Return the Java object property value with the given name.
```

Note that this method can be used to return in objectified format, an object that had been stored as a property in the `Message` with the equivalent `setObject` method call, or it's equivalent primitive set method.

Specified By

`javax.jms.Message.getObjectProperty(java.lang.String)` in interface `javax.jms.Message`

Overrides

`getObjectProperty(String)` in class `AQjmsMessage`

Parameters

`name` - the name of the Java object property

Returns

the Java object property value with the given name, in objectified format (i.e. if it set as an int, then a Integer is returned). If there is no property by this name, a null value is returned.

Throws

`JMSEException` - if JMS fails to get Property due to some internal JMS error.

getPropertyNames()

```
public synchronized java.util.Enumeration getPropertyNames()
```

Return an Enumeration of all the property names.

Specified By

`javax.jms.Message.getPropertyNames()` in interface `javax.jms.Message`

Overrides

`getPropertyNames()` in class `AQjmsMessage`

Returns

an enumeration of all the names of property values.

Throws

`JMSEException` - if JMS fails to get Property names due to some internal JMS error.

getShortProperty(String)

```
public short getShortProperty(java.lang.String name)
```

Return the short property value with the given name.

Specified By

`javax.jms.Message.getShortProperty(java.lang.String)` in interface `javax.jms.Message`

Overrides

`getShortProperty(String)` in class `AQjmsMessage`

Parameters

`name` - the name of the short property

Returns

the short property value with the given name.

Throws

`JMSEException` - if JMS fails to get Property due to some internal JMS error.

`MessageFormatException` - if this type conversion is invalid.

getStringProperty(String)

```
public java.lang.String getStringProperty(java.lang.String name)
```

Return the String property value with the given name.

Specified By

`javax.jms.Message.getStringProperty(java.lang.String)` in interface `javax.jms.Message`

Overrides

`getStringProperty(String)` in class `AQjmsMessage`

Parameters

`name` - the name of the String property

Returns

the String property value with the given name. If there is no property by this name, a null value is returned.

Throws

`JMSEException` - if JMS fails to get Property due to some internal JMS error.

`MessageFormatException` - if this type conversion is invalid.

propertyExists(String)

```
public boolean propertyExists(java.lang.String name)
```

Check if a property value exists.

Specified By

`javax.jms.Message.propertyExists(java.lang.String)` in interface `javax.jms.Message`

Overrides

propertyExists(String) in class AQjmsMessage

Parameters

name - the name of the property to test

Returns

true if the property does exist.

Throws

JMSEException - if JMS fails to check if property exists due to some internal JMS error.

setAdtPayload(CustomDatum)

```
public void setAdtPayload(oracle.sql.CustomDatum payload)
set the CustomDatum object containing this Adt message's data
```

Specified By

setAdtPayload(CustomDatum) in interface AdtMessage

Parameters

payload - the message's data (the object must implement the CustomDatum interface). This payload must be a java object that represents the ADT that is defined as the queue/topic payload type

Throws

JMSEException - if JMS fails to set the adt payload

MessageNotWriteableException - if message in read-only mode.

setBooleanProperty(String, boolean)

```
public void setBooleanProperty(java.lang.String name, boolean value)
Set a boolean property value with the given name, into the Message.
```

Specified By

javax.jms.Message.setBooleanProperty(java.lang.String, boolean) in interface javax.jms.Message

Overrides

`setBooleanProperty(String, boolean)` in class `AQjmsMessage`

Parameters

`name` - the name of the boolean property

`value` - the boolean property value to set in the `Message`.

Throws

`JMSEException` - if JMS fails to set Property due to some internal JMS error.

`MessageNotWriteableException` - if properties are read-only

setByteProperty(String, byte)

```
public void setByteProperty(java.lang.String name, byte value)
```

Set a byte property value with the given name, into the `Message`.

Specified By

`javax.jms.Message.setByteProperty(java.lang.String, byte)` in interface

`javax.jms.Message`

Overrides

`setByteProperty(String, byte)` in class `AQjmsMessage`

Parameters

`name` - the name of the byte property

`value` - the byte property value to set in the `Message`.

Throws

`JMSEException` - if JMS fails to set Property due to some internal JMS error.

`MessageNotWriteableException` - if properties are read-only

setDoubleProperty(String, double)

```
public void setDoubleProperty(java.lang.String name, double value)
```

Set a double property value with the given name, into the `Message`.

Specified By

`javax.jms.Message.setDoubleProperty(java.lang.String, double)` in interface `javax.jms.Message`

Overrides

`setDoubleProperty(String, double)` in class `AQjmsMessage`

Parameters

`name` - the name of the double property

`value` - the double property value to set in the `Message`.

Throws

`JMSEException` - if JMS fails to set Property due to some internal JMS error.

`MessageNotWriteableException` - if properties are read-only

setFloatProperty(String, float)

```
public void setFloatProperty(java.lang.String name, float value)
```

Set a float property value with the given name, into the `Message`.

Specified By

`javax.jms.Message.setFloatProperty(java.lang.String, float)` in interface `javax.jms.Message`

Overrides

`setFloatProperty(String, float)` in class `AQjmsMessage`

Parameters

`name` - the name of the float property

`value` - the float property value to set in the `Message`.

Throws

`JMSEException` - if JMS fails to set Property due to some internal JMS error.

`MessageNotWriteableException` - if properties are read-only

setIntProperty(String, int)

```
public void setIntProperty(java.lang.String name, int value)
```

Set an integer property value with the given name, into the Message.

Specified By

javax.jms.Message.setIntProperty(java.lang.String, int) in interface javax.jms.Message

Overrides

setIntProperty(String, int) in class AQjmsMessage

Parameters

name - the name of the integer property

value - the integer property value to set in the Message.

Throws

JMSException - if JMS fails to set Property due to some internal JMS error.

MessageNotWriteableException - if properties are read-only

setJMSReplyTo(Destination)

```
public void setJMSReplyTo(javax.jms.Destination replyTo)
```

Set where a reply to this message should be sent. This method is not supported for AdtMessage in this release

Specified By

javax.jms.Message.setJMSReplyTo(javax.jms.Destination) in interface javax.jms.Message

Overrides

setJMSReplyTo(Destination) in class AQjmsMessage

Throws

JMSException - NOT_SUPPORTED for AdtMessage

setJMSType(String)

```
public void setJMSType(java.lang.String type)
```

Set the message type. This method is not supported for AdtMessages in this release

Specified By

`javax.jms.Message.setJMSType(java.lang.String)` in interface `javax.jms.Message`

Overrides

`setJMSType(String)` in class `AQjmsMessage`

Parameters

`type` - of the message

Throws

`JMSEException` - `NOT_SUPPORTED` for `AdtMessage`

setLongProperty(String, long)

```
public void setLongProperty(java.lang.String name, long value)
```

Set a long property value with the given name, into the Message.

Specified By

`javax.jms.Message.setLongProperty(java.lang.String, long)` in interface `javax.jms.Message`

Overrides

`setLongProperty(String, long)` in class `AQjmsMessage`

Parameters

`name` - the name of the long property

`value` - the long property value to set in the Message.

Throws

`JMSEException` - if JMS fails to set Property due to some internal JMS error.

`MessageNotWriteableException` - if properties are read-only

setObjectProperty(String, Object)

```
public void setObjectProperty(java.lang.String name, java.lang.Object value)
```

Set a Java object property value with the given name, into the Message.

Note that this method only works for the objectified primitive object types (Integer, Double, Long ...) and String's.

Specified By

`javax.jms.Message.setObjectProperty(java.lang.String, java.lang.Object)` in interface `javax.jms.Message`

Overrides

`setObjectProperty(String, Object)` in class `AQjmsMessage`

Parameters

`name` - the name of the Java object property.

`value` - the Java object property value to set in the Message.

Throws

`JMSEException` - if JMS fails to set Property due to some internal JMS error.

`MessageFormatException` - if object is invalid

`MessageNotWriteableException` - if properties are read-only

setShortProperty(String, short)

```
public void setShortProperty(java.lang.String name, short value)
```

Set a short property value with the given name, into the Message.

Specified By

`javax.jms.Message.setShortProperty(java.lang.String, short)` in interface `javax.jms.Message`

Overrides

`setShortProperty(String, short)` in class `AQjmsMessage`

Parameters

`name` - the name of the short property

`value` - the short property value to set in the Message.

Throws

`JMSEException` - if JMS fails to set Property due to some internal JMS error.

`MessageNotWriteableException` - if properties are read-only

setStringProperty(String, String)

```
public void setStringProperty(java.lang.String name, java.lang.String value)
```

Set a String property value with the given name, into the Message.

Specified By

`javax.jms.Message.setStringProperty(java.lang.String, java.lang.String)` in interface `javax.jms.Message`

Overrides

`setStringProperty(String, String)` in class `AQjmsMessage`

Parameters

`name` - the name of the String property

`value` - the String property value to set in the Message.

Throws

`JMSEException` - if JMS fails to set Property due to some internal JMS error.

`MessageNotWriteableException` - if properties are read-only

AQjmsAgent

Syntax

```
public class AQjmsAgent implements javax.jms.Destination
```

```
oracle.jms.AQjmsAgent
```

All Implemented Interfaces

```
javax.jms.Destination
```

Description

This class implements the Destination interface. It is used to define remote subscribers and ReplyTo Destinations

Member Summary	Description
Constructors	-
<code>AQjmsAgent(String, String)</code>	Constructor
<code>AQjmsAgent(String, String, int)</code>	Constructor
Methods	-
<code>getAddress()</code>	Get the address of the agent
<code>getName()</code>	Get the name of the agent
<code>getProtocol()</code>	Get the protocol of the agent
<code>setAddress(String)</code>	Set the address of the agent
<code>setName(String)</code>	Set the name of the agent
<code>setProtocol(int)</code>	Set the protocol of the agent
<code>toString()</code>	Convert the agent to its string representation which is of the form: "[AQjmsAgent] \n name: NAME \n address: ADDRESS \n protocol: PROTOCOL"

Constructors

AQjmsAgent(String, String)

```
public AQjmsAgent(java.lang.String name, java.lang.String address)
Constructor
```

Parameters

`name` - Name of the agent

`address` - Address of the agent

Throws

`SQLException` - if it fails to create an agent

AQjmsAgent(String, String, int)

```
public AQjmsAgent(java.lang.String name, java.lang.String address, int
protocol)
Constructor
```

Parameters

`name` - Name of the agent

`address` - Address of the agent

`protocol` - Protocol of the agent

Throws

`SQLException` - if it fails to create an agent

Methods

getAddress()

```
public java.lang.String getAddress()
Get the address of the agent
```

Returns

the address of the agent

Throws

`SQLException` - if there was an error in getting the address

getName()

```
public java.lang.String getName()  
Get the name of the agent
```

Returns

the name of the agent

Throws

`SQLException` - if there was an error in getting the name

getProtocol()

```
public int getProtocol()  
Get the protocol of the agent
```

Returns

the protocol of the agent

Throws

`SQLException` - if there was an error in getting the protocol

setAddress(String)

```
public void setAddress(java.lang.String address)  
Set the address of the agent
```

Parameters

`address` - the address of the agent

Throws

`SQLException` - if there was an error in setting the address

setName(String)

```
public void setName(java.lang.String name)
```

Set the name of the agent

Parameters

name - the name of the agent

Throws

`SQLException` - if there was an error in setting the name

setProtocol(int)

```
public void setProtocol(int protocol)
```

Set the protocol of the agent

Parameters

protocol - the protocol of the agent

Throws

`SQLException` - if there was an error in setting the address

toString()

```
public java.lang.String toString()
```

Convert the agent to its string representation which is of the form: "[AQjmsAgent] \n name: NAME \n address: ADDRESS \n protocol: PROTOCOL"

Returns

the string representation of the agent

Throws

`SQLException` - if there was an error in setting the address

AQjmsBytesMessage

Syntax

```
public class AQjmsBytesMessage extends AQjmsMessage
    implements javax.jms.BytesMessage
```

```
java.lang.Object
|
+--AQjmsMessage
|
+--oracle.jms.AQjmsBytesMessage
```

All Implemented Interfaces

javax.jms.BytesMessage, javax.jms.Message

Description

This class implements the BytesMessage interface. A BytesMessage is used to send a message containing a stream of uninterpreted bytes

Member Summary	Description
Methods	-
<code>clearBody()</code>	Clear out the message body.
<code>clearProperties()</code>	Clear a message's properties.
<code>readBoolean()</code>	Read a <code>boolean</code> from the stream message.
<code>readByte()</code>	Read a signed 8-bit value from the stream message.
<code>readBytes(byte[])</code>	Read a byte array from the stream message.
<code>readBytes(byte[], int)</code>	Read a portion of the bytes message.
<code>readChar()</code>	Read a Unicode character value from the stream message.
<code>readDouble()</code>	Read a <code>double</code> from the stream message.
<code>readFloat()</code>	Read a <code>float</code> from the stream message.
<code>readInt()</code>	Read a signed 32-bit integer from the stream message.
<code>readLong()</code>	Read a signed 64-bit integer from the stream message.

Member Summary	Description
<code>readShort()</code>	Put the message in read-only mode, and reposition the stream of bytes to the beginning.
<code>readUnsignedByte()</code>	Read an unsigned 8-bit number from the stream message.
<code>readUnsignedShort()</code>	Read an unsigned 16-bit number from the stream message.
<code>readUTF()</code>	Read in a string that has been encoded using a modified UTF-8 format from the stream message.
<code>reset()</code>	Put the message in read-only mode, and reposition the stream of bytes to the beginning.
<code>writeBoolean(boolean)</code>	Write a <code>boolean</code> to the stream message as a 1-byte value.
<code>writeByte(byte)</code>	Write out a <code>byte</code> to the stream message as a 1-byte value.
<code>writeBytes(byte[])</code>	Write a byte array to the stream message.
<code>writeBytes(byte, int, int)</code>	Write a portion of a byte array to the stream message.
<code>writeChar(char)</code>	Write a <code>char</code> to the stream message as a 2-byte value, high byte first.
<code>writeDouble(double)</code>	Convert the <code>double</code> argument to a <code>long</code> using the <code>doubleToLongBits</code> method in class <code>Double</code> , and then writes that <code>long</code> value to the stream message as an 8-byte quantity, high byte first.
<code>writeFloat(float)</code>	Convert the <code>float</code> argument to an <code>int</code> using the <code>floatToIntBits</code> method in class <code>Float</code> , and then writes that <code>int</code> value to the stream message as a 4-byte quantity, high byte first.
<code>writeInt(int)</code>	Write an <code>int</code> to the stream message as four bytes, high byte first.
<code>writeLong(long)</code>	Write a <code>long</code> to the stream message as eight bytes, high byte first.
<code>writeObject(Object)</code>	Write a Java object to the stream message.
<code>writeShort(short)</code>	Write a <code>short</code> to the stream message as two bytes, high byte first.
<code>writeUTF(String)</code>	Write a string to the stream message using UTF-8 encoding in a machine-independent manner.

Inherited Member Summary

Fields inherited from interface javax.jms.Message

DEFAULT_DELIVERY_MODE, DEFAULT_PRIORITY, DEFAULT_TIME_TO_LIVE

Methods inherited from class AQjmsMessage

getBooleanProperty(String), getByteProperty(String),
getDoubleProperty(String), getFloatProperty(String),
getIntProperty(String), getJMSCorrelationID(),
getJMSCorrelationIDAsBytes(), getJMSDeliveryMode(),
getJMSDestination(), getJMSExpiration(), getJMSMessageID(),
getJMSMessageIDAsBytes(), getJMSPriority(), getJMSRedelivered(),
getJMSReplyTo(), getJMSTimestamp(), getJMSType(),
getLongProperty(String), getObjectProperty(String),
getPropertyNames(), getSenderID(), getShortProperty(String),
getStringProperty(String), propertyExists(String),
setBooleanProperty(String, boolean), setByteProperty(String, byte),
setDoubleProperty(String, double), setFloatProperty(String, float),
setIntProperty(String, int), setJMSCorrelationID(String),
setJMSCorrelationIDAsBytes(byte[]), setJMSDestination(Destination),
setJMSExpiration(long), setJMSMessageID(String),
setJMSPriority(int), setJMSRedelivered(boolean),
setJMSReplyTo(Destination), setJMSTimestamp(long),
setJMSType(String), setLongProperty(String, long),
setObjectProperty(String, Object), setSenderID(AQjmsAgent),
setShortProperty(String, short), setStringProperty(String, String)

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll,
toString, wait, wait, wait

Methods inherited from interface javax.jms.Message

getBooleanProperty, getByteProperty, getDoubleProperty,
getFloatProperty, getIntProperty, getJMSCorrelationID,
getJMSCorrelationIDAsBytes, getJMSDeliveryMode, getJMSDestination,
getJMSExpiration, getJMSMessageID, getJMSPriority,
getJMSRedelivered, getJMSReplyTo, getJMSTimestamp, getJMSType,
getLongProperty, getObjectProperty, getPropertyNames,
getShortProperty, getStringProperty, propertyExists,
setBooleanProperty, setByteProperty, setDoubleProperty,
setFloatProperty, setIntProperty, setJMSCorrelationID,
setJMSCorrelationIDAsBytes, setJMSDeliveryMode, setJMSDestination,
setJMSExpiration, setJMSMessageID, setJMSPriority,
setJMSRedelivered, setJMSReplyTo, setJMSTimestamp, setJMSType,
setLongProperty, setObjectProperty, setShortProperty,
setStringProperty

Methods

clearBody()

```
public void clearBody()
```

Clear out the message body. All other parts of the message are left untouched.

Specified By

`javax.jms.Message.clearBody()` in interface `javax.jms.Message`

Overrides

`clearBody()` in class `AQjmsMessage`

Throws

`JMSEException` - if JMS fails to due to some internal JMS error.

clearProperties()

```
public void clearProperties()
```

Clear a message's properties.

Specified By

`javax.jms.Message.clearProperties()` in interface `javax.jms.Message`

Overrides

`clearProperties()` in class `AQjmsMessage`

Throws

`JMSEException` - if JMS fails to clear JMS message properties due to some internal JMS error.

readBoolean()

```
public boolean readBoolean()
```

Read a `boolean` from the stream message.

Specified By

`javax.jms.BytesMessage.readBoolean()` in interface `javax.jms.BytesMessage`

Returns

the `boolean` value read.

Throws

`MessageNotReadableException` - if message in write-only mode.

`JMSEException` - if JMS fails to read message due to some internal JMS error.

`MessageEOFException` - if end of message stream

readByte()

```
public byte readByte()
```

Read a signed 8-bit value from the stream message.

Specified By

`javax.jms.BytesMessage.readByte()` in interface `javax.jms.BytesMessage`

Returns

the next byte from the stream message as a signed 8-bit byte.

Throws

`MessageNotReadableException` - if message in write-only mode.

`MessageEOFException` - if end of message stream

`JMSEException` - if JMS fails to read message due to some internal JMS error.

readBytes(byte[])

```
public int readBytes(byte[] value)
```

Read a byte array from the stream message.

Specified By

`javax.jms.BytesMessage.readBytes(byte[])` in interface `javax.jms.BytesMessage`

Parameters

`value` - the buffer into which the data is read.

Returns

the total number of bytes read into the buffer, or -1 if there is no more data because the end of the stream has been reached.

Throws

`MessageNotReadableException` - if message in write-only mode.

`MessageEOFException` - if end of message stream

`JMSEException` - if JMS fails to read message due to some internal JMS error.

readBytes(byte[], int)

```
public int readBytes(byte[] value, int length)
```

Read a portion of the bytes message.

Specified By

`javax.jms.BytesMessage.readBytes(byte[], int)` in interface `javax.jms.BytesMessage`

Parameters

`value` - the buffer into which the data is read.

`length` - the number of bytes to read.

Returns

the total number of bytes read into the buffer, or -1 if there is no more data because the end of the stream has been reached.

Throws

`MessageNotReadableException` - if message in write-only mode.

`MessageEOFException` - if end of message stream

`JMSEException` - if JMS fails to read message due to some internal JMS error.

readChar()

```
public char readChar()
```

Read a Unicode character value from the stream message.

Specified By

`javax.jms.BytesMessage.readChar()` in interface `javax.jms.BytesMessage`

Returns

the next two bytes from the stream message as a Unicode character.

Throws

`MessageNotReadableException` - if message in write-only mode.

`MessageEOFException` - if end of message stream

`JMSEException` - if JMS fails to read message due to some internal JMS error.

readDouble()

```
public double readDouble()
```

Read a double from the stream message.

Specified By

`javax.jms.BytesMessage.readDouble()` in interface `javax.jms.BytesMessage`

Returns

the next eight bytes from the stream message, interpreted as a `double`.

Throws

`MessageNotReadableException` - if message in write-only mode.

`MessageEOFException` - if end of message stream

`JMSEException` - if JMS fails to read message due to some internal JMS error.

readFloat()

```
public float readFloat()
```

Read a float from the stream message.

Specified By

`javax.jms.BytesMessage.readFloat()` in interface `javax.jms.BytesMessage`

Returns

the next four bytes from the stream message, interpreted as a `float`.

Throws

`MessageNotReadableException` - if message in write-only mode.

`MessageEOFException` - if end of message stream

`JMSEException` - if JMS fails to read message due to some internal JMS error.

readInt()

```
public int readInt()
```

Read a signed 32-bit integer from the stream message.

Specified By

`javax.jms.BytesMessage.readInt()` in interface `javax.jms.BytesMessage`

Returns

the next four bytes from the stream message, interpreted as an `int`.

Throws

`MessageNotReadableException` - if message in write-only mode.

`MessageEOFException` - if end of message stream

`JMSEException` - if JMS fails to read message due to some internal JMS error.

readLong()

```
public long readLong()
```

Read a signed 64-bit integer from the stream message.

Specified By

`javax.jms.BytesMessage.readLong()` in interface `javax.jms.BytesMessage`

Returns

the next eight bytes from the stream message, interpreted as a `long`.

Throws

`MessageNotReadableException` - if message in write-only mode.

`MessageEOFException` - if end of message stream

`JMSEException` - if JMS fails to read message due to some internal JMS error.

readShort()

```
public short readShort()
```

Put the message in read-only mode, and reposition the stream of bytes to the beginning. Throws `MessageNotWritableException` - if message in write-only mode. `JMSEException` - if JMS fails to read message due to some internal JMS error.

Specified By

`javax.jms.BytesMessage.readShort()` in interface `javax.jms.BytesMessage`

readUnsignedByte()

```
public int readUnsignedByte()
```

Read an unsigned 8-bit number from the stream message.

Specified By

`javax.jms.BytesMessage.readUnsignedByte()` in interface `javax.jms.BytesMessage`

Returns

the next byte from the stream message, interpreted as an unsigned 8-bit number.

Throws

`MessageNotReadableException` - if message in write-only mode.

`MessageEOFException` - if end of message stream

`JMSEXception` - if JMS fails to read message due to some internal JMS error.

readUnsignedShort()

```
public int readUnsignedShort()
```

Read an unsigned 16-bit number from the stream message.

Specified By

`javax.jms.BytesMessage.readUnsignedShort()` in interface `javax.jms.BytesMessage`

Returns

the next two bytes from the stream message, interpreted as an unsigned 16-bit integer.

Throws

`MessageNotReadableException` - if message in write-only mode.

`MessageEOFException` - if end of message stream

`JMSEXception` - if JMS fails to read message due to some internal JMS error.

readUTF()

```
public java.lang.String readUTF()
```

Read in a string that has been encoded using a modified UTF-8 format from the stream message.

Specified By

`javax.jms.BytesMessage.readUTF()` in interface `javax.jms.BytesMessage`

Returns

a Unicode string from the stream message.

Throws

`MessageNotReadableException` - if message in write-only mode.

`MessageEOFException` - if end of message stream

`JMSEException` - if JMS fails to read message due to some internal JMS error.

reset()

```
public void reset()
```

Put the message in read-only mode, and reposition the stream of bytes to the beginning.

Specified By

`javax.jms.BytesMessage.reset()` in interface `javax.jms.BytesMessage`

Throws

`JMSEException` - if JMS fails to reset the message due to some internal JMS error.

`MessageFormatException` - if message has an invalid format

writeBoolean(boolean)

```
public void writeBoolean(boolean value)
```

Write a `boolean` to the stream message as a 1-byte value. The value `true` is written out as the value `(byte)1`; the value `false` is written out as the value `(byte)0`.

Specified By

`javax.jms.BytesMessage.writeBoolean(boolean)` in interface `javax.jms.BytesMessage`

Parameters

`value` - the boolean value to be written.

Throws

`MessageNotWriteableException` - if message in read-only mode.

`JMSEException` - if JMS fails to write message due to some internal JMS error.

writeByte(byte)

```
public void writeByte(byte value)
```

Write out a byte to the stream message as a 1-byte value.

Specified By

`javax.jms.BytesMessage.writeByte(byte)` in interface `javax.jms.BytesMessage`

Parameters

`value` - the byte value to be written.

Throws

`MessageNotWriteableException` - if message in read-only mode.

`JMSEException` - if JMS fails to write message due to some internal JMS error.

writeBytes(byte[])

```
public void writeBytes(byte[] value)
```

Write a byte array to the stream message.

Specified By

`javax.jms.BytesMessage.writeBytes(byte[])` in interface `javax.jms.BytesMessage`

Parameters

`value` - the byte array to be written.

Throws

`MessageNotWriteableException` - if message in read-only mode.

`JMSEException` - if JMS fails to write message due to some internal JMS error.

writeBytes(byte, int, int)

```
public void writeBytes(byte[] value, int offset, int length)
```

Write a portion of a byte array to the stream message.

Specified By

`javax.jms.BytesMessage.writeBytes(byte[], int, int)` in interface `javax.jms.BytesMessage`

Parameters

`value` - the byte array value to be written.
`offset` - the initial offset within the byte array.
`length` - the number of bytes to use.

Throws

`MessageNotWriteableException` - if message in read-only mode.
`JMSEException` - if JMS fails to write message due to some internal JMS error.

writeChar(char)

```
public void writeChar(char value)
```

Write a char to the stream message as a 2-byte value, high byte first.

Specified By

`javax.jms.BytesMessage.writeChar(char)` in interface `javax.jms.BytesMessage`

Parameters

`value` - the char value to be written.

Throws

`MessageNotWriteableException` - if message in read-only mode.
`JMSEException` - if JMS fails to write message due to some internal JMS error.

writeDouble(double)

```
public void writeDouble(double value)
```

Convert the double argument to a long using the `doubleToLongBits` method in class `Double`, and then writes that long value to the stream message as an 8-byte quantity, high byte first.

Specified By

`javax.jms.BytesMessage.writeDouble(double)` in interface `javax.jms.BytesMessage`

Parameters

`value` - the double value to be written.

Throws

`MessageNotWriteableException` - if message in read-only mode.

`JMSEException` - if JMS fails to write message due to some internal JMS error.

writeFloat(float)

```
public void writeFloat(float value)
```

Convert the float argument to an int using the `floatToIntBits` method in class `Float`, and then writes that int value to the stream message as a 4-byte quantity, high byte first.

Specified By

`javax.jms.BytesMessage.writeFloat(float)` in interface `javax.jms.BytesMessage`

Parameters

`value` - the float value to be written.

Throws

`MessageNotWriteableException` - if message in read-only mode.

`JMSEException` - if JMS fails to write message due to some internal JMS error.

writeInt(int)

```
public void writeInt(int value)
```

Write an int to the stream message as four bytes, high byte first.

Specified By

`javax.jms.BytesMessage.writeInt(int)` in interface `javax.jms.BytesMessage`

Parameters

value - the int to be written.

Throws

`MessageNotWriteableException` - if message in read-only mode.

`JMSEException` - if JMS fails to write message due to some internal JMS error.

writeLong(long)

```
public void writeLong(long value)
```

Write a long to the stream message as eight bytes, high byte first.

Specified By

`javax.jms.BytesMessage.writeLong(long)` in interface `javax.jms.BytesMessage`

Parameters

value - the long to be written.

Throws

`MessageNotWriteableException` - if message in read-only mode.

`JMSEException` - if JMS fails to write message due to some internal JMS error.

writeObject(Object)

```
public void writeObject(java.lang.Object value)
```

Write a Java object to the stream message.

Note that this method only works for the objectified primitive object types (Integer, Double, Long ...), String's and byte arrays.

Specified By

`javax.jms.BytesMessage.writeObject(java.lang.Object)` in interface `javax.jms.BytesMessage`

Parameters

value - the Java object to be written.

Throws

`MessageNotWriteableException` - if message in read-only mode.

`MessageFormatException` - if object is invalid type.

`JMSEException` - if JMS fails to write message due to some internal JMS error.

writeShort(short)

```
public void writeShort(short value)
```

Write a `short` to the stream message as two bytes, high byte first.

Specified By

`javax.jms.BytesMessage.writeShort(short)` in interface `javax.jms.BytesMessage`

Parameters

`value` - the `short` to be written.

Throws

`MessageNotWriteableException` - if message in read-only mode.

`JMSEException` - if JMS fails to write message due to some internal JMS error.

writeUTF(String)

```
public void writeUTF(java.lang.String value)
```

Write a string to the stream message using UTF-8 encoding in a machine-independent manner.

Specified By

`javax.jms.BytesMessage.writeUTF(java.lang.String)` in interface `javax.jms.BytesMessage`

Parameters

`value` - the `String` value to be written.

Throws

`MessageNotWriteableException` - if message in read-only mode.

`JMSEException` - if JMS fails to write message due to some internal JMS error.

AQjmsConnection

Syntax

```
public class AQjmsConnection extends java.lang.Object
    implements javax.jms.QueueConnection, javax.jms.TopicConnection
```

```
java.lang.Object
|
+--oracle.jms.AQjmsConnection
```

All Implemented Interfaces

javax.jms.Connection, javax.jms.QueueConnection, javax.jms.TopicConnection

Description

This class implements the Connection interface. This is an active connection to the JMS provider

Member Summary	Description
Methods	-
<code>close()</code>	Since a provider typically allocates significant resources outside the JVM on behalf of a Connection, clients should close them when they are not needed.
<code>createQueueSession(boolean, int)</code>	create a queue session
<code>createTopicSession(boolean, int)</code>	Create a TopicSession
<code>getClientID()</code>	Get the client identifier for this connection.
<code>getCurrentJmsSession()</code>	gets the current session
<code>getMetaData()</code>	Get the meta data for this connection.
<code>setClientID(String)</code>	Set the client identifier for this connection.
<code>start()</code>	Start (or restart) a Connection's delivery of incoming messages.
<code>stop()</code>	Used to temporarily stop a Connection's delivery of incoming messages.

Member Summary	Description
setExceptionListener(ExceptionListener)	Set the exception listener for this connection.
getExceptionListener()	Get the exception listener for this connection.
setPingPeriod(long)	Set the sleep period between each 'ping' of the exception listener.
getPingPeriod()	Get the sleep period between each 'ping' of the exception listener.

Inherited Member Summary
Methods inherited from class <code>java.lang.Object</code>
<code>clone, equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait</code>

Methods

close()

```
public void close()
```

Since a provider typically allocates significant resources outside the JVM on behalf of a `Connection`, clients should close them when they are not needed. Relying on garbage collection to eventually reclaim these resources may not be timely enough.

Specified By

`javax.jms.Connection.close()` in interface `javax.jms.Connection`

Specified By

`javax.jms.Connection.close()` in interface `javax.jms.Connection`

Throws

`JMSEException` - if JMS implementation fails to close the connection due to internal error. For example, a failure to release resources or to close socket connection can lead to throwing of this exception.

createQueueSession(boolean, int)

```
public javax.jms.QueueSession createQueueSession(boolean transacted, int ack_
mode)
```

create a queue session

Specified By

`javax.jms.QueueConnection.createQueueSession(boolean, int)` in interface `javax.jms.QueueConnection`

Parameters

`transacted` - is session transacted?

`ack_mode` - acknowledgement mode

Returns

`QueueSession`. A `QueueSession` provides methods for creating `QueueReceiver`'s, `QueueSender`'s, `QueueBrowser`'s.

Throws

`JMSException` - if JMS fails to create queue session

createTopicSession(boolean, int)

```
public javax.jms.TopicSession createTopicSession(boolean transacted, int ack_
mode)
```

Create a `TopicSession`

Specified By

`javax.jms.TopicConnection.createTopicSession(boolean, int)` in interface `javax.jms.TopicConnection`

Parameters

`transacted` - if true, the session is transacted.

`acknowledgeMode` - indicates whether the consumer or the client will acknowledge any messages it receives. This parameter will be ignored if the session is transacted.

Returns

a newly created topic session.

Throws

`JMSEException` - if JMS Connection fails to create a session due to some internal error or lack of support for specific transaction and acknowledgement mode.

getClientID()

```
public java.lang.String getClientID()  
Get the client identifier for this connection.
```

Specified By

`javax.jms.Connection.getClientID()` in interface `javax.jms.Connection`

Returns

the unique client identifier.

Throws

`JMSEException` - if JMS implementation fails to return the client ID for this Connection due to some internal error.

getCurrentJmsSession()

```
public javax.jms.Session getCurrentJmsSession()  
gets the current session
```

Returns

Session The current JMS session

getMetaData()

```
public javax.jms.ConnectionMetaData getMetaData()  
Get the meta data for this connection.
```

Specified By

`javax.jms.Connection.getMetaData()` in interface `javax.jms.Connection`

Returns

the connection meta data.

Throws

`JMSEException` - general exception if JMS implementation fails to get the Connection meta-data for this Connection.

See Also

`javax.jms.ConnectionMetaData`

setClientID(String)

```
public void setClientID(java.lang.String clientID)
```

Set the client identifier for this connection.

The preferred way to assign a Client's client identifier is for it to be configured in a client-specific ConnectionFactory and transparently assigned to the Connection it creates. Alternatively, a client can set a Connections's client identifier using a provider-specific value.

The purpose of client identifier is to associate a session and its objects with a state maintained on behalf of the client by a provider. The only such state identified by JMS is that required to support durable subscriptions

Specified By

`javax.jms.Connection.setClientID(java.lang.String)` in interface `javax.jms.Connection`

Parameters

`clientID` - the unique client identifier

Throws

`JMSEException` - general exception if JMS implementation fails to set the client ID for this Connection due to some internal error.

`InvalidClientIDException` - if JMS client specifies an invalid or duplicate client id.

start()

```
public void start()
```

Start (or restart) a Connection's delivery of incoming messages. Restart begins with the oldest unacknowledged message. Starting a started session is ignored.

Specified By

`javax.jms.Connection.start()` in interface `javax.jms.Connection`

Throws

`JMSEException` - if JMS implementation fails to start the message delivery due to some internal error.

See Also

`javax.jms.Connection.stop()`

stop()

```
public void stop()
```

Used to temporarily stop a Connection's delivery of incoming messages. It can be restarted using its `start` method. When stopped, delivery to all the Connection's message consumers is inhibited: synchronous receive's block and messages are not delivered to message listeners.

After `stop` is called there may still be some messages delivered.

Stopping a Session has no affect on its ability to send messages. Stopping a stopped session is ignored.

Specified By

`javax.jms.Connection.stop()` in interface `javax.jms.Connection`

Throws

`JMSEException` - if JMS implementation fails to stop the message delivery due to some internal error.

See Also

`javax.jms.Connection.start()`

setExceptionHandler(ExceptionListener)

```
public void setExceptionHandler(javax.jms.ExceptionListener listener)
```

Set an exception listener for this connection.

If a JMS provider detects a serious problem with a connection it will inform the connection's `ExceptionHandler` if one has been registered. It does this by calling the listener's `onException()` method passing it a `JMSEException` describing the problem.

This allows a client to be asynchronously notified of a problem. Some connections only consume messages so they would have no other way to learn their connection has failed.

A Connection serializes execution of its `ExceptionListener`.

Specified By

`javax.jms.Connection.setExceptionListener(javax.jms.ExceptionListener listener)` in interface `javax.jms.Connection`.

Parameters

`listener` - the exception listener.

Throws

`JMSEException` - general exception if JMS implementation fails to set the `ExceptionListener` for this `Connection`.

getExceptionListener()

```
public javax.jms.ExceptionListener getExceptionListener()
```

Get the `ExceptionListener` for this `Connection`.

Specified By

`javax.jms.Connection.getExceptionListener()` in interface `javax.jms.Connection`

Returns

The `ExceptionListener` for this `Connection` if registered, else null

Throws

`JMSEException` - general exception if JMS implementation fails to get the `ExceptionListener` for this `Connection`.

setPingPeriod(long)

```
public void setPingPeriod(long period)
```

Set the sleep period (in milliseconds) between each 'ping' of the exception listener for this connection.

If a exception listener is registered, the connection 'pings' the server periodically to ensure that the server is alive. These 'pings' can result in performance degradation. A trade-off has to be made in selecting a good 'ping' period value. The greater the

value the larger the time period an asynchronous client may have to wait before it is aware of a fatal exception. The smaller the value, more the overhead of the 'pings'. If an exception listener is not registered for this connection, then 'ping' period is of no relevance. The default value of the ping period is 2 minutes.

Parameters

`period` - the sleep period between each 'ping' in milliseconds.

getPingPeriod()

```
public long getPingPeriod()
```

Get the sleep period (in milliseconds) between each 'ping' of the exception listener for this connection.

This method Returns the value set by a previous call to `setPingPeriod()` or the default value (2 minutes) if `setPingPeriod` is not calle.

Returns

The sleep period between each 'ping' in milliseconds.

AQjmsConnectionMetaData

Syntax

```
public class AQjmsConnectionMetaData extends java.lang.Object
    implements javax.jms.ConnectionMetaData
```

```
java.lang.Object
|
+--oracle.jms.AQjmsConnectionMetaData
```

All Implemented Interfaces

```
javax.jms.ConnectionMetaData
```

Description

This class represents the Meta Data information available for a JMS Connection.

Member Summary	Description
Constructors	-
AQjmsConnectionMetaData()	-
Methods	-
getJMSMajorVersion()	Get the JMS major version number.
getJMSMinorVersion()	Get the JMS minor version number.
getJMSProviderName()	Get the JMS provider name.
getJMSVersion()	Get the JMS version.
getProviderMajorVersion()	Get the JMS provider major version number.
getProviderMinorVersion()	Get the JMS provider minor version number.
getProviderVersion()	Get the JMS provider version.

Inherited Member Summary

Methods inherited from class java.lang.Object

```
clone, equals, finalize, getClass, hashCode, notify, notifyAll,
toString, wait, wait, wait
```

Constructors

AQjmsConnectionMetaData()

```
public AQjmsConnectionMetaData()
```

Methods

getJMSMajorVersion()

```
public int getJMSMajorVersion()  
Get the JMS major version number.
```

Specified By

javax.jms.ConnectionMetaData.getJMSMajorVersion() in interface
javax.jms.ConnectionMetaData

Returns

the JMS major version number.

Throws

JMSEException - if some internal error occurs in JMS implementation during the meta-data retrieval.

getJMSMinorVersion()

```
public int getJMSMinorVersion()  
Get the JMS minor version number.
```

Specified By

javax.jms.ConnectionMetaData.getJMSMinorVersion() in interface
javax.jms.ConnectionMetaData

Returns

the JMS minor version number.

Throws

JMSEException - if some internal error occurs in JMS implementation during the meta-data retrieval.

getJMSProviderName()

```
public java.lang.String getJMSProviderName()
```

Get the JMS provider name.

Specified By

`javax.jms.ConnectionMetaData.getJMSProviderName()` in interface
`javax.jms.ConnectionMetaData`

Returns

the JMS provider name.

Throws

`JMSException` - if some internal error occurs in JMS implementation during the meta-data retrieval.

getJMSVersion()

```
public java.lang.String getJMSVersion()
```

Get the JMS version.

Specified By

`javax.jms.ConnectionMetaData.getJMSVersion()` in interface
`javax.jms.ConnectionMetaData`

Returns

the JMS version.

Throws

`JMSException` - if some internal error occurs in JMS implementation during the meta-data retrieval.

getProviderMajorVersion()

```
public int getProviderMajorVersion()
```

Get the JMS provider major version number.

Specified By

javax.jms.ConnectionMetaData.getProviderMajorVersion() in interface
javax.jms.ConnectionMetaData

Returns

the JMS provider major version number.

Throws

JMSEException - if some internal error occurs in JMS implementation during the meta-data retrieval.

getProviderMinorVersion()

```
public int getProviderMinorVersion()  
Get the JMS provider minor version number.
```

Specified By

javax.jms.ConnectionMetaData.getProviderMinorVersion() in interface
javax.jms.ConnectionMetaData

Returns

the JMS provider minor version number.

Throws

JMSEException - if some internal error occurs in JMS implementation during the meta-data retrieval.

getProviderVersion()

```
public java.lang.String getProviderVersion()  
Get the JMS provider version.
```

Specified By

javax.jms.ConnectionMetaData.getProviderVersion() in interface
javax.jms.ConnectionMetaData

Returns

the JMS provider version.

Throws

`JMSException` - if some internal error occurs in JMS implementation during the meta-data retrieval.

AQjmsConstants

Syntax

```
public class AQjmsConstants
```

```
oracle.jms.AQjmsConstants
```

Description

This class defines the constants used in the oracle.jms package

Members

Fields

[EXCEPTION](#)

[NONE](#)

[NORMAL](#)

[STATE_EXPIRED](#)

[STATE_PROCESSED](#)

[STATE_READY](#)

[STATE_WAITING](#)

[TRANSACTIONAL](#)

[WAIT_FOREVER](#)

[WAIT_NONE](#)

Constructors

[AQjmsConstants\(\)](#)

Methods

[isJ2eeCompliant\(\)](#)

Fields

EXCEPTION

```
public static final int EXCEPTION
```


NONE

```
public static final int NONE
```

NORMAL

```
public static final int NORMAL
```

STATE_EXPIRED

```
public static final int STATE_EXPIRED
```

STATE_PROCESSED

```
public static final int STATE_PROCESSED
```

STATE_READY

```
public static final int STATE_READY
```

STATE_WAITING

```
public static final int STATE_WAITING
```

TRANSACTIONAL

```
public static final int TRANSACTIONAL
```

WAIT_FOREVER

```
public static final int WAIT_FOREVER
```

WAIT_NONE

```
public static final int WAIT_NONE
```

Constructors**AQjmsConstants()**

```
public AQjmsConstants()
```

Methods

isJ2eeCompliant()

Return `true` if the JMS client is run in the J2EE/JMS 1.3 compliance mode and `false` otherwise.

The client can define the j2EE compliance mode used by OJMS by setting the java property "oracle.jms.j2eeCompliant" to either `true` or `false` at run time. When running with the `j2eeCompliant` flag set to `false`, OJMS clients will support older (non j2ee compliant) OJMS behavior for priority, expiration and non durable subscriber semantics. This allows older clients to run without code modifications.

AQjmsConsumer

Syntax

```
public class AQjmsConsumer extends java.lang.Object
    implements AQjmsQueueReceiver, AQjmsTopicSubscriber, AQjmsTopicReceiver
```

```
java.lang.Object
|
+--oracle.jms.AQjmsConsumer
```

All Implemented Interfaces

AQjmsQueueReceiver, AQjmsTopicReceiver, AQjmsTopicSubscriber,
 javax.jms.MessageConsumer, javax.jms.QueueReceiver, TopicReceiver,
 javax.jms.TopicSubscriber

Description

This class implements the MessageConsumer interface

Member Summary	Description
Methods	-
<code>close()</code>	Since a provider may allocate some resources on behalf of a MessageConsumer outside the JVM, clients should close them when they are not needed.
<code>getMessageListener()</code>	Get the message consumer's MessageListener.
<code>getMessageSelector()</code>	Get the message consumer's message selector expression.
<code>getNavigationMode()</code>	Get the navigation mode for the consumer
<code>getNoLocal()</code>	Get the NoLocal attribute for this TopicSubscriber.
<code>getQueue()</code>	Get the queue associated with this queue receiver.
<code>getTopic()</code>	Get the topic associated with this subscriber.
<code>receive()</code>	Receive the next message produced for this message consumer.
<code>receive(long)</code>	Receive the next message that arrives within the specified timeout interval.
<code>receiveNoData()</code>	Consume the message without returning it to the user.

Member Summary	Description
<code>receiveNoData()</code>	Consume the message without returning it to the user.
<code>receiveNoWait()</code>	Receive the next message if one is immediately available.
<code>setMessageListener(MessageListener)</code>	Set the message consumer's MessageListener.
<code>setNavigationMode(int)</code>	Set the navigation mode for the consumer

Inherited Member Summary

Methods inherited from class `java.lang.Object`

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

Methods

close()

```
public void close()
```

Since a provider may allocate some resources on behalf of a `MessageConsumer` outside the JVM, clients should close them when they are not needed. Relying on garbage collection to eventually reclaim these resources may not be timely enough.

Specified By

`javax.jms.MessageConsumer.close()` in interface `javax.jms.MessageConsumer`

Specified By

`javax.jms.MessageConsumer.close()` in interface `javax.jms.MessageConsumer`

Throws

`JMSException` - if JMS fails to close the consumer due to some error.

getMessageListener()

```
public synchronized javax.jms.MessageListener getMessageListener()
```

Get the message consumer's `MessageListener`.

Specified By

`javax.jms.MessageConsumer.getMessageListener()` in interface `javax.jms.MessageConsumer`

Specified By

`javax.jms.MessageConsumer.getMessageListener()` in interface `javax.jms.MessageConsumer`

Returns

the listener for the message consumer, or null if this isn't one set.

Throws

`JMSEException` - if JMS fails to get message listener due to some JMS error

getMessageSelector()

```
public synchronized java.lang.String getMessageSelector()  
Get the message consumer's message selector expression.
```

Specified By

`javax.jms.MessageConsumer.getMessageSelector()` in interface `javax.jms.MessageConsumer`

Specified By

`javax.jms.MessageConsumer.getMessageSelector()` in interface `javax.jms.MessageConsumer`

Returns

this message consumer's message selector

Throws

`JMSEException` - if JMS fails to get message selector due to some JMS error

getNavigationMode()

```
public synchronized int getNavigationMode()  
Get the navigation mode for the consumer
```

Specified By

getNavigationMode() in interface AQjmsTopicSubscriber

Specified By

getNavigationMode() in interface AQjmsTopicReceiver

Returns

the navigation mode of the consumer

Throws

if - the navigation mode could not be got

getNoLocal()

```
public synchronized boolean getNoLocal()
```

Get the NoLocal attribute for this TopicSubscriber. The default value for this attribute is false.

Specified By

javax.jms.TopicSubscriber.getNoLocal() in interface javax.jms.TopicSubscriber

Returns

set to true if locally published messages are being inhibited.

Throws

JMSException - if JMS fails to get noLocal attribute for this topic subscriber due to some internal error.

getQueue()

```
public synchronized javax.jms.Queue getQueue()
```

Get the queue associated with this queue receiver.

Specified By

javax.jms.QueueReceiver.getQueue() in interface javax.jms.QueueReceiver

Returns

the queue associated with the receiver

Throws

`JMSEException` - if JMS fails to get queue for this queue receiver due to some internal error.

getTopic()

```
public synchronized javax.jms.Topic getTopic()
```

Get the topic associated with this subscriber.

Specified By

`javax.jms.TopicSubscriber.getTopic()` in interface `javax.jms.TopicSubscriber`
`getTopic()` in interface `TopicReceiver`

Returns

this subscriber's topic

Throws

`JMSEException` - if JMS fails to get topic for this topic subscriber due to some internal error.

receive()

```
public synchronized javax.jms.Message receive()
```

Receive the next message produced for this message consumer.

This call blocks indefinitely until a message is produced.

Specified By

`javax.jms.MessageConsumer.receive()` in interface `javax.jms.MessageConsumer`

Returns

the next message produced for this message consumer.

Throws

`JMSEException` - if JMS fails to receive the next message due to some error.

receive(long)

```
public synchronized javax.jms.Message receive(long timeout)
```

Receive the next message that arrives within the specified timeout interval. This call blocks until either a message arrives or the timeout expires.

Specified By

`javax.jms.MessageConsumer.receive(long)` in interface `javax.jms.MessageConsumer`

Parameters

`timeout` - the timeout value (in milliseconds)

Returns

the next message produced for this message consumer, or null if one is not available.

Throws

`JMSEException` - if JMS fails to receive the next message due to some error.

receiveNoData()

```
public synchronized void receiveNoData()
```

Consume the message without returning it to the user. This call will avoid the overhead of fetching the message from the database. It can be used as an optimization by jms clients who have already read the message, for example using a queue browser.

Specified By

`receiveNoData()` in interface `AQjmsQueueReceiver`

Throws

`JMSEException` - if the message could not be received due to an error

receiveNoData(long)

```
public synchronized void receiveNoData(long timeout)
```

Consume the message without returning it to the user. This call will avoid the overhead of fetching the message from the database. It can be used as an optimization by jms clients who have already read the message, for example using a queue browser. This call will block until a message arrives or the timeout expires

Specified By

receiveNoData(long) in interface AQjmsQueueReceiver

Parameters

timeout - the timeout value in milliseconds

Throws

JMSException - if the message could not be received due to an error

receiveNoWait()

```
public synchronized javax.jms.Message receiveNoWait()  
Receive the next message if one is immediately available.
```

Specified By

javax.jms.MessageConsumer.receiveNoWait() in interface
javax.jms.MessageConsumer

Returns

the next message produced for this message consumer, or null if one is not available.

Throws

JMSException - if JMS fails to receive the next message due to some error.

setMessageListener(MessageListener)

```
public synchronized void setMessageListener(javax.jms.MessageListener  
myListener)
```

Set the message consumer's MessageListener. The onMessage method of this object is called when there are messages for this consumer.

Specified By

javax.jms.MessageConsumer.setMessageListener(javax.jms.MessageListener) in
interface javax.jms.MessageConsumer

Parameters

myListener - set the consumer's message listener

Throws

`JMSEException` - if JMS fails to get message listener due to some JMS error

setNavigationMode(int)

```
public synchronized void setNavigationMode(int mode)
```

Set the navigation mode for the consumer

Specified By

`setNavigationMode(int)` in interface `AQjmsQueueReceiver`

Parameters

`mode` - the navigation mode of the consumer

Throws

`if` - the navigation mode could not be set

AQjmsDestination

Syntax

```
public class AQjmsDestination extends java.lang.Object
    implements javax.jms.Queue, javax.jms.Topic
```

```
java.lang.Object
|
+--oracle.jms.AQjmsDestination
```

All Implemented Interfaces

```
javax.jms.Destination, javax.jms.Queue, javax.jms.Topic
```

Description

This class implements administered objects, Queue and Topic

Member Summary	Description
Methods	-
<code>alter(Session, AQjmsDestinationProperty)</code>	alter the properties of the queue/topic
<code>alterPropagationSchedule(Session, String, Double, String, Double)</code>	alter propagation schedule between the topic and the destination database
<code>delete()</code>	Deletes a temporary destination and makes it unusable for further operations.
<code>disablePropagationSchedule(Session, String)</code>	disable propagation schedule
<code>drop(Session)</code>	drop the queue/topic
<code>enablePropagationSchedule(Session, String)</code>	enable propagation schedule
<code>getCompleteName()</code>	Get the complete name of the queue/topic, of the form, [schema].name
<code>getCompleteTableName()</code>	Get the complete name of the queue table of the queue/topic of the form, [schema].name

(Cont.) Member Summary	Description
<code>getQueueName()</code>	Get the name of the queue
<code>getQueueOwner()</code>	Get the owner of the queue
<code>getTopicName()</code>	Get the name of the Topic
<code>getTopicOwner()</code>	Get the schema of the topic
<code>grantQueuePrivilege(Session, String, String, boolean)</code>	Grant enqueue or dequeue privilege on the queue to a database user
<code>grantTopicPrivilege(Session, String, String, boolean)</code>	Grant a topic privilege
<code>revokeQueuePrivilege(Session, String, String)</code>	Revoke a queue privilege
<code>revokeTopicPrivilege(Session, String, String)</code>	Revoke a topic privilege
<code>schedulePropagation(Session, String, Date, Double, String, Double)</code>	Schedule propagation from the topic for the given destination database
<code>start(Session, boolean, boolean)</code>	start the queue/topic for enqueue or dequeue or both
<code>stop(Session, boolean, boolean, boolean)</code>	stop the queue/topic for enqueue or dequeue or both
<code>toString()</code>	Get the queue/topic as a string, of the form [schema].name
<code>unschedulePropagation(Session, String)</code>	Unschedule propagation between the topic and the specified destination

Inherited Member Summary

Methods inherited from class `java.lang.Object`

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `wait`, `wait`, `wait`

Methods

alter(Session, AQjmsDestinationProperty)

```
public void alter(javax.jms.Session session, AQjmsDestinationProperty dest_
```

property)
alter the properties of the queue/topic

Parameters

session - the jms session
dest_property - the new properties of the queue/topic

alterPropagationSchedule(Session, String, Double, String, Double)

```
public void alterPropagationSchedule(javax.jms.Session session, java.lang.String destination, java.lang.Double duration, java.lang.String next_time, java.lang.Double latency)
```

alter propagation schedule between the topic and the destination database

Parameters

session - the jms session
destination - the dblink of the destination database
duration - the new duration
next_time - the new next_time for propagation
latency - the new latency

delete()

```
public void delete()
```

Deletes a temporary destination and makes it unusable for further operations.

Specified By

javax.jms.TemporaryQueue.delete() in interface javax.jms.TemporaryQueue
and javax.jms.TemporaryTopic.delete() in interface javax.jms.TemporaryTopic.

Throws

JMSException - if OJMS fails to delete the temporary queue/topic due to some internal error or if there are existing receivers still using the temporary destination.

disablePropagationSchedule(Session, String)

```
public void disablePropagationSchedule(javax.jms.Session session, java.lang.String destination)
```

disable propagation schedule

Parameters

`session` - the jms session

`destination` - the dblink to the destination database

Throws

`JMSEException` - if the propagation schedule could not be disabled

drop(Session)

```
public void drop(javax.jms.Session session)
```

drop the queue/topic

Parameters

`session` - the jms session

Throws

`JMSEException` - if the queue/topic could not be dropped

enablePropagationSchedule(Session, String)

```
public void enablePropagationSchedule(javax.jms.Session session,
```

```
java.lang.String destination)
```

enable propagation schedule

Parameters

`session` - the JMS session

`destination` - the dblink of the destination database

Throws

`JMSEException` - if the propagation could not be enabled

getCompleteName()

```
public java.lang.String getCompleteName()
```

Get the complete name of the queue/topic, of the form, [schema].name

Returns

the complete name of the queue/topic

getCompleteTableName()

```
public java.lang.String getCompleteTableName()
```

Get the complete name of the queue table of the queue/topic of the form, [schema].name

Returns

the complete name of the queue/topic's queue table

getQueueName()

```
public java.lang.String getQueueName()
```

Get the name of the queue

Specified By

javax.jms.Queue.getQueueName() in interface javax.jms.Queue

Returns

the name of the queue

Throws

JMSException - if the queue is not a single consumer queue

getQueueOwner()

```
public java.lang.String getQueueOwner()
```

Get the owner of the queue

Returns

the schema of the queue

Throws

JMSException - if the schema could not be retrieved

getTopicName()

```
public java.lang.String getTopicName()
```

Get the name of the Topic

Specified By

`javax.jms.Topic.getTopicName()` in interface `javax.jms.Topic`

Returns

the name of the topic

Throws

`JMSEException` - if the queue is not a multi consumer queue (topic)

getTopicOwner()

```
public java.lang.String getTopicOwner()
```

Get the schema of the topic

Returns

the schema of the topic

Throws

`JMSEException` - if the schema could not be retrieved

grantQueuePrivilege(Session, String, String, boolean)

```
public void grantQueuePrivilege(javax.jms.Session session, java.lang.String  
privilege, java.lang.String grantee, boolean grant_option)
```

Grant enqueue or dequeue privilege on the queue to a database user

Parameters

`session` - the jms session

`privilege` - the privilege (ENQUEUE or DEQUEUE)

`grantee` - the user being granted the privilege

`grant_option` - whether the grantee can grant the privilege to others

Throws

`JMSEException` - if the privilege could not be granted

grantTopicPrivilege(Session, String, String, boolean)

```
public void grantTopicPrivilege(javax.jms.Session session, java.lang.String
```


privilege, java.lang.String grantee, boolean grant_option)
Grant a topic privilege

Parameters

session - the jms session

privilege - the privilege (ENQUEUE or DEQUEUE) being granted

grantee - the database user being granted the privilege

grant_option - whether the grantee can grant the privilege to other users

Throws

JMSEException - if the privilege could not be granted

revokeQueuePrivilege(Session, String, String)

public void revokeQueuePrivilege(javax.jms.Session session, java.lang.String
privilege, java.lang.String grantee)
Revoke a queue privilege

Parameters

session - the jms session

privilege - the privilege (ENQUEUE or DEQUEUE) being revoked

grantee - the database user from whom the privilege is being revoked

Throws

JMSEException - if the privilege could not be revoked

revokeTopicPrivilege(Session, String, String)

public void revokeTopicPrivilege(javax.jms.Session session, java.lang.String
privilege, java.lang.String grantee)
Revoke a topic privilege

Parameters

session - the jms session

privilege - the privilege (ENQUEUE or DEQUEUE) being revoked

grantee - the database user from whom the privilege is being revoked

Throws

`JMSEException` - if the privilege could not be revoked

schedulePropagation(Session, String, Date, Double, String, Double)

```
public void schedulePropagation(javax.jms.Session session, java.lang.String destination, java.util.Date start_time, java.lang.Double duration, java.lang.String next_time, java.lang.Double latency)
```

Schedule propagation from the topic for the given destination database

Parameters

`session` - the JMS session

`destination` - the dblink of the remote database for which propagation is being scheduled. A null string means that propagation will be scheduled for all subscribers in the database of the topic

`start_time` - the time propagation must be started

`duration` - the duration of propagation

`next_time` - the next time propagation must be done

`latency` - the latency in seconds that can be tolerated latency is the difference between the time a message was enqueued and the time it was propagated

Throws

`JMSEException` - if propagation could not be scheduled

start(Session, boolean, boolean)

```
public void start(javax.jms.Session session, boolean enqueue, boolean dequeue)
```

start the queue/topic for enqueue or dequeue or both

Parameters

`session` - the jms session

`enqueue` - whether enqueue should be enabled

`dequeue` - whether dequeue should be enabled

Throws

`JMSEException` - if failed to start the queue/topic

stop(Session, boolean, boolean, boolean)

```
public void stop(javax.jms.Session session, boolean enqueue, boolean dequeue,
boolean wait)
```

stop the queue/topic for enqueue or dequeue or both

Parameters

`session` - the jms session

`enqueue` - whether enqueue should be disabled

`dequeue` - whether dequeue should be disabled

`wait` - whether to wait for pending transactions on the queue/topic to complete

Throws

`JMSEException` - if failed to stop the queue/topic

toString()

```
public java.lang.String toString()
```

Get the queue/topic as a string, of the form [schema].name

Specified By

`javax.jms.Queue.toString()` in interface `javax.jms.Queue`

Overrides

`java.lang.Object.toString()` in class `java.lang.Object`

Returns

the queue/topic as a string

unschedulePropagation(Session, String)

```
public void unschedulePropagation(javax.jms.Session session, java.lang.String
destination)
```

Unschedule propagation between the topic and the specified destination

Parameters

`session` - the jms session

`destination` - the dblink of the destination database for which propagation must be unscheduled

Throws

`JMSEException` - if propagation could not be unscheduled

AQjmsDestinationProperty

```
public class AQjmsDestinationProperty
oracle.jms.AQjmsDestinationProperty
```

This class defines Destination properties

Member Summary	Description
Fields	-
NORMAL_QUEUE	-
EXCEPTION_QUEUE	-
INFINITE	infinite retention
Constructors	-
AQjmsDestinationProperty()	Constructor - initializes object with default destination properties
Methods	-
getQueueType	This method gets the queue type.
setQueueType	This method is used to set the queue type.
getMaxRetries	This method gets the maximum retries for dequeue with REMOVE mode.
setMaxRetries	This method sets the maximum retries for dequeue with REMOVE mode.
setRetryInterval	This method sets the retry interval, that is the time before this message is scheduled for processing after an application rollback. Default is 0.
getRetryInterval	This method gets the retry interval.
getRetentionTime	This method gets the retention time.
setRetentionTime	This method gets the retention time.
getComment	This method gets the queue comment.
setComment	This method sets the queue comment.

Constants

```
public static final int NORMAL_QUEUE
```

```
public static final int EXCEPTION_QUEUE
public static final int INFINITE /* infinite retention */
```

Constructors

AQjmsDestinationProperty()

```
public AQjmsDestinationProperty()
Constructor - initializes object with default destination properties
```

Methods

getQueueType

```
public int getQueueType() throws AQException
This method gets the queue type.
```

Returns

NORMAL_QUEUE or EXCEPTION_QUEUE

setQueueType

```
public void setQueueType(int q_type) throws AQException
This method is used to set the queue type.
```

Parameter	Meaning
q_type	NORMAL_QUEUE or EXCEPTION_QUEUE

getMaxRetries

```
public int getMaxRetries() throws AQException
This method gets the maximum retries for dequeue with REMOVE mode.
```

setMaxRetries

```
public void setMaxRetries(int retries) throws AQException
public void setMaxRetries(Integer retries) throws AQException
This method sets the maximum retries for dequeue with REMOVE mode.
```

Parameter	Meaning
retries	maximum retries for dequeue with REMOVE mode; specifying NULL will use the default. The default applies to single consumer queues and 8.1. compatible multiconsumer queues. Max_retries is not supported for 8.0 compatible multiconsumer queues.

setRetryInterval

public void setRetryInterval(double interval) throws AQException

public void setRetryInterval(Double interval) throws AQException

This method sets the retry interval, that is the time before this message is scheduled for processing after an application rollback. Default is 0.

Parameter	Meaning
interval	retry interval; specifying NULL will use the default

getRetryInterval

public double getRetryInterval() throws AQException

This method gets the retry interval.

getRetentionTime

public double getRetentionTime() throws AQException

This method gets the retention time.

setRetentionTime

public void setRetentionTime(double r_time) throws AQException

public void setRetentionTime(Double r_time) throws AQException

This method gets the retention time.

Parameter	Meaning
r_time	retention time; specifying NULL will use the default

getComment

public java.lang.String getComment() throws AQException

This method gets the queue comment.

setComment

`public void setComment(java.lang.String qt_comment) throws AQException`

This method sets the queue comment.

Parameter	Meaning
<code>qt_comment</code>	queue comment

AQjmsException

Syntax

```
public class AQjmsException extends javax.jms.JMSEException
```

```

java.lang.Object
|
+--java.lang.Throwable
    |
    +--java.lang.Exception
        |
        +--javax.jms.JMSEException
            |
            +--oracle.jms.AQjmsException

```

All Implemented Interfaces

```
java.io.Serializable
```

Description

This exception extends JMSEException - adds Oracle error codes. This is the root of all JMS exceptions

Member Summary	Description
Methods	-
getErrorNumber()	Get the Oracle Error code for the exception

Inherited Member Summary

Methods inherited from interface javax.jms.JMSEException

```
getErrorCode, getLinkedException, setLinkedException
```

Methods inherited from class java.lang.Throwable

```
fillInStackTrace, getLocalizedMessage, getMessage, printStackTrace,
printStackTrace, printStackTrace, toString
```

Methods inherited from class java.lang.Object

```
clone, equals, finalize, getClass, hashCode, notify, notifyAll,
wait, wait, wait
```

Methods

getErrorNumber()

```
public int getErrorNumber()
```

Get the Oracle Error code for the exception

AQjmsFactory

Syntax

```
public class AQjmsFactory extends java.lang.Object
```

```
java.lang.Object
|
+--oracle.jms.AQjmsFactory
```

Description

This class is used for accessing administered ConnectionFactory objects in Oracle's implementation of JMS

Member Summary	Description
Methods	-
<code>getQueueConnectionFactory(String, Properties)</code>	get a Queue Connection Factory
<code>getQueueConnectionFactory(String, String, int, String)</code>	get a Queue Connection Factory
<code>getTopicConnectionFactory(String, Properties)</code>	get a Topic Connection Factory
<code>getTopicConnectionFactory(String, String, int, String)</code>	get a Topic Connection Factory
<code>registerConnectionFactory(java.sql.Connection, String, String, String, int, String, String)</code>	Register a Queue or Topic Connection Factory in LDAP through the database
<code>registerConnectionFactory(java.sql.Connection, String, String, java.util.Properties, String)</code>	Register a Queue or Topic Connection Factory in LDAP through the database
<code>registerConnectionFactory(java.util.Hashtable, String, String, String, int, String, String)</code>	Register a Queue or Topic Connection Factory to LDAP
<code>registerConnectionFactory(java.util.Hashtable, String, String, java.util.Properties, String)</code>	Register a Queue or Topic Connection Factory in LDAP

Inherited Member Summary

Methods inherited from class java.lang.Object

(Cont.) Inherited Member Summary

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Methods

getQueueConnectionFactory(String, Properties)

```
public static javax.jms.QueueConnectionFactory  
getQueueConnectionFactory(java.lang.String jdbc_url, java.util.Properties info)  
Get a Queue Connection Factory
```

Parameters

`jdbc_url` - url to connect to
`info` - properties information

Returns

a Queue Connection Factory

Throws

`JMSEException` - if JMS fails to get a queue connection factory due to some JMS error

getQueueConnectionFactory(String, String, int, String)

```
public static javax.jms.QueueConnectionFactory  
getQueueConnectionFactory(java.lang.String hostname, java.lang.String oracle_  
sid, int portno, java.lang.String driver)  
Get a Queue Connection Factory
```

Parameters

`hostname` - the name of the host running Oracle
`oracle_sid` - the oracle system identifier
`portno` - the port number
`driver` - the type of jdbc driver (thin or oci8)

Returns

a Queue Connection Factory

Throws

`JMSEException` - if JMS fails to get a queue connection factory due to some JMS error

getTopicConnectionFactory(String, Properties)

```
public static javax.jms.TopicConnectionFactory  
getTopicConnectionFactory(java.lang.String jdbc_url, java.util.Properties info)  
get a Topic Connection Factory
```

Parameters

`jdbc_url` - url to connect to
`info` - properties information

Returns

a Topic Connection Factory

Throws

`JMSEException` - if JMS fails to get a queue connection factory due to some JMS error

getTopicConnectionFactory(String, String, int, String)

```
public static javax.jms.TopicConnectionFactory  
getTopicConnectionFactory(java.lang.String hostname, java.lang.String oracle_  
sid, int portno, java.lang.String driver)  
get a Topic Connection Factory
```

Parameters

`hostname` - the name of the host running Oracle
`oracle_sid` - the oracle system identifier
`portno` - the port number
`driver` - the type of jdbc driver (thin or oci8)

Returns

a Topic Connection Factory

Throws

`JMSEException` - if JMS fails to get a queue connection factory due to some JMS error

registerConnectionFactory(java.sql.Connection, String, String, String, int, String, String)

```
public static void registerConnectionFactory(  
    java.sql.Connection connection, String conn_name,  
    String hostname, String oracle_sid, int portno,  
    String driver, String type)  
    throws JMSEException
```

Register a Queue or Topic Connection Factory in the LDAP server associated with the Oracle database. The user can log on to the Oracle9i database first and then have the database update the LDAP entry. The user that logs on to the database must have the AQ_ADMINISTRATOR_ROLE to perform this operation.

Parameters

`connection` - a valid database connection

`conn_name` - the name of the Connection Factory to be registered

`hostname` - host name of the machine which hosts the database that the connection factory represents

`oracle_sid` - the oracle SID of the database that the connection factory represents

`portno` - the port number of the database

`driver` - the type of jdbc driver ("thin" or "oci8") to be used to connect to the database (JMS provider)

`type` - Specify "queue" to register a QueueConnectionFactory. Specify "topic" to register a TopicConnectionFactory

Throws

`JMSEException` - if JMS fails to register connection factory due to some JMS error

registerConnectionFactory(java.sql.Connection, String, String, java.util.Properties, String)

```
public static void registerConnectionFactory(  
    java.sql.Connection connection, String conn_name,  
    String jdbc_url, Properties info, String type)  
    throws JMSEException
```

Register a Queue or Topic Connection Factory in the LDAP server associated with the Oracle database. The user can log on to the Oracle9i database first and then have the database update the LDAP entry. The user that logs on to the database must have the AQ_ADMINISTRATOR_ROLE to perform this operation.

Parameters

`connection` - a valid database connection

`conn_name` - the name of the Connection Factory to be registered

`jdbc_url` - the JDBC URL to connect to the database that this factory represents

`info` - JDBC connection properties

`type` - Specify "queue" to register a QueueConnectionFactory. Specify "topic" to register a TopicConnectionFactory

Throws

`JMSEException` - if JMS fails to register connection factory due to some JMS error

registerConnectionFactory(java.util.Hashtable, String, String, String, int, String, String)

```
public static void registerConnectionFactory(
    java.util.Hashtable env, String conn_name,
    String hostname, String oracle_sid, int portno,
    String driver, String type)
    throws JMSEException
```

Register a Queue or Topic Connection Factory in LDAP server. This method allows you to register a connection factory in LDAP directly without connecting to the database.

The user must have the GLOBAL_AQ_USER_ROLE to register connection factories in LDAP

Parameters

`env` - a valid LDAP environment

`conn_name` - the name of the Connection Factory to be registered

`hostname` - host name of the machine which hosts the database that the connection factory represents

`oracle_sid` - the oracle SID of the database that the connection factory represents

`portno` - the port number of the database

`driver` - the type of jdbc driver ("thin" or "oci8") to be used to connect to the database (JMS provider)

`type` - Specify "queue" to register a QueueConnectionFactory. Specify "topic" to register a TopicConnectionFactory

Throws

`JMSEException` - if JMS fails to register connection factory due to some JMS error

registerConnectionFactory(`java.util.Hashtable`, `String`, `String`, `java.util.Properties`, `String`)

```
public static void registerConnectionFactory(  
    java.util.Hashtable env, String conn_name,  
    String jdbc_url, Properties info, String type)  
    throws JMSEException
```

Register a Queue or Topic Connection Factory in LDAP server. This method allows you to register a connection factory in LDAP directly without connecting to the database.

The user must have the `GLOBAL_AQ_USER_ROLE` to register connection factories in LDAP

Parameters

`env` - a valid LDAP environment

`conn_name` - the name of the Connection Factory to be registered

`jdbc_url` - the JDBC URL to connect to the database that this factory represents

`info` - JDBC connection properties

`type` - Specify "queue" to register a `QueueConnectionFactory`. Specify "topic" to register a `TopicConnectionFactory`

Throws

`JMSEException` - if JMS fails to register connection factory due to some JMS error

AQjmsInvalidDestinationException

Syntax

```
public class AQjmsInvalidDestinationException
    extends javax.jms.InvalidDestinationException
```

```
java.lang.Object
|
+--java.lang.Throwable
    |
    +--java.lang.Exception
        |
        +--javax.jms.JMSException
            |
            +--javax.jms.InvalidDestinationException
                |
                +--oracle.jms.AQjmsInvalidDestinationException
```

All Implemented Interfaces

```
java.io.Serializable
```

Description

This exception extends `InvalidDestinationException`. It is thrown when a Destination is not valid

Inherited Member Summary

Methods inherited from interface `javax.jms.JMSException`

```
getErrorCode, getLinkedException, setLinkedException
```

Methods inherited from class `java.lang.Throwable`

```
fillInStackTrace, getLocalizedMessage, getMessage, printStackTrace,
printStackTrace, printStackTrace, toString
```

Methods inherited from class `java.lang.Object`

```
clone, equals, finalize, getClass, hashCode, notify, notifyAll,
wait, wait, wait
```

AQjmsInvalidSelectorException

Syntax

```
public class AQjmsInvalidSelectorException
    extends javax.jms.InvalidSelectorException

java.lang.Object
|
+--java.lang.Throwable
    |
    +--java.lang.Exception
        |
        +--javax.jms.JMSException
            |
            +--javax.jms.InvalidSelectorException
                |
                +--oracle.jms.AQjmsInvalidSelectorException
```

All Implemented Interfaces

```
java.io.Serializable
```

Description

This exception extends `InvalidSelectorException`. It is thrown when the specified `MessageSelector` is not valid

Inherited Member Summary

Methods inherited from interface `javax.jms.JMSException`

```
getErrorCode, getLinkedException, setLinkedException
```

Methods inherited from class `java.lang.Throwable`

```
fillInStackTrace, getLocalizedMessage, getMessage, printStackTrace,
printStackTrace, printStackTrace, toString
```

Methods inherited from class `java.lang.Object`

```
clone, equals, finalize, getClass, hashCode, notify, notifyAll,
wait, wait, wait
```

AQjmsMapMessage

Syntax

```
public class AQjmsMapMessage extends AQjmsMessage
    implements javax.jms.MapMessage
```

```
java.lang.Object
|
+--AQjmsMessage
|
+---oracle.jms.AQjmsMapMessage
```

All Implemented Interfaces

javax.jms.MapMessage, javax.jms.Message

Description

This class implements the MapMessage interface. A MapMessage is used to send a set of name-value pairs where names are Strings and values are java primitive types

Member Summary	Description
Methods	-
<code>clearBody()</code>	Clear out the message body.
<code>clearProperties()</code>	Clear a message's properties.
<code>getBoolean(String)</code>	Return the boolean value with the given name.
<code>getByte(String)</code>	Return the byte value with the given name.
<code>getBytes(String)</code>	Return the byte array value with the given name.
<code>getChar(String)</code>	Return the Unicode character value with the given name.
<code>getDouble(String)</code>	Return the double value with the given name.
<code>getFloat(String)</code>	Return the float value with the given name.
<code>getInt(String)</code>	Return the integer value with the given name.
<code>getLong(String)</code>	Return the long value with the given name.
<code>getMapNames()</code>	Return an Enumeration of all the Map message's names.

(Cont.) Member Summary	Description
<code>getObject(String)</code>	Return the Java object value with the given name.
<code>getShort(String)</code>	Return the short value with the given name.
<code>getString(String)</code>	Set a String value with the given name, into the Map.
<code>itemExists(String)</code>	Check if an item exists in this MapMessage.
<code>setBoolean(String, boolean)</code>	Set a boolean value with the given name, into the Map.
<code>setByte(String, byte)</code>	Set a byte value with the given name, into the Map.
<code>setBytes(String, byte[])</code>	Set a byte array value with the given name, into the Map.
<code>setBytes(String, byte[], int, int)</code>	Set a portion of the byte array value with the given name, into the Map.
<code>setChar(String, char)</code>	Set a Unicode character value with the given name, into the Map.
<code>setDouble(String, double)</code>	Set a double value with the given name, into the Map.
<code>setFloat(String, float)</code>	Set a float value with the given name, into the Map.
<code>setInt(String, int)</code>	Set an integer value with the given name, into the Map.
<code>setLong(String, long)</code>	Set a long value with the given name, into the Map.
<code>setObject(String, Object)</code>	Set a Java object value with the given name, into the Map.
<code>setShort(String, short)</code>	Set a short value with the given name, into the Map.
<code>setString(String, String)</code>	Set a String value with the given name, into the Map.

Inherited Member Summary

Fields inherited from interface `javax.jms.Message`

`DEFAULT_DELIVERY_MODE`, `DEFAULT_PRIORITY`, `DEFAULT_TIME_TO_LIVE`

Methods inherited from class `AQjmsMessage`

(Cont.) Inherited Member Summary

```

getBooleanProperty(String), getByteProperty(String),
getDoubleProperty(String), getFloatProperty(String),
getIntProperty(String), getJMSCorrelationID(),
getJMSCorrelationIDAsBytes(), getJMSDeliveryMode(),
getJMSDestination(), getJMSExpiration(), getJMSMessageID(),
getJMSMessageIDAsBytes(), getJMSPriority(), getJMSRedelivered(),
getJMSReplyTo(), getJMSTimestamp(), getJMSType(),
getLongProperty(String), getObjectProperty(String),
getPropertyNames(), getSenderID(), getShortProperty(String),
getStringProperty(String), propertyExists(String),
setBooleanProperty(String, boolean), setByteProperty(String, byte),
setDoubleProperty(String, double), setFloatProperty(String, float),
setIntProperty(String, int), setJMSCorrelationID(String),
setJMSCorrelationIDAsBytes(byte[]), setJMSDestination(Destination),
setJMSExpiration(long), setJMSMessageID(String),
setJMSPriority(int), setJMSRedelivered(boolean),
setJMSReplyTo(Destination), setJMSTimestamp(long),
setJMSType(String), setLongProperty(String, long),
setObjectProperty(String, Object), setSenderID(AQjmsAgent),
setShortProperty(String, short), setStringProperty(String, String)

```

Methods inherited from class java.lang.Object

```

clone, equals, finalize, getClass, hashCode, notify, notifyAll,
toString, wait, wait, wait

```

Methods inherited from interface javax.jms.Message

```

getBooleanProperty, getByteProperty, getDoubleProperty,
getFloatProperty, getIntProperty, getJMSCorrelationID,
getJMSCorrelationIDAsBytes, getJMSDeliveryMode, getJMSDestination,
getJMSExpiration, getJMSMessageID, getJMSPriority,
getJMSRedelivered, getJMSReplyTo, getJMSTimestamp, getJMSType,
getLongProperty, getObjectProperty, getPropertyNames,
getShortProperty, getStringProperty, propertyExists,
setBooleanProperty, setByteProperty, setDoubleProperty,
setFloatProperty, setIntProperty, setJMSCorrelationID,
setJMSCorrelationIDAsBytes, setJMSDeliveryMode, setJMSDestination,
setJMSExpiration, setJMSMessageID, setJMSPriority,
setJMSRedelivered, setJMSReplyTo, setJMSTimestamp, setJMSType,
setLongProperty, setObjectProperty, setShortProperty,
setStringProperty

```

Methods

clearBody()

```
public void clearBody()
```

Clear out the message body. All other parts of the message are left untouched. The message can now be both read and written to.

Specified By

`javax.jms.Message.clearBody()` in interface `javax.jms.Message`

Overrides

`clearBody()` in class `AQjmsMessage`

Throws

`JMSEException` - if JMS fails to due to some internal JMS error.

clearProperties()

```
public void clearProperties()
```

Clear a message's properties.

Specified By

`javax.jms.Message.clearProperties()` in interface `javax.jms.Message`

Overrides

`clearProperties()` in class `AQjmsMessage`

Throws

`JMSEException` - if JMS fails to clear JMS message properties due to some internal JMS error.

getBoolean(String)

```
public boolean getBoolean(java.lang.String name)
```

Return the boolean value with the given name.

Specified By

`javax.jms.MapMessage.getBoolean(java.lang.String)` in interface `javax.jms.MapMessage`

Parameters

`name` - the name of the boolean

Returns

the boolean value with the given name.

Throws

`JMSEException` - if JMS fails to read message due to some internal JMS error.

`MessageFormatException` - if this type conversion is invalid.

getBytes(String)

```
public byte getByte(java.lang.String name)
Return the byte value with the given name.
```

Specified By

`javax.jms.MapMessage.getBytes(java.lang.String)` in interface `javax.jms.MapMessage`

Parameters

`name` - the name of the byte

Returns

the byte value with the given name.

Throws

`JMSEException` - if JMS fails to read message due to some internal JMS error.

`MessageFormatException` - if this type conversion is invalid.

getBytes(String)

```
public byte[] getBytes(java.lang.String name)
Return the byte array value with the given name.
```

Specified By

`javax.jms.MapMessage.getBytes(java.lang.String)` in interface `javax.jms.MapMessage`

Parameters

`name` - the name of the byte array

Returns

the byte array value with the given name. If there is no item by this name, a null value is returned.

Throws

`JMSEException` - if JMS fails to read message due to some internal JMS error.

`MessageFormatException` - if this type conversion is invalid.

getChar(String)

```
public char getChar(java.lang.String name)
```

Return the Unicode character value with the given name.

Specified By

`javax.jms.MapMessage.getChar(java.lang.String)` in interface `javax.jms.MapMessage`

Parameters

`name` - the name of the Unicode character

Returns

the Unicode character value with the given name.

Throws

`JMSEException` - if JMS fails to read message due to some internal JMS error.

`MessageFormatException` - if this type conversion is invalid.

getDouble(String)

```
public double getDouble(java.lang.String name)
```

Return the double value with the given name.

Specified By

`javax.jms.MapMessage.getDouble(java.lang.String)` in interface `javax.jms.MapMessage`

Parameters

`name` - the name of the double

Returns

the double value with the given name.

Throws

`JMSEException` - if JMS fails to read message due to some internal JMS error.

`MessageFormatException` - if this type conversion is invalid.

getFloat(String)

```
public float getFloat(java.lang.String name)
```

Return the float value with the given name.

Specified By

`javax.jms.MapMessage.getFloat(java.lang.String)` in interface `javax.jms.MapMessage`

Parameters

`name` - the name of the float

Returns

the float value with the given name.

Throws

`JMSEException` - if JMS fails to read message due to some internal JMS error.

`MessageFormatException` - if this type conversion is invalid.

getInt(String)

```
public int getInt(java.lang.String name)
```

Return the integer value with the given name.

Specified By

`javax.jms.MapMessage.getInt(java.lang.String)` in interface `javax.jms.MapMessage`

Parameters

`name` - the name of the integer

Returns

the integer value with the given name.

Throws

`JMSEException` - if JMS fails to read message due to some internal JMS error.

`MessageFormatException` - if this type conversion is invalid.

getLong(String)

```
public long getLong(java.lang.String name)
```

Return the long value with the given name.

Specified By

`javax.jms.MapMessage.getLong(java.lang.String)` in interface `javax.jms.MapMessage`

Parameters

`name` - the name of the long

Returns

the long value with the given name.

Throws

`JMSEException` - if JMS fails to read message due to some internal JMS error.

`MessageFormatException` - if this type conversion is invalid.

getMapNames()

```
public java.util.Enumeration getMapNames()
```

Return an Enumeration of all the Map message's names.

Specified By

`javax.jms.MapMessage.getMapNames()` in interface `javax.jms.MapMessage`

Returns

an enumeration of all the names in this Map message.

Throws

`JMSEException` - if JMS fails to read message due to some internal JMS error.

getObject(String)

```
public java.lang.Object getObject(java.lang.String name)
```

Return the Java object value with the given name.

Note that this method can be used to return in objectified format, an object that had been stored in the Map with the equivalent `setObject` method call, or it's equivalent primitive set method.

Specified By

`javax.jms.MapMessage.getObject(java.lang.String)` in interface `javax.jms.MapMessage`

Parameters

`name` - the name of the Java object

Returns

the Java object value with the given name, in objectified format (i.e. if it set as an int, then a Integer is returned). If there is no item by this name, a null value is returned.

Throws

`JMSEException` - if JMS fails to read message due to some internal JMS error.

getShort(String)

```
public short getShort(java.lang.String name)
```

Return the short value with the given name.

Specified By

`javax.jms.MapMessage.getShort(java.lang.String)` in interface `javax.jms.MapMessage`

Parameters

`name` - the name of the short

Returns

the short value with the given name.

Throws

`JMSEException` - if JMS fails to read message due to some internal JMS error.

`MessageFormatException` - if this type conversion is invalid.

getString(String)

```
public java.lang.String getString(java.lang.String name)
Set a String value with the given name, into the Map.
```

Specified By

`javax.jms.MapMessage.getString(java.lang.String)` in interface `javax.jms.MapMessage`

Parameters

`name` - the name of the String

`value` - the String value to set in the Map.

Throws

`JMSEException` - if JMS fails to write message due to some internal JMS error.

`MessageNotWriteableException` - if message in read-only mode.

itemExists(String)

```
public boolean itemExists(java.lang.String name)
Check if an item exists in this MapMessage.
```

Specified By

`javax.jms.MapMessage.itemExists(java.lang.String)` in interface `javax.jms.MapMessage`

Parameters

`name` - the name of the item to test

Returns

true if the item does exist.

Throws

`JMSEException` - if a JMS error occurs.

setBoolean(String, boolean)

```
public void setBoolean(java.lang.String name, boolean value)
Set a boolean value with the given name, into the Map.
```

Specified By

`javax.jms.MapMessage.setBoolean(java.lang.String, boolean)` in interface `javax.jms.MapMessage`

Parameters

`name` - the name of the boolean

`value` - the boolean value to set in the Map.

Throws

`JMSEException` - if JMS fails to write message due to some internal JMS error.

`MessageNotWriteableException` - if message in read-only mode.

setByte(String, byte)

```
public void setByte(java.lang.String name, byte value)
Set a byte value with the given name, into the Map.
```

Specified By

`javax.jms.MapMessage.setByte(java.lang.String, byte)` in interface `javax.jms.MapMessage`

Parameters

name - the name of the byte

value - the byte value to set in the Map.

Throws

`JMSEException` - if JMS fails to write message due to some internal JMS error.

`MessageNotWriteableException` - if message in read-only mode.

setBytes(String, byte[])

```
public void setBytes(java.lang.String name, byte[] value)
```

Set a byte array value with the given name, into the Map.

Specified By

`javax.jms.MapMessage.setBytes(java.lang.String, byte[])` in interface `javax.jms.MapMessage`

Parameters

name - the name of the byte array

value - the byte array value to set in the Map.

Throws

`JMSEException` - if JMS fails to write message due to some internal JMS error.

`MessageNotWriteableException` - if message in read-only mode.

setBytes(String, byte[], int, int)

```
public void setBytes(java.lang.String name, byte[] value, int offset, int length)
```

Set a portion of the byte array value with the given name, into the Map.

Specified By

`javax.jms.MapMessage.setBytes(java.lang.String, byte[], int, int)` in interface `javax.jms.MapMessage`

Parameters

name - the name of the byte array

value - the byte array value to set in the Map.

`offset` - the initial offset within the byte array.

`length` - the number of bytes to use.

Throws

`JMSEException` - if JMS fails to write message due to some internal JMS error.

`MessageNotWriteableException` - if message in read-only mode.

setChar(String, char)

```
public void setChar(java.lang.String name, char value)
```

Set a Unicode character value with the given name, into the Map.

Specified By

`javax.jms.MapMessage.setChar(java.lang.String, char)` in interface `javax.jms.MapMessage`

Parameters

`name` - the name of the Unicode character

`value` - the Unicode character value to set in the Map.

Throws

`JMSEException` - if JMS fails to write message due to some internal JMS error.

`MessageNotWriteableException` - if message in read-only mode.

setDouble(String, double)

```
public void setDouble(java.lang.String name, double value)
```

Set a double value with the given name, into the Map.

Specified By

`javax.jms.MapMessage.setDouble(java.lang.String, double)` in interface `javax.jms.MapMessage`

Parameters

`name` - the name of the double

`value` - the double value to set in the Map.

Throws

`JMSEException` - if JMS fails to write message due to some internal JMS error.

`MessageNotWriteableException` - if message in read-only mode.

setFloat(String, float)

```
public void setFloat(java.lang.String name, float value)
```

Set a float value with the given name, into the Map.

Specified By

`javax.jms.MapMessage.setFloat(java.lang.String, float)` in interface `javax.jms.MapMessage`

Parameters

`name` - the name of the float

`value` - the float value to set in the Map.

Throws

`JMSEException` - if JMS fails to write message due to some internal JMS error.

`MessageNotWriteableException` - if message in read-only mode.

setInt(String, int)

```
public void setInt(java.lang.String name, int value)
```

Set an integer value with the given name, into the Map.

Specified By

`javax.jms.MapMessage.setInt(java.lang.String, int)` in interface `javax.jms.MapMessage`

Parameters

`name` - the name of the integer

`value` - the integer value to set in the Map.

Throws

`JMSEException` - if JMS fails to write message due to some internal JMS error.

`MessageNotWriteableException` - if message in read-only mode.

setLong(String, long)

```
public void setLong(java.lang.String name, long value)
```

Set a long value with the given name, into the Map.

Specified By

`javax.jms.MapMessage.setLong(java.lang.String, long)` in interface `javax.jms.MapMessage`

Parameters

`name` - the name of the long

`value` - the long value to set in the Map.

Throws

`JMSEException` - if JMS fails to write message due to some internal JMS error.

`MessageNotWriteableException` - if message in read-only mode.

setObject(String, Object)

```
public void setObject(java.lang.String name, java.lang.Object value)
```

Set a Java object value with the given name, into the Map.

Note that this method only works for the objectified primitive object types (Integer, Double, Long ...), String's and byte arrays.

Specified By

`javax.jms.MapMessage.setObject(java.lang.String, java.lang.Object)` in interface `javax.jms.MapMessage`

Parameters

`name` - the name of the Java object

`value` - the Java object value to set in the Map.

Throws

`JMSEException` - if JMS fails to write message due to some internal JMS error.

`MessageFormatException` - if object is invalid

`MessageNotWriteableException` - if message in read-only mode.

setShort(String, short)

```
public void setShort(java.lang.String name, short value)
Set a short value with the given name, into the Map.
```

Specified By

`javax.jms.MapMessage.setShort(java.lang.String, short)` in interface `javax.jms.MapMessage`

Parameters

`name` - the name of the short
`value` - the short value to set in the Map.

Throws

`JMSEException` - if JMS fails to write message due to some internal JMS error.
`MessageNotWriteableException` - if message in read-only mode.

setString(String, String)

```
public void setString(java.lang.String name, java.lang.String value)
Set a String value with the given name, into the Map.
```

Specified By

`javax.jms.MapMessage.setString(java.lang.String, java.lang.String)` in interface `javax.jms.MapMessage`

Parameters

`name` - the name of the String
`value` - the String value to set in the Map.

Throws

`JMSEException` - if JMS fails to write message due to some internal JMS error.
`MessageNotWriteableException` - if message in read-only mode.

AQjmsMessage

Syntax

```
public class AQjmsMessage extends java.lang.Object
    implements javax.jms.Message
```

```
java.lang.Object
|
+--oracle.jms.AQjmsMessage
```

Direct Known Subclasses

AQjmsAdtMessage, AQjmsBytesMessage, AQjmsMapMessage, AQjmsObjectMessage, AQjmsStreamMessage, AQjmsTextMessage

All Implemented Interfaces

javax.jms.Message

Description

This class implements the Message interface. This is the superclass of all JMS messages

Member Summary	Description
Methods	-
<code>clearBody()</code>	Clear out the message body.
<code>clearProperties()</code>	Clear a message's properties.
<code>getBooleanProperty(String)</code>	Return the boolean property value with the given name.
<code>getByteProperty(String)</code>	Return the byte property value with the given name.
<code>getDoubleProperty(String)</code>	Return the double property value with the given name.
<code>getFloatProperty(String)</code>	Return the float property value with the given name.
<code>getIntProperty(String)</code>	Return the integer property value with the given name.

Member Summary	Description
<code>getJMSCorrelationID()</code>	Get the correlation ID for the message.
<code>getJMSCorrelationIDAsBytes()</code>	Get the correlation ID as an array of bytes for the message.
<code>getJMSDeliveryMode()</code>	Get the delivery mode for this message.
<code>getJMSDestination()</code>	Get the destination for this message.
<code>getJMSExpiration()</code>	Get the message's expiration value.
<code>getJMSMessageID()</code>	Get the message ID.
<code>getJMSMessageIDAsBytes()</code>	Get the message ID.
<code>getJMSPriority()</code>	Get the message priority.
<code>getJMSRedelivered()</code>	Get an indication of whether this message is being redelivered.
<code>getJMSReplyTo()</code>	Get the replyTo field for this message
<code>getJMSTimestamp()</code>	Get the message timestamp.
<code>getJMSType()</code>	Get the message type.
<code>getLongProperty(String)</code>	Return the long property value with the given name.
<code>getObjectProperty(String)</code>	Return the Java object property value with the given name.
<code>getPropertyNames()</code>	Return an Enumeration of all the property names.
<code>getSenderID()</code>	Get the message's senderID.
<code>getShortProperty(String)</code>	Return the short property value with the given name.
<code>getStringProperty(String)</code>	Return the String property value with the given name.
<code>propertyExists(String)</code>	Check if a property value exists.
<code>setBooleanProperty(String, boolean)</code>	Set a boolean property value with the given name, into the Message.
<code>setByteProperty(String, byte)</code>	Set a byte property value with the given name, into the Message.
<code>setDoubleProperty(String, double)</code>	Set a double property value with the given name, into the Message.

Member Summary	Description
<code>setFloatProperty(String, float)</code>	Set a float property value with the given name, into the Message.
<code>setIntProperty(String, int)</code>	Set an integer property value with the given name, into the Message.
<code>setJMSCorrelationID(String)</code>	Set the correlation ID for the message.
<code>setJMSCorrelationIDAsBytes(byte[])</code>	Set the correlation ID as an array of bytes for the message.
<code>setJMSDestination(Destination)</code>	Set the destination for this message.
<code>setJMSExpiration(long)</code>	Set the message's expiration value Providers set this field when a message is sent.
<code>setJMSMessageID(String)</code>	Set the message ID.
<code>setJMSPriority(int)</code>	Set the priority for this message.
<code>setJMSRedelivered(boolean)</code>	Set to indicate whether this message is being redelivered.
<code>setJMSReplyTo(Destination)</code>	Set where a reply to this message should be sent.
<code>setJMSTimestamp(long)</code>	Set the message timestamp.
<code>setJMSType(String)</code>	Set the message type.
<code>setLongProperty(String, long)</code>	Set a long property value with the given name, into the Message.
<code>setObjectProperty(String, Object)</code>	Set a Java object property value with the given name, into the Message.
<code>setSenderID(AQjmsAgent)</code>	Set the message's senderID.
<code>setShortProperty(String, short)</code>	Set a short property value with the given name, into the Message.
<code>setStringProperty(String, String)</code>	Set a String property value with the given name, into the Message.

Inherited Member Summary

Fields inherited from interface `javax.jms.Message`

`DEFAULT_DELIVERY_MODE`, `DEFAULT_PRIORITY`, `DEFAULT_TIME_TO_LIVE`

Methods inherited from class `java.lang.Object`

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`,
`toString`, `wait`, `wait`, `wait`

Methods

acknowledge()

```
public void acknowledge()
```

Used to acknowledge message receipt when using the `CLIENT_ACKNOWLEDGE` mode.

Specified By

`javax.jms.Message.acknowledge()` in interface `javax.jms.Message`

Throws

`JMSEException` - if internal exceptions occur during message acknowledge.

clearBody()

```
public void clearBody()
```

Clear out the message body. All other parts of the message are left untouched.

Specified By

`javax.jms.Message.clearBody()` in interface `javax.jms.Message`

Throws

`JMSEException` - if JMS fails to clear message

clearProperties()

```
public void clearProperties()
```

Clear a message's properties.

Specified By

`javax.jms.Message.clearProperties()` in interface `javax.jms.Message`

Throws

`JMSEException` - if JMS fails to clear JMS message properties due to some internal JMS error.

getBooleanProperty(String)

```
public boolean getBooleanProperty(java.lang.String name)
```

Return the boolean property value with the given name.

Specified By

`javax.jms.Message.getBooleanProperty(java.lang.String)` in interface `javax.jms.Message`

Parameters

`name` - the name of the boolean property

Returns

the boolean property value with the given name.

Throws

`JMSEException` - if JMS fails to get Property

`MessageFormatException` - if this type conversion is invalid.

getBytesProperty(String)

```
public byte[] getBytesProperty(java.lang.String name)
```

Return the byte property value with the given name.

Specified By

`javax.jms.Message.getBytesProperty(java.lang.String)` in interface `javax.jms.Message`

Parameters

`name` - the name of the byte property

Returns

the byte property value with the given name.

Throws

`JMSEException` - if JMS fails to get Property

`MessageFormatException` - if this type conversion is invalid.

getDoubleProperty(String)

```
public double getDoubleProperty(java.lang.String name)
```

Return the double property value with the given name.

Specified By

`javax.jms.Message.getDoubleProperty(java.lang.String)` in interface `javax.jms.Message`

Parameters

`name` - the name of the double property

Returns

the double property value with the given name.

Throws

`JMSEException` - if JMS fails to get Property

`MessageFormatException` - if this type conversion is invalid.

getFloatProperty(String)

```
public float getFloatProperty(java.lang.String name)
```

Return the float property value with the given name.

Specified By

`javax.jms.Message.getFloatProperty(java.lang.String)` in interface `javax.jms.Message`

Parameters

`name` - the name of the float property

Returns

the float property value with the given name.

Throws

`JMSEException` - if JMS fails to get Property

`MessageFormatException` - if this type conversion is invalid.

getIntProperty(String)

```
public int getIntProperty(java.lang.String name)
```

Return the integer property value with the given name.

Specified By

`javax.jms.Message.getIntProperty(java.lang.String)` in interface `javax.jms.Message`

Parameters

`name` - the name of the integer property

Returns

the integer property value with the given name.

Throws

`JMSEException` - if JMS fails to get Property

`MessageFormatException` - if this type conversion is invalid.

getJMSCorrelationID()

```
public java.lang.String getJMSCorrelationID()
```

Get the correlation ID for the message.

Specified By

`javax.jms.Message.getJMSCorrelationID()` in interface `javax.jms.Message`

Returns

the correlation ID of a message as a String.

Throws

`JMSEException` - if JMS fails to get correlationId due to some internal JMS error.

getJMSCorrelationIDsAsBytes()

```
public byte[] getJMSCorrelationIDsAsBytes()
```

Get the correlation ID as an array of bytes for the message.

Specified By

`javax.jms.Message.getJMSCorrelationIDsAsBytes()` in interface `javax.jms.Message`

Returns

the correlation ID of a message as an array of bytes.

Throws

`JMSEException` - if JMS fails to get `correlationId` due to some internal JMS error.

getJMSDeliveryMode()

```
public int getJMSDeliveryMode()
```

Get the delivery mode for this message.

Specified By

`javax.jms.Message.getJMSDeliveryMode()` in interface `javax.jms.Message`.

Returns

The delivery mode of this message, which is either `DeliverMode.PERSISTENT` or `DeliveryMode.NON_PERSISTENT`.

Throws

`JMSEException` - if JMS fails to get JMS `DeliveryMode` due to some internal JMS error.

getJMSDestination()

```
public javax.jms.Destination getJMSDestination()
```

Get the destination for this message. The destination field contains the destination to which the message is being sent. When a message is sent this value is ignored. After completion of the send method it holds the destination specified by the send. When a message is received, its destination value must be equivalent to the value assigned when it was sent.

Specified By

`javax.jms.Message.getJMSDestination()` in interface `javax.jms.Message`

Returns

the destination of this message.

Throws

`JMSException` - if JMS fails to get JMS Destination due to some internal JMS error.

getJMSExpiration()

```
public long getJMSExpiration()
```

Get the message's expiration value. When a message is sent, expiration is left unassigned. After completion of the send method, it holds the expiration time of the message. This is the sum of the time-to-live value specified by the client and the GMT at the time of the send. If the time-to-live is specified as zero, expiration is set to zero which indicates the message does not expire. When a message's expiration time is reached, the message is moved to the exception queue corresponding to the destination queue/topic

Specified By

`javax.jms.Message.getJMSExpiration()` in interface `javax.jms.Message`

Returns

the time the message expires. It is the sum of the time-to-live value specified by the client, and the GMT at the time of the send.

Throws

`JMSException` - if JMS fails to get JMS message expiration due to some internal JMS error.

See Also

`javax.jms.Message#setJMSExpiration()`

getJMSMessageID()

```
public java.lang.String getJMSMessageID()
```

Get the message ID. The `messageID` header field contains a value that uniquely identifies each message sent by a provider. When the send method Returnsit

contains a provider-assigned value. All JMSMessageID string values start with the prefix 'ID:'

Specified By

javax.jms.Message.getJMSMessageID() in interface javax.jms.Message

Returns

the message ID as a string (prefixed with 'ID:')

Throws

JMSEException - if JMS fails to get the message Id due to internal JMS error.

getJMSMessageIDAsBytes()

```
public byte[] getJMSMessageIDAsBytes()  
Get the message ID.
```

Returns

the message ID as a byte array

Throws

JMSEException - if JMS fails to get the message Id due to internal JMS error.

getJMSPriority()

```
public int getJMSPriority()  
Get the message priority. JMS defines a ten level priority value with 0 as the lowest  
priority and 9 as the highest.
```

Specified By

javax.jms.Message.getJMSPriority() in interface javax.jms.Message

Returns

the default message priority

getJMSRedelivered()

```
public boolean getJMSRedelivered()  
Get an indication of whether this message is being redelivered.
```

If a client receives a message with the redelivered indicator set, it is likely, but not guaranteed, that this message was delivered to the client earlier but the client did not commit the transaction

Specified By

`javax.jms.Message.getJMSRedelivered()` in interface `javax.jms.Message`

Returns

set to true if this message is being redelivered.

Throws

`JMSEException` - if JMS fails to get JMS Redelivered flag due to some internal JMS error.

getJMSReplyTo()

```
public javax.jms.Destination getJMSReplyTo()
```

Get the `replyTo` field for this message

Specified By

`javax.jms.Message.getJMSReplyTo()` in interface `javax.jms.Message`

Returns

`replyTo` destination (the format is a `AQjmsAgent`)

getJMSTimestamp()

```
public long getJMSTimestamp()
```

Get the message timestamp. The `JMSTimestamp` header field contains the time a message was handed off to a provider to be sent. When a message is sent, `JMSTimestamp` is ignored. When the send is complete - this method will contain the time the message was enqueued.

Specified By

`javax.jms.Message.getJMSTimestamp()` in interface `javax.jms.Message`

Throws

`JMSEException` - if JMS fails to get the Timestamp

getJMSType()

```
public java.lang.String getJMSType()  
Get the message type.
```

Specified By

javax.jms.Message.getJMSType() in interface javax.jms.Message

Returns

the message type

Throws

`JMSEException` - if JMS fails to get JMS message type due to some internal JMS error.

getLongProperty(String)

```
public long getLongProperty(java.lang.String name)  
Return the long property value with the given name.
```

Specified By

javax.jms.Message.getLongProperty(java.lang.String) in interface javax.jms.Message

Parameters

`name` - the name of the long property

Returns

the long property value with the given name.

Throws

`JMSEException` - if JMS fails to get Property

`MessageFormatException` - if this type conversion is invalid.

getObjectProperty(String)

```
public java.lang.Object getObjectProperty(java.lang.String name)  
Return the Java object property value with the given name. Note that this method  
can be used to return in objectified format, an object that had been stored as a  
property in the Message with the equivalent setObject method call, or it's  
equivalent primitive set method.
```

Specified By

`javax.jms.Message.getObjectProperty(java.lang.String)` in interface `javax.jms.Message`

Parameters

`name` - the name of the Java object property

Returns

the Java object property value with the given name, in objectified format (i.e. if it set as an int, then a Integer is returned). If there is no property by this name, a null value is returned.

Throws

`JMSEException` - if JMS fails to get Property due to some internal JMS error.

getPropertyNames()

```
public synchronized java.util.Enumeration getPropertyNames()  
Return an Enumeration of all the property names.
```

Specified By

`javax.jms.Message.getPropertyNames()` in interface `javax.jms.Message`

Returns

an enumeration of all the names of property values.

Throws

`JMSEException` - if JMS fails to get Property names due to some internal JMS error.

getSenderID()

```
public AQjmsAgent getSenderID()  
Get the message's senderID. This value is available only if it was set by the sender before sending the message
```

Throws

`JMSEException` - if JMS fails to get SenderID

getShortProperty(String)

```
public short getShortProperty(java.lang.String name)
```

Return the short property value with the given name.

Specified By

`javax.jms.Message.getShortProperty(java.lang.String)` in interface `javax.jms.Message`

Parameters

`name` - the name of the short property

Returns

the short property value with the given name.

Throws

`JMSEException` - if JMS fails to get Property

`MessageFormatException` - if this type conversion is invalid.

getStringProperty(String)

```
public java.lang.String getStringProperty(java.lang.String name)
```

Return the String property value with the given name.

Specified By

`javax.jms.Message.getStringProperty(java.lang.String)` in interface `javax.jms.Message`

Parameters

`name` - the name of the String property

Returns

the String property value with the given name. If there is no property by this name, a null value is returned.

Throws

`JMSEException` - if JMS fails to get Property

`MessageFormatException` - if this type conversion is invalid.

propertyExists(String)

```
public boolean propertyExists(java.lang.String name)
```

Check if a property value exists.

Specified By

`javax.jms.Message.propertyExists(java.lang.String)` in interface `javax.jms.Message`

Parameters

`name` - the name of the property to test

Returns

true if the property does exist.

Throws

`JMSEException` - if JMS fails to check if property exists due to some internal JMS error.

setBooleanProperty(String, boolean)

```
public void setBooleanProperty(java.lang.String name, boolean value)
```

Set a boolean property value with the given name, into the Message.

Specified By

`javax.jms.Message.setBooleanProperty(java.lang.String, boolean)` in interface `javax.jms.Message`

Parameters

`name` - the name of the boolean property

`value` - the boolean property value to set in the Message.

Throws

`JMSEException` - if JMS fails to set Property

`MessageNotWriteableException` - if properties are read-only

setByteProperty(String, byte)

```
public void setByteProperty(java.lang.String name, byte value)
```

Set a byte property value with the given name, into the Message.

Specified By

`javax.jms.Message.setByteProperty(java.lang.String, byte)` in interface `javax.jms.Message`

Parameters

`name` - the name of the byte property

`value` - the byte property value to set in the `Message`.

Throws

`JMSEException` - if JMS fails to set Property

`MessageNotWriteableException` - if properties are read-only

setDoubleProperty(String, double)

`public void setDoubleProperty(java.lang.String name, double value)`
Set a double property value with the given name, into the `Message`.

Specified By

`javax.jms.Message.setDoubleProperty(java.lang.String, double)` in interface `javax.jms.Message`

Parameters

`name` - the name of the double property

`value` - the double property value to set in the `Message`.

Throws

`JMSEException` - if JMS fails to set Property

`MessageNotWriteableException` - if properties are read-only

setFloatProperty(String, float)

`public void setFloatProperty(java.lang.String name, float value)`
Set a float property value with the given name, into the `Message`.

Specified By

`javax.jms.Message.setFloatProperty(java.lang.String, float)` in interface `javax.jms.Message`

Parameters

name - the name of the float property

value - the float property value to set in the Message.

Throws

`JMSEException` - if JMS fails to set Property

`MessageNotWriteableException` - if properties are read-only

setIntProperty(String, int)

```
public void setIntProperty(java.lang.String name, int value)
```

Set an integer property value with the given name, into the Message.

Specified By

`javax.jms.Message.setIntProperty(java.lang.String, int)` in interface `javax.jms.Message`

Parameters

name - the name of the integer property

value - the integer property value to set in the Message.

Throws

`JMSEException` - if JMS fails to set Property

`MessageNotWriteableException` - if properties are read-only

setJMSCorrelationID(String)

```
public void setJMSCorrelationID(java.lang.String correlationID)
```

Set the correlation ID for the message. A client can use the `JMSCorrelationID` header field to link one message with another.

Specified By

`javax.jms.Message.setJMSCorrelationID(java.lang.String)` in interface `javax.jms.Message`

Parameters

correlationID - the message ID of a message being referred to.

Throws

`JMSEException` - if JMS fails to set correlationId due to some internal JMS error.

setJMSCorrelationIDAsBytes(byte[])

```
public void setJMSCorrelationIDAsBytes(byte[] correlationID)
```

Set the correlation ID as an array of bytes for the message.

Specified By

`javax.jms.Message.setJMSCorrelationIDAsBytes(byte[])` in interface `javax.jms.Message`

Parameters

`correlationID` - the correlation ID value as an array of bytes.

Throws

`JMSEException` - if JMS fails to set correlationId due to some internal JMS error.

setJMSDeliveryMode()

```
public void setJMSDeliveryMode(int deliveryMode)
```

Used to set the delivery mode for the current mode. The delivery mode could be either `PERSISTENT` or `NON_PERSISTENT`.

Specified By

`javax.jms.Message.setJMSDeliveryMode(java.lang.int)` in interface `javax.jms.Message`

setJMSDestination(Destination)

```
public void setJMSDestination(javax.jms.Destination destination)
```

Set the destination for this message. Providers set this field when a message is sent.

Specified By

`javax.jms.Message.setJMSDestination(javax.jms.Destination)` in interface `javax.jms.Message`

Parameters

`destination` - the destination for this message.

Throws

`JMSEException` - if JMS fails to set JMS Destination due to some internal JMS error.

setJMSExpiration(long)

```
public void setJMSExpiration(long expiration)
```

Set the message's expiration value Providers set this field when a message is sent.

Specified By

`javax.jms.Message.setJMSExpiration(long)` in interface `javax.jms.Message`

Parameters

`expiration` - the message's expiration time

Throws

`JMSEException` - if JMS fails to set JMS message expiration due to some internal JMS error.

setJMSMessageID(String)

```
public void setJMSMessageID(java.lang.String id)
```

Set the message ID. Providers set this field when a message is sent.

Specified By

`javax.jms.Message.setJMSMessageID(java.lang.String)` in interface `javax.jms.Message`

Parameters

`id` - the ID of the message

Throws

`JMSEException` - if JMS fails to set the message Id due to internal JMS error.

setJMSPriority(int)

```
public void setJMSPriority(int priority)
```

Set the priority for this message. Providers set this field when a message is sent.

Specified By

`javax.jms.Message.setJMSPriority(int)` in interface `javax.jms.Message`

Parameters

`priority` - the priority of this message

Throws

`JMSEException` - if JMS fails to set JMS message priority due to some internal JMS error.

setJMSRedelivered(boolean)

```
public void setJMSRedelivered(boolean redelivered)
```

Set to indicate whether this message is being redelivered. This field is set at the time the message is delivered.

Specified By

`javax.jms.Message.setJMSRedelivered(boolean)` in interface `javax.jms.Message`

Parameters

`redelivered` - an indication of whether this message is being redelivered.

Throws

`JMSEException` - if JMS fails to set JMS Redelivered flag due to some internal JMS error.

setJMSReplyTo(Destination)

```
public void setJMSReplyTo(javax.jms.Destination replyTo)
```

Set where a reply to this message should be sent.

Specified By

`javax.jms.Message.setJMSReplyTo(javax.jms.Destination)` in interface `javax.jms.Message`

Parameters

`replyTo` - where to send a response to this message. The destination must be specified as an `AQJmsAgent` (with `consumer_name` and `queue/topic address`)

Throws

`JMSEException` - if JMS fails to set ReplyTo Destination due to some internal JMS error.

setJMSTimestamp(long)

```
public void setJMSTimestamp(long timestamp)
```

Set the message timestamp. Providers set this field when a message is sent.

Specified By

`javax.jms.Message.setJMSTimestamp(long)` in interface `javax.jms.Message`

Parameters

`timestamp` - the timestamp for this message

Throws

`JMSEException` - if JMS fails to set the timestamp due to some internal JMS error.

setJMSType(String)

```
public void setJMSType(java.lang.String type)
```

Set the message type.

Specified By

`javax.jms.Message.setJMSType(java.lang.String)` in interface `javax.jms.Message`

Parameters

`type` - of the message

Throws

`JMSEException` - if JMS fails to set JMS message type due to some internal JMS error.

setLongProperty(String, long)

```
public void setLongProperty(java.lang.String name, long value)
```

Set a long property value with the given name, into the Message.

Specified By

`javax.jms.Message.setLongProperty(java.lang.String, long)` in interface `javax.jms.Message`

Parameters

`name` - the name of the long property

`value` - the long property value to set in the `Message`.

Throws

`JMSEException` - if JMS fails to set Property

`MessageNotWriteableException` - if properties are read-only

setObjectProperty(String, Object)

```
public void setObjectProperty(java.lang.String name, java.lang.Object value)
```

Set a Java object property value with the given name, into the `Message`. Note that this method only works for the objectified primitive object types (`Integer`, `Double`, `Long` ...) and `String`'s.

Specified By

`javax.jms.Message.setObjectProperty(java.lang.String, java.lang.Object)` in interface `javax.jms.Message`

Parameters

`name` - the name of the Java object property.

`value` - the Java object property value to set in the `Message`.

Throws

`JMSEException` - if JMS fails to set Property

`MessageFormatException` - if object is invalid

`MessageNotWriteableException` - if properties are read-only

setSenderID(AQjmsAgent)

```
public void setSenderID(AQjmsAgent sender)
```

Set the message's `senderID`.

Throws

`JMSEException` - if JMS fails to set SenderID

setShortProperty(String, short)

```
public void setShortProperty(java.lang.String name, short value)
```

Set a short property value with the given name, into the Message.

Specified By

`javax.jms.Message.setShortProperty(java.lang.String, short)` in interface `javax.jms.Message`

Parameters

`name` - the name of the short property

`value` - the short property value to set in the Message.

Throws

`JMSEException` - if JMS fails to set Property

`MessageNotWriteableException` - if properties are read-only

setStringProperty(String, String)

```
public void setStringProperty(java.lang.String name, java.lang.String value)
```

Set a String property value with the given name, into the Message.

Specified By

`javax.jms.Message.setStringProperty(java.lang.String, java.lang.String)` in interface `javax.jms.Message`

Parameters

`name` - the name of the String property

`value` - the String property value to set in the Message.

Throws

`JMSEException` - if JMS fails to set Property

`MessageNotWriteableException` - if properties are read-only

AQjmsMessageEOFException

Syntax

```
public class AQjmsMessageEOFException
    extends javax.jms.MessageEOFException
```

```
java.lang.Object
|
+--java.lang.Throwable
|
+--java.lang.Exception
|
+--javax.jms.JMSEException
|
+--javax.jms.MessageEOFException
|
+--oracle.jms.AQjmsMessageEOFException
```

All Implemented Interfaces

```
java.io.Serializable
```

Description

This exception extends `MessageEOFException`. It is thrown when an unexpected end of stream has been reached when a `StreamMessage` or `BytesMessage` is being read

Inherited Member Summary

Methods inherited from interface `javax.jms.JMSEException`

```
getErrorCode, getLinkedException, setLinkedException
```

Methods inherited from class `java.lang.Throwable`

```
fillInStackTrace, getLocalizedMessage, getMessage, printStackTrace,
printStackTrace, printStackTrace, toString
```

Methods inherited from class `java.lang.Object`

```
clone, equals, finalize, getClass, hashCode, notify, notifyAll,
wait, wait, wait
```

AQjmsMessageFormatException

Syntax

```
public class AQjmsMessageFormatException
    extends javax.jms.MessageFormatException
```

```
java.lang.Object
|
+--java.lang.Throwable
    |
    +--java.lang.Exception
        |
        +--javax.jms.JMSException
            |
            +--javax.jms.MessageFormatException
                |
                +--oracle.jms.AQjmsMessageFormatException
```

All Implemented Interfaces

```
java.io.Serializable
```

Description

This exception extends MessageFormatException. It is thrown when a client attempts to use a datatype not supported by a message or attempts to read data in the message as the wrong type

Inherited Member Summary

Methods inherited from interface javax.jms.JMSException

```
getErrorCode, getLinkedException, setLinkedException
```

Methods inherited from class java.lang.Throwable

```
fillInStackTrace, getLocalizedMessage, getMessage, printStackTrace,
printStackTrace, printStackTrace, toString
```

Methods inherited from class java.lang.Object

```
clone, equals, finalize, getClass, hashCode, notify, notifyAll,
wait, wait, wait
```

AQjmsMessageNotReadableException

Syntax

```
public class AQjmsMessageNotReadableException
    extends javax.jms.MessageNotReadableException
```

```
java.lang.Object
|
+--java.lang.Throwable
    |
    +--java.lang.Exception
        |
        +--javax.jms.JMSEException
            |
            +--javax.jms.MessageNotReadableException
                |
                +--oracle.jms.AQjmsMessageNotReadableException
```

All Implemented Interfaces

```
java.io.Serializable
```

Description

This exception extends `MessageNotReadableException`. It is thrown when a client attempts to read a write-only message

Inherited Member Summary

Methods inherited from interface `javax.jms.JMSEException`

```
getErrorCode, getLinkedException, setLinkedException
```

Methods inherited from class `java.lang.Throwable`

```
fillInStackTrace, getLocalizedMessage, getMessage, printStackTrace,
printStackTrace, printStackTrace, toString
```

Methods inherited from class `java.lang.Object`

```
clone, equals, finalize, getClass, hashCode, notify, notifyAll,
wait, wait, wait
```

AQjmsMessageNotWriteableException

Syntax

```
public class AQjmsMessageNotWriteableException
    extends javax.jms.MessageNotWriteableException
```

```
java.lang.Object
|
+--java.lang.Throwable
    |
    +--java.lang.Exception
        |
        +--javax.jms.JMSException
            |
            +--javax.jms.MessageNotWriteableException
                |
                +--oracle.jms.AQjmsMessageNotWriteableException
```

All Implemented Interfaces

```
java.io.Serializable
```

Description

This exception extends `MessageNotWriteableException`. It is thrown when a client attempts to write a read-only message

Inherited Member Summary

Methods inherited from interface `javax.jms.JMSException`

```
getErrorCode, getLinkedException, setLinkedException
```

Methods inherited from class `java.lang.Throwable`

```
fillInStackTrace, getLocalizedMessage, getMessage, printStackTrace,
printStackTrace, printStackTrace, toString
```

Methods inherited from class `java.lang.Object`

```
clone, equals, finalize, getClass, hashCode, notify, notifyAll,
wait, wait, wait
```

AQjmsObjectMessage

Syntax

```
public class AQjmsObjectMessage extends AQjmsMessage
    implements javax.jms.ObjectMessage
```

```
java.lang.Object
|
+--AQjmsMessage
|
+--oracle.jms.AQjmsObjectMessage
```

All Implemented Interfaces

javax.jms.Message, javax.jms.ObjectMessage

Description

This class implements the ObjectMessage interface. An ObjectMessage is used to send a message that contains a serializable java object

Member Summary	Description
Methods	-
<code>clearBody()</code>	Clear out the message body.
<code>clearProperties()</code>	Clear a message's properties.
<code>getObject()</code>	Get the serializable object containing this message's data.
<code>setObject(Serializable)</code>	Set the serializable object containing this message's data.

Inherited Member Summary

Fields inherited from interface javax.jms.Message

DEFAULT_DELIVERY_MODE, DEFAULT_PRIORITY, DEFAULT_TIME_TO_LIVE

Methods inherited from class AQjmsMessage

Inherited Member Summary

```

getBooleanProperty(String), getByteProperty(String),
getDoubleProperty(String), getFloatProperty(String),
getIntProperty(String), getJMSCorrelationID(),
getJMSCorrelationIDAsBytes(), getJMSDeliveryMode(),
getJMSDestination(), getJMSExpiration(), getJMSMessageID(),
getJMSMessageIDAsBytes(), getJMSPriority(), getJMSRedelivered(),
getJMSReplyTo(), getJMSTimestamp(), getJMSType(),
getLongProperty(String), getObjectProperty(String),
getPropertyNames(), getSenderID(), getShortProperty(String),
getStringProperty(String), propertyExists(String),
setBooleanProperty(String, boolean), setByteProperty(String, byte),
setDoubleProperty(String, double), setFloatProperty(String, float),
setIntProperty(String, int), setJMSCorrelationID(String),
setJMSCorrelationIDAsBytes(byte[]), setJMSDestination(Destination),
setJMSExpiration(long), setJMSMessageID(String),
setJMSPriority(int), setJMSRedelivered(boolean),
setJMSReplyTo(Destination), setJMSTimestamp(long),
setJMSType(String), setLongProperty(String, long),
setObjectProperty(String, Object), setSenderID(AQjmsAgent),
setShortProperty(String, short), setStringProperty(String, String)

```

Methods inherited from class java.lang.Object

```

clone, equals, finalize, getClass, hashCode, notify, notifyAll,
toString, wait, wait, wait

```

Methods inherited from interface javax.jms.Message

```

getBooleanProperty, getByteProperty, getDoubleProperty,
getFloatProperty, getIntProperty, getJMSCorrelationID,
getJMSCorrelationIDAsBytes, getJMSDeliveryMode, getJMSDestination,
getJMSExpiration, getJMSMessageID, getJMSPriority,
getJMSRedelivered, getJMSReplyTo, getJMSTimestamp, getJMSType,
getLongProperty, getObjectProperty, getPropertyNames,
getShortProperty, getStringProperty, propertyExists,
setBooleanProperty, setByteProperty, setDoubleProperty,
setFloatProperty, setIntProperty, setJMSCorrelationID,
setJMSCorrelationIDAsBytes, setJMSDeliveryMode, setJMSDestination,
setJMSExpiration, setJMSMessageID, setJMSPriority,
setJMSRedelivered, setJMSReplyTo, setJMSTimestamp, setJMSType,
setLongProperty, setObjectProperty, setShortProperty,
setStringProperty

```

Methods

clearBody()

```
public void clearBody()
```

Clear out the message body. All other parts of the message are left untouched.

Specified By

`javax.jms.Message.clearBody()` in interface `javax.jms.Message`

Overrides

`clearBody()` in class `AQjmsMessage`

Throws

`JMSEException` - if JMS fails to due to some internal JMS error.

clearProperties()

```
public void clearProperties()
```

Clear a message's properties.

Specified By

`javax.jms.Message.clearProperties()` in interface `javax.jms.Message`

Overrides

`clearProperties()` in class `AQjmsMessage`

Throws

`JMSEException` - if JMS fails to clear JMS message properties due to some internal JMS error.

getObject()

```
public java.io.Serializable getObject()
```

Get the serializable object containing this message's data. The default value is null.

Specified By

`javax.jms.ObjectMessage.getObject()` in interface `javax.jms.ObjectMessage`

Returns

the serializable object containing this message's data

Throws

`JMSEException` - if JMS fails to get object due to some internal JMS error.

`MessageFormatException` - if object deserialization fails

setObject(Serializable)

```
public void setObject(java.io.Serializable object)
```

Set the serializable object containing this message's data.

Specified By

`javax.jms.ObjectMessage.setObject(java.io.Serializable)` in interface
`javax.jms.ObjectMessage`

Parameters

`object` - the message's data

Throws

`JMSEException` - if JMS fails to set object due to some internal JMS error.

`MessageFormatException` - if object serialization fails

`MessageNotWriteableException` - if message in read-only mode.

AQjmsOracleDebug

Syntax

```
public class AQjmsOracleDebug extends java.lang.Object
```

```
java.lang.Object  
|  
+--oracle.jms.AQjmsOracleDebug
```

Description

AQ Oracle Debug class - Do not use unless instructed by Oracle Support

Member Summary	Description
Methods	-
<code>getLogStream()</code>	Get log stream
<code>setLogStream(OutputStream)</code>	Set log stream
<code>setTraceLevel(int)</code>	Set trace level 0 - no tracing (default) 1 - fatal errors 2 - other errors, imp messages 3 - exception trace, other trace info 4 - method entry/exit 5 - print stack traces, variables

Inherited Member Summary

Methods inherited from class java.lang.Object

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

Methods

getLogStream()

```
public static java.io.OutputStream getLogStream()  
Get log stream
```

setLogStream(OutputStream)

```
public static void setLogStream(java.io.OutputStream output_stream)  
Set log stream
```

Parameters

output - log stream

setTraceLevel(int)

```
public static void setTraceLevel(int level)  
Set trace level 0 - no tracing (default) 1 - fatal errors 2 - other errors, imp messages 3  
- exception trace, other trace info 4 - method entry/exit 5 - print stack traces,  
variables
```

AQjmsProducer

Syntax

```
public class AQjmsProducer extends java.lang.Object
    implements AQjmsQueueSender, AQjmsTopicPublisher
```

```
java.lang.Object
|
+--oracle.jms.AQjmsProducer
```

All Implemented Interfaces

AQjmsQueueSender, AQjmsTopicPublisher, javax.jms.MessageProducer, javax.jms.QueueSender, javax.jms.TopicPublisher

Description

This class implements the MessageProducer interface. A MessageProducer is used to send messages to a Destination

Member Summary	Description
Methods	-
<code>close()</code>	Since a provider may allocate some resources on behalf of a MessageProducer outside the JVM, clients should close them when they are not needed.
<code>getDeliveryMode()</code>	Get the producer's default delivery mode.
<code>getDisableMessageID()</code>	Get an indication of whether message IDs are disabled.
<code>getDisableMessageTimestamp()</code>	Get an indication of whether message timestamps are disabled.
<code>getPriority()</code>	Get the producer's default priority.
<code>getQueue()</code>	Get the queue associated with this queue sender.
<code>getTimeToLive()</code>	Get the default length of time in milliseconds from its dispatch time that a produced message should be retained by the message system.
<code>getTopic()</code>	Get the topic associated with this publisher.
<code>publish(Message)</code>	Publish a Message to the topic

(Cont.) Member Summary	Description
<code>publish(Message, AQjmsAgent[])</code>	Publish a Message to a specific list of recipients
<code>publish(Message, AQjmsAgent[], int, int, long)</code>	Publish a Message to a topic by specifying a list of recipients, delivery mode, priority and time to live
<code>publish(Message, int, int, long)</code>	Publish a Message to the topic specifying delivery mode, priority and time to live to the topic.
<code>publish(Topic, Message)</code>	Publish a Message to a topic for an unidentified message producer.
<code>publish(Topic, Message, AQjmsAgent[])</code>	Publish a Message to a topic by specifying a list of recipients
<code>publish(Topic, Message, AQjmsAgent[], int, int, long)</code>	Publish a Message to a topic by specifying a list of recipients, delivery mode, priority and time to live
<code>publish(Topic, Message, int, int, long)</code>	Publish a Message to a topic for an unidentified message producer, specifying delivery mode, priority and time to live.
<code>send(Message)</code>	Send a message
<code>send(Message, int, int, long)</code>	Send a message.
<code>send(Queue, Message)</code>	Send a message.
<code>send(Queue, Message, int, int, long)</code>	Send a message.
<code>setDeliveryMode(int)</code>	Set the producer's default delivery mode.
<code>setDisableMessageID(boolean)</code>	Set whether message IDs are disabled.
<code>setDisableMessageTimestamp(boolean)</code>	Set whether message timestamps are disabled.
<code>setPriority(int)</code>	Set the producer's default priority.
<code>setTimeToLive(int)</code>	Set the default length of time in milliseconds from its dispatch time that a produced message should be retained by the message system.

Inherited Member Summary

Methods inherited from class `java.lang.Object`

(Cont.) Inherited Member Summary

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`,
`toString`, `wait`, `wait`, `wait`

Methods

close()

```
public void close()
```

Since a provider may allocate some resources on behalf of a `MessageProducer` outside the JVM, clients should close them when they are not needed. Relying on garbage collection to eventually reclaim these resources may not be timely enough.

Specified By

`javax.jms.MessageProducer.close()` in interface `javax.jms.MessageProducer`

Throws

`JMSEException` - if JMS fails to close the producer due to some error.

getDeliveryMode()

```
public synchronized int getDeliveryMode()
```

Get the producer's default delivery mode.

Specified By

`javax.jms.MessageProducer.getDeliveryMode()` in interface `javax.jms.MessageProducer`

Returns

the message delivery mode for this message producer.

Throws

`JMSEException` - if JMS fails to get delivery mode due to some internal error.

getDisableMessageID()

```
public synchronized boolean getDisableMessageID()
```

Get an indication of whether message IDs are disabled.

Specified By

`javax.jms.MessageProducer.getDisableMessageID()` in interface `javax.jms.MessageProducer`

Returns

an indication of whether message IDs are disabled.

Throws

`JMSEException` - if JMS fails to get disabled message Id due to some internal error.

getDisableMessageTimestamp()

```
public synchronized boolean getDisableMessageTimestamp()  
Get an indication of whether message timestamps are disabled.
```

Specified By

`javax.jms.MessageProducer.getDisableMessageTimestamp()` in interface `javax.jms.MessageProducer`

Returns

an indication of whether message IDs are disabled.

Throws

`JMSEException` - if JMS fails to get disabled message timestamp due to some internal error.

getPriority()

```
public synchronized int getPriority()  
Get the producer's default priority.
```

Specified By

`javax.jms.MessageProducer.getPriority()` in interface `javax.jms.MessageProducer`

Returns

the message priority for this message producer.

Throws

`JMSEException` - if JMS fails to get priority due to some internal error.

getQueue()

```
public synchronized javax.jms.Queue getQueue()
```

Get the queue associated with this queue sender.

Specified By

`javax.jms.QueueSender.getQueue()` in interface `javax.jms.QueueSender`

Returns

the queue

Throws

`JMSEException` - if JMS fails to get queue for this queue sender due to some internal error.

getTimeToLive()

```
public synchronized int getTimeToLive()
```

Get the default length of time in milliseconds from its dispatch time that a produced message should be retained by the message system.

Specified By

`javax.jms.MessageProducer.getTimeToLive()` in interface `javax.jms.MessageProducer`

Returns

the message time to live in milliseconds; zero is unlimited

Throws

`JMSEException` - if JMS fails to get Time to Live due to some internal error.

getTopic()

```
public synchronized javax.jms.Topic getTopic()
```

Get the topic associated with this publisher.

Specified By

`javax.jms.TopicPublisher.getTopic()` in interface `javax.jms.TopicPublisher`

Returns

this publisher's topic

Throws

`JMSEException` - if JMS fails to get topic for this topic publisher due to some internal error.

publish(Message)

```
public synchronized void publish(javax.jms.Message message)
Publish a Message to the topic
```

Specified By

`javax.jms.TopicPublisher.publish(javax.jms.Message)` in interface `javax.jms.TopicPublisher`

Parameters

`message` - The message to be published

Throws

`JMSEException` - if JMS fails to publish the message due to some internal error.

publish(Message, AQjmsAgent[])

```
public synchronized void publish(javax.jms.Message message, AQjmsAgent
recipient_list)
Publish a Message to a specific list of recipients
```

Specified By

`publish(Message, AQjmsAgent[])` in interface `AQjmsTopicPublisher`

Parameters

`message` - The message to be published

`recipient_list` - The list of recipients to which the message is published. The recipients are of type `AQjmsAgent`.

Throws

`JMSEException` - if JMS fails to publish the message due to some internal error.

publish(Message, AQjmsAgent[], int, int, long)

```
public synchronized void publish(javax.jms.Message message, AQjmsAgent
recipient_list, int deliveryMode, int priority, long timeToLive)
```

Publish a Message to a topic by specifying a list of recipients, delivery mode, priority and time to live

Specified By

`publish(Message, AQjmsAgent[], int, int, long)` in interface `AQjmsTopicPublisher`

Parameters

`message` - The message to be published

`recipient_list` - The list of recipients to which the message is published. The recipients are of type `AQjmsAgent`.

`deliveryMode` - The delivery mode - persistent or non_persistent

`priority` - The priority of the message

`timeToLive` - the message time to live in milliseconds; zero is unlimited

Throws

`JMSEException` - if JMS fails to publish the message due to some internal error.

publish(Message, int, int, long)

```
public synchronized void publish(javax.jms.Message message, int deliveryMode,
int priority, long timeToLive)
```

Publish a Message to the topic specifying delivery mode, priority and time to live to the topic.

Specified By

`javax.jms.TopicPublisher.publish(javax.jms.Message, int, int, long)` in interface `javax.jms.TopicPublisher`

Parameters

`message` - The message to be published

`deliveryMode` - The message delivery mode - persistent or non_persistent

`priority` - The priority of the message

`timeToLive` - the message time to live in milliseconds; zero is unlimited

Throws

`JMSEException` - if JMS fails to publish the message due to some internal error.

publish(Topic, Message)

```
public synchronized void publish(javax.jms.Topic topic, javax.jms.Message message)
```

Publish a Message to a topic for an unidentified message producer. Use the producer's default delivery mode, `timeToLive` and `priority`.

Specified By

`javax.jms.TopicPublisher.publish(javax.jms.Topic, javax.jms.Message)` in interface `javax.jms.TopicPublisher`

Parameters

`topic` - The topic to which to publish the message. This overrides the default topic of the Message Producer

`message` - The message to be published

Throws

`JMSEException` - if JMS fails to publish the message due to some internal error.

publish(Topic, Message, AQjmsAgent[])

```
public synchronized void publish(javax.jms.Topic topic, javax.jms.Message message, AQjmsAgent recipient_list)
```

Publish a Message to a topic by specifying a list of recipients

Specified By

`publish(Topic, Message, AQjmsAgent[])` in interface `AQjmsTopicPublisher`

Parameters

`topic` - The topic to which to publish the message. This overrides the default topic of the Message Producer

`message` - The message to be published

`recipient_list` - The list of recipients to which the message is published. The recipients are of type `AQjmsAgent`.

Throws

`JMSEException` - if JMS fails to publish the message due to some internal error.

publish(Topic, Message, AQjmsAgent[], int, int, long)

```
public synchronized void publish(javax.jms.Topic topic, javax.jms.Message message, AQjmsAgent recipient_list, int deliveryMode, int priority, long timeToLive)
```

Publish a Message to a topic by specifying a list of recipients, delivery mode, priority and time to live

Specified By

`publish(Topic, Message, AQjmsAgent[], int, int, long)` in interface `AQjmsTopicPublisher`

Parameters

`topic` - The topic to which to publish the message. This overrides the default topic of the Message Producer

`message` - The message to be published

`recipient_list` - The list of recipients to which the message is published. The recipients are of type `AQjmsAgent`.

`deliveryMode` - The delivery mode - persistent or non_persistent

`priority` - The priority of the message

`timeToLive` - the message time to live in milliseconds; zero is unlimited

Throws

`JMSEException` - if JMS fails to publish the message due to some internal error.

publish(Topic, Message, int, int, long)

```
public synchronized void publish(javax.jms.Topic topic, javax.jms.Message message, int deliveryMode, int priority, long timeToLive)
```

Publish a Message to a topic for an unidentified message producer, specifying delivery mode, priority and time to live.

Specified By

`javax.jms.TopicPublisher.publish(javax.jms.Topic, javax.jms.Message, int, int, long)`
in interface `javax.jms.TopicPublisher`

Parameters

`topic` - The topic to which to publish the message. This overrides the default topic of the Message Producer

`message` - The message to be published

`deliveryMode` - The message delivery mode - persistent or non_persistent

`priority` - The priority of the message

`timeToLive` - the message time to live in milliseconds; zero is unlimited

Throws

`JMSEException` - if JMS fails to publish the message due to some internal error.

send(Message)

```
public synchronized void send(javax.jms.Message message)
```

Send a message

Specified By

`javax.jms.QueueSender.send(javax.jms.Message)` in interface
`javax.jms.QueueSender`

Parameters

`message` - The message that has to be sent

Throws

`JMSEException` - if JMS fails to send the message due to some internal error.

send(Message, int, int, long)

```
public synchronized void send(javax.jms.Message message, int deliveryMode, int  
priority, long timeToLive)
```

Send a message.

Specified By

`javax.jms.QueueSender.send(javax.jms.Message, int, int, long)` in interface `javax.jms.QueueSender`

Parameters

`message` - The message that has to be sent

`deliverMode` - The message delivery mode - `persistent` or `non_persistent`

Throws

`JMSEException` - if JMS fails to send the message due to some internal error.

send(Queue, Message)

```
public synchronized void send(javax.jms.Queue queue, javax.jms.Message message)
Send a message.
```

Specified By

`javax.jms.QueueSender.send(javax.jms.Queue, javax.jms.Message)` in interface `javax.jms.QueueSender`

Parameters

`queue` - The destination queue where the message has to be sent. This overrides the default queue of the Message Producer.

`message` - The message that has to be sent

Throws

`JMSEException` - if JMS fails to send the message due to some internal error.

send(Queue, Message, int, int, long)

```
public synchronized void send(javax.jms.Queue queue, javax.jms.Message message,
int deliverMode, int priority, long timeToLive)
Send a message.
```

Specified By

`javax.jms.QueueSender.send(javax.jms.Queue, javax.jms.Message, int, int, long)` in interface `javax.jms.QueueSender`

Parameters

`queue` - The destination queue where the message has to be sent. This overrides the default queue of the Message Producer.

`message` - The message that has to be sent

`deliveryMode` - The message delivery mode - persistent or non_persistent

`priority` - The priority of the message

`timeToLive` - the message time to live in milliseconds; zero is unlimited

Throws

`JMSEException` - if JMS fails to send the message due to some internal error.

setDeliveryMode(int)

```
public synchronized void setDeliveryMode(int deliveryMode)
```

Set the producer's default delivery mode.

Delivery mode is set to PERSISTENT by default.

Specified By

`javax.jms.MessageProducer.setDeliveryMode(int)` in interface

`javax.jms.MessageProducer`

Parameters

`deliveryMode` - the message delivery mode for this message producer.

Throws

`JMSEException` - if JMS fails to set delivery mode due to some internal error.

setDisableMessageID(boolean)

```
public synchronized void setDisableMessageID(boolean value)
```

Set whether message IDs are disabled.

Since message ID's take some effort to create and increase a message's size, some JMS providers may be able to optimize message overhead if they are given a hint that message ID is not used by an application. JMS message Producers provide a hint to disable message ID. When a client sets a Producer to disable message ID they are saying that they do not depend on the value of message ID for the messages it produces. These messages must either have message ID set to null or, if the hint is ignored, messageID must be set to its normal unique value.

Message IDs are enabled by default.

Specified By

`javax.jms.MessageProducer.setDisableMessageID(boolean)` in interface `javax.jms.MessageProducer`

Parameters

`value` - indicates if message IDs are disabled.

Throws

`JMSException` - if JMS fails to set disabled message Id due to some internal error.

setDisableMessageTimestamp(boolean)

```
public synchronized void setDisableMessageTimestamp(boolean value)
Set whether message timestamps are disabled.
```

Specified By

`javax.jms.MessageProducer.setDisableMessageTimestamp(boolean)` in interface `javax.jms.MessageProducer`

Parameters

`value` - indicates if message timestamps are disabled.

Throws

`JMSException` - if JMS fails to set disabled message timestamp due to some internal error.

setPriority(int)

```
public synchronized void setPriority(int priority)
Set the producer's default priority.
```


Priority is set to 4, by default.

Specified By

`javax.jms.MessageProducer.setPriority(int)` in interface `javax.jms.MessageProducer`

Parameters

`priority` - the message priority for this message producer.

Throws

`JMSEException` - if JMS fails to set priority due to some internal error.

setTimeToLive(int)

```
public synchronized void setTimeToLive(int timeToLive)
```

Set the default length of time in milliseconds from its dispatch time that a produced message should be retained by the message system.

Time to live is set to zero by default.

Specified By

`javax.jms.MessageProducer.setTimeToLive(int)` in interface `javax.jms.MessageProducer`

Parameters

`timeToLive` - the message time to live in milliseconds; zero is unlimited

Throws

`JMSEException` - if JMS fails to set Time to Live due to some internal error.

AQjmsQueueBrowser

Syntax

```
public class AQjmsQueueBrowser extends java.lang.Object
    implements javax.jms.QueueBrowser, java.util.Enumeration
```

```
java.lang.Object
|
+--oracle.jms.AQjmsQueueBrowser
```

All Implemented Interfaces

java.util.Enumeration, javax.jms.QueueBrowser

Description

This class implements the QueueBrowser interface. A QueueBrowser is used to look at messages in a Queue without removing them

Member Summary	Description
Methods	-
close()	Since a provider may allocate some resources on behalf of a QueueBrowser outside the JVM, clients should close them when they are not needed.
getEnumeration()	Get an enumeration for browsing the current queue messages in the order they would be received.
getMessageSelector()	Get this queue browser's message selector expression.
getQueue()	Get the queue associated with this queue browser.
getTransformation()	get the transformation for this queue browser
hasMoreElements()	Tests if this enumeration contains more elements.
nextElement()	Returns the next element of this enumeration.
setTransformation(String)	set the transformation for this queue browser

Inherited Member Summary

Methods inherited from class java.lang.Object

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`,
`toString`, `wait`, `wait`, `wait`

Methods

close()

```
public void close()
```

Since a provider may allocate some resources on behalf of a QueueBrowser outside the JVM, clients should close them when they are not needed. Relying on garbage collection to eventually reclaim these resources may not be timely enough.

Specified By

`javax.jms.QueueBrowser.close()` in interface `javax.jms.QueueBrowser`

getEnumeration()

```
public java.util.Enumeration getEnumeration()
```

Get an enumeration for browsing the current queue messages in the order they would be received.

Specified By

`javax.jms.QueueBrowser.getEnumeration()` in interface `javax.jms.QueueBrowser`

Returns

an enumeration for browsing the messages

Throws

`JMSEException` - if JMS fails to get the enumeration for this browser due to some JMS error.

getMessageSelector()

```
public java.lang.String getMessageSelector()
```

Get this queue browser's message selector expression.

Specified By

`javax.jms.QueueBrowser.getMessageSelector()` in interface `javax.jms.QueueBrowser`

Returns

this queue browser's message selector

Throws

`JMSEException` - if JMS fails to get message selector due to some JMS error

getQueue()

```
public javax.jms.Queue getQueue()
```

Get the queue associated with this queue browser.

Specified By

`javax.jms.QueueBrowser.getQueue()` in interface `javax.jms.QueueBrowser`

Returns

the queue

Throws

`JMSEException` - if JMS fails to get the queue associated with this Browser due to some JMS error.

getTransformation()

```
public String getTransformation()
```

Get the transformation for this browser

Returns

the transformation

Throws

`JMSEException` - if there was an error in getting the transformation

hasMoreElements()

```
public boolean hasMoreElements()
Tests if this enumeration contains more elements.
```

Specified By

java.util.Enumeration.hasMoreElements() in interface java.util.Enumeration

Returns

true if more elements exist in the enumeration false otherwise.

nextElement()

```
public java.lang.Object nextElement()
Returns the next element of this enumeration.
```

Specified By

java.util.Enumeration.nextElement() in interface java.util.Enumeration

Returns

the next element of this enumeration

Throws

NoSuchElementException - if no more elements exist.

setTransformation(String)

```
public void setTransformation(String transformation)
Set transformation for this browser. This transformation will be applied before the
message is returned to the user.
```

Parameters

transformation - transformation to be applied before returning the message

Throws

JMSException - if there was an error in setting the transformation

AQjmsQueueConnectionFactory

Syntax

```
public class AQjmsQueueConnectionFactory extends java.lang.Object
    implements javax.jms.QueueConnectionFactory
```

```
java.lang.Object
|
+--oracle.jms.AQjmsQueueConnectionFactory
```

All Implemented Interfaces

```
javax.jms.ConnectionFactory, javax.jms.QueueConnectionFactory,
java.lang.Referenceable, java.lang.Serializable
```

Description

This class implements the QueueConnectionFactory interface. A QueueConnectionFactory is used to create QueueConnections

Member Summary	Description
Methods	-
createQueueConnection()	create a Queue Connection to the JMS Server hosting this Queue- ConnectionFactory.
createQueueConnection(Connection)	create a Queue Connection using the already open JDBC connection.
createQueueConnection(String, String)	create a Queue Connection using the given username and password for authentication during creation of the Connection.

Inherited Member Summary

Methods inherited from class java.lang.Object

```
clone, equals, finalize, getClass, hashCode, notify, notifyAll,
toString, wait, wait, wait
```

Methods

createQueueConnection()

```
public javax.jms.QueueConnection createQueueConnection()  
create a Queue Connection to the JMS Server hosting this Queue-  
ConnectionFactory.
```

Specified By

`javax.jms.QueueConnectionFactory.createQueueConnection()` in interface `javax.jms.QueueConnectionFactory`

Returns

a Queue Connection

Throws

`JMSEException` - if JMS fails to get a queue connection due to some JMS error

createQueueConnection(Connection)

```
public static javax.jms.QueueConnection  
createQueueConnection(java.sql.Connection jdbc_connection)  
create a Queue Connection using the already open JDBC connection. This creation  
does NOT result in creation of another connection to the database. Instead JMS  
binds to the given connection to the database and provides an interface to the  
Queuing mechanism defined by JMS.
```

Parameters

`jdbc_connection` - a valid open connection to the database.

Returns

a Queue Connection

Throws

`JMSEException` - if JMS fails to get a queue connection due to some JMS error

createQueueConnection(String, String)

```
public javax.jms.QueueConnection createQueueConnection(java.lang.String
```

`username, java.lang.String password)`

create a Queue Connection using the given username and password for authentication during creation of the Connection.

Specified By

`javax.jms.QueueConnectionFactory.createQueueConnection(java.lang.String, java.lang.String)` in interface `javax.jms.QueueConnectionFactory`

Parameters

`username` - name of the user connecting to the DB for Queueing. `password` password for the creating the connection to server.

Returns

a Queue Connection

Throws

`JMSException` - if JMS fails to get a queue connection due to some JMS error

AQjmsQueueReceiver

Syntax

```
public interface AQjmsQueueReceiver extends javax.jms.QueueReceiver
```

All Superinterfaces

```
javax.jms.MessageConsumer, javax.jms.QueueReceiver
```

All Known Implementing Classes

```
AQjmsConsumer
```

Description

This interface extends `javax.jms.QueueReceiver` and defines AQ extensions to JMS. A client uses a `QueueReceiver` for receiving messages that have been delivered to a `Queue`.

Member Summary	Description
Methods	-
<code>getNavigationMode()</code>	get the navigation mode used for receiving messages
<code>getTransformation()</code>	get the transformation for this receiver
<code>receiveNoData()</code>	Consume the message without returning it to the user.
<code>receiveNoData(long)</code>	Consume the message without returning it to the user.
<code>setNavigationMode(int)</code>	set the navigation mode used for receiving messages
<code>setTransformation(String)</code>	set the transformation for this receiver

Inherited Member Summary

Methods inherited from interface `javax.jms.QueueReceiver`

```
getQueue
```

Methods inherited from interface `javax.jms.MessageConsumer`

```
close, getMessageListener, getMessageSelector, receive, receive, receiveNoWait, setMessageListener
```

Methods

getNavigationMode()

```
public int getNavigationMode()  
get the navigation mode used for receiving messages
```

Returns

the navigation mode

Throws

`JMSEException` - if there was an error in getting the navigation mode

getTransformation()

```
public String getTransformation()  
Get the transformation for this receiver
```

Returns

the transformation

Throws

`JMSEException` - if there was an error in getting the transformation

receiveNoData()

```
public void receiveNoData()  
Consume the message without returning it to the user. This call will avoid the  
overhead of fetching the message from the database and hence can be used as an  
optimization by jms clients who have already got the message for example using a  
queue browser.
```

Throws

`JMSEException` - if the message could not be received due to an error

receiveNoData(long)

```
public void receiveNoData(long timeout)
```

Consume the message without returning it to the user. This call will avoid the overhead of fetching the message from the database and hence can be used as an optimization by jms clients who have already got the message for example using a queue browser. This call will block until a message arrives or the timeout expires

Parameters

`timeout` - the timeout value in milliseconds

Throws

`JMSEException` - if the message could not be received due to an error

setNavigationMode(int)

```
public void setNavigationMode(int mode)
set the navigation mode used for receiving messages
```

Parameters

`mode` - the new value of the navigation mode

Throws

`JMSEException` - if there was an error in getting the navigation mode

setTransformation(String)

```
public void setTransformation(String transformation)
Set transformation for this receiver. This transformation will be applied before the
message is returned to the user.
```

Parameters

`transformation` - transformation to be applied before returning the message

Throws

`JMSEException` - if there was an error in setting the transformation

AQjmsQueueSender

Syntax

```
public interface AQjmsQueueSender extends javax.jms.QueueSender
```

All Superinterfaces

`javax.jms.MessageProducer`, `javax.jms.QueueSender`

All Known Implementing Classes

`AQjmsProducer`

Description

This interface extends `QueueSender` and defines AQ extensions to JMS. A client uses a `QueueSender` to send messages to a `Queue`

Member Summary	Description
Methods	-
getTransformation()	get the transformation for this sender
setTransformation(String)	set the transformation for this sender

Inherited Member Summary

Methods inherited from interface `javax.jms.QueueSender`

`getQueue`, `send`, `send`, `send`, `send`

Methods inherited from interface `javax.jms.MessageProducer`

`close`, `getDeliveryMode`, `getDisableMessageID`,
`getDisableMessageTimestamp`, `getPriority`, `getTimeToLive`,
`setDeliveryMode`, `setDisableMessageID`, `setDisableMessageTimestamp`,
`setPriority`, `setTimeToLive`

Methods

getTransformation()

```
public String getTransformation()
```

Get the transformation for this sender

Returns

the transformation

Throws

`JMSEException` - if there was an error in getting the transformation

setTransformation(String)

```
public void setTransformation(String transformation)
```

Set transformation for this sender. This transformation will be applied before the message is inserted in the queue

Parameters

`transformation` - transformation to be applied before sending the message

Throws

`JMSEException` - if there was an error in setting the transformation

AQjmsSession

Syntax

```
public class AQjmsSession extends java.lang.Object
    implements javax.jms.QueueSession, javax.jms.TopicSession
```

```
java.lang.Object
|
+--oracle.jms.AQjmsSession
```

All Implemented Interfaces

```
javax.jms.QueueSession, java.lang.Runnable, javax.jms.Session,
javax.jms.TopicSession
```

Description

This class implements the `javax.jms.Session` interface. A JMS Session is a single threaded context for producing and consuming messages.

Member Summary	Description
Methods	-
<code>close()</code>	Close a JMS session Since a provider may allocate some resources on behalf of a Session outside the JVM, clients should close them when they are not needed.
<code>commit()</code>	Commit all messages done in this transaction and releases any locks currently held.
<code>createAdtMessage()</code>	Create an AdtMessage.
<code>createAdtMessage(CustomDatum)</code>	Create an initialized AdtMessage.
<code>createBrowser(Queue)</code>	Create a QueueBrowser to peek at the messages on the specified queue.
<code>createBrowser(Queue, CustomDatumFactory)</code>	Create a QueueBrowser to peek at the messages on the specified queue containing ADT messages.
<code>createBrowser(Queue, String)</code>	Create a QueueBrowser to peek at the messages on the specified queue.

(Cont.) Member Summary	Description
<code>createBrowser(Queue, String, boolean)</code>	Create a <code>QueueBrowser</code> to peek at the messages on the specified queue.
<code>createBrowser(Queue, String, CustomDatumFactory)</code>	Create a <code>QueueBrowser</code> to peek at the messages on the specified queue containing ADT messages.
<code>createBrowser(Queue, String, CustomDatumFactory, boolean)</code>	Create a <code>QueueBrowser</code> to peek at the messages on the specified queue containing ADT messages.
<code>createBytesMessage()</code>	Create a <code>BytesMessage</code> .
<code>createDurableSubscriber(Topic, String)</code>	Create a durable <code>Subscriber</code> to the specified topic.
<code>createDurableSubscriber(Topic, String, CustomDatumFactory)</code>	Create a durable <code>Subscriber</code> to the specified topic.
<code>createDurableSubscriber(Topic, String, String, boolean)</code>	Create a durable <code>Subscriber</code> to the specified topic.
<code>createDurableSubscriber(Topic, String, String, boolean, String)</code>	Create a durable <code>Subscriber</code> to the specified topic. Specify transformation for the subscriber
<code>createDurableSubscriber(Topic, String, String, boolean, CustomDatumFactory, String)</code>	Create a durable <code>Subscriber</code> to the specified Oracle Object (ADT) topic.
<code>createDurableSubscriber(Topic, String, String, boolean, CustomDatumFactory, String)</code>	Create a durable <code>Subscriber</code> to the specified Oracle Object (ADT) topic. Specify transformation for the subscriber
<code>createMapMessage()</code>	Create a <code>MapMessage</code> .
<code>createObjectMessage()</code>	Create an <code>ObjectMessage</code> .
<code>createObjectMessage(Serializable)</code>	Create an initialized <code>ObjectMessage</code> .
<code>createPublisher(Topic)</code>	Create a <code>Publisher</code> for the specified topic.
<code>createQueue(AQQueueTable, String, AQjmsDestinationProperty)</code>	Create a queue.
<code>createQueueTable(String, String, AQQueueTableProperty)</code>	Create a <code>Queue Table</code> .
<code>createReceiver(Queue)</code>	Create a <code>QueueReceiver</code> to receive messages from the specified queue.

(Cont.) Member Summary	Description
<code>createReceiver(Queue, CustomDatumFactory)</code>	Create a QueueReceiver to receive messages from the specified queue containing ADT messages.
<code>createReceiver(Queue, String)</code>	Create a QueueReceiver to receive messages from the specified queue.
<code>createReceiver(Queue, String, CustomDatumFactory)</code>	Create a QueueReceiver to receive messages from the specified queue containing ADT messages.
<code>createRemoteSubscriber(Topic, AQjmsAgent, String)</code>	Create a remote subscriber for a topic.
<code>createRemoteSubscriber(Topic, AQjmsAgent, String, String)</code>	Create a remote subscriber for a topic. Specify transformation for the remote subscriber
<code>createRemoteSubscriber(Topic, AQjmsAgent, String, CustomDatumFactory)</code>	Create a remote subscriber for a topic.
<code>createRemoteSubscriber(Topic, AQjmsAgent, String, CustomDatumFactory, String)</code>	Create a remote subscriber for a Oracle Object (ADT) topic. Specify transformation for the remote subscriber
<code>createRemoteSubscriber(Topic, AQjmsAgent, String, CustomDatumFactory, String)</code>	Create a QueueSender to send messages to the specified queue.
<code>createStreamMessage()</code>	Create a StreamMessage.
<code>createSubscriber(Topic)</code>	Create a non-durable Subscriber to the specified topic.
<code>createSubscriber(Topic, String, boolean)</code>	Create a non-durable Subscriber to the specified topic.
<code>createTextMessage()</code>	Create a TextMessage.
<code>createTextMessage(StringBuffer)</code>	Create an initialized TextMessage.
<code>createTopic(AQQueueTable, String, AQjmsDestinationProperty)</code>	Create a topic
<code>createTopicReceiver(Topic, String, String)</code>	Create a TopicReceiver to receive messages from the specified topic.
<code>createTopicReceiver(Topic, String, String, CustomDatumFactory)</code>	

(Cont.) Member Summary	Description
<code>getDBConnection()</code>	
<code>getJmsConnection()</code>	
<code>getMessageListener()</code>	Return the session's distinguished message listener.
<code>getQueue(String, String)</code>	Get an existing queue.
<code>getQueueTable(String, String)</code>	Get a handle to an existing queue-table If owner of queue-table is not the same as the user which opened the connection, the caller must have AQ enqueue/dequeue privileges on queues/topics in the queue table.
<code>getTopic(String, String)</code>	Get an existing topic.
<code>getTransacted()</code>	Checks if the session in transacted mode?
<code>grantSystemPrivilege(String, String, boolean)</code>	Grant AQ system privileges to users/roles.
<code>revokeSystemPrivilege(String, String)</code>	Revoke AQ system privilege from user/roles
<code>rollback()</code>	Rollback any messages done in this transaction and releases any locks currently held.
<code>run()</code>	
<code>setMessageListener(MessageListener)</code>	Set the session's distinguished message listener.
<code>unsubscribe(Topic, AQjmsAgent)</code>	Unsubscribe a remote durable subscription that has been created by a client on the specified topic
<code>unsubscribe(Topic, String)</code>	Unsubscribe a durable subscription that has been created by a client on the specified topic

Inherited Member Summary

Fields inherited from interface `javax.jms.Session`

`AUTO_ACKNOWLEDGE`, `CLIENT_ACKNOWLEDGE`, `DUPS_OK_ACKNOWLEDGE`

Methods inherited from class `java.lang.Object`

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

Methods

close()

```
public void close()
```

Close a JMS session. Since a provider may allocate some resources on behalf of a Session outside the JVM, clients should close them when they are not needed. Relying on garbage collection to eventually reclaim these resources may not be timely enough. This call may take a couple of minutes if there are receivers blocked on a receive call with infinite timeout.

Specified By

`javax.jms.Session.close()` in interface `javax.jms.Session`

Specified By

`javax.jms.Session.close()` in interface `javax.jms.Session`

Throws

`JMSEXception` - if JMS implementation fails to close a Session due to some internal error.

commit()

```
public synchronized void commit()
```

Commit all messages done in this transaction and releases any locks currently held.

Specified By

`javax.jms.Session.commit()` in interface `javax.jms.Session`

Throws

`JMSEXception` - if JMS implementation fails to commit the transaction due to some internal error. The linked SQL exception has more info about the error.

createAdtMessage()

```
public synchronized AdtMessage createAdtMessage()
```

Create an `AdtMessage`. An `AdtMessage` is used to send a message that containing a Java object that maps to a Oracle SQL ADT. This object must support the `OracleCustomDatum` interface.

Throws

`JMSEException` - if some error occurs during message creation

createAdtMessage(CustomDatum)

```
public synchronized AQjmsAdtMessage createAdtMessage(oracle.sql.CustomDatum  
payload)
```

Create an initialized `AdtMessage`. An `AQjmsAdtMessage` is used to send a message that containing an Java object that maps to a Oracle SQL ADT. This object must support the `OracleCustomDatum` interface.

Parameters

`payload` - the object to use to initialize this message.

Throws

`JMSEException` - if some error occurs during message creation

createBrowser(Queue)

```
public synchronized javax.jms.QueueBrowser createBrowser(javax.jms.Queue queue)
```

Create a `QueueBrowser` to peek at the messages on the specified queue. This method can be used to create browsers for queues that contain payloads of type `AQ$JMS_TEXT_MESSAGE`, `AQ$JMS_STREAM_MESSAGE`, `AQ$JMS_BYTES_MESSAGE`, `AQ$JMS_MAP_MESSAGE` or `AQ$JMS_OBJECT_MESSAGE`

Specified By

`javax.jms.QueueSession.createBrowser(javax.jms.Queue)` in interface `javax.jms.QueueSession`

Parameters

`queue` - the queue to access

Throws

`JMSEException` - if a session fails to create a browser due to some JMS error.

`InvalidDestinationException` - if invalid Queue specified.

createBrowser(Queue, CustomDatumFactory)

```
public synchronized javax.jms.QueueBrowser createBrowser(javax.jms.Queue queue,
```

```
oracle.sql.CustomDatumFactory payload_factory)
```

Create a `QueueBrowser` to peek at the messages on the specified queue containing ADT messages. This method is used to create receivers for queues that contain Oracle ADT payloads (instead of the standard JMS defined payloads)

Parameters

`queue` - the queue to access

`payload_factory` - `CustomDatumFactory` for the java class that maps to the Oracle ADT

Throws

`JMSException` - if a session fails to create a browser due to some JMS error.

`InvalidDestinationException` - if invalid Queue specified.

createBrowser(Queue, String)

```
public synchronized javax.jms.QueueBrowser createBrowser(javax.jms.Queue queue,  
java.lang.String messageSelector)
```

Create a `QueueBrowser` to peek at the messages on the specified queue. This method can be used to create browsers for queues that contain payloads of type `AQ$JMS_TEXT_MESSAGE`, `AQ$JMS_STREAM_MESSAGE`, `AQ$JMS_BYTES_MESSAGE`, `AQ$JMS_MAP_MESSAGE` or `AQ$JMS_OBJECT_MESSAGE`

Specified By

`javax.jms.QueueSession.createBrowser(javax.jms.Queue, java.lang.String)` in interface `javax.jms.QueueSession`

Parameters

`queue` - the queue to access

`messageSelector` - only messages with properties matching the message selector expression are delivered. The selector can be any expression that has a combination of one or more of the following:

- `JMSMessageID = 'ID:23452345'` to retrieve messages that have a specified message ID
- JMS Message header fields or properties:
`JMSPriority < 3 AND JMSCorrelationID = 'Fiction'`
- User defined message properties:

```
color IN ('RED', 'BLUE', 'GREEN') AND price < 30000
```

All message IDs must be prefixed with "ID:"

Throws

`JMSEException` - if a session fails to create a browser due to some JMS error.

`InvalidDestinationException` - if invalid Queue specified.

`InvalidSelectorException` - if the message selector is invalid.

createBrowser(Queue, String, boolean)

```
public synchronized javax.jms.QueueBrowser createBrowser(javax.jms.Queue queue,
    java.lang.String messageSelector, boolean locked)
```

Create a `QueueBrowser` to peek at the messages on the specified queue. This method can be used to create browsers for queues that contain payloads of type `AQ$JMS_TEXT_MESSAGE`, `AQ$JMS_STREAM_MESSAGE`, `AQ$JMS_BYTES_MESSAGE`, `AQ$JMS_MAP_MESSAGE` or `AQ$JMS_OBJECT_MESSAGE`

Parameters

`queue` - the queue to access

`messageSelector` - only messages with properties matching the message selector expression are delivered

The selector can be any expression that has a combination of one or more of the following:

- `JMSMessageID = 'ID:23452345'` to retrieve messages that have a specified message ID
- JMS Message header fields or properties:


```
JMSPriority < 3 AND JMSCorrelationID = 'Fiction'
```
- User defined message properties:


```
color IN ('RED', 'BLUE', 'GREEN') AND price < 30000
```

All message IDs must be prefixed with "ID:"

`locked` - if true then messages are locked as they are browsed (similar to a `SELECT` for `UPDATE`)

Throws

`JMSEException` - if a session fails to create a browser due to some JMS error.

`InvalidDestinationException` - if invalid Queue specified.

`InvalidSelectorException` - if the message selector is invalid.

createBrowser(Queue, String, CustomDatumFactory)

```
public synchronized javax.jms.QueueBrowser createBrowser(javax.jms.Queue queue,  
java.lang.String messageSelector, oracle.sql.CustomDatumFactory payload_factory)
```

Create a `QueueBrowser` to peek at the messages on the specified queue containing ADT messages. This method is used to create browsers for queues that contain Oracle ADT payloads (instead of the standard JMS defined payloads)

Parameters

`queue` - the queue to access

`messageSelector` - only messages with properties matching the message selector expression are delivered. For queues containing `AdtMessages` the selector for `QueueBrowser` can be a SQL expression on the message payload contents or `messageID` or `priority` or `correlationID`.

- Selector on message id - to retrieve messages that have a specific `messageID`

```
msgid = '23434556566767676'
```

Note: in this case message IDs must NOT be prefixed with 'ID:'

- Selector on priority or correlation is specified as follows

```
priority < 3 AND corrid = 'Fiction'
```

- Selector on message payload is specified as follows

```
tab.user_data.color = 'GREEN' AND tab.user_data.price < 30000
```

`payload_factory` - `CustomDatumFactory` for the java class that maps to the Oracle ADT

Throws

`JMSEException` - if a session fails to create a browser due to some JMS error.

`InvalidDestinationException` - if invalid Queue specified.

createBrowser(Queue, String, CustomDatumFactory, boolean)

```
public synchronized javax.jms.QueueBrowser createBrowser(javax.jms.Queue queue,  
java.lang.String messageSelector, oracle.sql.CustomDatumFactory payload_factory,  
boolean locked)
```

Create a `QueueBrowser` to peek at the messages on the specified queue containing ADT messages. This method is used to create browsers for queues that contain Oracle ADT payloads (instead of the standard JMS defined payloads)

Parameters

`queue` - the queue to access

`messageSelector` - only messages with properties matching the message selector expression are delivered. For queues containing `AdtMessages` the selector for `QueueBrowser` can be a SQL expression on the message payload contents or `messageID` or `priority` or `correlationID`.

- Selector on message id - to retrieve messages that have a specific `messageID`

```
msgid = '234345566767676'
```

Note: in this case message IDs must NOT be prefixed with 'ID:'

- Selector on priority or correlation is specified as follows

```
priority < 3 AND corrid = 'Fiction'
```

- Selector on message payload is specified as follows

```
tab.user_data.color = 'GREEN' AND tab.user_data.price < 30000
```

`payload_factory` - `CustomDatumFactory` for the java class that maps to the Oracle ADT

`locked` - if true then messages are locked as they are browsed (similar to a `SELECT` for `UPDATE`)

Throws

`JMSEException` - if a session fails to create a browser due to some JMS error.

`InvalidDestinationException` - if invalid Queue specified.

createBytesMessage()

```
public synchronized javax.jms.BytesMessage createBytesMessage()
```

Create a `BytesMessage`. A `BytesMessage` is used to send a message containing a stream of uninterpreted bytes.

Specified By

`javax.jms.Session.createBytesMessage()` in interface `javax.jms.Session`

Throws

`JMSEException` - if some error occurs during message creation

createDurableSubscriber(Topic, String)

```
public synchronized javax.jms.TopicSubscriber  
createDurableSubscriber(javax.jms.Topic topic, java.lang.String subs_name)
```

Create a durable Subscriber to the specified topic. This method can be used to create subscribers for queues that contain payloads of type `AQS_JMS_TEXT_MESSAGE`, `AQS_JMS_STREAM_MESSAGE`, `AQS_JMS_BYTES_MESSAGE`, `AQS_JMS_MAP_MESSAGE` or `AQS_JMS_OBJECT_MESSAGE`. A client can change an existing durable subscription by creating a durable `TopicSubscriber` with the same name and message selector.

Specified By

`javax.jms.TopicSession.createDurableSubscriber(javax.jms.Topic, java.lang.String)` in interface `javax.jms.TopicSession`

Parameters

`topic` - the topic to subscribe to

`name` - the name used to identify this subscription.

Throws

`JMSEException` - if a session fails to create a subscriber due to some JMS error.

`InvalidDestinationException` - if invalid Topic specified.

createDurableSubscriber(Topic, String, CustomDatumFactory)

```
public synchronized javax.jms.TopicSubscriber  
createDurableSubscriber(javax.jms.Topic topic, java.lang.String subs_name,  
oracle.sql.CustomDatumFactory payload_factory)
```

Create a durable Subscriber to the specified topic. This method is used to create browsers for queues that contain Oracle ADT payloads (instead of the standard JMS defined payloads). A client can change an existing durable subscription by creating a durable `TopicSubscriber` with the same name and message selector.

Parameters

`topic` - the topic to subscribe to

`name` - the name used to identify this subscription.

`payload_factory` - CustomDatumFactory for the java class that maps to the Oracle ADT

Throws

`JMSEException` - if a session fails to create a subscriber due to some JMS error.

`InvalidDestinationException` - if invalid Topic specified.

createDurableSubscriber(Topic, String, String, boolean)

```
public synchronized javax.jms.TopicSubscriber
createDurableSubscriber(javax.jms.Topic topic, java.lang.String subs_name,
java.lang.String messageSelector, boolean noLocal)
```

Create a durable Subscriber to the specified topic. This method can be used to create subscribers for queues that contain payloads of type `AQ$JMS_TEXT_MESSAGE`, `AQ$JMS_STREAM_MESSAGE`, `AQ$JMS_BYTES_MESSAGE`, `AQ$JMS_MAP_MESSAGE` or `AQ$JMS_OBJECT_MESSAGE`. A client can change an existing durable subscription by creating a durable `TopicSubscriber` with the same name and message selector.

Specified By

`javax.jms.TopicSession.createDurableSubscriber(javax.jms.Topic, java.lang.String, java.lang.String, boolean)` in interface `javax.jms.TopicSession`

Parameters

`topic` - the topic to subscribe to

`name` - the name used to identify this subscription.

`messageSelector` - only messages with properties matching the message selector expression are delivered. This value may be null.

The selector can contain any SQL92 expression which has a combination of one or more of the following:

- a. JMS Message header fields or properties: `JMSPriority` (int), `JMSCorrelationID` (string), `JMSType` (string), `JMSXUserID` (string), `JMSXAppID` (string), `JMSXGroupID` (string) `JMSXGroupSeq` (int)

Example: `JMSPriority < 3 AND JMSCorrelationID = 'Fiction'`

- b. User defined message properties

Example: `color IN ('RED', 'BLUE', 'GREEN') AND price < 30000`

Operators allowed are:

- logical operators in precedence order NOT, AND, OR
- comparison operators =, >, >=, <, <=, <>, ! (both <> and ! can be used for not equal)
- arithmetic operators in precedence order +, - unary, *, /, +, -
- identifier [NOT] IN (string-literal1, string-literal2, ..)
- arithmetic-expr1 [NOT] BETWEEN arithmetic-expr2 and arithmetic-expr3
- identifier [NOT] LIKE pattern-value [ESCAPE escape-character]
pattern-value is a string literal where % refers to any sequence of characters and _ refers to any single character. The optional escape-character is used to escape the special meaning of the '-' and '%' in pattern-value
- identifier IS [NOT] NULL

noLocal -- must be set to false. noLocal=true not supported.

Throws

`JMSEException` - if a session fails to create a subscriber due to some JMS error.

`InvalidDestinationException` - if invalid Topic specified.

`InvalidSelectorException` - if the message selector is invalid.

createDurableSubscriber(Topic, String, String, boolean, String)

```
public synchronized javax.jms.TopicSubscriber  
createDurableSubscriber(javax.jms.Topic topic, java.lang.String subs_name,  
java.lang.String messageSelector, boolean noLocal, String transformation)  
Create a durable Subscriber to the specified topic and specify a transformation for  
the subscriber
```

This method can be used to create subscribers for topics that contain payloads of type `AQ$JMS_TEXT_MESSAGE`, `AQ$JMS_STREAM_MESSAGE`, `AQ$JMS_BYTES_MESSAGE`, `AQ$JMS_MAP_MESSAGE` or `AQ$JMS_OBJECT_MESSAGE`. A client can change an existing durable subscription by creating a durable `TopicSubscriber` with the same name and message selector.

Specified By

`javax.jms.TopicSession.createDurableSubscriber(javax.jms.Topic, java.lang.String, java.lang.String, boolean)` in interface `javax.jms.TopicSession`

Parameters

`topic` - the topic to subscribe to

`name` - the name used to identify this subscription.

`messageSelector` - only messages with properties matching the message selector expression are delivered. This value may be null.

The selector can contain any SQL92 expression which has a combination of one or more of the following:

- a. JMS Message header fields or properties: JMSPriority (int), JMSCorrelationID (string), JMSType (string), JMSXUserID (string), JMSXAppID (string), JMSXGroupID (string) JMSXGroupSeq (int)

Example: `JMSPriority < 3 AND JMSCorrelationID = 'Fiction'`

- b. User defined message properties

Example: `color IN ('RED', 'BLUE', 'GREEN') AND price < 30000`

Operators allowed are:

- logical operators in precedence order NOT, AND, OR
- comparison operators =, >, >=, <, <=, <>, ! (both <> and ! can be used for not equal)
- arithmetic operators in precedence order +, -, unary *, /, +, -
- identifier [NOT] IN (string-literal1, string-literal2, ..)
- arithmetic-expr1 [NOT] BETWEEN arithmetic-expr2 and arithmetic-expr3
- identifier [NOT] LIKE pattern-value [ESCAPE escape-character]
pattern-value is a string literal where % refers to any sequence of characters and _ refers to any single character. The optional escape-character is used to escape the special meaning of the '-' and '%' in pattern-value
- identifier IS [NOT] NULL

`noLocal` -- must be set to false. `noLocal=true` not supported.

`transformation` -- transformation associated with this subscriber. This transformation is applied before messages are delivered to this subscriber

Throws

`JMSException` - if a session fails to create a subscriber due to some JMS error.

`InvalidDestinationException` - if invalid Topic specified.

`InvalidSelectorException` - if the message selector is invalid.

createDurableSubscriber(Topic, String, String, boolean, CustomDatumFactory, String)

```
public synchronized javax.jms.TopicSubscriber
createDurableSubscriber(javax.jms.Topic topic, java.lang.String subs_name,
java.lang.String messageSelector, boolean noLocal, oracle.sql.CustomDatumFactory
payload_factory, String transformation)
```

Create a durable Subscriber to the specified topic and specify a transformation for the subscriber.

This method is used to create subscribers for topics that contain Oracle ADT payloads (instead of the standard JMS defined payloads) A client can change an existing durable subscription by creating a durable `TopicSubscriber` with the same name and message selector.

Parameters

`topic` - the topic to subscribe to

`name` - the name used to identify this subscription.

`messageSelector` - only messages with attributes matching the message selector expression are delivered. This value may be null.

The syntax for the selector for queues containing ADT messages is different from the syntax for selectors on queues containing standard JMS payloads (text, stream, object, bytes, map) The selector is similar to the AQ rules syntax

a. Selector on priority or correlation is specified as follows. Example:

```
priority < 3 AND corrid = 'Fiction'
```

b. Selector on message payload is specified as follows. The attribute name must be prefixed with `tab.user_data`. Example: `tab.user_data.color =`

```
'GREEN' AND tab.user_data.price < 30000
```

`noLocal` - must be set to false. `noLocal=true` not supported

`payload_factory` - `CustomDatumFactory` for the java class that maps to the Oracle ADT messages do not contain any user defined properties.

`transformation` - transformation associated with this subscriber. This transformation is applied before messages are delivered to this subscriber

Throws

`JMSException` - if a session fails to create a subscriber due to some JMS error.

`InvalidDestinationException` - if invalid Topic specified.

createMessage()

```
public synchronized javax.jms.Message createMessage()
```

Create a generic header only Message.

Specified By

`javax.jms.Session.createMessage()` in interface `javax.jms.Session`

Throws

`JMSEException` - if some error occurs during message creation.

createMapMessage()

```
public synchronized javax.jms.MapMessage createMapMessage()
```

Create a `MapMessage`. A `MapMessage` is used to send a self-defining set of name-value pairs where names are `Strings` and values are Java primitive types.

Specified By

`javax.jms.Session.createMapMessage()` in interface `javax.jms.Session`

Throws

`JMSEException` - if some error occurs during message creation

createObjectMessage()

```
public synchronized javax.jms.ObjectMessage createObjectMessage()
```

Create an `ObjectMessage`. An `ObjectMessage` is used to send a message that containing a serializable Java object.

Specified By

`javax.jms.Session.createObjectMessage()` in interface `javax.jms.Session`

Throws

`JMSEException` - if some error occurs during message creation

createObjectMessage(Serializable)

```
public synchronized javax.jms.ObjectMessage  
createObjectMessage(java.io.Serializable object)
```

Create an initialized `ObjectMessage`. An `ObjectMessage` is used to send a message that containing a serializable Java object.

Specified By

`javax.jms.Session.createObjectMessage(java.io.Serializable)` in interface `javax.jms.Session`

Parameters

`object` - the object to use to initialize this message.

Throws

`JMSEException` - if some error occurs during message creation

createPublisher(Topic)

```
public synchronized javax.jms.TopicPublisher createPublisher(javax.jms.Topic topic)
```

Create a `Publisher` for the specified topic. A client uses a `TopicPublisher` for publishing messages on a topic.

Specified By

`javax.jms.TopicSession.createPublisher(javax.jms.Topic)` in interface `javax.jms.TopicSession`

Parameters

`topic` - the topic to publish to, or null if this is an unidentified producer.

Throws

`JMSEException` - if a session fails to create a publisher due to some JMS error.

`InvalidDestinationException` - if invalid Topic specified.

createQueue()

```
public synchronized javax.jms.Queue createQueue(String)
```

Create a queue given a `Queue` name. This facility is provided for the rare cases where clients need to dynamically manipulate queue identity. It allows the creation of a queue identity with a provider-specific name. Clients that depend on this ability are not portable. Note that this method is not for creating the physical queue. The physical creation of queues is an administrative task.

The Queue name is of the form "[schema].name". If "schema" is not specified, current session "user", is utilised.

Specified By

`javax.jms.QueueSession.createQueue()` in interface `javax.jms.QueueSession`.

Throws

`JMSEException` - if the queue could not be created

createQueue(AQQueueTable, String, AQjmsDestinationProperty)

```
public synchronized javax.jms.Queue createQueue(oracle.jms.AQQueueTable q_table,  
java.lang.String queue_name, AQjmsDestinationProperty dest_property)  
Create a queue.
```

Parameters

`q_table` - Queue-Table in which the queue is to be created. The queue-table must not be multiconsumer enabled

`queue_name` - name of the queue to be created

`dest_property` - Queue properties.

Throws

`JMSEException` - if the queue could not be created

See Also

[AQjmsDestinationProperty](#)

createQueueTable(String, String, AQQueueTableProperty)

```
public synchronized oracle.jms.AQQueueTable createQueueTable(java.lang.String  
owner, java.lang.String name, oracle.jms.AQQueueTableProperty property)  
Create a Queue Table. A QueueTable holds both queues or topics
```

Parameters

`owner` - the queue table owner (schema)

`name` - queue table name

`property` - queue table properties. If the queue table will be used to hold queues, then the queue table must not be multiconsumer enabled (default). If the queue table will be used to hold topics the queue table must be multiconsumer enabled

Throws

`JMSEException` - if the `QueueTable` cannot be created

See Also

`oracle.AQ.AQQueueTableProperty`

createReceiver(Queue)

```
public synchronized javax.jms.QueueReceiver createReceiver(javax.jms.Queue queue)
```

Create a `QueueReceiver` to receive messages from the specified queue. This method can be used to create receivers for queues that contain payloads of type `AQ$_JMS_TEXT_MESSAGE`, `AQ$_JMS_STREAM_MESSAGE`, `AQ$_JMS_BYTES_MESSAGE`, `AQ$_JMS_MAP_MESSAGE` or `AQ$_JMS_OBJECT_MESSAGE`

Specified By

`javax.jms.QueueSession.createReceiver(javax.jms.Queue)` in interface `javax.jms.QueueSession`

Parameters

`queue` - the queue to access

Throws

`JMSEException` - if a session fails to create a receiver due to some JMS error.

`InvalidDestinationException` - if invalid `Queue` specified.

createReceiver(Queue, CustomDatumFactory)

```
public synchronized javax.jms.QueueReceiver createReceiver(javax.jms.Queue queue, oracle.sql.CustomDatumFactory payload_factory)
```

Create a `QueueReceiver` to receive messages from the specified queue containing ADT messages. This method is used to create receivers for queues that contain Oracle ADT payloads (instead of the standard JMS defined payloads)

Parameters

`queue` - the queue to access

`payload_factory` - CustomDatumFactory for the java class that maps to the Oracle ADT

Throws

`JMSException` - if a session fails to create a receiver due to some JMS error.

`InvalidDestinationException` - if invalid Queue specified.

createReceiver(Queue, String)

```
public synchronized javax.jms.QueueReceiver createReceiver(javax.jms.Queue
queue, java.lang.String messageSelector)
```

Create a `QueueReceiver` to receive messages from the specified queue. This method can be used to create receivers for queues that contain payloads of type `AQ$_JMS_TEXT_MESSAGE`, `AQ$_JMS_STREAM_MESSAGE`, `AQ$_JMS_BYTES_MESSAGE`, `AQ$_JMS_MAP_MESSAGE` or `AQ$_JMS_OBJECT_MESSAGE`

Specified By

`javax.jms.QueueSession.createReceiver(javax.jms.Queue, java.lang.String)` in interface `javax.jms.QueueSession`

Parameters

`queue` - the queue to access

`messageSelector` - only messages with properties matching the message selector expression are delivered. The selector can be any expression that has a combination of one or more of the following:

- `JMSMessageID = 'ID:23452345'` to retrieve messages that have a specified message ID
- JMS Message header fields or properties:
 - `JMSPriority < 3 AND JMSCorrelationID = 'Fiction'`
- User defined message properties:
 - `color IN ('RED', 'BLUE', 'GREEN') AND price < 30000`

All message IDs must be prefixed with "ID:"

Throws

`JMSEException` - if a session fails to create a receiver due to some JMS error.

`InvalidDestinationException` - if invalid Queue specified.

`InvalidSelectorException` - if the message selector is invalid.

createReceiver(Queue, String, CustomDatumFactory)

```
public synchronized javax.jms.QueueReceiver createReceiver(javax.jms.Queue
queue, java.lang.String messageSelector, oracle.sql.CustomDatumFactory payload_
factory)
```

Create a `QueueReceiver` to receive messages from the specified queue containing ADT messages. This method is used to create receivers for queues that contain Oracle ADT payloads (instead of the standard JMS defined payloads)

Parameters

`queue` - the queue to access

`messageSelector` - only messages with properties matching the message selector expression are delivered. For queues containing `AdtMessages` the selector for `QueueReceiver` can be a SQL expression on the message payload contents or `messageID` or `priority` or `correlationID`.

- Selector on message id - to retrieve messages that have a specific `messageID`

```
msgid = '23434556566767676'
```

Note: in this case message IDs must NOT be prefixed with 'ID:'

- Selector on priority or correlation is specified as follows

```
priority < 3 AND corrid = 'Fiction'
```

- Selector on message payload is specified as follows

```
tab.user_data.color = 'GREEN' AND tab.user_data.price < 30000
```

`payload_factory` - `CustomDatumFactory` for the java class that maps to the Oracle ADT

Throws

`JMSEException` - if a session fails to create a receiver due to some JMS error.

`InvalidDestinationException` - if invalid Queue specified.

`InvalidSelectorException` - if the message selector is invalid.

createRemoteSubscriber(Topic, AQjmsAgent, String)

```
public synchronized void createRemoteSubscriber(javax.jms.Topic topic,
AQjmsAgent remote_subscriber, java.lang.String messageSelector)
```

Create a remote subscriber for a topic. This method can be used to remote subscribers for queues that contain payloads of type `AQ$JMS_TEXT_MESSAGE`, `AQ$JMS_STREAM_MESSAGE`, `AQ$JMS_BYTES_MESSAGE`, `AQ$JMS_MAP_MESSAGE` or `AQ$JMS_OBJECT_MESSAGE`.

AQ allows topics to have remote subscribers, ie subscribers at other topics in the same or different database. In order to use remote subscribers, you must set up propagation between the two local and remote topic.

Parameters

`topic` - the topic to subscribe to

`remote_subscriber` - `AQjmsAgent` that refers to the remote subscriber

`messageSelector` - only messages with properties matching the message selector expression are delivered. This value may be null.

The selector syntax is the same as that for `createDurableSubscriber`. Remote subscribers may be a specific consumer at the remote topic or all subscribers at the remote topic.

A remote subscriber is defined using the `AQjmsAgent` structure. An `AQjmsAgent` consists of a name and address. The name refers to the `consumer_name` at the remote topic. The address refers to the remote topic - the syntax is `(schema).(topic_name)[@dblink]`.

1. To publish messages to a particular consumer at the remote topic, the `subscription_name` of the recipient at the remote topic must be specified in the name field of `AQjmsAgent`. The remote topic must be specified in the address field of `AQjmsAgent`.
2. To publish messages to all subscribers of the remote topic, the name field of `AQjmsAgent` must be set to null. The remote topic must be specified in the address field of `AQjmsAgent`.

createRemoteSubscriber(Topic, AQjmsAgent, String, String)

```
public synchronized void createRemoteSubscriber(javax.jms.Topic topic,
AQjmsAgent remote_subscriber, java.lang.String messageSelector, java.lang.String
transformation)
```

Create a remote subscriber for a topic and specify a transformation for the subscriber. This method can be used to remote subscribers for queues that contain payloads of type `AQ$JMS_TEXT_MESSAGE`, `AQ$JMS_STREAM_MESSAGE`,

AQ\$_JMS_BYTES_MESSAGE, AQ\$_JMS_MAP_MESSAGE or AQ\$_JMS_OBJECT_MESSAGE.

AQ allows topics to have remote subscribers, ie subscribers at other topics in the same or different database. In order to use remote subscribers, you must set up propagation between the two local and remote topic.

Parameters

`topic` - the topic to subscribe to

`remote_subscriber` - `AQjmsAgent` that refers to the remote subscriber

`messageSelector` - only messages with properties matching the message selector expression are delivered. This value may be null.

The selector syntax is the same as that for `createDurableSubscriber`. Remote subscribers may be a specific consumer at the remote topic or all subscribers at the remote topic.

`transformation` - the transformation for this subscriber. This transformation will be applied before the message is delivered to this subscriber.

A remote subscriber is defined using the `AQjmsAgent` structure. An `AQjmsAgent` consists of a name and address. The name refers to the `consumer_name` at the remote topic. The address refers to the remote topic - the syntax is `(schema).(topic_name)[@dblink]`.

1. To publish messages to a particular consumer at the remote topic, the `subscription_name` of the recipient at the remote topic must be specified in the name field of `AQjmsAgent`. The remote topic must be specified in the address field of `AQjmsAgent`.
2. To publish messages to all subscribers of the remote topic, the name field of `AQjmsAgent` must be set to null. The remote topic must be specified in the address field of `AQjmsAgent`.

createRemoteSubscriber(Topic, AQjmsAgent, String, CustomDatumFactory)

```
public synchronized void createRemoteSubscriber(javax.jms.Topic topic,  
AQjmsAgent remote_subscriber, java.lang.String messageSelector,  
oracle.sql.CustomDatumFactory payload_factory)
```

Create a remote subscriber for a topic. This method is used to create browsers for queues that contain Oracle ADT payloads (instead of the standard JMS defined payloads).

AQ allows topics to have remote subscribers, ie subscribers at other topics in the same or different database. In order to use remote subscribers, you must set up propagation between the two local and remote topic.

Parameters

`topic` - the topic to subscribe to

`remote_subscriber` - AQjmsAgent that refers to the remote subscriber

`messageSelector` - only messages with properties matching the message selector expression are delivered. This value may be null. The selector syntax is the same as that for `createDurableSubscriber` for topics with ADT messages

`payload_factory` - CustomDatumFactory for the java class that maps to the Oracle ADT.

Remote subscribers may be a specific consumer at the remote topic or all subscribers at the remote topic. A remote subscriber is defined using the AQjmsAgent structure. An AQjmsAgent consists of a name and address. The name refers to the consumer_name at the remote topic. The address refers to the remote topic - the syntax is (schema).(topic_name)[@dblink].

1. To publish messages to a particular consumer at the remote topic, the subscription_name of the recipient at the remote topic must be specified in the name field of AQjmsAgent. The remote topic must be specified in the address field of AQjmsAgent.

2. To publish messages to all subscribers of the remote topic, the name field of AQjmsAgent must be set to null. The remote topic must be specified in the address field of AQjmsAgent

createRemoteSubscriber(Topic, AQjmsAgent, String, CustomDatumFactory, String)

```
public synchronized void createRemoteSubscriber(javax.jms.Topic topic,
AQjmsAgent remote_subscriber, java.lang.String messageSelector,
oracle.sql.CustomDatumFactory payload_factory, String transformation)
```

Create a remote subscriber for a topic and specify a transformation for this subscriber. This method is used to create browsers for queues that contain Oracle ADT payloads (instead of the standard JMS defined payloads).

AQ allows topics to have remote subscribers, ie subscribers at other topics in the same or different database. In order to use remote subscribers, you must set up propagation between the two local and remote topic.

Parameters

`topic` - the topic to subscribe to

`remote_subscriber` - AQjmsAgent that refers to the remote subscriber

`messageSelector` - only messages with properties matching the message selector expression are delivered. This value may be null. The selector syntax is the same as that for `createDurableSubscriber` for topics with ADT messages

`payload_factory` - `CustomDatumFactory` for the java class that maps to the Oracle ADT.

`transformation` - the transformation for this subscriber. This transformation will be applied before the message is delivered to this subscriber.

Remote subscribers may be a specific consumer at the remote topic or all subscribers at the remote topic. A remote subscriber is defined using the `AQjmsAgent` structure. An `AQjmsAgent` consists of a name and address. The name refers to the `consumer_name` at the remote topic. The address refers to the remote topic - the syntax is `(schema).(topic_name)[@dblink]`.

1. To publish messages to a particular consumer at the remote topic, the `subscription_name` of the recipient at the remote topic must be specified in the name field of `AQjmsAgent`. The remote topic must be specified in the address field of `AQjmsAgent`.

2. To publish messages to all subscribers of the remote topic, the name field of `AQjmsAgent` must be set to null. The remote topic must be specified in the address field of `AQjmsAgent`

createSender(Queue)

```
public synchronized javax.jms.QueueSender createSender(javax.jms.Queue queue)
```

Create a `QueueSender` to send messages to the specified queue.

Specified By

`javax.jms.QueueSession.createSender(javax.jms.Queue)` in interface `javax.jms.QueueSession`

Parameters

`queue` - the queue to access, or null if this is an unidentified producer.

Throws

`JMSException` - if a session fails to create a sender due to some JMS error.

`InvalidDestinationException` - if invalid Queue specified.

createStreamMessage()

```
public synchronized javax.jms.StreamMessage createStreamMessage()
```

Create a `StreamMessage`. A `StreamMessage` is used to send a self-defining stream of Java primitives.

Specified By

`javax.jms.Session.createStreamMessage()` in interface `javax.jms.Session`

Throws

`JMSEException` - if some error occurs during message creation

createSubscriber(Topic)

```
public synchronized javax.jms.TopicSubscriber createSubscriber(javax.jms.Topic topic)
```

Create a non-durable Subscriber to the specified topic.

Specified By

`javax.jms.TopicSession.createSubscriber(javax.jms.Topic)` in interface `javax.jms.TopicSession`

Throws

`JMSEException` - if some error occurs during non durable subscriber creation.

createSubscriber(Topic, String, boolean)

```
public synchronized javax.jms.TopicSubscriber createSubscriber(javax.jms.Topic topic, java.lang.String messageSelector, boolean noLocal)
```

Create a non-durable Subscriber to the specified topic.

Specified By

`javax.jms.TopicSession.createSubscriber(javax.jms.Topic, java.lang.String, boolean)` in interface `javax.jms.TopicSession`

Throws

`JMSEException` - if some error occurs during non durable subscriber creation.

createTemporaryQueue()

```
public synchronized javax.jms.TemporaryQueue createTemporaryQueue()
```

Create a temporary queue. Its lifetime will be that of the `QueueConnection` unless it is deleted earlier. Specified By `javax.jms.Session.createTemporaryQueue()` in interface `javax.jms.Session`.

Throws

`JMSEException` - if the temporary queue could not be created.

createTemporaryTopic()

```
public synchronized javax.jms.Topic createTemporaryTopic()
```

Create a temporary topic. Its lifetime will be that of the `TopicConnection` unless it is deleted earlier.

Specified By

`javax.jms.Session.createTemporaryTopic()` in interface `javax.jms.Session`.

Throws

`JMSEException` - if the temporary topic could not be created.

createTextMessage()

```
public synchronized javax.jms.TextMessage createTextMessage()
```

Create a `TextMessage`. A `TextMessage` is used to send a message containing a `StringBuffer`.

Specified By

`javax.jms.Session.createTextMessage()` in interface `javax.jms.Session`

Throws

`JMSEException` - if some error occurs during message creation

createTextMessage(StringBuffer)

```
public synchronized javax.jms.TextMessage  
createTextMessage(java.lang.StringBuffer stringBuffer)
```

Create an initialized `TextMessage`. A `TextMessage` is used to send a message containing a `StringBuffer`.

Specified By

`javax.jms.Session.createTextMessage(java.lang.StringBuffer)` in interface `javax.jms.Session`

Parameters

`stringBuffer` - the string buffer used to initialize this message.

Throws

`JMSEException` - if some error occurs during message creation

createTopic()

```
public synchronized javax.jms.Topic createTopic(String)
```

Create a Topic given a Topic name. This facility is provided for the rare cases where clients need to dynamically manipulate topic identity. It allows the creation of a topic identity with a provider-specific name. Clients that depend on this ability are not portable. Note that this method is not for creating the physical topic. The physical creation of topic is an administrative task.

The Topic name is of the form "[schema].name." If "schema" is not specified, current session "user," is utilized.

Specified By

`javax.jms.QueueSession.createQueue()` in interface `javax.jms.QueueSession`.

Throws

`JMSEException` - if the queue could not be created.

createTopic(AQQueueTable, String, AQjmsDestinationProperty)

```
public synchronized javax.jms.Topic createTopic(oracle.jms.AQQueueTable q_table,  
java.lang.String topic_name, AQjmsDestinationProperty dest_property)
```

Create a topic

Parameters

`q_table` - Queue-Table in which the topic is to be created. The queue-table must be multiconsumer enabled

`topic_name` - name of the topic to be created

`dest_property` - Topic properties.

Throws

`JMSEException` - if the topic could not be created

See Also

[AQjmsDestinationProperty](#)

createTopicReceiver(Topic, String, String)

```
public synchronized TopicReceiver createTopicReceiver(javax.jms.Topic topic,  
java.lang.String receiver_name, java.lang.String messageSelector)
```

Create a `TopicReceiver` to receive messages from the specified topic. AQ allows messages to be sent to all subscribers of a topic or to specified recipients. These receivers may or may not be subscribers of the topic. If the receiver is not a subscriber to the topic, it will receive only those messages that are explicitly This method must be used order to create a `TopicReceiver` object for consumers that are not durable subscribers of the topic

This method can be used to create `TopicReceivers` for topics that contain payloads of type `AQ$JMS_TEXT_MESSAGE`, `AQ$JMS_STREAM_MESSAGE`, `AQ$JMS_BYTES_MESSAGE`, `AQ$JMS_MAP_MESSAGE` or `AQ$JMS_OBJECT_MESSAGE`

Parameters

`topic` - the topic to access

`receiver_name` - the name of the recipient (or subscriber)

`messageSelector` - only messages with properties matching the message selector expression are delivered. The selector can be any expression that has a combination of one or more of the following:

- `JMSMessageID = 'ID:23452345'` to retrieve messages that have a specified message ID
- JMS Message header fields or properties:
`JMSPriority < 3 AND JMSCorrelationID = 'Fiction'`
- User defined message properties:
`color IN ('RED', 'BLUE', 'GREEN') AND price < 30000`

All message IDs must be prefixed with "ID:"

Throws

`JMSEException` - if a session fails to create a receiver due to some JMS error.

`InvalidDestinationException` - if invalid Topic specified.

`InvalidSelectorException` - if the message selector is invalid.

createTopicReceiver(Topic, String, String, CustomDatumFactory)

```
public synchronized TopicReceiver createTopicReceiver(javax.jms.Topic topic,  
java.lang.String receiver_name, java.lang.String messageSelector,  
oracle.sql.CustomDatumFactory payload_factory)
```

Create a `TopicReceiver` to receive messages from the specified topic containing ADT messages. AQ allows messages to be sent to all subscribers of a topic or to specified recipients. These receivers may or may not be subscribers of the topic. If the receiver is not a subscriber to the topic, it will receive only those messages that are explicitly This method must be used order to create a `TopicReceiver` object for consumers that are not durable subscribers of the topic

This method is used to create `TopicReceivers` for topics that contain Oracle ADT payloads (instead of the standard JMS defined payloads)

Parameters

`topic` - the topic to access

`receiver_name` - the name of the recipient (or subscriber)

`messageSelector` - only messages with properties matching the message selector expression are delivered. For queues containing `AdtMessages` the selector can be a SQL expression on the message payload contents or `messageID` or priority or `correlationID`.

- Selector on message id - to retrieve messages that have a specific `messageID`

```
msgid = '23434556566767676'
```

Note: in this case message IDs must NOT be prefixed with 'ID:'

- Selector on priority or correlation is specified as follows

```
priority < 3 AND corrid = 'Fiction'
```

- Selector on message payload is specified as follows

```
tab.user_data.color = 'GREEN' AND tab.user_data.price < 30000
```

`payload_factory` - `CustomDatumFactory` for the java class that maps to the Oracle ADT

createQueue()

getDBConnection()

```
public synchronized java.sql.Connection getDBConnection()
```

getJmsConnection()

```
public AQJmsConnection getJmsConnection()
```

getMessageListener()

```
public synchronized javax.jms.MessageListener getMessageListener()  
Return the session's distinguished message listener.
```

Specified By

javax.jms.Session.getMessageListener() in interface javax.jms.Session

Returns

the message listener associated with this session.

Throws

JMSEException - if JMS fails to get the message listener due to an internal error in JMS Provider.

getQueue(String, String)

```
public synchronized javax.jms.Queue getQueue(java.lang.String owner,  
java.lang.String name)  
Get an existing queue. The Queue is returned only if the user has created the queue  
or as enqueue/dequeue privileges on the specified queue
```

Parameters

owner - queue owner (schema)

name - queue name

Throws

JMSEException - if the queue could not be returned due to some error

getQueueTable(String, String)

```
public synchronized oracle.jms.AQQueueTable getQueueTable(java.lang.String
```

owner, java.lang.String name)

Get a handle to an existing queue-table. If owner of queue-table is not the same as the user which opened the connection, the caller must have AQ enqueue/dequeue privileges on queues/topics in the queue table. Otherwise the queue-table will not be returned.

Parameters

owner - the owner (schema) of the queue-table

name - queue-table name

Throws

JMSEException - if the queue table does not exist or if the user does not have privileges on any queue/topic in the queue-table

getTopic(String, String)

```
public synchronized javax.jms.Topic getTopic(java.lang.String owner,  
java.lang.String name)
```

Get an existing topic. The Topic is returned only if the user has created the topic or as enqueue/dequeue privileges on the specified topic.

Parameters

owner - topic owner (schema)

name - topic name

Throws

JMSEException - if the topic could not be returned due to some error

getTransacted()

```
public synchronized boolean getTransacted()
```

Checks if the session is in transacted mode?

Specified By

javax.jms.Session.getTransacted() in interface javax.jms.Session

Returns

true if in transacted mode

Throws

`JMSEException` - if session is closed

grantSystemPrivilege(String, String, boolean)

```
public void grantSystemPrivilege(java.lang.String privilege, java.lang.String grantee, boolean admin_option)
```

Grant AQ system privileges to users/roles. Initially only SYS and SYSTEM can use this procedure successfully

Parameters

`privilege` - options are `ENQUEUE_ANY`, `DEQUEUE_ANY` and `MANAGE_ANY`
`ENQUEUE_ANY` - users with this privilege are allowed to enqueue messages to any queue/topic in the database.

`DEQUEUE_ANY` - users with this privilege are allowed to dequeue messages from any queue/topic in the database.

`MANAGE_ANY` - users with this privilege are allowed to access and make admin calls on any queue/topic in the database.

`grantee` - specifies the grantee. The grantee can be a user, role or the `PUBLIC` role

`admin_option` - if this is set to true, the grantee is allowed to use this procedure to grant the system privilege to other users or roles

Throws

`JMSEException` - if the system privilege could not be granted.

recover()

```
public synchronized void recover()
```

Stops message delivery in this session, and restarts message delivery with the oldest unacknowledged message.

All consumers deliver messages in a serial order. Acknowledging a received message automatically acknowledges all messages that have been delivered to the client.

Restarting a session causes it to take the following actions:

- Stop message delivery.
- Mark all messages that might have been delivered but not acknowledged as "redelivered."

- Restart the delivery sequence including all unacknowledged messages that had been previously delivered. Redelivered messages need not be delivered in their exact original delivery order.

Specified By

javax.jms.Session.recover() in interface javax.jms.Session

Throws

JMSEException - if JMS implementation fails to stop and restart message delivery due to some internal error.

IllegalStateException-if the method is called by a transacted session.

revokeSystemPrivilege(String, String)

```
public void revokeSystemPrivilege(java.lang.String privilege, java.lang.String grantee)
```

Revoke AQ system privilege from user/roles

Parameters

privilege - options are ENQUEUE_ANY, DEQUEUE_ANY and MANAGE_ANY

grantee - specifies the grantee. The grantee can be a user, role or the PUBLIC role

Throws

JMSEException - if the system privilege could not be revoked

rollback()

```
public synchronized void rollback()
```

Rollback any messages done in this transaction and releases any locks currently held.

Specified By

javax.jms.Session.rollback() in interface javax.jms.Session

Throws

JMSEException - if JMS implementation fails to rollback the the transaction due to some internal error.

run()

```
public void run()
```

Specified By

java.lang.Runnable.run() in interface java.lang.Runnable

setMessageListener(MessageListener)

```
public synchronized void setMessageListener(javax.jms.MessageListener listener)
```

Set the session's distinguished message listener. When it is set no other form of message receipt in the session can be used; however, all forms of sending messages are still supported.

Specified By

javax.jms.Session.setMessageListener(javax.jms.MessageListener) in interface javax.jms.Session

Parameters

listener - the message listener to associate with this session.

Throws

JMSEException - if JMS fails to set the message listener due to an internal error in JMS Provider.

unsubscribe(String)

```
public synchronized void unsubscribe(String subs_name)
```

Unsubscribe a durable subscription with a given subscriber name that has been created by a client.

Parameters

subs_name - name of the durable subscriber that need to be unsubscribed.

Throws

JMSEException - if JMS fails to unsubscribe to durable subscription due to some JMS error.

unsubscribe(Topic, AQjmsAgent)

```
public synchronized void unsubscribe(javax.jms.Topic topic, AQjmsAgent remote_subscriber)
```

Unsubscribe a remote durable subscription that has been created by a client on the specified topic

Parameters

`topic` - the topic subscribed to

`remote_subscriber` - AQjmsAgent that refers to the remote subscriber. the address field of the AQjmsAgent cannot be null

Throws

`JMSException` - if JMS fails to unsubscribe to durable subscription due to some JMS error.

`InvalidDestinationException` - if invalid Topic specified.

unsubscribe(Topic, String)

```
public synchronized void unsubscribe(javax.jms.Topic topic, java.lang.String subs_name)
```

Unsubscribe a durable subscription that has been created by a client on the specified topic

Parameters

`topic` - the topic subscribed to

`subs_name` - the name used to identify this subscription.

Throws

`JMSException` - if JMS fails to unsubscribe to durable subscription due to some JMS error.

`InvalidDestinationException` - if invalid Topic specified.

AQjmsStreamMessage

Syntax

```
public class AQjmsStreamMessage extends AQjmsMessage
    implements javax.jms.StreamMessage
```

```
java.lang.Object
|
+--AQjmsMessage
    |
    +--oracle.jms.AQjmsStreamMessage
```

All Implemented Interfaces

javax.jms.Message, javax.jms.StreamMessage

Description

This class implements the StreamMessage interface. A StreamMessage is used to send a stream of java primitives

Member Summary	Description
Methods	-
<code>clearBody()</code>	Clear out the message body.
<code>clearProperties()</code>	-
<code>readBoolean()</code>	Reads a boolean from the stream message
<code>readByte()</code>	Reads a signed 8-bit value from the stream message
<code>readBytes(byte[])</code>	Read a byte array from the stream message
<code>readChar()</code>	Read a Unicode character value from the stream message
<code>readDouble()</code>	Read a double from the stream message
<code>readFloat()</code>	Read a float from the stream message
<code>readInt()</code>	Read a signed 32 bit integer from the stream message
<code>readLong()</code>	Read a signed 64 bit integer from the stream message
<code>readObject()</code>	Read a Java object from the stream message.

(Cont.) Member Summary	Description
<code>readShort()</code>	Reads a signed 16-bit value from the stream message
<code>readString()</code>	Read a string from the stream message
<code>reset()</code>	Put the message in read-only mode, and reposition the stream of bytes to the beginning.
<code>writeBoolean(boolean)</code>	Write a boolean to the stream message as a 1-byte value.
<code>writeByte(byte)</code>	Write out a byte to the stream message as a 1-byte value.
<code>writeBytes(byte[])</code>	Write a byte array to the stream message
<code>writeBytes(byte[], int, int)</code>	Write a portion of byte array to the stream message
<code>writeChar(char)</code>	Write a char to the stream as a 2-byte, high byte first
<code>writeDouble(double)</code>	Write a double to the stream as a 8-byte, high byte first
<code>writeFloat(float)</code>	Write a float to the stream as a 4-byte, high byte first
<code>writeInt(int)</code>	Write a int to the stream as a 4-byte, high byte first
<code>writeLong(long)</code>	Write a int to the stream as a 4-byte, high byte first
<code>writeObject(Object)</code>	Write a java object to the stream message
<code>writeShort(short)</code>	Write a short to the stream as a 2-byte, high byte first
<code>writeString(String)</code>	Writes a string to the underlying output stream

Inherited Member Summary

Fields inherited from interface `javax.jms.Message`

`DEFAULT_DELIVERY_MODE`, `DEFAULT_PRIORITY`, `DEFAULT_TIME_TO_LIVE`

Methods inherited from class `AQjmsMessage`

(Cont.) Inherited Member Summary

```

getBooleanProperty(String), getByteProperty(String),
getDoubleProperty(String), getFloatProperty(String),
getIntProperty(String), getJMSCorrelationID(),
getJMSCorrelationIDAsBytes(), getJMSDeliveryMode(),
getJMSDestination(), getJMSExpiration(), getJMSMessageID(),
getJMSMessageIDAsBytes(), getJMSPriority(), getJMSRedelivered(),
getJMSReplyTo(), getJMSTimestamp(), getJMSType(),
getLongProperty(String), getObjectProperty(String),
getPropertyNames(), getSenderID(), getShortProperty(String),
getStringProperty(String), propertyExists(String),
setBooleanProperty(String, boolean), setByteProperty(String, byte),
setDoubleProperty(String, double), setFloatProperty(String, float),
setIntProperty(String, int), setJMSCorrelationID(String),
setJMSCorrelationIDAsBytes(byte[]), setJMSDestination(Destination),
setJMSExpiration(long), setJMSMessageID(String),
setJMSPriority(int), setJMSRedelivered(boolean),
setJMSReplyTo(Destination), setJMSTimestamp(long),
setJMSType(String), setLongProperty(String, long),
setObjectProperty(String, Object), setSenderID(AQjmsAgent),
setShortProperty(String, short), setStringProperty(String, String)

```

Methods inherited from class java.lang.Object

```

clone, equals, finalize, getClass, hashCode, notify, notifyAll,
toString, wait, wait, wait

```

Methods inherited from interface javax.jms.Message

```

getBooleanProperty, getByteProperty, getDoubleProperty,
getFloatProperty, getIntProperty, getJMSCorrelationID,
getJMSCorrelationIDAsBytes, getJMSDeliveryMode, getJMSDestination,
getJMSExpiration, getJMSMessageID, getJMSPriority,
getJMSRedelivered, getJMSReplyTo, getJMSTimestamp, getJMSType,
getLongProperty, getObjectProperty, getPropertyNames,
getShortProperty, getStringProperty, propertyExists,
setBooleanProperty, setByteProperty, setDoubleProperty,
setFloatProperty, setIntProperty, setJMSCorrelationID,
setJMSCorrelationIDAsBytes, setJMSDeliveryMode, setJMSDestination,
setJMSExpiration, setJMSMessageID, setJMSPriority,
setJMSRedelivered, setJMSReplyTo, setJMSTimestamp, setJMSType,
setLongProperty, setObjectProperty, setShortProperty,
setStringProperty

```

Methods

clearBody()

```
public void clearBody()
```

Clear out the message body. All other parts of the message are left untouched.

Specified By

`javax.jms.Message.clearBody()` in interface `javax.jms.Message`

Overrides

`clearBody()` in class `AQjmsMessage`

Throws

`JMSEException` - if JMS fails to due to some internal JMS error.

clearProperties()

```
public void clearProperties()
```

Specified By

`javax.jms.Message.clearProperties()` in interface `javax.jms.Message`

Overrides

`clearProperties()` in class `AQjmsMessage`

readBoolean()

```
public boolean readBoolean()
```

Reads a boolean from the stream message

Specified By

`javax.jms.StreamMessage.readBoolean()` in interface `javax.jms.StreamMessage`

Throws

`MessageNotReadableException` - if message in write-only mode.

`JMSEException` - if JMS fails to read message due to some internal JMS error.

`MessageEOFException` - if end of message stream

readByte()

```
public byte readByte()  
Reads a signed 8-bit value from the stream message
```

Specified By

`javax.jms.StreamMessage.readByte()` in interface `javax.jms.StreamMessage`

Returns

the next byte from the stream message as a signed 8-bit byte

Throws

`MessageNotReadableException` - if message in write-only mode.
`JMSEException` - if JMS fails to read message due to some internal JMS error.
`MessageEOFException` - if end of message stream

readBytes(byte[])

```
public int readBytes(byte[] value)  
Read a byte array from the stream message
```

Specified By

`javax.jms.StreamMessage.readBytes(byte[])` in interface `javax.jms.StreamMessage`

Parameters

`value` - the buffer into which the data is read

Throws

`MessageNotReadableException` - if message in write-only mode.
`JMSEException` - if JMS fails to read message due to some internal JMS error.
`MessageEOFException` - if end of message stream

readChar()

```
public char readChar()  
Read a Unicode character value from the stream message
```

Specified By

`javax.jms.StreamMessage.readChar()` in interface `javax.jms.StreamMessage`

Returns

the next two bytes from the stream message as a Unicode character.

Throws

`MessageNotReadableException` - if message in write-only mode.

`JMSException` - if JMS fails to read message due to some internal JMS error.

`MessageEOFException` - if end of message stream

readDouble()

```
public double readDouble()
```

Read a double from the stream message

Specified By

`javax.jms.StreamMessage.readDouble()` in interface `javax.jms.StreamMessage`

Returns

the next eight bytes from the stream message, interpreted as a double.

Throws

`MessageNotReadableException` - if message in write-only mode.

`JMSException` - if JMS fails to read message due to some internal JMS error.

`MessageEOFException` - if end of message stream

readFloat()

```
public float readFloat()
```

Read a float from the stream message

Specified By

`javax.jms.StreamMessage.readFloat()` in interface `javax.jms.StreamMessage`

Returns

the next four bytes from the stream message, interpreted as a float.

Throws

`MessageNotReadableException` - if message in write-only mode.

`JMSEException` - if JMS fails to read message due to some internal JMS error.

`MessageEOFException` - if end of message stream

readInt()

```
public int readInt()
```

Read a signed 32 bit integer from the stream message

Specified By

`javax.jms.StreamMessage.readInt()` in interface `javax.jms.StreamMessage`

Returns

the next four bytes from the stream message, interpreted as a `int`.

Throws

`MessageNotReadableException` - if message in write-only mode.

`JMSEException` - if JMS fails to read message due to some internal JMS error.

`MessageEOFException` - if end of message stream

readLong()

```
public long readLong()
```

Read a signed 64 bit integer from the stream message

Specified By

`javax.jms.StreamMessage.readLong()` in interface `javax.jms.StreamMessage`

Returns

the next eight bytes from the stream message, interpreted as a `long`.

Throws

`MessageNotReadableException` - if message in write-only mode.

`JMSEException` - if JMS fails to read message due to some internal JMS error.

`MessageEOFException` - if end of message stream

readObject()

```
public java.lang.Object readObject()
```

Read a Java object from the stream message.

Note that this method can be used to return in objectified format, an object that had been written to the Stream with the equivalent `writeObject` method call, or it's equivalent primitive write method.

Specified By

`javax.jms.StreamMessage.readObject()` in interface `javax.jms.StreamMessage`

Returns

a Java object from the stream message, in objectified format (ie. if it set as an int, then a Integer is returned).

Throws

`JMSEException` - if JMS fails to read message due to some internal JMS error.

`MessageEOFException` - if an end of message stream

`MessageNotReadableException` - if message in write-only mode.

readShort()

```
public short readShort()
```

Reads a signed 16-bit value from the stream message

Specified By

`javax.jms.StreamMessage.readShort()` in interface `javax.jms.StreamMessage`

Returns

the next two bytes from the stream message, interpreted as a 16-bit number

Throws

`MessageNotReadableException` - if message in write-only mode.

`JMSEException` - if JMS fails to read message due to some internal JMS error.

`MessageEOFException` - if end of message stream

readString()

```
public java.lang.String readString()  
Read a string from the stream message
```

Specified By

`javax.jms.StreamMessage.readString()` in interface `javax.jms.StreamMessage`

Returns

string from the stream message

Throws

`MessageNotReadableException` - if message in write-only mode.

`JMSException` - if JMS fails to read message due to some internal JMS error.

`MessageEOFException` - if end of message stream

reset()

```
public void reset()  
Put the message in read-only mode, and reposition the stream of bytes to the  
beginning.
```

Specified By

`javax.jms.StreamMessage.reset()` in interface `javax.jms.StreamMessage`

Throws

`MessageNotWriteableException` - if message in write-only mode.

`JMSException` - if JMS fails to read message due to some internal JMS error.

writeBoolean(boolean)

```
public void writeBoolean(boolean value)  
Write a boolean to the stream message as a 1-byte value. The value true is written  
out as the value (byte)1; the value false is written out as the value (byte)0.
```

Specified By

`javax.jms.StreamMessage.writeBoolean(boolean)` in interface
`javax.jms.StreamMessage`

Parameters

value - the boolean value to be written

Throws

`MessageNotWritableException` - if message in write-only mode.

`JMSEException` - if JMS fails to read message due to some internal JMS error.

writeByte(byte)

```
public void writeByte(byte value)
```

Write out a byte to the stream message as a 1-byte value.

Specified By

`javax.jms.StreamMessage.writeByte(byte)` in interface `javax.jms.StreamMessage`

Parameters

value - the byte value to be written

Throws

`MessageNotWritableException` - if message in write-only mode.

`JMSEException` - if JMS fails to read message due to some internal JMS error.

writeBytes(byte[])

```
public void writeBytes(byte[] value)
```

Write a byte array to the stream message

Specified By

`javax.jms.StreamMessage.writeBytes(byte[])` in interface `javax.jms.StreamMessage`

Parameters

value - The byte array to be written

Throws

`MessageNotWritableException` - if message in write-only mode.

`JMSEException` - if JMS fails to read message due to some internal JMS error.

writeBytes(byte[], int, int)

```
public void writeBytes(byte[] value, int offset, int length)
Write a portion of byte array to the stream message
```

Specified By

`javax.jms.StreamMessage.writeBytes(byte[], int, int)` in interface `javax.jms.StreamMessage`

Parameters

`value` - the byte array to be written
`offset` - the initial offset within the byte array
`length` - the number of bytes to use

Throws

`MessageNotWritableException` - if message in write-only mode.
`JMSEException` - if JMS fails to read message due to some internal JMS error.

writeChar(char)

```
public void writeChar(char value)
Write a char to the stream as a 2-byte, high byte first
```

Specified By

`javax.jms.StreamMessage.writeChar(char)` in interface `javax.jms.StreamMessage`

Parameters

`value` - the char to be written

Throws

`MessageNotWritableException` - if message in write-only mode.
`JMSEException` - if JMS fails to read message due to some internal JMS error.

writeDouble(double)

```
public void writeDouble(double value)
Write a double to the stream as a 8-byte, high byte first
```

Specified By

`javax.jms.StreamMessage.writeDouble(double)` in interface `javax.jms.StreamMessage`

Parameters

`value` - The double to be written

Throws

`MessageNotWritableException` - if message in write-only mode.

`JMSEException` - if JMS fails to read message due to some internal JMS error.

writeFloat(float)

```
public void writeFloat(float value)
```

Write a float to the stream as a 4-byte, high byte first

Specified By

`javax.jms.StreamMessage.writeFloat(float)` in interface `javax.jms.StreamMessage`

Parameters

`value` - the float to be written

Throws

`MessageNotWritableException` - if message in write-only mode.

`JMSEException` - if JMS fails to read message due to some internal JMS error.

writeInt(int)

```
public void writeInt(int value)
```

Write a int to the stream as a 4-byte, high byte first

Specified By

`javax.jms.StreamMessage.writeInt(int)` in interface `javax.jms.StreamMessage`

Parameters

`value` - the int to be written

Throws

`MessageNotWritableException` - if message in write-only mode.

`JMSEException` - if JMS fails to read message due to some internal JMS error.

writeLong(long)

```
public void writeLong(long value)
```

Write a int to the stream as a 4-byte, high byte first

Specified By

`javax.jms.StreamMessage.writeLong(long)` in interface `javax.jms.StreamMessage`

Parameters

`value` - the int to be written

Throws

`MessageNotWritableException` - if message in write-only mode.

`JMSEException` - if JMS fails to read message due to some internal JMS error.

writeObject(Object)

```
public void writeObject(java.lang.Object value)
```

Write a java object to the stream message

Specified By

`javax.jms.StreamMessage.writeObject(java.lang.Object)` in interface `javax.jms.StreamMessage`

Parameters

`value` - the java object to be written.

Throws

`MessageNotWritableException` - if message in write-only mode.

`MessageFormatException` - if object is invalid type

`JMSEException` - if JMS fails to read message due to some internal JMS error.

writeShort(short)

```
public void writeShort(short value)
```

Write a short to the stream as a 2-byte, high byte first

Specified By

`javax.jms.StreamMessage.writeShort(short)` in interface `javax.jms.StreamMessage`

Parameters

`value` - the short to be written

Throws

`MessageNotWritableException` - if message in write-only mode.

`JMSEException` - if JMS fails to read message due to some internal JMS error.

writeString(String)

```
public void writeString(java.lang.String value)
```

Writes a string to the underlying output stream

Specified By

`javax.jms.StreamMessage.writeString(java.lang.String)` in interface `javax.jms.StreamMessage`

Parameters

`value` - The string to be written

Throws

`MessageNotWritableException` - if message in write-only mode.

`JMSEException` - if JMS fails to read message due to some internal JMS error.

AQjmsTextMessage

Syntax

```
public class AQjmsTextMessage extends AQjmsMessage
    implements javax.jms.TextMessage
```

```
java.lang.Object
|
+--AQjmsMessage
|
+--oracle.jms.AQjmsTextMessage
```

All Implemented Interfaces

javax.jms.Message, javax.jms.TextMessage

Description

This class implements the TextMessage interface. A TextMessage is used to send a message containing a java.lang.StringBuffer

Member Summary	Description
Methods	-
<code>clearBody()</code>	Clear the body
<code>clearProperties()</code>	Clear a message's properties.
<code>getText()</code>	Get the string containing this message's data.
<code>setText(String)</code>	Set the string containing this message's data.

Inherited Member Summary

Fields inherited from interface javax.jms.Message

DEFAULT_DELIVERY_MODE, DEFAULT_PRIORITY, DEFAULT_TIME_TO_LIVE

Methods inherited from class AQjmsMessage

Inherited Member Summary

getBooleanProperty(String), getByteProperty(String),
getDoubleProperty(String), getFloatProperty(String),
getIntProperty(String), getJMSCorrelationID(),
getJMSCorrelationIDAsBytes(), getJMSDeliveryMode(),
getJMSDestination(), getJMSExpiration(), getJMSMessageID(),
getJMSMessageIDAsBytes(), getJMSPriority(), getJMSRedelivered(),
getJMSReplyTo(), getJMSTimestamp(), getJMSType(),
getLongProperty(String), getObjectProperty(String),
getPropertyNames(), getSenderID(), getShortProperty(String),
getStringProperty(String), propertyExists(String),
setBooleanProperty(String, boolean), setByteProperty(String, byte),
setDoubleProperty(String, double), setFloatProperty(String, float),
setIntProperty(String, int), setJMSCorrelationID(String),
setJMSCorrelationIDAsBytes(byte[]), setJMSDestination(Destination),
setJMSExpiration(long), setJMSMessageID(String),
setJMSPriority(int), setJMSRedelivered(boolean),
setJMSReplyTo(Destination), setJMSTimestamp(long),
setJMSType(String), setLongProperty(String, long),
setObjectProperty(String, Object), setSenderID(AQjmsAgent),
setShortProperty(String, short), setStringProperty(String, String)

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll,
toString, wait, wait, wait

Methods inherited from interface javax.jms.Message

getBooleanProperty, getByteProperty, getDoubleProperty,
getFloatProperty, getIntProperty, getJMSCorrelationID,
getJMSCorrelationIDAsBytes, getJMSDeliveryMode, getJMSDestination,
getJMSExpiration, getJMSMessageID, getJMSPriority,
getJMSRedelivered, getJMSReplyTo, getJMSTimestamp, getJMSType,
getLongProperty, getObjectProperty, getPropertyNames,
getShortProperty, getStringProperty, propertyExists,
setBooleanProperty, setByteProperty, setDoubleProperty,
setFloatProperty, setIntProperty, setJMSCorrelationID,
setJMSCorrelationIDAsBytes, setJMSDeliveryMode, setJMSDestination,
setJMSExpiration, setJMSMessageID, setJMSPriority,
setJMSRedelivered, setJMSReplyTo, setJMSTimestamp, setJMSType,
setLongProperty, setObjectProperty, setShortProperty,
setStringProperty

Methods

clearBody()

```
public void clearBody()
```

Specified By

`javax.jms.Message.clearBody()` in interface `javax.jms.Message`

Overrides

`clearBody()` in class `AQjmsMessage`

clearProperties()

```
public void clearProperties()
```

Clear a message's properties.

Specified By

`javax.jms.Message.clearProperties()` in interface `javax.jms.Message`

Overrides

`clearProperties()` in class `AQjmsMessage`

Throws

`JMSEException` - if JMS fails to clear JMS message properties due to some internal JMS error.

getText()

```
public java.lang.String getText()
```

Get the string containing this message's data. The default value is null.

Specified By

`javax.jms.TextMessage.getText()` in interface `javax.jms.TextMessage`

Returns

the `String` containing the message's data

Throws

`JMSEException` - if JMS fails to get text due to some internal JMS error.

setText(String)

```
public void setText(java.lang.String string)
```

Set the string containing this message's data.

Specified By

`javax.jms.TextMessage.setText(java.lang.String)` in interface `javax.jms.TextMessage`

Parameters

`string` - the String containing the message's data

Throws

`JMSEException` - if JMS fails to set text due to some internal JMS error.

`MessageNotWriteableException` - if message in read-only mode.

AQjmsTopicBrowser

```
java.lang.Object
|
+ -- oracle.jms.AQjmsTopicBrowser
```

Description

A client uses an instance of `AQjmsTopicBrowser` to look at messages on a topic without removing them. This implementation is an Oracle-specific extension to JMS.

Member Summary	Description
Methods	-
getTopic()	Get the topic associated with this topic browser.
getEnumeration()	Get an enumeration for browsing the current topic messages in the order they would be received.
getMessageSelector()	Get this topic browser's message selector expression.
nextElement()	Return the next element of this enumeration.
hasMoreElements()	Check if this enumeration contains more elements.
purgeSeen()	Purge messages seen so far during browse.
setTransformation(String transformation)	Set transformation for the browser.
getTransformation()	Get the transformation for the Consumer.

Methods

close()

```
public void close()
```

Close the topic browser. Since OJMS allocates resources on behalf of a `TopicBrowser` outside the JVM, clients should close them when they are not needed. Relying on garbage collection to eventually reclaim these resources may not be timely enough. All errors during the operations will be silently ignored.

getTopic()

```
public Topic getTopic()
```

Get the topic associated with this topic browser

Returns

the topic associated with this topic browser

Throws

JMSEException-if JMS fails to get the topic associated with this Browser due to some JMS error.

getEnumeration()

```
public Enumeration getEnumeration()
```

Get an enumeration for browsing the current topic messages in the order they would be received. If `getEnumeration()` is called twice on the same `TopicBrowser`, the same enumeration object is returned. Hence the `nextElement()` call on one enumeration object would modify the state second enumeration object too

Returns

an enumeration for browsing the messages

getMessageSelector()

```
public String getMessageSelector()
```

Get this topic browser's message selector expression.

Returns

this topic browser's message selector

Throws

JMSEException-if JMS fails to get message selector due to some JMS error.

nextElement()

```
public Object nextElement()
```

Returns the next element of this enumeration. Attempt to use cached messages (if one is available from a previous call to `hasMoreElements()`). If the browser's selector used message ID, only one message can ever be returned during the browse

Returns

the next element of this enumeration

Throws

`NoSuchElementException`-if no more elements exist

hasMoreElements()

```
public boolean hasMoreElements()
```

Check if this enumeration contains more elements.

Returns

true if more elements exist in the enumeration false otherwise.

purgeSeen()

```
public void purgeSeen()
```

Purge messages seen so far during browse. A message is considered seen if it was returned to the client via a call to `nextElement()` during the browse. Thus, it is possible for a client to create a topic browser, call `purge` immediately, and not change the state of the topic (since no messages were seen as specified by this method).

- Purging a topic also does not affect the state of messages yet to be seen by the client during a browse.
- Purging is supported only on topic browsers that were created in the `LOCKED` mode. Attempting to purge topics that were not created in the `LOCKED` mode will result in an exception.
- The purge operation will take effect only if the session for this topic browser is committed. If the session is rolled back, the purge operation will be undone and the messages will become visible again.
- The purge will be finalized only when the session for the topic browse is committed.

Throws

`JMSEException` - if a JMS error occurred during the purge operation.

setTransformation(String transformation)

```
public void setTransformation(String transformation)
```

Set transformation for the browser. This transformation will be applied before the message is returned to the user.

Parameters

`transformation` - transformation name

getTransformation()

```
public String getTransformation()
```

Get the transformation for the Consumer.

Returns

the transformation for the Consumer

AQjmsIllegalStateException

Syntax

```
public class AQjmsIllegalStateException
    extends javax.jms.IllegalStateException
```

```
java.lang.Object
|
+--java.lang.Throwable
    |
    +--java.lang.Exception
        |
        +--javax.jms.JMSException
            |
            +--javax.jms.IllegalStateException
                |
                +--oracle.jms.AQjmsIllegalStateException
```

All Implemented Interfaces

```
java.io.Serializable
```

Description

This exception extends `IllegalStateException`. It is thrown when a method is invoked at an illegal or inappropriate time or if OJMS is not in an appropriate state for the requested operation. For example, this exception must be thrown if `Session.commit` is called on a non-transacted session.

Inherited Member Summary

Methods inherited from interface `javax.jms.JMSException`

```
getErrorCode, getLinkedException, setLinkedException
```

Methods inherited from class `java.lang.Throwable`

```
fillInStackTrace, getLocalizedMessage, getMessage, printStackTrace,
printStackTrace, printStackTrace, toString
```

Methods inherited from class `java.lang.Object`

```
clone, equals, finalize, getClass, hashCode, notify, notifyAll,
wait, wait, wait
```

AQjmsTopicConnectionFactory

Syntax

```
public class AQjmsTopicConnectionFactory extends java.lang.Object
    implements javax.jms.TopicConnectionFactory
```

```
java.lang.Object
|
+--oracle.jms.AQjmsTopicConnectionFactory
```

All Implemented Interfaces

javax.jms.ConnectionFactory, javax.jms.TopicConnectionFactory,
java.lang.Referenceable, java.lang.Serializable

Description

This class implements the TopicConnectionFactory interface. A TopicConnectionFactory is used to create TopicConnections

Member Summary	Description
Methods	-
<code>createTopicConnection()</code>	create a Topic Connection to the JMS Server hosting this Topic- connection factory.
<code>createTopicConnection(Connection)</code>	create a TopicConnection using the already open JDBC connection.
<code>createTopicConnection(String, String)</code>	create a Topic Connection using the given username and password for authentication during creation of the Connection.

Inherited Member Summary

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Methods

createTopicConnection()

```
public javax.jms.TopicConnection createTopicConnection()  
create a Topic Connection to the JMS Server hosting this Topic- connection factory.
```

Specified By

`javax.jms.TopicConnectionFactory.createTopicConnection()` in interface `javax.jms.TopicConnectionFactory`

Returns

a Topic Connection

Throws

`JMSEException` - if JMS fails to get a topic connection due to some JMS error.

createTopicConnection(Connection)

```
public static javax.jms.TopicConnection  
createTopicConnection(java.sql.Connection jdbc_connection)  
create a TopicConnection using the already open JDBC connection. This creation  
does NOT result in creation of another connection to the database. Instead JMS  
binds to the given connection to the database and provides an interface to the  
Pub/Sub mechanism defined by JMS.
```

Parameters

`jdbc_connection` - a valid open connection to the database.

Returns

a TopicConnection

Throws

`JMSEException` - if JMS fails to get a topic connection due to some JMS error

createTopicConnection(String, String)

```
public javax.jms.TopicConnection createTopicConnection(java.lang.String  
username, java.lang.String password)
```

create a Topic Connection using the given username and password for authentication during creation of the Connection.

Specified By

`javax.jms.TopicConnectionFactory.createTopicConnection(java.lang.String, java.lang.String)` in interface `javax.jms.TopicConnectionFactory`

Parameters

`username` - name of the user connecting to the DB for Queueing. `password` password for the user creating the connection.

Returns

a Topic Connection

Throws

`JMSEException` - if JMS fails to get a topic connection due to some JMS error

AQjmsTopicPublisher

Syntax

```
public interface AQjmsTopicPublisher extends javax.jms.TopicPublisher
```

All Superinterface

```
javax.jms.MessageProducer, javax.jms.TopicPublisher
```

All Known Implementing Classes

```
AQjmsProducer
```

Description

This interface extends `TopicPublisher` and defines AQ extensions to JMS. A client uses a `TopicPublisher` for publishing messages to a `Topic`

Member Summary	Description
Methods	-
getTransformation()	get the transformation for this publisher
publish(Message, AQjmsAgent[])	Publish a Message to a specific list of recipients
publish(Message, AQjmsAgent[], int, int, long)	Publish a Message to a topic by specifying a list of recipients, delivery mode, priority and time to live
publish(Topic, Message, AQjmsAgent[])	Publish a Message to a topic by specifying a list of recipients
publish(Topic, Message, AQjmsAgent[], int, int, long)	Publish a Message to a topic by specifying a list of recipients, delivery mode, priority and time to live
setTransformation(String)	set the transformation for this publisher

Inherited Member Summary

Methods inherited from interface `javax.jms.TopicPublisher`

```
getTopic, publish, publish, publish, publish
```

Methods inherited from interface `javax.jms.MessageProducer`

(Cont.) Inherited Member Summary

```
close, getDeliveryMode, getDisableMessageID,  
getDisableMessageTimestamp, getPriority, getTimeToLive,  
setDeliveryMode, setDisableMessageID, setDisableMessageTimestamp,  
setPriority, setTimeToLive
```

Methods

getTransformation()

```
public String getTransformation()  
Get the transformation for this publisher
```

Returns

the transformation

Throws

`JMSEException` - if there was an error in getting the transformation

publish(Message, AQjmsAgent[])

```
public void publish(javax.jms.Message message, AQjmsAgent recipient_list)  
Publish a Message to a specific list of recipients
```

Parameters

`message` - The message to be published

`recipient_list` - The list of recipients to which the message is published. The recipients are of type `AQjmsAgent`.

Throws

`JMSEException` - if JMS fails to publish the message due to some internal error.

publish(Message, AQjmsAgent[], int, int, long)

```
public void publish(javax.jms.Message message, AQjmsAgent recipient_list, int  
deliveryMode, int priority, long timeToLive)
```

Publish a Message to a topic by specifying a list of recipients, delivery mode, priority and time to live

Parameters

`message` - The message to be published

`recipient_list` - The list of recipients to which the message is published. The recipients are of type `AQjmsAgent`.

`deliveryMode` - The delivery mode - `persistent` or `non_persistent`

`priority` - The priority of the message

`timeToLive` - the message time to live in milliseconds; zero is unlimited

Throws

`JMSEException` - if JMS fails to publish the message due to some internal error.

publish(Topic, Message, AQjmsAgent[])

```
public void publish(javax.jms.Topic topic, javax.jms.Message message, AQjmsAgent recipient_list)
```

Publish a Message to a topic by specifying a list of recipients

Parameters

`topic` - The topic to which to publish the message. This overrides the default topic of the Message Producer

`message` - The message to be published

`recipient_list` - The list of recipients to which the message is published. The recipients are of type `AQjmsAgent`.

Throws

`JMSEException` - if JMS fails to publish the message due to some internal error.

publish(Topic, Message, AQjmsAgent[], int, int, long)

```
public void publish(javax.jms.Topic topic, javax.jms.Message message, AQjmsAgent recipient_list, int deliveryMode, int priority, long timeToLive)
```

Publish a Message to a topic by specifying a list of recipients, delivery mode, priority and time to live

Parameters

`topic` - The topic to which to publish the message. This overrides the default topic of the Message Producer

`message` - The message to be published

`recipient_list` - The list of recipients to which the message is published. The recipients are of type `AQjmsAgent`.

`deliveryMode` - The delivery mode - persistent or non_persistent

`priority` - The priority of the message

`timeToLive` - the message time to live in milliseconds; zero is unlimited

Throws

`JMSEException` - if JMS fails to publish the message due to some internal error.

setTransformation(String)

```
public void setTransformation(String transformation)
```

Set transformation for this sender. This transformation will be applied before the message is published to the topic

Parameters

`transformation` - transformation to be applied before publishing the message

Throws

`JMSEException` - if there was an error in setting the transformation

AQjmsTopicReceiver

Syntax

```
public interface AQjmsTopicReceiver extends TopicReceiver
```

All Superinterfaces

```
javax.jms.MessageConsumer, TopicReceiver
```

All Known Implementing Classes

```
AQjmsConsumer
```

Description

This interface extends the `TopicReceiver` interface that defines AQ extensions for remote subscribers and explicitly specified recipients (in point-to-multipoint communication). A `TopicReceiver` is used to receive messages from a `Topic`.

Member Summary	Description
Methods	-
<code>getNavigationMode()</code>	get the navigation mode used for receiving messages
<code>getTransformation()</code>	get the transformation for this receiver.
<code>receiveNoData()</code>	Consume the message without returning it to the user.
<code>receiveNoData(long)</code>	Consume the message without returning it to the user.
<code>setNavigationMode(int)</code>	set the navigation mode used for receiving messages.
<code>setTransformation(String)</code>	set the transformation for this receiver.

Inherited Member Summary

Methods inherited from interface `TopicReceiver`

```
getTopic()
```

Methods inherited from interface `javax.jms.MessageConsumer`

```
close, getMessageListener, getMessageSelector, receive, receive,  
receiveNoWait, setMessageListener
```

Methods

getNavigationMode()

```
public int getNavigationMode()  
Get the navigation mode used for receiving messages
```

Returns

the navigation mode

Throws

`JMSEException` - if there was an error in getting the navigation mode

getTransformation()

```
public String getTransformation()  
Get the transformation for this receiver
```

Returns

the transformation

Throws

`JMSEException` - if there was an error in getting the transformation

receiveNoData()

```
public void receiveNoData()  
Consume the message without returning it to the user. This call will avoid the overhead of fetching the message from the database and hence can be used as an optimization by jms clients who have already got the message for example using a queue browser.
```

Throws

`JMSEException` - if the message could not be received due to an error

receiveNoData(long)

```
public void receiveNoData(long tomeOut)
```

Consume the message without returning it to the user. This call will avoid the overhead of fetching the message from the database and hence can be used as an optimization by jms clients who have already got the message for example using a queue browser. This call will block until a message arrives or the timeout expires

Parameters

`timeout` - the timeout value in milliseconds

Throws

`JMSEException` - if the message could not be received due to an error

setTransformation(String)

```
public void setTransformation(String transformation)
```

Set transformation for this receiver. This transformation will be applied before the message is returned to the user.

Parameters

`transformation` - transformation to be applied before returning the message

Throws

`JMSEException` - if there was an error in setting the transformation

setNavigationMode(int)

```
public void setNavigationMode(int mode)
```

set the navigation mode used for receiving messages

Parameters

`mode` - the new value of the navigation mode

Throws

`JMSEException` - if there was an error in getting the navigation mode

AQjmsTopicSubscriber

Syntax

```
public interface AQjmsTopicSubscriber extends javax.jms.TopicSubscriber
```

All Superinterfaces

```
javax.jms.MessageConsumer, javax.jms.TopicSubscriber
```

All Known Implementing Classes

```
AQjmsConsumer
```

Description

This interface extends `TopicSubscriber` and defines AQ extensions to JMS. A client uses a `TopicSubscriber` to receive messages published on a `Topic`

Member Summary	Description
Methods	-
<code>getNavigationMode()</code>	Consume the message without returning it to the user.
<code>receiveNoData()</code>	Consume the message without returning it to the user.
<code>receiveNoData(long)</code>	Consume the message without returning it to the user.
<code>setNavigationMode(int)</code>	set the navigation mode used for receiving messages

Inherited Member Summary

Methods inherited from interface `javax.jms.TopicSubscriber`

```
getNoLocal, getTopic
```

Methods inherited from interface `javax.jms.MessageConsumer`

```
close, getMessageListener, getMessageSelector, receive, receive, receiveNoWait, setMessageListener
```

Methods

getNavigationMode()

```
public int getNavigationMode()
```

Consume the message without returning it to the user. This call will avoid the overhead of fetching the message from the database and hence can be used as an optimization by jms clients who have already got the message for example using a queue browser. This call will block until a message arrives or the timeout expires

Parameters

`timeout` - the timeout value in milliseconds

Throws

`JMSEException` - if the message could not be received due to an error

receiveNoData()

```
public void receiveNoData()
```

Consume the message without returning it to the user. This call will avoid the overhead of fetching the message from the database and hence can be used as an optimization by jms clients who have already got the message for example using a queue browser.

Throws

`JMSEException` - if the message could not be received due to an error

receiveNoData(long)

```
public void receiveNoData(long timeOut)
```

Consume the message without returning it to the user. This call will avoid the overhead of fetching the message from the database and hence can be used as an optimization by jms clients who have already got the message for example using a queue browser. This call will block until a message arrives or the timeout expires

Parameters

`timeout` - the timeout value in milliseconds

Throws

`JMSEException` - if the message could not be received due to an error

setNavigationMode(int)

```
public void setNavigationMode(int mode)
set the navigation mode used for receiving messages
```

Parameters

`mode` - the new value of the navigation mode

Throws

`JMSEException` - if there was an error in getting the navigation mode

TopicBrowser

Syntax

```
public interface TopicBrowser extends javax.jms.MessageConsumer
```

All Known Subinterfaces

AQjmsTopicBrowser

All Superinterfaces

javax.jms.MessageConsumer

Description

This interface extends MessageConsumer to allow remote subscribers to look at messages on a topic without removing them.

TopicReceiver

Syntax

```
public interface TopicReceiver extends javax.jms.MessageConsumer
```

All Known Subinterfaces

AQjmsTopicReceiver

All Superinterfaces

javax.jms.MessageConsumer

Description

This interface extends MessageConsumer to allow remote subscribers and explicitly specified recipients (in point-to-multipoint communication) to receive messages.

Member Summary

Description

Methods

-

[getTopic\(\)](#)

Get the topic associated with this receiver.

Inherited Member Summary

Methods inherited from interface javax.jms.MessageConsumer

close, getMessageListener, getMessageSelector, receive, receive,
receiveNoWait, setMessageListener

Methods

getTopic()

```
public javax.jms.Topic getTopic()  
Get the topic associated with this receiver.
```

Returns

this subscriber's topic

Throws

`JMSEException` - if JMS fails to get topic for this topic receiver due to some internal error.

Package oracle.ODCI

This chapter describes Java language ODCI (Oracle Data Cartridge Interface) extensible indexing interfaces that are provided in package oracle.ODCI.

This chapter includes these sections:

- [Package oracle.ODCI Description](#)
- [Installing ODCI.jar and CartridgeServices.jar Files](#)
- [Package oracle.ODCI Summary](#)

Package oracle.ODCI Description

In addition to the efficient and secure management of data ordered under the relational model, Oracle supports data organized under the object model. Object types and other Oracle database features, such as large objects (LOBs), external procedures, extensible indexing, and query optimization, can be used to build reusable server-based components called data cartridges.

Within the framework of the Oracle Extensibility Architecture, data cartridges are a pure object-oriented mechanism for extending the capabilities of the Oracle server.

See Also: *Oracle9i Data Cartridge Developer's Guide* for information about creating and using data cartridges.

Installing ODCI.jar and CartridgeServices.jar Files

The `ODCI.jar` and `CartridgeServices.jar` files must be installed into the SYS schema in order to use the Java classes described in this chapter.

If you installed the Java option, then you must install the `ODCI.jar` and `CartridgeServices.jar` files. You do not need to perform this task if you did not install the Java option.

To install `ODCI.jar` and `CartridgeServices.jar` files, run the following commands from the command line:

```
loadjava -user sys/PASSWORD -resolve -synonym -grant public  
-verbose ORACLE_HOME/vobs/jlib/CartridgeServices.jar
```

```
loadjava -user sys/PASSWORD -resolve -synonym -grant public  
-verbose ORACLE_HOME/vobs/jlib/ODCI.jar
```

Substitute the SYS password for `PASSWORD`, and substitute the Oracle home directory for `ORACLE_HOME`. These commands install the classes and create the synonyms in the SYS schema.

Package oracle.ODCI Summary

Table 5–1 Package oracle.ODCI summary

Class	Description
ODCIArgDesc	Argument description
ODCIArgDescList	Argument description list
ODCIArgDescRef	Reference argument description
ODCIColInfo	Column info
ODCIColInfoList	Column info list
ODCIColInfoRef	Column info reference
ODCICost	Cost
ODCICostRef	Cost reference
ODCIEnv	Environment
ODCIEnvRef	Environment reference
ODCIFuncInfo	Function info
ODCIFuncInfoRef	Function info reference
ODCIIndexCtx	Index context
ODCIIndexCtxRef	Index context reference
ODCIIndexInfo	Index info
ODCIQueryInfoRef	Query info reference
ODCIRidList	Rid list
ODCIStatsOptions	Stats options
ODCIStatsOptionsRef	Stats options reference

ODCIArgDesc

```
oracle.ODCI.ODCIArgDesc  
public class ODCIArgDesc
```

Fields

_SQL_NAME

```
public static final java.lang.String _SQL_NAME
```

_SQL_TYPECODE

```
public static final int _SQL_TYPECODE
```

Constructors

ODCIArgDesc

```
public ODCIArgDesc()
```

Methods

getORADDataFactory

```
public static oracle.sql.ORADDataFactory getORADDataFactory()
```

toDatum

```
public oracle.sql.Datum toDatum(java.sql.Connection c)  
    throws java.sql.SQLException
```

create

```
public oracle.sql.ORADData create(oracle.sql.Datum d,  
    int sqlType)  
    throws java.sql.SQLException
```

```
getArgType  
public java.math.BigDecimal getArgType()  
    throws java.sql.SQLException
```

setArgType

```
public void setArgType(java.math.BigDecimal ArgType)  
    throws java.sql.SQLException
```

getTableName

```
public java.lang.String getTableName()  
    throws java.sql.SQLException
```

setTableName

```
public void setTableName(java.lang.String TableName)  
    throws java.sql.SQLException
```

getTableSchema

```
public java.lang.String getTableSchema()  
    throws java.sql.SQLException
```

setTableSchema

```
public void setTableSchema(java.lang.String TableSchema)  
    throws java.sql.SQLException
```

getColName

```
public java.lang.String getColName()  
    throws java.sql.SQLException
```

setColName

```
public void setColName(java.lang.String ColName)  
    throws java.sql.SQLException
```

getTablePartitionLower

```
public java.lang.String getTablePartitionLower()
```

throws java.sql.SQLException

setTablePartitionLower

```
public void setTablePartitionLower(java.lang.String TablePartitionLower)
    throws java.sql.SQLException
```

getTablePartitionUpper

```
public java.lang.String getTablePartitionUpper()
    throws java.sql.SQLException
```

setTablePartitionUpper

```
public void setTablePartitionUpper(java.lang.String TablePartitionUpper)
    throws java.sql.SQLException
```

ODCIArgDescList

```
oracle.ODCI.ODCIArgDescList  
  
public class ODCIArgDescList
```

Fields

_SQL_NAME

```
public static final java.lang.String _SQL_NAME
```

_SQL_TYPECODE

```
public static final int _SQL_TYPECODE
```

Constructors

ODCIArgDescList

```
public ODCIArgDescList()
```

ODCIArgDescList

```
public ODCIArgDescList(ODCIArgDesc[] a)
```

Methods

getORADataFactory

```
public static oracle.sql.ORADataFactory getORADataFactory()
```

toDatum

```
public oracle.sql.Datum toDatum(java.sql.Connection c)  
    throws java.sql.SQLException
```

create

```
public oracle.sql.ORADATA create(oracle.sql.Datum d,  
                                int sqlType)  
    throws java.sql.SQLException
```

length

```
public int length()  
    throws java.sql.SQLException
```

getBaseType

```
public int getBaseType()  
    throws java.sql.SQLException
```

getBaseTypeName

```
public java.lang.String getBaseTypeName()  
    throws java.sql.SQLException
```

getDescriptor

```
public oracle.sql.ArrayDescriptor getDescriptor()  
    throws java.sql.SQLException
```

getArray

```
public ODCIArgDesc[] getArray()  
    throws java.sql.SQLException
```

setArray

```
public void setArray(ODCIArgDesc[] a)  
    throws java.sql.SQLException
```

getArray

```
public ODCIArgDesc[] getArray(long index,  
                                int count)  
    throws java.sql.SQLException
```


setArray

```
public void setArray(ODCIArgDesc[] a,  
                    long index)  
    throws java.sql.SQLException
```

getElement

```
public ODCIArgDesc getElement(long index)  
    throws java.sql.SQLException
```

setElement

```
public void setElement(ODCIArgDesc a,  
                      long index)  
    throws java.sql.SQLException
```

ODCIArgDescRef

```
oracle.ODCI.ODCIArgDescRef  
  
public class ODCIArgDescRef
```

Fields

_SQL_BASETYPE

```
public static final java.lang.String _SQL_BASETYPE
```

_SQL_TYPECODE

```
public static final int _SQL_TYPECODE
```

Constructors

ODCIArgDescRef

```
public ODCIArgDescRef()
```

Methods

getORADDataFactory

```
public static oracle.sql.ORADDataFactory getORADDataFactory()
```

toDatum

```
public oracle.sql.Datum toDatum(java.sql.Connection c)  
    throws java.sql.SQLException
```

create

```
public oracle.sql.ORADData create(oracle.sql.Datum d,  
    int sqlType)  
    throws java.sql.SQLException
```

getValue

```
public ODCIArgDesc getValue()  
    throws java.sql.SQLException
```

setValue

```
public void setValue(ODCIArgDesc c)  
    throws java.sql.SQLException
```

ODCICollInfo

```
oracle.ODCI.ODCICollInfo  
public class ODCICollInfo
```

Fields

_SQL_NAME

```
public static final java.lang.String _SQL_NAME
```

_SQL_TYPECODE

```
public static final int _SQL_TYPECODE
```

Constructors

ODCICollInfo

```
public ODCICollInfo()
```

Methods

getORADDataFactory

```
public static oracle.sql.ORADDataFactory getORADDataFactory()
```

toDatum

```
public oracle.sql.Datum toDatum(java.sql.Connection c)  
    throws java.sql.SQLException
```

create

```
public oracle.sql.ORADData create(oracle.sql.Datum d,  
    int sqlType)  
    throws java.sql.SQLException
```

getTableSchema

```
public java.lang.String getTableSchema()  
    throws java.sql.SQLException
```

setTableSchema

```
public void setTableSchema(java.lang.String TableSchema)  
    throws java.sql.SQLException
```

getTableName

```
public java.lang.String getTableName()  
    throws java.sql.SQLException
```

setTableName

```
public void setTableName(java.lang.String TableName)  
    throws java.sql.SQLException
```

getColName

```
public java.lang.String getColName()  
    throws java.sql.SQLException
```

setColName

```
public void setColName(java.lang.String ColName)  
    throws java.sql.SQLException
```

getColTypeName

```
public java.lang.String getColTypeName()  
    throws java.sql.SQLException
```

setColTypeName

```
public void setColTypeName(java.lang.String ColTypeName)  
    throws java.sql.SQLException
```

getColTypeSchema

```
public java.lang.String getColTypeSchema()  
    throws java.sql.SQLException
```

setColTypeSchema

```
public void setColTypeSchema(java.lang.String ColTypeSchema)  
    throws java.sql.SQLException
```

getTablePartition

```
public java.lang.String getTablePartition()  
    throws java.sql.SQLException
```

setTablePartition

```
public void setTablePartition(java.lang.String TablePartition)  
    throws java.sql.SQLException
```

ODCIColInfoList

```
oracle.ODCI.ODCIColInfoList  
public class ODCIColInfoList
```

Fields

_SQL_NAME

```
public static final java.lang.String _SQL_NAME
```

_SQL_TYPECODE

```
public static final int _SQL_TYPECODE
```

Constructors

ODCIColInfoList

```
public ODCIColInfoList()
```

ODCIColInfoList

```
public ODCIColInfoList(ODCIColInfo[] a)
```

Methods

getORADDataFactory

```
public static oracle.sql.ORADDataFactory getORADDataFactory()
```

toDatum

```
public oracle.sql.Datum toDatum(java.sql.Connection c)  
    throws java.sql.SQLException
```

create

```
public oracle.sql.ORADATA create(oracle.sql.Datum d,  
                                int sqlType)  
    throws java.sql.SQLException
```

length

```
public int length()  
    throws java.sql.SQLException
```

getBaseType

```
public int getBaseType()  
    throws java.sql.SQLException
```

getBaseTypeName

```
public java.lang.String getBaseTypeName()  
    throws java.sql.SQLException
```

getDescriptor

```
public oracle.sql.ArrayDescriptor getDescriptor()  
    throws java.sql.SQLException
```

getArray

```
public ODCIColInfo[] getArray()  
    throws java.sql.SQLException
```

setArray

```
public void setArray(ODCIColInfo[] a)  
    throws java.sql.SQLException
```

getArray

```
public ODCIColInfo[] getArray(long index,  
                                int count)  
    throws java.sql.SQLException
```


setArray

```
public void setArray(ODCIColInfo[] a,  
                    long index)  
    throws java.sql.SQLException
```

getElement

```
public ODCIColInfo getElement(long index)  
    throws java.sql.SQLException
```

setElement

```
public void setElement(ODCIColInfo a,  
                      long index)  
    throws java.sql.SQLException
```

ODCIColInfoRef

```
oracle.ODCI.ODCIColInfoRef  
public class ODCIColInfoRef
```

Fields

_SQL_BASETYPE

```
public static final java.lang.String _SQL_BASETYPE
```

_SQL_TYPECODE

```
public static final int _SQL_TYPECODE
```

Constructors

ODCIColInfoRef

```
public ODCIColInfoRef()
```

Methods

getORADDataFactory

```
public static oracle.sql.ORADDataFactory getORADDataFactory()
```

toDatum

```
public oracle.sql.Datum toDatum(java.sql.Connection c)  
throws java.sql.SQLException
```

create

```
public oracle.sql.ORADData create(oracle.sql.Datum d,  
int sqlType)  
throws java.sql.SQLException
```

getValue

```
public ODCIColInfo getValue()  
    throws java.sql.SQLException
```

setValue

```
public void setValue(ODCIColInfo c)  
    throws java.sql.SQLException
```

ODCICost

```
oracle.ODCI.ODCICost  
public class ODCICost
```

Fields

_SQL_NAME

```
public static final java.lang.String _SQL_NAME
```

_SQL_TYPECODE

```
public static final int _SQL_TYPECODE
```

Constructors

ODCICost

```
public ODCICost()
```

Methods

getORADDataFactory

```
public static oracle.sql.ORADDataFactory getORADDataFactory()
```

toDatum

```
public oracle.sql.Datum toDatum(java.sql.Connection c)  
    throws java.sql.SQLException
```

create

```
public oracle.sql.ORADData create(oracle.sql.Datum d,  
    int sqlType)  
    throws java.sql.SQLException
```

getCPUcost

```
public java.math.BigDecimal getCPUcost()  
    throws java.sql.SQLException
```

setCPUcost

```
public void setCPUcost(java.math.BigDecimal CPUcost)  
    throws java.sql.SQLException
```

getIOcost

```
public java.math.BigDecimal getIOcost()  
    throws java.sql.SQLException
```

setIOcost

```
public void setIOcost(java.math.BigDecimal IOcost)  
    throws java.sql.SQLException
```

getNetworkCost

```
public java.math.BigDecimal getNetworkCost()  
    throws java.sql.SQLException
```

setNetworkCost

```
public void setNetworkCost(java.math.BigDecimal NetworkCost)  
    throws java.sql.SQLException
```

getIndexCostInfo

```
public java.lang.String getIndexCostInfo()  
    throws java.sql.SQLException
```

setIndexCostInfo

```
public void setIndexCostInfo(java.lang.String IndexCostInfo)  
    throws java.sql.SQLException
```

ODCICostRef

```
oracle.ODCI.ODCICostRef  
public class ODCICostRef
```

Fields

_SQL_BASETYPE

```
public static final java.lang.String _SQL_BASETYPE
```

_SQL_TYPECODE

```
public static final int _SQL_TYPECODE
```

Constructors

ODCICostRef

```
public ODCICostRef()
```

Methods

getORADDataFactory

```
public static oracle.sql.ORADDataFactory getORADDataFactory()
```

toDatum

```
public oracle.sql.Datum toDatum(java.sql.Connection c)  
throws java.sql.SQLException
```

create

```
public oracle.sql.ORADData create(oracle.sql.Datum d,  
int sqlType)  
throws java.sql.SQLException
```

getValue

```
public ODCICost getValue()  
    throws java.sql.SQLException
```

setValue

```
public void setValue(ODCICost c)  
    throws java.sql.SQLException
```

ODCIEnv

```
oracle.ODCI.ODCIEnv  
public class ODCIEnv
```

Fields

SQL_NAME

```
public static final java.lang.String _SQL_NAME
```

_SQL_TYPECODE

```
public static final int _SQL_TYPECODE
```

Constructors

ODCIEnv

```
public ODCIEnv()
```

Methods

getORADDataFactory

```
public static oracle.sql.ORADDataFactory getORADDataFactory()
```

toDatum

```
public oracle.sql.Datum toDatum(java.sql.Connection c)  
    throws java.sql.SQLException
```

create

```
public oracle.sql.ORADData create(oracle.sql.Datum d,  
    int sqlType)  
    throws java.sql.SQLException
```


getEnvFlags

```
public java.math.BigDecimal getEnvFlags()  
    throws java.sql.SQLException
```

setEnvFlags

```
public void setEnvFlags(java.math.BigDecimal EnvFlags)  
    throws java.sql.SQLException
```

getCallProperty

```
public java.math.BigDecimal getCallProperty()  
    throws java.sql.SQLException
```

setCallProperty

```
public void setCallProperty(java.math.BigDecimal CallProperty)  
    throws java.sql.SQLException
```

ODCIEnvRef

```
oracle.ODCI.ODCIEnvRef  
public class ODCIEnvRef
```

Fields

_SQL_BASETYPE

```
public static final java.lang.String _SQL_BASETYPE
```

_SQL_TYPECODE

```
public static final int _SQL_TYPECODE
```

Constructors

ODCIEnvRef

```
public ODCIEnvRef()
```

Methods

getORADDataFactory

```
public static oracle.sql.ORADDataFactory getORADDataFactory()
```

toDatum

```
public oracle.sql.Datum toDatum(java.sql.Connection c)  
throws java.sql.SQLException
```

create

```
public oracle.sql.ORADData create(oracle.sql.Datum d,  
int sqlType)  
throws java.sql.SQLException
```

getValue

```
public ODCIEnv getValue()  
    throws java.sql.SQLException
```

setValue

```
public void setValue(ODCIEnv c)  
    throws java.sql.SQLException
```

ODCIFuncInfo

```
oracle.ODCI.ODCIFuncInfo  
public class ODCIFuncInfo
```

Fields

_SQL_NAME

```
public static final java.lang.String _SQL_NAME
```

_SQL_TYPECODE

```
public static final int _SQL_TYPECODE
```

Constructors

ODCIFuncInfo

```
public ODCIFuncInfo()
```

Methods

getORADDataFactory

```
public static oracle.sql.ORADDataFactory getORADDataFactory()
```

toDatum

```
public oracle.sql.Datum toDatum(java.sql.Connection c)  
    throws java.sql.SQLException
```

create

```
public oracle.sql.ORADData create(oracle.sql.Datum d,  
    int sqlType)  
    throws java.sql.SQLException
```

getObjectSchema

```
public java.lang.String getObjectSchema()  
    throws java.sql.SQLException
```

setObjectSchema

```
public void setObjectSchema(java.lang.String ObjectSchema)  
    throws java.sql.SQLException
```

getObjectName

```
public java.lang.String getObjectName()  
    throws java.sql.SQLException
```

setObjectName

```
public void setObjectName(java.lang.String ObjectName)  
    throws java.sql.SQLException
```

getMethodName

```
public java.lang.String getMethodName()  
    throws java.sql.SQLException
```

setMethodName

```
public void setMethodName(java.lang.String MethodName)  
    throws java.sql.SQLException
```

getFlags

```
public java.math.BigDecimal getFlags()  
    throws java.sql.SQLException
```

setFlags

```
public void setFlags(java.math.BigDecimal Flags)  
    throws java.sql.SQLException
```

ODCIFuncInfoRef

```
oracle.ODCI.ODCIFuncInfoRef  
public class ODCIFuncInfoRef
```

Fields

_SQL_BASETYPE

```
public static final java.lang.String _SQL_BASETYPE
```

_SQL_TYPECODE

```
public static final int _SQL_TYPECODE
```

Constructors

ODCIFuncInfoRef

```
public ODCIFuncInfoRef()
```

Methods

getORADDataFactory

```
public static oracle.sql.ORADDataFactory getORADDataFactory()
```

toDatum

```
public oracle.sql.Datum toDatum(java.sql.Connection c)  
    throws java.sql.SQLException
```

create

```
public oracle.sql.ORADData create(oracle.sql.Datum d,  
    int sqlType)  
    throws java.sql.SQLException
```

getValue

```
public ODCIFuncInfo getValue()  
    throws java.sql.SQLException
```

setValue

```
public void setValue(ODCIFuncInfo c)  
    throws java.sql.SQLException
```

ODCIIndexCtx

```
oracle.ODCI.ODCIIndexCtx  
public class ODCIIndexCtx
```

Fields

_SQL_NAME

```
public static final java.lang.String _SQL_NAME
```

_SQL_TYPECODE

```
public static final int _SQL_TYPECODE
```

Constructors

ODCIIndexCtx

```
public ODCIIndexCtx()
```

Methods

getORADDataFactory

```
public static oracle.sql.ORADDataFactory getORADDataFactory()
```

toDatum

```
public oracle.sql.Datum toDatum(java.sql.Connection c)  
throws java.sql.SQLException
```

create

```
public oracle.sql.ORADData create(oracle.sql.Datum d,  
int sqlType)  
throws java.sql.SQLException
```


getIndexInfo

```
public ODCIIndexInfo getIndexInfo()  
    throws java.sql.SQLException
```

setIndexInfo

```
public void setIndexInfo(ODCIIndexInfo IndexInfo)  
    throws java.sql.SQLException
```

getRid

```
public java.lang.String getRid()  
    throws java.sql.SQLException
```

setRid

```
public void setRid(java.lang.String Rid)  
    throws java.sql.SQLException
```

ODCIIndexCtxRef

```
oracle.ODCI.ODCIIndexCtxRef  
public class ODCIIndexCtxRef
```

Fields

_SQL_BASETYPE

```
public static final java.lang.String _SQL_BASETYPE
```

_SQL_TYPECODE

```
public static final int _SQL_TYPECODE
```

Constructors

ODCIIndexCtxRef

```
public ODCIIndexCtxRef()
```

Methods

getORADDataFactory

```
public static oracle.sql.ORADDataFactory getORADDataFactory()
```

toDatum

```
public oracle.sql.Datum toDatum(java.sql.Connection c)  
    throws java.sql.SQLException
```

create

```
public oracle.sql.ORADData create(oracle.sql.Datum d,  
    int sqlType)  
    throws java.sql.SQLException
```

getValue

```
public ODCIIndexCtx getValue()  
    throws java.sql.SQLException
```

setValue

```
public void setValue(ODCIIndexCtx c)  
    throws java.sql.SQLException
```

ODCIIndexInfo

```
oracle.ODCI.ODCIIndexInfo  
public class ODCIIndexInfo
```

Fields

_SQL_NAME

```
public static final java.lang.String _SQL_NAME
```

_SQL_TYPECODE

```
public static final int _SQL_TYPECODE
```

Constructors

ODCIIndexInfo

```
public ODCIIndexInfo()
```

Methods

getORADDataFactory

```
public static oracle.sql.ORADDataFactory getORADDataFactory()
```

toDatum

```
public oracle.sql.Datum toDatum(java.sql.Connection c)  
throws java.sql.SQLException
```

create

```
public oracle.sql.ORADData create(oracle.sql.Datum d,  
int sqlType)  
throws java.sql.SQLException
```

getIndexSchema

```
public java.lang.String getIndexSchema()  
    throws java.sql.SQLException
```

setIndexSchema

```
public void setIndexSchema(java.lang.String IndexSchema)  
    throws java.sql.SQLException
```

getIndexName

```
public java.lang.String getIndexName()  
    throws java.sql.SQLException
```

setIndexName

```
public void setIndexName(java.lang.String IndexName)  
    throws java.sql.SQLException
```

getIndexCols

```
public ODCIColInfoList getIndexCols()  
    throws java.sql.SQLException
```

setIndexCols

```
public void setIndexCols(ODCIColInfoList IndexCols)  
    throws java.sql.SQLException
```

getIndexPartition

```
public java.lang.String getIndexPartition()  
    throws java.sql.SQLException
```

setIndexPartition

```
public void setIndexPartition(java.lang.String IndexPartition)  
    throws java.sql.SQLException
```

getIndexInfoFlags

```
public java.math.BigDecimal getIndexInfoFlags()  
    throws java.sql.SQLException
```

setIndexInfoFlags

```
public void setIndexInfoFlags(java.math.BigDecimal IndexInfoFlags)  
    throws java.sql.SQLException
```

ODCIIndexInfoRef

```
oracle.ODCI.ODCIIndexInfoRef  
public class ODCIIndexInfoRef
```

Fields

_SQL_BASETYPE

```
public static final java.lang.String _SQL_BASETYPE
```

_SQL_TYPECODE

```
public static final int _SQL_TYPECODE
```

Constructors

ODCIIndexInfoRef

```
public ODCIIndexInfoRef()
```

Methods

getORADataFactory

```
public static oracle.sql.ORADataFactory getORADataFactory()
```

toDatum

```
public oracle.sql.Datum toDatum(java.sql.Connection c)  
throws java.sql.SQLException
```

create

```
public oracle.sql.ORAData create(oracle.sql.Datum d,  
int sqlType)  
throws java.sql.SQLException
```

getValue

```
public ODCIIndexInfo getValue()  
    throws java.sql.SQLException
```

setValue

```
public void setValue(ODCIIndexInfo c)  
    throws java.sql.SQLException
```


ODCIObject

```
oracle.ODCI.ODCIObject  
public class ODCIObject
```

Fields

_SQL_NAME

```
public static final java.lang.String _SQL_NAME
```

_SQL_TYPECODE

```
public static final int _SQL_TYPECODE
```

Constructors

ODCIObject

```
public ODCIObject()
```

Methods

getORADDataFactory

```
public static oracle.sql.ORADDataFactory getORADDataFactory()
```

toDatum

```
public oracle.sql.Datum toDatum(java.sql.Connection c)  
throws java.sql.SQLException
```

create

```
public oracle.sql.ORADData create(oracle.sql.Datum d,  
int sqlType)  
throws java.sql.SQLException
```

getObjectSchema

```
public java.lang.String getObjectSchema()  
    throws java.sql.SQLException
```

setObjectSchema

```
public void setObjectSchema(java.lang.String ObjectSchema)  
    throws java.sql.SQLException
```

getObjectName

```
public java.lang.String getObjectName()  
    throws java.sql.SQLException
```

setObjectName

```
public void setObjectName(java.lang.String ObjectName)  
    throws java.sql.SQLException
```

ODCIObjectList

```
oracle.ODCI.ODCIObjectList  
public class ODCIObjectList
```

Fields

_SQL_NAME

```
public static final java.lang.String _SQL_NAME
```

_SQL_TYPECODE

```
public static final int _SQL_TYPECODE
```

Constructors

ODCIObjectList

```
public ODCIObjectList()
```

ODCIObjectList

```
public ODCIObjectList(ODCIObject[] a)
```

Methods

getORADDataFactory

```
public static oracle.sql.ORADataFactory getORADDataFactory()
```

toDatum

```
public oracle.sql.Datum toDatum(java.sql.Connection c)  
    throws java.sql.SQLException
```

create

```
public oracle.sql.ORAData create(oracle.sql.Datum d,  
                                int sqlType)  
    throws java.sql.SQLException
```

length

```
public int length()  
    throws java.sql.SQLException
```

getBaseType

```
public int getBaseType()  
    throws java.sql.SQLException
```

getBaseTypeName

```
public java.lang.String getBaseTypeName()  
    throws java.sql.SQLException
```

getDescriptor

```
public oracle.sql.ArrayDescriptor getDescriptor()  
    throws java.sql.SQLException
```

getArray

```
public ODCIObject[] getArray()  
    throws java.sql.SQLException
```

setArray

```
public void setArray(ODCIObject[] a)  
    throws java.sql.SQLException
```

getArray

```
public ODCIObject[] getArray(long index,  
                               int count)  
    throws java.sql.SQLException
```

setArray

```
public void setArray(ODCIObject[] a,  
                    long index)  
    throws java.sql.SQLException
```

getElement

```
public ODCIObject getElement(long index)  
    throws java.sql.SQLException
```

setElement

```
public void setElement(ODCIObject a,  
                      long index)  
    throws java.sql.SQLException
```

ODCIObjectRef

```
oracle.ODCI.ODCIObjectRef  
public class ODCIObjectRef
```

Fields

_SQL_BASETYPE

```
public static final java.lang.String _SQL_BASETYPE
```

_SQL_TYPECODE

```
public static final int _SQL_TYPECODE
```

Constructors

ODCIObjectRef

```
public ODCIObjectRef()
```

Methods

getORADDataFactory

```
public static oracle.sql.ORADDataFactory getORADDataFactory()
```

toDatum

```
public oracle.sql.Datum toDatum(java.sql.Connection c)  
    throws java.sql.SQLException
```

create

```
public oracle.sql.ORADData create(oracle.sql.Datum d,  
    int sqlType)  
    throws java.sql.SQLException
```

getValue

```
public ODCIObject getValue()  
    throws java.sql.SQLException
```

setValue

```
public void setValue(ODCIObject c)  
    throws java.sql.SQLException
```

ODCIPartInfo

```
oracle.ODCI.ODCIPartInfo  
public class ODCIPartInfo
```

Fields

_SQL_NAME

```
public static final java.lang.String _SQL_NAME
```

_SQL_TYPECODE

```
public static final int _SQL_TYPECODE
```

Constructors

ODCIPartInfo

```
public ODCIPartInfo()
```

Methods

getORADDataFactory

```
public static oracle.sql.ORADDataFactory getORADDataFactory()
```

toDatum

```
public oracle.sql.Datum toDatum(java.sql.Connection c)  
    throws java.sql.SQLException
```

create

```
public oracle.sql.ORADData create(oracle.sql.Datum d,  
    int sqlType)  
    throws java.sql.SQLException
```


getTablePartition

```
public java.lang.String getTablePartition()  
    throws java.sql.SQLException
```

setTablePartition

```
public void setTablePartition(java.lang.String TablePartition)  
    throws java.sql.SQLException
```

getIndexPartition

```
public java.lang.String getIndexPartition()  
    throws java.sql.SQLException
```

setIndexPartition

```
public void setIndexPartition(java.lang.String IndexPartition)  
    throws java.sql.SQLException
```

ODCIPartInfoRef

```
oracle.ODCI.ODCIPartInfoRef  
public class ODCIPartInfoRef
```

Fields

_SQL_BASETYPE

```
public static final java.lang.String _SQL_BASETYPE
```

_SQL_TYPECODE

```
public static final int _SQL_TYPECODE
```

Constructors

ODCIPartInfoRef

```
public ODCIPartInfoRef()
```

Methods

getORADDataFactory

```
public static oracle.sql.ORADDataFactory getORADDataFactory()
```

toDatum

```
public oracle.sql.Datum toDatum(java.sql.Connection c)  
    throws java.sql.SQLException
```

create

```
public oracle.sql.ORADData create(oracle.sql.Datum d,  
    int sqlType)  
    throws java.sql.SQLException
```

getValue

```
public ODCIPartInfo getValue()  
    throws java.sql.SQLException
```

setValue

```
public void setValue(ODCIPartInfo c)  
    throws java.sql.SQLException
```

ODCIPredInfo

```
oracle.ODCI.ODCIPredInfo  
public class ODCIPredInfo
```

Fields

_SQL_NAME

```
public static final java.lang.String _SQL_NAME
```

_SQL_TYPECODE

```
public static final int _SQL_TYPECODE
```

Constructors

ODCIPredInfo

```
public ODCIPredInfo()
```

Methods

getORADDataFactory

```
public static oracle.sql.ORADDataFactory getORADDataFactory()
```

toDatum

```
public oracle.sql.Datum toDatum(java.sql.Connection c)  
    throws java.sql.SQLException
```

create

```
public oracle.sql.ORADData create(oracle.sql.Datum d,  
    int sqlType)  
    throws java.sql.SQLException
```

getObjectSchema

```
public java.lang.String getObjectSchema()  
    throws java.sql.SQLException
```

setObjectSchema

```
public void setObjectSchema(java.lang.String ObjectSchema)  
    throws java.sql.SQLException
```

getObjectName

```
public java.lang.String getObjectName()  
    throws java.sql.SQLException
```

setObjectName

```
public void setObjectName(java.lang.String ObjectName)  
    throws java.sql.SQLException
```

getMethodName

```
public java.lang.String getMethodName()  
    throws java.sql.SQLException
```

setMethodName

```
public void setMethodName(java.lang.String MethodName)  
    throws java.sql.SQLException
```

getFlags

```
public java.math.BigDecimal getFlags()  
    throws java.sql.SQLException
```

setFlags

```
public void setFlags(java.math.BigDecimal Flags)  
    throws java.sql.SQLException
```

ODCIPredInfoRef

```
oracle.ODCI.ODCIPredInfoRef  
public class ODCIPredInfoRef
```

Fields

_SQL_BASETYPE

```
public static final java.lang.String _SQL_BASETYPE
```

_SQL_TYPECODE

```
public static final int _SQL_TYPECODE
```

Constructors

ODCIPredInfoRef

```
public ODCIPredInfoRef()
```

Methods

getORADDataFactory

```
public static oracle.sql.ORADDataFactory getORADDataFactory()
```

toDatum

```
public oracle.sql.Datum toDatum(java.sql.Connection c)  
    throws java.sql.SQLException
```

create

```
public oracle.sql.ORADData create(oracle.sql.Datum d,  
    int sqlType)  
    throws java.sql.SQLException
```

getValue

```
public ODCIPredInfo getValue()  
    throws java.sql.SQLException
```

setValue

```
public void setValue(ODCIPredInfo c)  
    throws java.sql.SQLException
```

ODCIQueryInfo

```
oracle.ODCI.ODCIQueryInfo  
public class ODCIQueryInfo
```

Fields

_SQL_NAME

```
public static final java.lang.String _SQL_NAME
```

_SQL_TYPECODE

```
public static final int _SQL_TYPECODE
```

Constructors

ODCIQueryInfo

```
public ODCIQueryInfo()
```

Methods

getORADDataFactory

```
public static oracle.sql.ORADDataFactory getORADDataFactory()
```

toDatum

```
public oracle.sql.Datum toDatum(java.sql.Connection c)  
    throws java.sql.SQLException
```

create

```
public oracle.sql.ORADData create(oracle.sql.Datum d,  
    int sqlType)  
    throws java.sql.SQLException
```


getFlags

```
public java.math.BigDecimal getFlags()  
    throws java.sql.SQLException
```

setFlags

```
public void setFlags(java.math.BigDecimal flags)  
    throws java.sql.SQLException
```

getAncOps

```
public ODCIObjectList getAncOps()  
    throws java.sql.SQLException
```

setAncOps

```
public void setAncOps(ODCIObjectList ancOps)  
    throws java.sql.SQLException
```

ODCIQueryInfoRef

```
oracle.ODCI.ODCIQueryInfoRef  
public class ODCIQueryInfoRef
```

Fields

_SQL_BASETYPE

```
public static final java.lang.String _SQL_BASETYPE
```

_SQL_TYPECODE

```
public static final int _SQL_TYPECODE
```

Constructors

ODCIQueryInfoRef

```
public ODCIQueryInfoRef()
```

Methods

getORADDataFactory

```
public static oracle.sql.ORADDataFactory getORADDataFactory()
```

toDatum

```
public oracle.sql.Datum toDatum(java.sql.Connection c)  
    throws java.sql.SQLException
```

create

```
public oracle.sql.ORADData create(oracle.sql.Datum d,  
    int sqlType)  
    throws java.sql.SQLException
```

getValue

```
public ODCIQueryInfo getValue()  
    throws java.sql.SQLException
```

setValue

```
public void setValue(ODCIQueryInfo c)  
    throws java.sql.SQLException
```

ODCIRidList

```
oracle.ODCI.ODCIRidList  
public class ODCIRidList
```

Fields

_SQL_NAME

```
public static final java.lang.String _SQL_NAME
```

_SQL_TYPECODE

```
public static final int _SQL_TYPECODE
```

Constructors

ODCIRidList

```
public ODCIRidList()
```

ODCIRidList

```
public ODCIRidList(java.lang.String[] a)
```

Methods

getORADDataFactory

```
public static oracle.sql.ORADDataFactory getORADDataFactory()
```

toDatum

```
public oracle.sql.Datum toDatum(java.sql.Connection c)  
throws java.sql.SQLException
```

create

```
public oracle.sql.ORADATA create(oracle.sql.Datum d,  
                                int sqlType)  
    throws java.sql.SQLException
```

length

```
public int length()  
    throws java.sql.SQLException
```

getBaseType

```
public int getBaseType()  
    throws java.sql.SQLException
```

getBaseTypeName

```
public java.lang.String getBaseTypeName()  
    throws java.sql.SQLException
```

getDescriptor

```
public oracle.sql.ArrayDescriptor getDescriptor()  
    throws java.sql.SQLException
```

getArray

```
public java.lang.String[] getArray()  
    throws java.sql.SQLException
```

setArray

```
public void setArray(java.lang.String[] a)  
    throws java.sql.SQLException
```

getArray

```
public java.lang.String[] getArray(long index,  
                                    int count)  
    throws java.sql.SQLException
```

setArray

```
public void setArray(java.lang.String[] a,  
                    long index)  
    throws java.sql.SQLException
```

getElement

```
public java.lang.String getElement(long index)  
    throws java.sql.SQLException
```

setElement

```
public void setElement(java.lang.String a,  
                      long index)  
    throws java.sql.SQLException
```

ODCIStatsOptions

```
oracle.ODCI.ODCIStatsOptions  
public class ODCIStatsOptions
```

Fields

_SQL_NAME

```
public static final java.lang.String _SQL_NAME
```

_SQL_TYPECODE

```
public static final int _SQL_TYPECODE
```

Constructors

ODCIStatsOptions

```
public ODCIStatsOptions()
```

Methods

getORADataFactory

```
public static oracle.sql.ORADataFactory getORADataFactory()
```

toDatum

```
public oracle.sql.Datum toDatum(java.sql.Connection c)  
throws java.sql.SQLException
```

create

```
public oracle.sql.ORAData create(oracle.sql.Datum d,  
int sqlType)  
throws java.sql.SQLException
```

getSample

```
public java.math.BigDecimal getSample()  
    throws java.sql.SQLException
```

setSample

```
public void setSample(java.math.BigDecimal Sample)  
    throws java.sql.SQLException
```

getOptions

```
public java.math.BigDecimal getOptions()  
    throws java.sql.SQLException
```

setOptions

```
public void setOptions(java.math.BigDecimal Options)  
    throws java.sql.SQLException
```

getFlags

```
public java.math.BigDecimal getFlags()  
    throws java.sql.SQLException
```

setFlags

```
public void setFlags(java.math.BigDecimal Flags)  
    throws java.sql.SQLException
```

ODCIStatsOptionsRef

```
oracle.ODCI.ODCIStatsOptionsRef  
public class ODCIStatsOptionsRef
```

Fields

_SQL_BASETYPE

```
public static final java.lang.String _SQL_BASETYPE
```

_SQL_TYPECODE

```
public static final int _SQL_TYPECODE
```

Constructors

ODCIStatsOptionsRef

```
public ODCIStatsOptionsRef()
```

Methods

getORADataFactory

```
public static oracle.sql.ORADataFactory getORADataFactory()
```

toDatum

```
public oracle.sql.Datum toDatum(java.sql.Connection c)  
    throws java.sql.SQLException
```

create

```
public oracle.sql.ORAData create(oracle.sql.Datum d,  
    int sqlType)  
    throws java.sql.SQLException
```

getValue

```
public ODCIStatsOptions getValue()  
    throws java.sql.SQLException
```

setValue

```
public void setValue(ODCIStatsOptions c)  
    throws java.sql.SQLException
```

Part II

Java Packages for Oracle9i XDK for Java

This part describes Java packages contained in the Oracle XDK for Java. The Oracle XML Developer's Kits (XDKs) contain the basic building blocks for reading, manipulating, transforming, and viewing XML documents.

This part contains these chapters:

- [Chapter 6, "Package oracle.xml.classgen"](#)
- [Chapter 11, "Package oracle.xml.parser.v2"](#)
- [Chapter 7, "Package oracle.XML.parser.schema"](#)
- [Chapter 8, "Package oracle.xml.sql.dml"](#)
- [Chapter 9, "Package oracle.xml.sql.query"](#)
- [Chapter 10, "Package oracle.xml.util"](#)

The production Oracle XDKs are fully supported and come with a commercial redistribution license. The production libraries are regularly updated on the OTN Web site. For more information, refer to the XDK for Java on the Oracle Technology Network (OTN) Web site as follows:

- Oracle XDK Home at
<http://otn.oracle.com/tech/xml/xdkhome.html>
- Oracle XML Developer's Kit for Java at
http://otn.oracle.com/tech/xml/xdk_java/content.html
- Oracle XML Developer's Kit for JavaBeans at
http://otn.oracle.com/tech/xml/xdk_jbeans/content.html

Package oracle.xml.classgen

This chapter describes package `oracle.xml.classgen`, which contains the classes for the XML Class Generator in the Oracle9i XDK for Java. A class generator is a utility that accepts an input file and creates a set of output classes that have corresponding functionality. For the XML Class Generator, the input file is a DTD and the output is a series of classes that can be used to create XML documents conforming with the DTD.

This chapter contains these sections:

- [Package oracle.xml.classgen Description](#)
- [Package oracle.xml.classgen Summary](#)

Package oracle.xml.classgen Description

The XML Class Generator for Java creates Java source files from an XML DTD or XML Schema. This is useful when an application wants to send an XML message to another application based on an agreed-upon DTD or Schema, or as the back end of a Web form to construct an XML document. Using these classes, Java applications can construct, validate, and print XML documents that comply with the input DTD or Schema. The Class Generator works in conjunction with the Oracle XML Parser for Java, which parses the DTD or Schema and passes the parsed document to the class generator.

For more information, refer to Oracle resources for the XDK for Java on the Oracle Technology Network (OTN) Web site.

See Also:

- *Oracle9i XML Developer's Kits Guide - XDK*
- *Oracle9i XML API Reference - XDK and Oracle XML DB*

Package oracle.xml.classgen Summary

Table 6–1 Summary of oracle.xml.classgen

Name	Description
CGDocument Class	Serves as the base document class for the DTD class Generator generated classes
CGNode Class	Serves as the base class for the classes corresponding to the nodes of XML document generated by the DTD class generator.
CGXSDElement Class	Serves as the base class for the all the generated classes corresponding to the XML Schema generated by Schema Class Generator.
DTDClassGenerator Class	Generates the data binding classes corresponding to a DTD or an XML file based on a DTD.
InvalidContentException Class	Defines the Exception thrown by DTD ClassGenerator and Schema Class Generator.
oracg Class	Provides a command-line interface to generate java classes corresponding to the DTD or XML
SchemaClassGenerator Class	Generates the classes corresponding to an XML Schema.

CGDocument Class

Description of CGDocument

This class serves as the base document class for the DTD class Generator generated classes.

Syntax of CGDocument

```
public abstract class CGDocument extends oracle.xml.classgen.CGNode implements
java.io.Externalizable
```

```
oracle.xml.classgen.CGNode
|
+--oracle.xml.classgen.CGDocument
```

Implemented Interfaces of CGDocument

java.io.Externalizable, java.io.Serializable

Methods of CGDocument

Table 6–2 Summary of Methods of CGDocument

Method	Description
CGDocument() , on page 6-4	Constructor for the root element of the DTD.
print() , on page 6-5	Prints the constructed XML document.
readExternal() , on page 6-5	Reads the compressed stream and creates the object corresponding to the root element.

CGDocument()

Description

Constructor for the root element of the DTD.

Syntax

```
protected CGDocument( java.lang.String doctype,
                       oracle.xml.parser.v2.DTD dtd );
```


Parameters

doctype	Name of the root Element of the DTD.
dtd	The DTD used to generate the classes.

print()

Description

Prints the constructed XML Document. Throws `InvalidContentException` if the document's content does not match the grammar specified by DTD; the validation mode should be set to `TRUE`. See also `setValidationMode()` in [DTDClassGenerator Class](#). The options are described in the following table.

Syntax	Description
<code>protected void print(java.io.OutputStream out);</code>	Prints the constructed XML Document to output stream.
<code>protected void print(java.io.OutputStream out, java.lang.String enc);</code>	Prints the constructed XML Document to output stream with user-defined encoding.

Parameters

out	Output stream to which the document will be printed.
enc	Encoding of the output stream.

readExternal()

Description

Reads the compressed stream and creates the object corresponding to the root element. Used for instantiating the generated classes with XML instance document.

Syntax

```
protected void readExternal( java.io.ObjectInput inArg,
                             oracle.xml.comp.CXMLContext cxmlContext);
```

Parameters

in	ObjectInput stream passed to read the compressed stream
----	---

cxmlContext

The context of the compressed stream

CGNode Class

Description of CGNode

This class serves as the base class for the classes corresponding to the nodes of XML document generated by the DTD class generator.

Syntax of CGNode

```
public abstract class CGNode
```

```
oracle.xml.classgen.CGNode
```

Direct Subclasses of CGNode

```
CGDocument
```

Fields of CGNode

isValidating

```
protected boolean isValidating
```

Boolean to indicate the validating mode.

Methods of CGNode

Table 6–3 Summary of Methods of CGNode

Method	Description
CGNode() , on page 6-8	Constructor for the Elements of the DOM Tree.
addCDATASection() , on page 6-9	Adds CDATA Section to the Element.
addData() , on page 6-9	Adds PCDATA to the element node.
addNode() , on page 6-9	Adds a node as a child to the element.
deleteData() , on page 6-10	Deletes PCDATA from an element node.
getAttribute() , on page 6-10	Retrieves the value of the attribute.
getCGDocument() , on page 6-10	Retrieves the base document.
getData() , on page 6-11	Retrieves the PCDATA of the element.

Table 6–3 (Cont.) Summary of Methods of CGNode

Method	Description
getDTDNode() , on page 6-11	Retrieves the static DTD from the base document.
getElementNode() , on page 6-11	Retrieves the XMLElement node corresponding to this CGNode.
getNode() , on page 6-11	Retrieves the CGNode which is one of the children of the element corresponding to this node whose name matches the input string.
readExternal() , on page 6-12	Reads the compressed stream and instantiates the corresponding node.
setAttribute() , on page 6-12	Sets the value of the attribute.
setDocument() , on page 6-12	Sets the base document.
setElementNode() , on page 6-13	Sets the XMLElement node corresponding to this CGNode.
storeID() , on page 6-13	Stores this value of ID identifier.
storeIDREF() , on page 6-13	Stores this value for an IDREF identifier.
validateContent() , on page 6-14	Checks if the content of the element is valid according to the Content Model specified in DTD.
validEntity() , on page 6-14	Checks if the ENTITY identifier is valid.
validID() , on page 6-14	Checks if the ID identifier is valid.
validNMTOKEN() , on page 6-15	Checks if the NMTOKEN identifier is valid.
writeExternal() , on page 6-15	Writes the compressed stream corresponding to this node.

CGNode()

Description

Constructor for the Elements of the DOM Tree.

Syntax

```
protected CGNode( java.lang.String elementName );
```

Parameters

elementName Name of the element.

addCDATASection()

Description

Adds CDATA Section to the Element. Throws `InvalidContentException` if `theData` has illegal characters; validation must be set to `TRUE`. See also [setValidationMode\(\)](#) in [DTDClassGenerator Class](#).

Syntax

```
protected void addCDATASection( java.lang.String theData);
```

Parameters

`theData` Text to be added as CDATA Section to the element.

addData()

Description

Adds PCDATA to the element node. Throws `InvalidContentException` if `theData` has illegal characters; validation must be set to `TRUE`. See also [setValidationMode\(\)](#) in [DTDClassGenerator Class](#).

Syntax

```
protected void addData( java.lang.String theData);
```

Parameters

`theData` Text to be added a to the element.

addNode()

Description

Adds a node as a child to the element. Throws `InvalidContentException` if `theData` has illegal characters; validation must be set to `TRUE`. See also [setValidationMode\(\)](#) in [DTDClassGenerator Class](#).

Syntax

```
protected void addNode( CGNode theNode);
```

Parameters

theNode The node to be added as child.

deleteData()

Description

Deletes PCDATA from the element node. Throws `InvalidContentException` if `theData` has illegal characters; validation must be set to `TRUE`. See also [setValidationMode\(\)](#) in [DTDClassGenerator Class](#).

Syntax

```
protected void deleteData( java.lang.String theData);
```

Parameters

theNode Text to be deleted from an element.

getAttribute()

Description

Returns the value of the attribute.

Syntax

```
protected java.lang.String getAttribute( java.lang.String attName);
```

Parameters

attName Name of the attribute.

getCGDocument()

Description

Gets the base document (root Element).

Syntax

```
protected CGDocument getCGDocument();
```

getData()

Description

Gets the PCDATA of the Element. Throws `InvalidContentException` if the data is not present.

Syntax

```
protected java.lang.String getData();
```

getDTDNode()

Description

Retrieves the static DTD from the base `CGDocument`.

Syntax

```
protected abstract oracle.xml.parser.v2.DTD getDTDNode();
```

getElementNode()

Description

Retrieves the `XMLElement` node corresponding to this `CGNode`.

Syntax

```
protected oracle.xml.parser.v2.XMLElement getElementNode();
```

getNode()

Description

Retrieves the `CGNode` which is one of the children of the element corresponding to the node whose name matches the input string.

Syntax

```
protected java.lang.Object getNode(java.lang.String theNode);
```

Parameters

`theNode` The name of the string corresponding to the `CGNode` returned.

readExternal()

Description

Reads the compressed stream and instantiate the corresponding node. Throws the following exceptions:

IOException	When an IO Error occurs
ClassNotFoundException	When the corresponding class could not be instantiated.

Syntax

```
protected void readExternal(oracle.xml.io.XMLObjectInput in,  
                           oracle.xml.comp.CXMLContext cxmlContext)
```

Parameters

in	The XMLObjectInput stream that is used to read the compressed stream.
cxmlContext	The context of the compressed stream.

setAttribute()

Description

Sets the value of the attribute.

Syntax

```
protected void setAttribute(java.lang.String attName,  
                           java.lang.String value);
```

Parameters

attName	Name of the attribute.
value	Value of the attribute.

setDocument()

Description

Sets the base document (root element).

Syntax

```
public void setDocument( CGDocument d);
```

Parameters

d Base CGDocument.

setElementNode()**Description**

Sets the XMLElement node corresponding to this CGNode.

Syntax

```
protected void setElementNode(oracle.xml.parser.v2.XMLElement node);
```

Parameters

node The XMLElement.

storeID()**Description**

Store this value for an ID identifier, which can be verified with IDREF values.

Syntax

```
protected void storeID(java.lang.String attName,  
                          java.lang.String id);
```

Parameters

attName Name of the ID attribute.

id Value of the ID

storeIDREF()**Description**

Store this value for an IDREF identifier, which can be verified by the corresponding ID.

Syntax

```
protected void storeIDREF( java.lang.String attName,  
                           java.lang.String idref);
```

Parameters

attName	Name of the IDREF attribute.
idref	Value of the IDREF

validateContent()

Description

Checks if the content of the element is valid according to the Content Model specified in DTD.

Syntax

```
protected void validateContent();
```

validEntity()

Description

Checks if the ENTITY identifier is valid. Returns `TRUE` if ENTITY is valid, `FALSE` otherwise.

Syntax

```
protected boolean validEntity( java.lang.String entity);
```

Parameters

name	Value of the ENTITY attribute
------	-------------------------------

validID()

Description

Checks if the ID identifier is valid. Returns `TRUE` if ID is valid, `FALSE` otherwise.

Syntax

```
protected boolean validID( java.lang.String name);
```

Parameters

name Value of the ID attribute.

validNMTOKEN()

Description

Checks if the NMTOKEN identifier is valid. Returns `TRUE` if NMTOKEN is valid, `FALSE` otherwise.

Syntax

```
protected boolean validNMTOKEN( java.lang.String name);
```

Parameters

name Value of the NMTOKEN attribute.

writeExternal()

Description

Writes the compressed stream corresponding to this node.

Syntax

```
protected void writeExternal( oracle.xml.io.XMLObjectOutput out,  
                             oracle.xml.comp.CXMLContext cxmlContext);
```

Parameters

out ObjectOutput stream to write the compressed data.

cxmlContext The context of the compressed stream.

CGXSDElement Class

Description of CGXSDElement

This class serves as the base class for all the generated classes corresponding to the XML Schema generated by Schema Class Generator.

Syntax of CGXSDElement

```
public abstract class CGXSDElement extends java.lang.Object
```

```
java.lang.Object
|
+--oracle.xml.classgen.CGXSDElement
```

Fields of CGXSDElement

Table 6–4 *Fields of ElementDecl*

Field	Syntax	Description
type	protected java.lang.Object type	Type information of a node

Methods of CGXSDElement

Table 6–5 *Summary of Methods of CGXSDElement*

Method	Description
CGXSDElement() , on page 6-17	Default constructor.
addAttribute() , on page 6-17	Adds the attribute of a given node to the hashtable.
addElement() , on page 6-17	Adds the local elements of an element node to the vector corresponding to the elements.
getAttributes() , on page 6-18	Returns the attributes as a hashtable of attribute names and values.
getChildElements() , on page 6-18	Retrieves the vector of all local elements.
getNodeValue() , on page 6-18	Returns the value of the node.
print() , on page 6-18	Prints an element node.
printAttributes() , on page 6-19	Prints an attribute node.

Table 6–5 Summary of Methods of CGXSDElement (Cont.)

Method	Description
setNodeValue() , on page 6-19	Sets the node value of an element.

CGXSDElement()

Description

Default constructor.

Syntax

```
public CGXSDElement();
```

addAttribute()

Description

Adds the attribute of a given node to the hashtable.

Syntax

```
protected void addAttribute(java.lang.String attName,  
                             java.lang.Object attValue);
```

Parameters

attName	The attribute name.
attValue	The attribute value.

addElement()

Description

Adds the local elements of an element node to the vector correspondig to the elements.

Syntax

```
protected void addElement( java.lang.Object elem);
```

Parameters

elem The object which needs to be added.

getAttributes()

Description

Returns the attributes as a hashtable of attribute names and values.

Syntax

```
public java.util.Hashtable getAttributes();
```

getChildElements()

Description

Retrieves the vector of all local elements.

Syntax

```
public java.util.Vector getChildElements();
```

getNodeValue()

Description

Returns the value of the node.

Syntax

```
public java.lang.String getNodeValue();
```

print()

Description

Prints an element node. Throws an `IOException` if not able to print to the output stream

Syntax

```
public void print( oracle.xml.parser.v2.XMLOutputStream out );
```

Parameters

out The XMLObjectOutput stream to which the output is printed.

printAttributes()

Description

Prints an attribute node. Throws an `IOException` if not able to print to the `XMLObjectOutput` stream.

Syntax

```
public void printAttributes( oracle.xml.parser.v2.XMLOutputStream out,  
                             java.lang.String name,  
                             java.lang.String namespace);
```

Parameters

out The XMLObjectOutput stream to which the output is printed.
name The attribute name
namespace The namespace

setNodeValue()

Description

Sets the node value of an element.

Syntax

```
protected void setNodeValue( java.lang.String value);
```

Parameters

value The node vale.

DTDClassGenerator Class

Description of DTDClassGenerator

Generates the data binding classes corresponding to a DTD or an XML file based on a DTD.

Syntax of DTDClassGenerator

```
public class DTDClassGenerator extends java.lang.Object
```

```
java.lang.Object  
|  
+--oracle.xml.classgen.DTDClassGenerator
```

Methods of DTDClassGenerator

Table 6–6 Summary of Methods of DTDClassGenerator

Method	Description
DTDClassGenerator() , on page 6-20	Default constructor for DTDClassGenerator.
generate() , on page 6-21	Traverses the DTD with element <code>doctype</code> as root and generates Java classes.
setGenerateComments() , on page 6-21	Sets the switch to determine whether to generate java doc comments for the generated classes.
setJavaPackage() , on page 6-21	Sets the package for the classes generated.
setOutputDirectory() , on page 6-22	Sets the output directory where the java source code for the DTD is generated.
setSerializationMode() , on page 6-22	Sets the switch to determine if the DTD should be saved as a serialized object or as text file.
setValidationMode() , on page 6-23	Sets the switch to determine whether the classes generated should validate the XML document.

DTDClassGenerator()

Description

Default constructor for DTDClassGenerator.

Syntax

```
public DTDClassGenerator();
```

generate()**Description**

Traverses the DTD with element `doctype` as root and generates Java classes.

Syntax

```
public void generate( oracle.xml.parser.v2.DTD dtd,  
                    java.lang.String doctype);
```

Parameters

DTD	The DTD used to generate the classes.
doctype	Name of the root element.

setGenerateComments()**Description**

Sets the switch to determine whether to generate java doc comments for the generated classes. Default value is TRUE.

Syntax

```
public void setGenerateComments(boolean comments);
```

Parameters

comments	The boolean flag for turning on/off the java doc comment generation.
----------	--

setJavaPackage()**Description**

Sets the package for the classes generated. Default - no package set.

Syntax

```
public void setJavaPackage( java.util.Vector packageName);
```

Parameters

packageName Name of the package.

setOutputDirectory()

Description

Sets the output directory where the java source code for the DTD is generated. Default value is the current directory.

Syntax

```
public void setOutputDirectory( java.lang.String dir);
```

Parameters

dir Output directory.

setSerializationMode()

Description

Sets the switch to determine if the DTD should be saved as a serialized object or as text file. Serializing the DTD improves the performance when the generated classes are used to author XML files.

Syntax

```
public void setSerializationMode( boolean yes);
```

Parameters

yes The boolean flag for turning on/off saving of DTD as serialized object (TRUE). Default is saving as a text file (FALSE).

setValidationMode()

Description

Sets the switch to determine whether the classes generated should validate the XML document being constructed. Default value is `TRUE`.

Syntax

```
public void setValidationMode( boolean yes );
```

Parameters

yes	The boolean flag for turning on/off validation of XML document. Default is <code>TRUE</code> .
-----	--

InvalidContentException Class

Description of InvalidContentException

Defines the Exception thrown by DTD ClassGenerator and Schema Class Generator.

Syntax of InvalidContentException

```
public class InvalidContentException extends java.lang.Exception
```

```
java.lang.Object
|
+--java.lang.Throwable
|
+--java.lang.Exception
|
+--oracle.xml.classgen.InvalidContentException
```

Implemented Interfaces of InvalidContentException

```
java.io.Serializable
```

Methods of InvalidContentException

InvalidContentException()

Description

Constructor. The options are described in the following table.

Syntax	Description
<code>public InvalidContentException();</code>	Default constructor.
<code>public InvalidContentException(java.lang.String s);</code>	This constructor takes an input String of information about the exception.

Parameters

`s` String that contains the information about the exception.

oracg Class

Description of oracg

Provides a command-line interface to generate java classes corresponding to the DTD or XML

Syntax of oracg

```
public class oracg extends java.lang.Object
```

```
java.lang.Object
|
+--oracle.xml.classgen.oracg
```

Command-line options of oracg

Table 6–7 *Command-line options of oracg*

Option	Description
-help	Prints the help message text.
-version	Prints the release version.
-dtd [-root <rootNmae>]	The input file is a DTD file or DTD based XML file.
-schema <Schema File>	The input file is a Schema file or Schema based XML file.
-outputDir <Output Dir>	The directory name where java source is generated.
-package <Package Name>	The package name(s) of the generated java classes.
-comment	Generate comments for the generated java source code.

SchemaClassGenerator Class

Description of SchemaClassGenerator

This class generates the classes corresponding to an XML Schema.

Syntax of SchemaClassGenerator

```
public class SchemaClassGenerator extends java.lang.Object
```

```
java.lang.Object
|
+--oracle.xml.classgen.SchemaClassGenerator
```

Methods of SchemaClassGenerator

Table 6–8 Summary of Methods of SchemaClassGenerator

Method	Description
SchemaClassGenerator() , on page 6-26	Constructor.
generate() , on page 6-27	Generates the Schema classes corresponding to top level elements, simpleType elements and complexType elements.
setGenerateComments() , on page 6-27	Sets the switch to determine whether to generate java doc comments.
setJavaPackage() , on page 6-28	Assigns a user-defined Java package name for each namespace.
setOutputDirectory() , on page 6-28	Sets the output directory where the java source code for the Schema class are generated.

SchemaClassGenerator()

Description

Constructor. The options are described in the following table.

Syntax	Description
<code>public SchemaClassGenerator();</code>	Default empty constructor for Schema Class Generator.

Syntax	Description
<pre>public SchemaClassGenerator(java.lang.String fileName)</pre>	This constructor takes an input String containing the description of the XML Schema.

Parameters

fileName The input XML Schema.

generate()

Description

Generates the Schema classes corresponding to top level elements, simpleType elements and complexType elements by calling createSchemaClass() on each of these nodes.

Syntax

```
public void generate( oracle.xml.parser.schema.XMLSchema schema );
```

Parameters

XML Schema object.

setGenerateComments()

Description

Sets the switch to determine whether to generate java doc comments. TRUE by default.

Syntax

```
public void setGenerateComments(boolean comments)
```

Parameters

comments Turns on/off the java doc comment generation. TRUE by default.

setJavaPackage()

Description

Assigns user-defined Java package name for each namespace. The Namespaces defined in the schema are queried, and their number should match the number of package names provided by the user; otherwise, an error is thrown.

Syntax

```
public void setJavaPackage( oracle.xml.parser.schema.XMLSchema schema,  
                           java.util.Vector pkgName);
```

Parameters

schema	The XML Schema
pkgName	A vector containing user defined package names given through command line.

setOutputDirectory()

Description

Sets the output directory where the java source code for the Schema class are generated. The current directory is the default.

Syntax

```
public void setOutputDirectory( java.lang.String dir);
```

Parameters

dir	The output directory.
-----	-----------------------

Package `oracle.XML.parser.schema`

This chapter describes the `oracle.XML.parser.schema` package. The classes contained in package `oracle.XML.parser.schema` implement the Oracle XML Schema Processor for Java.

In addition to the class reference, this chapter contains these sections:

- [Package `oracle.XML.parser.schema` Description](#)
- [Package `oracle.XML.parser.schema` Summary](#)

Package oracle.XML.parser.schema Description

The classes contained in package oracle.XML.parser.schema implement the Oracle XML Schema Processor for Java and support the World Wide Web Consortium (W3C) XML Schema specification.

XML Schema can be used to define a class of XML documents. The term *instance document* describes an XML document that conforms to a particular XML Schema definition or XSD. This document assumes familiarity with the W3C recommendations for XML Schema. The W3C recommendations can be found at <http://www.w3.org/>.

Features in Oracle XML Schema Processor for Java

The Oracle XML Schema Processor for Java is built on the Oracle XML Parser for Java v2 and includes these features:

- Built-In Data Types: XML Schema specifies a set of built-in datatypes. Some are defined and called primitive datatypes and they form the basis of the type system.
- Support for Simple API for XML (SAX) processing: streaming, constant memory usage, and linear processing time. SAX is an event-based API between an XML parser and an XML application. The object-based interface is supplied by the DOM.
- Fully supports the W3C.org XML Schema specifications of the candidate recommendation published October 24, 2000 and the final recommendation published May 2, 2001.

What's Needed to Run XML Schema Processor for Java

To run XML Schema Processor for Java, you need an operating system that supports Java 1.1.x or above and the JDK 1.1.x. or above.

See Also:

- <http://www.w3.org/>
- *Oracle9i XML Developer's Kits Guide - XDK*
- *Oracle9i XML API Reference - XDK and Oracle XML DB*

Package oracle.XML.parser.schema Summary

The classes contained in package oracle.xml.parser.schema implement support for XML Schema and for XML Schema Definition.

Table 7–1 Summary of oracle.XML.parser.schema Classes and Interfaces

Class/Interface	Description
XMLSchema Class	Sets top-level XMLSchema document declarations, definitions, schema location, and schema target namespace.
XMLSchemaNode	Sets top-level XMLSchema document declarations and definitions plus schema location and schema target namespace.
XSDAttribute Class	Represents the complexType attribute group.
XSDBuilder Class	Builds an XMLSchema object from XMLSchema document.
XSDComplexType Class	Manages XML Schema Definition (XSD) for complexType for the XML document.
XSDConstants Interface	Implements the XSDConstants interface.
XSDConstrainingFacet Class	Implements XSDTypeConstants.
XSDDataValue Class	Implements XSDTypeConstants.
XSDElement Class	Presents XMLSchema Definition for element.
XSDException	Indicates that an exception occurred during XMLSchema validation.
XSDGroup Class	Presents the complexType model group.
XSDIdentity Class	Presents identity parameters for the XSD.
XSDNode Class	Root class for most of XSD classes.
XSDSimpleType Class	Implements XSDTypeConstants to derive a type.
XSDTypeConstants Interface	Implements the interface for XSDTypeConstants.
XSDValidator Class	Validates an <i>instance</i> XML document against an XMLSchema.

XMLSchema Class

Description

This class contains a set of XML Schemas for different target namespaces. They are used by XSDParser for validation of instance XML documents, and by XSDBuilder as imported schemas.

Syntax

```
public class XMLSchema extends oracle.xml.parser.schema.XSDNode
```

```
oracle.xml.parser.schema.XSDNode
|
+--oracle.xml.parser.schema.XMLSchema
```

Methods

Table 7-2 Methods of XMLSchemaNode

Method	Description
XMLSchema()	Class constructor.
getAllTargetNS()	Get all target namespaces for this set of Schemas.
getSchemaByTargetNS()	Get Schema node for the given namespace.
getSchemaTargetNS()	Get the top level Schema's target namespace.
getXMLSchemaNodeTable()	Get Schema node hashtable.
getXMLSchemaURLS()	Get the Schema URL.
printSchema()	Print the given Schema.

XMLSchema()

Description

XMLSchema class constructor.

Syntax

Table 7–3 *Versions of XMLSchema Constructor*

Syntax

```
public XMLSchema() throws XSDException
```

```
public XMLSchema(int n) throws XSDException
```

Parameters

Table 7–4 *Parameters of XMLSchema Constructor*

Parameter	Description
n	Initial size of schemanode set

getAllTargetNS()

Description

Get all the Target Name spaces defined in the Schema.

Syntax

```
public java.lang.String[] getAllTargetNS()
```

getSchemaByTargetNS()

Description

Get schemaNode for the given namespace.

Syntax

```
public XMLSchemaNode getSchemaByTargetNS(java.lang.String namespace)
```

Parameters

namespace - Target namespace of the required schema

Returns

XMLSchemaNode

getSchemaTargetNS()

Description

Get the top level schema's target Namespace. In case there are more than one top level schema, the last one being built is returned.

Syntax

```
public java.lang.String getSchemaTargetNS()
```

getXMLSchemaNodeTable()

Description

Get XMLSchemaNode table

Syntax

```
public java.util.Hashtable getXMLSchemaNodeTable()
```

Returns

Hashtable

getXMLSchemaURLS()

Description

Get XMLSchema URLs

Syntax

```
public java.lang.String[] getXMLSchemaURLS()
```

Returns

Array of schema URLs

printSchema()

Description

Print schema information.

Syntax

Table 7–5 *Versions of printSchema()*

Syntax	Description
<code>public void printSchema()</code>	Prints schema information
<code>public void printSchema(boolean all)</code>	Prints schema information, including buildins

Parameters

Table 7–6 *Parameters of printSchema*

Parameter	Description
<code>all</code>	Flag to indicate that all information, including built-ins, should be printed.

XMLSchemaNode

Description of XMLSchemaNode

XMLSchemaNode class sets top-level XMLSchema document declarations and definitions plus schema location and schema target namespace. XMLSchema objects are created by XSDBuilder as a result of processing XMLSchema documents.

Syntax of XMLSchemaNode

```
public class XMLSchemaNode extends oracle.xml.parser.schema.XSDNode
```

```
oracle.xml.parser.schema.XSDNode
|
+--oracle.xml.parser.schema.XMLSchemaNode
```

Methods of XMLSchemaNode

Table 7-7 Summary of Methods of XMLSchemaNode

Method	Description
XMLSchemaNode()	XNLSchema constructor
getAttributeDeclarations()	Get all the top level attributes in the schema
getComplexTypeSet()	Get all the top level attributes in the schema
getComplexTypeSet()	Get all the top level complexType elements in the schema
getComplexTypeTable()	Get the complexType definitions
getElementSet()	Get all the top level elements in the schema
getSimpleTypeSet()	Get all the top level simpleType elements in the schema
getSimpleTypeTable()	Get the simple type definitions
getTargetNS()	Get targetNS of the schema
getTypeDefinitionTable()	Get the type definitions

XMLSchemaNode()

Description

XNLSchema constructor.

Syntax

```
public XMLSchemaNode()
```

getAttributeDeclarations()**Description**

Get all the top level attributes in the schema

Syntax

```
public XSDAttribute getAttributeDeclarations()
```

Returns

an array of top level attribute definitions

getComplexTypeSet()**Description**

Get all the top level complexType elements in the schema

Syntax

```
public XSDNode getComplexTypeSet()
```

Returns

an array of top level complexType nodes

getComplexTypeTable()**Description**

Get the complexType definitions

Syntax

```
public java.util.Hashtable getComplexTypeTable()
```

Returns

Hashtable of complexTypes

getElementSet()

Description

Get all the top level elements in the schema

Syntax

```
public XSDNode getElementSet()
```

Returns

an array of top level XSDNode elements

getSimpleTypeSet()

Description

Get all the top level simpleType elements in the schema

Syntax

```
public XSDNode getSimpleTypeSet()
```

Returns

an array of top level simpleType nodes

getSimpleTypeTable()

Description

Get the simple type definitions

Syntax

```
public java.util.Hashtable getSimpleTypeTable()
```

Returns

Hashtable of simpleTypes

getTargetNS()

Description

Get targetNS of the schema. Overrides XSDNode.getTargetNS() in class XSDNode.

Syntax

```
public java.lang.String getTargetNS()
```

Returns

value targetNS

getTypeDefinitionTable()**Description**

Get the type definitions

Syntax

```
public java.util.Hashtable getTypeDefinitionTable()
```

Returns

Hashtable of type definitions

XSDAttribute Class

Description of XSDAttribute

XSDAttribute class. Represents the complexType attribute group for XMLSchema.

Syntax of XSDAttribute

```
public class XSDAttribute extends oracle.xml.parser.schema.XSDNode
```

```
oracle.xml.parser.schema.XSDNode  
|  
+--oracle.xml.parser.schema.XSDAttribute
```

Methods of XSDAttribute

Table 7-8 Summary of Methods of XSDAttribute

Method	Description
<code>getDefaultVal()</code>	Get the value of 'default' attr in case of element, and the value of 'value' attr based on 'use' attribute
<code>getFixedVal()</code>	Get the value of 'fixed' attr in case of element, and the value of 'value' attr based on 'use' attribute
<code>getName()</code>	Get the name of the node
<code>getRefLocalname()</code>	Get the local name of the resolved 'ref' attribute
<code>getRefNamespace()</code>	Get the namespace of the resolved 'ref' attribute
<code>getRefState()</code>	Get refState
<code>getTargetNS()</code>	Get target namespace
<code>getType()</code>	Get the node type
<code>isRequired()</code>	Check if the attribute is required

getDefaultVal()

Description

Get the value of 'default' attr in case of element, and the value of 'value' attr based on 'use' attribute

Syntax

```
public java.lang.String getDefaultVal()
```

Returns

defaultVal

getFixedVal()**Description**

Get the value of 'fixed' attr in case of element, and the value of 'value' attr based on 'use' attribute

Syntax

```
public java.lang.String getFixedVal()
```

Returns

defaultVal

getName()**Description**

Get the name of the node. Overrides `XSDNode.getName()` in class `XSDNode`

Syntax

```
public java.lang.String getName()
```

Returns

name

getRefLocalname()**Description**

Get the local name of the resolved 'ref' attribute

Syntax

```
public java.lang.String getRefLocalname()
```

Returns

refLocalname

getRefNamespace()

Description

Get the namespace of the resolved 'ref' attribute

Syntax

```
public java.lang.String getRefNamespace()
```

Returns

refNamespace

getRefState()

Description

Get refState. The return value is one of the following: TYPE_UNRESOLVED, TYPE_RESOLVED, REF_UNRESOLVED, REF_RESOLVED

Syntax

```
public int getRefState()
```

Returns

refstate value

getTargetNS()

Description

Get target namespace

Syntax

```
public java.lang.String getTargetNS()
```

Overrides

XSDNode.getTargetNS() in class XSDNode

getType()

Description

Get the node type

Syntax

```
public XSDNode getType()
```

Returns

nodeType, which is either simpleType or complexType.

isRequired()

Description

Check if the attribute is required.

Syntax

```
public boolean isRequired()
```

XSDBuilder Class

Description of XSDBuilder

Builds an XMLSchema object from XMLSchema document. XMLSchema object is a set of objects (infoset) corresponding to top-level schema declarations and definitions. Schema document is 'XML' parsed and converted to a DOM tree. This schema DOM tree is 'Schema' parsed in a following order: (if any) builds a schema object and makes it visible. (if any) is replaced by corresponding DOM tree. Top-level declarations and definitions are registered as a current schema infoset of items. Finally, top-level tree elements (infoset items) are 'Schema' parsed. The resulting XMLSchema object is a set (infoset) of objects (top-level input elements). Object's contents is a tree with nodes corresponding to low-level element/group decls/refs preceded by node/object of type SNode containing cardinality info (min/maxOccurs).

Syntax of XSDBuilder

```
public class XSDBuilder extends oracle.xml.parser.schema.XSDNode
```

```
oracle.xml.parser.schema.XSDNode
|
+--oracle.xml.parser.schema.XSDBuilder
```

Methods of XSDBuilder

Table 7-9 Summary of Methods of XSDBuilder

Method	Description
XSDBuilder()	Class constructor
build()	Build an XMLSchema object or document
getObject()	Returns XML schema object.
setEntityResolver()	Set an EntityResolver for resolving imports/include
setError()	Sets XMLError object.
setLocale()	Sets locale for error reporting.

XSDBuilder()

Description

XSDBuilder constructor.

Syntax

```
public XSDBuilder() throws XSDException
```

build()

Description

Build an XMLSchema object/document.

Syntax

Table 7–10 *Syntax of build()*

Syntax

```
public Object build(InputStream in, URL baseurl) throws Exception
```

```
public Object build(Reader r, URL baseurl) throws Exception
```

```
public Object build(String sysid) throws Exception
```

```
public Object build(String ns, String sysid) throws Exception
```

```
public Object build(String ns, URL sysid) throws Exception
```

```
public Object build(URL schemaurl) throws Exception
```

```
public Object build(XMLDocument schemaDoc) throws Exception
```

```
public Object build(XMLDocument[] schemaDocs, URL baseurl)
```

```
public Object build(XMLDocument doc, String fragment, String ns, URL sysid)
```

```
public Object build(XMLDocument schemaDoc, URL baseurl)
```

Parameters

Table 7–11 *Parameters of build()*

Parameter	Description
-----------	-------------

baseurl	URL used to resolve any relative refs; used for any import/include in document
---------	--

Table 7–11 Parameters of build()

Parameter	Description
doc	XMLdocument contain the schema element
fragment	Fragment ID of the schema element
in	Inputstream of Schema
ns	Schema target namespace used to validate targetNamespace
r	Reader of Schema
schemaDoc	XMLDocument
schemaDocs	Array of XMLDocuments
sysId	Schema location
url	URL of Schema

Returns

Object - XMLSchema

Throws

An Exception is thrown if Builder fails to build an XMLSchema object.

getObject()**Description**

Returns XML schema object.

Syntax

```
public Object getObject()
```

Returns

XMLSchema object.

setEntityResolver()**Description**

Set an EntityResolver for resolving imports/include. See also `org.xml.sax.EntityResolver`.

Syntax

```
public void setEntityResolver( org.xml.sax.entityResolver entResolver)
```

Parameters**Table 7–12 Parameters of setEntityResolver**

Parameter	Description
entResolver	EntityResolver

setError()**Description**

Sets XMLError error object.

Syntax

```
public void setError(XMLError er)
```

Parameters**Table 7–13 Parameters of setError**

Parameter	Description
er	XMLError object

setLocale()**Description**

Sets locale for error reporting.

Syntax

```
public void setLocale(Locale locale)
```

Parameters**Table 7–14 Parameters of setLocale**

Parameter	Description
locale	Locale object

XSDComplexType Class

Description of XSDComplexType

XSDComplexType class manages the complexType for XML Schema Definition (XSD) for XML document. In XML Schema, the structure of the instance document or an element is called complexType.

Syntax of XSDComplexType

```
public class XSDComplexType extends oracle.xml.parser.schema.XSDNode
```

```

oracle.xml.parser.schema.XSDNode
|
+--oracle.xml.parser.schema.XSDComplexType

```

Methods XSDComplexType

Table 7–15 Summary of Methods of XSDComplexType

Method	Description
<code>getAttributeDeclarations()</code>	Get attribute declarations of this complexType does not include wild card array of attribute declarations
<code>getAttributeWildcard()</code>	Get all the attributes of the complexType elements
<code>getAttributeWildcard()</code>	get attribute wildcard of this complexType
<code>getBaseType()</code>	Get all the local elements of the base Type of this complexType element
<code>getBaseType()</code>	Get the base type of this complexType
<code>getContent()</code>	Get the content of this complexType.
<code>getDerivationMethod()</code>	Get numeric code indicating how this type was derived from its parent type.
<code>getElementSet()</code>	Get all the local elements inside a complexType element If the complexType element extends another complexType element
<code>getGroup()</code>	Get the attribute group or the child and attribute group
<code>getRefLocalname()</code>	Get the local name of resolved 'base' attr
<code>getTypeGroup()</code>	Get the kind of group for this complexType.

Table 7–15 (Cont.) Summary of Methods of XSDComplexType

Method	Description
<code>init()</code>	Initialize this group.
<code>isAbstract()</code>	Declare abstract or not abstract for this complexType, with Boolean true or false.

getAttributeDeclarations()**Description**

Get attribute declarations of this complexType; does not include wild card array of attribute declarations

Syntax

```
public XSDAttribute getAttributeDeclarations()
```

getAttributeWildcard()**Description**

Get attribute wildcard of this complexType.

Syntax

```
public oracle.xml.parser.schema.XSDAny getAttributeWildcard()
```

Returns

The attribute wildcard, if this type has one.

getBaseType()**Description**

Get the base type of this complexType.

Syntax

```
public XSDNode getBaseType()
```

Returns

XSDNode - base type

getContent()

Description

Get the content of this complexType.

Syntax

```
public int getContent()
```

getDerivationMethod()

Description

Returns a numeric code indicating the kind of derivation used to construct this type.

Syntax

```
public short getDerivationMethod()
```

Returns

A code number for: EXTENSION_DERIVATION or RESTRICTION_DERIVATION.

getElementSet()

Description

Gets all the local elements inside a complexType element if the complexType element extends another complexType element

Syntax

```
public XSDNode getElementSet()
```

Returns

an array of local elements

getGroup()

Description

Get the attribute group or the child and attribute group

Syntax

```
public XSDGroup getGroup()
```

Returns

group

getRefLocalname()**Description**

Get the local name of resolved 'base' attr

Syntax

```
public java.lang.String getRefLocalname()
```

Returns

refLocalname

getTypeGroup()**Description**

Get the kind of group for this complexType as either: model group or attribute group.

Syntax

```
public XSDGroup getTypeGroup()
```

init()**Description**

Initialize this group.

Syntax

```
public static void init()
```

isAbstract()**Description**

Make the group abstract or not abstract, with Boolean true or false.

Syntax

```
public boolean isAbstract()
```

XSDConstants Interface

Description of XSDConstants

Implements the XSDConstants interface.

Syntax of XSDConstants

```
public class XSDConstants
```

```
oracle.xml.parser.schema.XSDConstants
```

Methods of XSDConstants

XSDConstants()

Description

Class constructor.

Syntax

```
public XSDConstants()
```

XSDConstrainingFacet Class

Description of XSDConstrainingFacet

Implements XSDTypeConstants. XML Schema defines fifteen facets for applying constraints when deriving a datatype through restriction. *Facets* constrain the permitted values of a datatype.

Some facets use a *value space* for defining the restrictions on a datatype. A *value space* is the set of values for a given datatype. A *lexical space* is the set of valid literals for a datatype. Enumeration constrains the value space to a specified set of values. Each value in the value space of a datatype is denoted by one or more literals in its lexical space.

Syntax of XSDConstrainingFacet

```
public class XSDConstrainingFacet extends java.lang.Object implements
oracle.xml.parser.schema.XSDTypeConstants
```

```
java.lang.Object
|
+--oracle.xml.parser.schema.XSDConstrainingFacet
```

Implemented Interfaces of XSDConstrainingFacet

XSDTypeConstants

Methods of XSDConstrainingFacet

Table 7–16 Summary of Methods of XSDConstrainingFacet

Method	Description
<code>getFacetId()</code>	Get ID for the facet.
<code>getLexicalEnumeration()</code>	Get the starting and ending points for the enumerated literals that define the value space for this facet.
<code>getLexicalValue()</code>	Get value for lexical space for facet.
<code>getName()</code>	Get name of facet.
<code>isFixed(boolean)</code>	Declare facet is fixed or not fixed, with Boolean true or false.
<code>validateFacet(XSDDataValue)</code>	Validate facet against datatype.

getFacetId()

Description

Get the facet ID.

Syntax

```
public int getFacetId()
```

getLexicalEnumeration()

Description

Get the starting and ending points for the enumerated literals that define the value space for this facet.

Syntax

```
public java.util.Vector getLexicalEnumeration()
```

getLexicalValue()

Description

Get value for lexical space for facet.

Syntax

```
public java.lang.String getLexicalValue()
```

getName()

Description

Get name of facet.

Syntax

```
public java.lang.String getName()
```

isFixed(boolean)

Description

Declare facet is fixed or not fixed, with Boolean true or false.

Syntax

```
public boolean isFixed(boolean fixed)
```

validateFacet(XSDDataValue)**Description**

Validate facet against XML Schema Definition.

Syntax

```
public void validateFacet(XSDDataValue value)
```

XSDDataValue Class

Description of XSDDataValue

Implements XSDTypeConstants

Syntax of XSDDataValue

```
public class XSDDataValue extends java.lang.Object implements
oracle.xml.parser.schema.XSDTypeConstants
```

```
java.lang.Object
|
+--oracle.xml.parser.schema.XSDDataValue
```

Implemented Interfaces of XSDDataValue

XSDTypeConstants

Methods of XSDDataValue

Table 7-17 Summary of Methods of XSDDataValue

Method	Description
<code>compareTo()</code>	Compare two values return int -1 -- smaller, 0 -- equal, 1 -- greater
<code>getLength()</code>	Gets the length of STRING/BINARY value.
<code>getLexicalValue()</code>	Gets LEXICAL value from the XSDDataValue class return String value
<code>getPrecision()</code>	Gets the precision of decimal value return int precision
<code>getScale()</code>	Gets the scale of decimal value return int scale

compareTo()

Description

Compare two values; return int -1 -- smaller, 0 -- equal, 1 -- greater

Syntax

```
public int compareTo(XSDDataValue val)
```

Throws

`XSDException` - if the data values are not comparable

getLength()**Description**

Gets the length of `STRING/BINARY` value; return `int` length.

Syntax

```
public int getLength()
```

Throws

`XSDException` - if the data value is not of `String/Binary` type

getLexicalValue()**Description**

Gets `LEXICAL` value from the `XSDDataValue` class; return `String` value

Syntax

```
public java.lang.String getLexicalValue()
```

getPrecision()**Description**

Gets the precision of decimal value return `int` precision

Syntax

```
public int getPrecision()
```

Throws

`XSDException` - if the data value is not of decimal type

getScale()**Description**

Gets the scale of decimal value return `int` scale

Syntax

```
public int getScale()
```

Throws

`XSDException` - if the data value is not decimal type

XSDElement Class

Description of XSDElement

XSDElement class. Represents XMLSchema Definition for element.

Syntax of XSDElement

```
public class XSDElement

oracle.xml.parser.schema.XSDElement
```

Methods of XSDElement

Table 7–18 Summary of Methods of XSDElement

Method	Description
<code>findEquivClass()</code>	Find the equivalent class corresponding to this class
<code>getDefaultVal()</code>	Get the value of 'default' attr in case of element, and the value of 'value' attr based on 'use' attribute
<code>getEquivClassRef()</code>	Get the local name of the resolved derive class
<code>getFixedVal()</code>	Get the value of 'fixed' attr in case of element, and the value of 'value' attr based on 'use' attribute
<code>getIdentities()</code>	Returns the set of identities
<code>getMaxOccurs()</code>	Get the maxOccurs
<code>getMinOccurs()</code>	Get the minOccurs
<code>getName()</code>	Get Name
<code>getRefLocalname()</code>	Get the local name of the resolved 'ref' attribute
<code>getRefNamespace()</code>	Get the namespace of the resolved 'ref' attribute
<code>getRefState()</code>	Get refState
<code>getSubstitutionGroup()</code>	Get the substitutionGroup
<code>getTargetNS()</code>	Get target namespace
<code>getType()</code>	Get the node type
<code>isAbstract()</code>	Abstract, true or false.

Table 7–18 Summary of Methods of XMLElement (Cont.)

Method	Description
<code>isNullable()</code>	Nullable, true or false.
<code>setMaxOccurs()</code>	Set the maxOccurs
<code>setMinOccurs()</code>	Set the minOccurs

findEquivClass()

Description

Find the equivalent class corresponding to this class

Syntax

```
public XMLElement findEquivClass(java.lang.String ns, java.lang.String nm)
```

Parameters

Table 7–19 Parameters of findEquivClass

Parameter	Description
ns	namespace for class
nm	name of class

Returns

XMLElement

getDefaultVal()

Description

Get the value of 'default' attr in case of element, and the value of 'value' attr based on 'use' attribute

Syntax

```
public String getDefaultVal()
```

Returns

defaultVal

getEquivClassRef()

Description

Get the local name of the resolved equiv class

Syntax

```
public String getEquivClassRef()
```

Returns

equivRefLocalname

getFixedVal()

Description

Get the value of 'fixed' attr in case of element, and the value of 'value' attr based on 'use' attribute

Syntax

```
public java.lang.String getFixedVal()
```

Returns

defaultVal

getIdentities()

Description

Returns the set of identities

Syntax

```
public XSDIdentity getIdentities()
```

Returns

array of identities

getMaxOccurs()

Description

Get the maxOccurs

Syntax

```
public int getMaxOccurs()
```

Returns

maxOccurs

getMinOccurs()

Description

Get the minOccurs

Syntax

```
public int getMinOccurs()
```

Returns

minOccurs value

getName()

Description

Get the name of the node

Syntax

```
public String getName()
```

Returns

name

getRefLocalname()

Description

Get the local name of the resolved 'ref' attribute

Syntax

```
public String getRefLocalname()
```

Returns

refLocalname

getRefNamespace()

Description

Get the namespace of the resolved 'ref' attribute

Syntax

```
public String getRefNamespace()
```

Returns

refNamespace

getRefState()

Description

Get refState. The return value is one of the following: TYPE_UNRESOLVED, TYPE_RESOLVED, REF_UNRESOLVED, REF_RESOLVED

Syntax

```
public int getRefState()
```

Returns

refstate value

getSubstitutionGroup()

Description

Get the substitutionGroup

Syntax

```
public java.util.Vector getSubstitutionGroup()
```

getTargetNS()

Description

Get target namespace

Syntax

```
public java.lang.String getTargetNS()
```

getType()

Description

Get the node type

Syntax

```
public XSDNode getType()
```

Returns

nodeType, which is either simpleType or complexType

isAbstract()

Description

Declares is or is not abstract.

Syntax

```
public boolean isAbstract()
```

isNullable()

Description

Declares is or is not nullable.

Syntax

```
public boolean isNullable()
```

setMaxOccurs()

Description

Set the maxOccurs

Syntax

```
public void setMaxOccurs(int max)
```

Parameters

Table 7–20 Parameters of setMaxOccurs

Parameter	Description
maxOccurs	value

setMinOccurs()

Description

Set the minOccurs

Syntax

```
public void setMinOccurs(int min)
```

Parameters

Table 7–21 Parameters of setMinOccurs

Parameter	Description
minOccurs	value

XSDException

Description of XSDException

Indicates that an exception occurred during XMLSchema validation.

Syntax of XSDException

```
java.lang.Object
|
+--- java.lang.Throwable
      |
      +--- java.lang.Exception
            |
            +--- oracle.xml.parser.schema.XSDException

public class XSDException
extends Exception
```

getMessage()

Description

Overrides `getMessage` in class `Throwable`, in order to construct error message from error id and error parameters. The options are described in [Table 7-22](#):

Table 7-22 *Versions of `getMessage()`*

Syntax	Description
<code>public String getMessage()</code>	Constructs error message from error id and error parameters
<code>public String getMessage(XMLError err)</code>	Constructs localized error message based on the <code>XMLError</code> sent as parameter

Parameters

Table 7-23 *Parameters of `getMessage()`*

Parameter	Description
<code>err</code>	<code>XMLError</code> class used to get the error message

XSDGroup Class

Description of XSDGroup

XSDGroup represents the model group for the XMLSchema. A model group may further contain model groups or element particles.

Syntax of XSDGroup

```
public class XSDGroup
```

```
oracle.xml.parser.schema.XSDGroup
```

Methods of XSDGroup

Table 7-24 Summary of Methods of XSDIdentity

Method	Description
<code>getMaxOccurs()</code>	Get the maxOccurs
<code>getMinOccurs()</code>	Get the minOccurs
<code>getNodeVector()</code>	Get the particles of the group stored in nodeVector
<code>getOrder()</code>	Get the composite type - ALL, SEQUENCE, CHOICE
<code>setMaxOccurs()</code>	Set maxOccurs
<code>setMinOccurs()</code>	Set minOccurs

getMaxOccurs()

Description

Get the maxOccurs

Syntax

```
public int getMaxOccurs()
```

Returns

maxOccurs

getMinOccurs()

Description

Get the minOccurs

Syntax

```
public int getMinOccurs()
```

Returns

minOccurs

getNodeVector()

Description

Get the particles of the group stored in nodeVector

Syntax

```
public java.util.Vector getNodeVector()
```

Returns

nodeVector

getOrder()

Description

Get the composite type - ALL, SEQUENCE, or CHOICE

Syntax

```
public int getOrder()
```

Returns

order

setMaxOccurs()

Description

Set maxOccurs

Syntax

```
public void setMaxOccurs(int max)
```

Parameters**Table 7–25 Parameters of setMaxOccurs**

Parameter	Description
maxOccurs	value

setMinOccurs()**Description**

Set the minOccurs

Syntax

```
public void setMinOccurs(int min)
```

Parameters**Table 7–26 Parameters of setMinOccurs**

Parameter	Description
minOccurs	value

XSDIdentity Class

Description of XSDIdentity

XSDIdentity presents identity parameters for the XSD for XMLSchema.

Syntax of XSDIdentity

```
public class XSDIdentity extends oracle.xml.parser.schema.XSDNode
```

```
oracle.xml.parser.schema.XSDNode  
|  
+--oracle.xml.parser.schema.XSDIdentity
```

Methods of XSDIdentity

Table 7–27 Summary of Methods of XSDIdentity

Method	Description
getFields()	Get the fields
getNodeType()	Get the node type
getRefer()	Get the reference key
getSelector()	Get the selector

getFields()

Description

Get the fields

Syntax

```
public java.lang.String[] getFields()
```

Returns

fields

getNodeType()

Description

Get the node Type. Overrides XSDNode.getNodeType() in class XSDNode

Syntax

```
public int getNodeTypes()
```

Returns

nodeTypes

getRefer()**Description**

Get the referenced key

Syntax

```
public java.lang.String getRefer()
```

Returns

referenced key

getSelector()**Description**

Get the selector

Syntax

```
public java.lang.String getSelector()
```

Returns

selector

XSDNode Class

Description of XSDNode

Root class for most of XSD classes. Contains fields and methods corresponding to XMLSchema definition attributes.

Syntax of XSDNode

```
public class XSDNode
```

```
oracle.xml.parser.schema.XSDNode
```

Direct Subclasses of XSDNode

XMLSchema, XMLSchemaNode, XSDAttribute, XSDComplexType, XSDIdentity

Methods of XSDNode

Table 7–28 Summary of Methods of XSDNode

Method	Description
<code>getName()</code>	Get the name of the node
<code>getNamespaceURI()</code>	Get namespace URI
<code>getNodeTypeInfo()</code>	Get the type of XSDNode.
<code>getTargetNS()</code>	Get target namespace
<code>isNodeTypeInfo()</code>	Checks if the node is of the give type

getName()

Description

Get the name of the node

Syntax

```
public java.lang.String getName()
```

Returns

name

getNamespaceURI()

Description

Get namespace URI.

Syntax

```
public java.lang.String getNamespaceURI()
```

Returns

targetNS

getNodeTypes()

Description

Get the type of XSDNode.

Syntax

```
public int getNodeTypes()
```

Returns

nodeType

getTargetNS()

Description

Get target namespace

Syntax

```
public java.lang.String getTargetNS()
```

Returns

targetNS

isNodeType()

Description

Checks if the node is of the give type

Syntax

```
public boolean isNodeType(int type)
```

Parameters

Table 7–29 *Parameters of isNodeType*

Parameter	Description
type	type of node that is being checked

XSDSimpleType Class

Description of XSDSimpleType

Implements XSDTypeConstants to derive a type.

Syntax of XSDSimpleType

```
public class XSDSimpleType implements oracle.xml.parser.schema.XSDTypeConstants
```

```
oracle.xml.parser.schema.XSDSimpleType
```

Implemented Interfaces of XSDSimpleType

XSDTypeConstants

Methods of XSDSimpleType

Table 7–30 Summary of Methods of XSDSimpleType

Method	Description
<code>XSDSimpleType()</code>	Class constructor
<code>derivedFrom()</code>	Derive a type from the given base type
<code>getBase()</code>	Gets base type.
<code>getBasicType()</code>	Gets the basic type from which this type was derived.
<code>getBuiltInDatatypes()</code>	Gets a built-in datatype
<code>getFacets()</code>	Get the facets
<code>getMaxOccurs()</code>	<code>public int getMaxOccurs()</code>
<code>getMinOccurs()</code>	Get the value of minOccurs
<code>getVariety()</code>	Get the variety of the type
<code>isAbstract()</code>	Declares Boolean for abstract, true or false.
<code>setFacet()</code>	Sets a facet for the datatype (Internal private API)
<code>setMaxOccurs()</code>	Set the value of maxOccurs
<code>setMinOccurs()</code>	Set the value of minOccurs
<code>setSource()</code>	Sets the base type of the datatype, or in case of aggregate types sets the type of the component of the aggregate type

Table 7–30 Summary of Methods of XSDSimpleType (Cont.)

Method	Description
<code>validateValue()</code>	Validates the string value with the facets defined for this type

XSDSimpleType()

Description

Class constructor. The options are described in [Table 7–31](#):

Table 7–31 Versions of XSDSimpleType constructor

Syntax	Description
<code>public XSDSimpleType()</code>	default constructor
<code>public XSDSimpleType(int basic, String nm)</code>	derived

derivedFrom()

Description

Derive a type from the given base type.

Syntax

```
public static XSDSimpleType derivedFrom(XSDSimpleType source, String nm, String var)
```

Parameters

Table 7–32 Parameters of derivedFrom

Parameter	Description
<code>source</code>	XSDSimpleType The base type
<code>nm</code>	String The name of the new type
<code>var</code>	String The method of derivation

Throws

XSDException - if new type can not be created.

getBase()

Description

Specify the base type from which to derive this type.

Syntax

```
public XSDSimpleType getBase()
```

getBasicType()

Description

Gets the basic type from which this type was derived.

Syntax

```
public int getBasicType()
```

Returns

basicType

getBuiltInDatatypes()

Description

Gets a built-in datatype

Syntax

```
public static Hashtable getBuiltInDatatypes()
```

Parameters

Table 7–33 Parameters of *getBuiltInDatatypes*

Parameter	Description
type	Name of the built-in type

Throws

XSDException if the type is not a valid name

getFacets()

Description

Get the facets for this datatype.

Syntax

```
public XSDConstrainingFacet getFacets()
```

Returns

facets

getMaxOccurs()

Description

```
public int getMaxOccurs()
```

Syntax

Get the value of maxOccurs

Returns

1

getMinOccurs()

Description

Get the value of minOccurs

Syntax

```
public int getMinOccurs()
```

Returns

1

getVariety()

Description

Get the variety of the type

Syntax

```
public java.lang.String getVariety()
```

Returns

variety

isAbstract()**Description**

Declares Boolean for abstract, true or false.

Syntax

```
public boolean isAbstract()
```

setFacet()**Description**

Sets a facet for the datatype (Internal private API)

Syntax

```
public void setFacet(String facetName, String value)
```

Parameters

Table 7–34 *Parameters of setFacet*

Parameter	Description
facetName	name of the facet being set
value	value of the facet

Throws

`XSDException` - if the facet is invalid

setMaxOccurs()**Description**

Set the value of maxOccurs

Syntax

```
public void setMaxOccurs(int max)
```

Parameters**Table 7–35 Parameters of setMaxOccurs**

Parameter	Description
max	number of maximum occurrences

setMinOccurs()**Description**

Set the value of minOccurs.

Syntax

```
public void setMinOccurs(int min)
```

Parameters**Table 7–36 Parameters of setMinOccurs**

Parameter	Description
min	number of minimum occurrences

setSource()**Description**

Sets the base type of the datatype, or for aggregate types, sets the type of the component of the aggregate type.

Syntax

```
public void setSource(XSDNode src)
```

Parameters**Table 7–37 Parameters of setSource**

Parameter	Description
src	XSDNode source

Throws

XSDException if the src is not a valid type

validateValue()**Description**

Validates the string value with the facets defined for this type.

Syntax

```
public void validateValue(java.lang.String val)
```

Parameters

Table 7–38 *Parameters of validateValue*

Parameter	Description
val	value to be validated

Throws

XSDException if the value is not valid

XSDTypeConstants Interface

Description of XSDTypeConstants

Implements the interface for XSDTypeConstants.

Syntax of XSDTypeConstants

```
public interface XSDTypeConstants
```

Implementing Classes of XSDTypeConstants

XSDDataValue, XSDConstrainingFacet, XSDSimpleType

Fields of XSDTypeConstants

Table 7–39 *Fields of XSDTypeConstants*

Field	Syntax
_atomic	public static final java.lang.String _atomic
_base64	public static final java.lang.String _base64
_collapse	public static final java.lang.String _collapse
_hex	public static final java.lang.String _hex
_preserve	public static final java.lang.String _preserve
_replace	public static final java.lang.String _replace
ANY_SIMPLE	public static final java.lang.String ANY_SIMPLE
ANY_URI	public static final java.lang.String ANY_URI
BASE64_BINARY	public static final java.lang.String BASE64_BINARY
BINARY	public static final java.lang.String BINARY
BOOLEAN	public static final java.lang.String BOOLEAN
BYTE	public static final java.lang.String BYTE
CDATA	public static final java.lang.String CDATA
CENTURY	public static final java.lang.String CENTURY
DATE	public static final java.lang.String DATE
DATE_TIME	public static final java.lang.String DATE_TIME

Table 7–39 Fields of XSDTypeConstants (Cont.)

Field	Syntax
DECIMAL	public static final java.lang.String DECIMAL
DOUBLE	public static final java.lang.String DOUBLE
DURATION	public static final java.lang.String DURATION
ENCODING	public static final java.lang.String ENCODING
ENTITIES	public static final java.lang.String ENTITIES
ENTITY	public static final java.lang.String ENTITY
ENUMERATION	public static final java.lang.String ENUMERATION
FLOAT	public static final java.lang.String FLOAT
FRACTION_DIGITS	public static final java.lang.String FRACTION_DIGITS
GDAY	public static final java.lang.String GDAY
GMONTH	public static final java.lang.String GMONTH
GMONTH_DAY	public static final java.lang.String GMONTH_DAY
GYEAR	public static final java.lang.String GYEAR
GYEAR_MONTH	public static final java.lang.String GYEAR_MONTH
HEX_BINARY	public static final java.lang.String HEX_BINARY
iANY_SIMPLE	public static final int iANY_SIMPLE
iANY_URI	public static final int iANY_URI
iBASE64_BINARY	public static final int iBASE64_BINARY
iBOOLEAN	public static final int iBOOLEAN
ID	public static final java.lang.String ID
iDATE	public static final int iDATE
iDATE_TIME	public static final int iDATE_TIME
iDECIMAL	public static final int iDECIMAL
iDOUBLE	public static final int iDOUBLE
IDREF	public static final java.lang.String IDREF
IDREFS	public static final java.lang.String IDREFS
iDUMMY	public static final int iDUMMY

Table 7–39 Fields of XSDTypeConstants (Cont.)

Field	Syntax
iDURATION	public static final int iDURATION
iENUMERATION	public static final int iENUMERATION
iFLOAT	public static final int iFLOAT
iFRACTION_DIGITS	public static final int iFRACTION_DIGITS
iGDAY	public static final int iGDAY
iGMONTH	public static final int iGMONTH
iGMONTH_DAY	public static final int iGMONTH_DAY
iGYEAR	public static final int iGYEAR
iGYEAR_MONTH	public static final int iGYEAR_MONTH
iHEX_BINARY	public static final int iHEX_BINARY
iLENGTH	public static final int iLENGTH
iMAXEXCLUSIVE	public static final int iMAXEXCLUSIVE
iMAXINCLUSIVE	public static final int iMAXINCLUSIVE
iMAXLENGTH	public static final int iMAXLENGTH
iMINEXCLUSIVE	public static final int iMINEXCLUSIVE
iMININCLUSIVE	public static final int iMININCLUSIVE
iMINLENGTH	public static final int iMINLENGTH
iNOTATION	public static final int iNOTATION
INT	public static final java.lang.String INT
INTEGER	public static final java.lang.String INTEGER
iPATTERN	public static final int iPATTERN
iQNAME	public static final int iQNAME
iSTRING	public static final int iSTRING
iTIME	public static final int iTIME
iTOTAL_DIGITS	public static final int iTOTAL_DIGITS
iWHITESPACE	public static final int iWHITESPACE
LANGUAGE	public static final java.lang.String LANGUAGE

Table 7–39 Fields of XSDTypeConstants (Cont.)

Field	Syntax
LENGTH	public static final java.lang.String LENGTH
LONG	public static final java.lang.String LONG
MAXEXCLUSIVE	public static final java.lang.String MAXEXCLUSIVE
MAXINCLUSIVE	public static final java.lang.String MAXINCLUSIVE
MAXLENGTH	public static final java.lang.String MAXLENGTH
MINEXCLUSIVE	public static final java.lang.String MINEXCLUSIVE
MININCLUSIVE	public static final java.lang.String MININCLUSIVE
MININCLUSIVE	public static final java.lang.String MININCLUSIVE
MINLENGTH	public static final java.lang.String MINLENGTH
MONTH	public static final java.lang.String MONTH
N_STRING	public static final java.lang.String N_STRING
NAME	public static final java.lang.String NAME
NCNAME	public static final java.lang.String NCNAME
NEGATIVE_INTEGER	public static final java.lang.String NEGATIVE_INTEGER
nFacets	public static final int nFacets
NMTOKEN	public static final java.lang.String NMTOKEN
NMTOKENS	public static final java.lang.String NMTOKENS
NON_NEGATIVE_INTEGER	public static final java.lang.String NON_NEGATIVE_INTEGER
NON_POSITIVE_INTEGER	public static final java.lang.String NON_POSITIVE_INTEGER
PATTERN	public static final java.lang.String PATTERN
PERIOD	public static final java.lang.String PERIOD
POSITIVE_INTEGER	public static final java.lang.String POSITIVE_INTEGER
PRECISION	public static final java.lang.String PRECISION
QNAME	public static final java.lang.String QNAME
RECURRING_DATE	public static final java.lang.String RECURRING_DATE
RECURRING_DAY	public static final java.lang.String RECURRING_DAY
RECURRING_DURATION	public static final java.lang.String RECURRING_DURATION

Table 7–39 Fields of XSDTypeConstants (Cont.)

Field	Syntax
SCALE	public static final java.lang.String SCALE
sFacets	public static final java.lang.String[] sFacets
SHORT	public static final java.lang.String SHORT
SNOTATION	public static final java.lang.String SNOTATION
STRING	public static final java.lang.String STRING
sTypes	public static final java.lang.String[] sTypes
TIME	public static final java.lang.String TIME
TIME_DURATION	public static final java.lang.String TIME_DURATION
TIME_INSTANT	public static final java.lang.String TIME_INSTANT
TIME_PERIOD	public static final java.lang.String TIME_PERIOD
TOKEN	public static final java.lang.String TOKEN
TOTAL_DIGITS	public static final java.lang.String TOTAL_DIGITS
UNSIGNED_BYTE	public static final java.lang.String UNSIGNED_BYTE
UNSIGNED_INT	public static final java.lang.String UNSIGNED_INT
UNSIGNED_LONG	public static final java.lang.String UNSIGNED_LONG
UNSIGNED_SHORT	public static final java.lang.String UNSIGNED_SHORT
URI_REFERENCE	public static final java.lang.String URI_REFERENCE
WHITESPACE	public static final java.lang.String WHITESPACE
YEAR	public static final java.lang.String YEAR

XSDValidator Class

Description of XSDValidator

XSDValidator validates an instance XML document against an XMLSchema.

- When registered, an XSDValidator object is inserted as a pipe-line node between XMLParser and XMLDocument event handlers (SAXHandler or DOMBuilder).
- It works with three events: startElement, characters and endElement. If defined, default element and default attribute values are added to the events contents (as XMLSchema additions to infoSet) and are propagated upwards.
- The XMLSchema object is a set or group of element declarations with the following structure: [element (name)] -> [snode (min/maxOccurs)] -> [type (group/simpleType)]
- XSDValidator is implemented as stack based state machine. Each state represents element type - group or simpleType.
- XMLSchema object (as a group) is loaded as a first state. Current element (event startElement) is matched against current state group elements. If matched the element type element name and snode info are loaded as new state.
- In a case of group, a vector of counters (int) is allocated in a parallel stack. This vector is used to count element occurrences.
 - State status can be:
 - NEW_STATE: just loaded and not tried.
 - ACCEPTED: minOccurs satisfied. Could still accept element occurrences.
 - DONE: maxOccurs satisfied. Doesn't accept element occurrences.
- Text element contents (event characters) is matched against simpleType (method validateValue). End element (event endElement) is matched against last named state.
- XMLSchema attributes are represented as a group (attrName -> attrType) forming the contents of special element: <_attrTag> attrType ... XMLParser converts attributes (event startElement) accordingly (see method startElement).
- XSDAny objects are used as Namespace frame descriptors (see XMLSchema definition of 'any' element).
- Fake states are loaded for errors or when wildcard ('any') contents are skipped.

Syntax of XSDValidator

```
public class XSDValidator
```

```
oracle.xml.parser.schema.XSDValidator
```

Methods of XSDValidator

Table 7–40 Summary of Methods of XSDValidator

Method	Description
<code>XSDValidator()</code> ,	Class constructor
<code>characters()</code>	Propagate notification of character data inside an element.
<code>endElement()</code>	Receive notification of the end of an element.
<code>setDocumentLocator()</code>	Propagates Locator object for document events.
<code>setError()</code>	Sets an XMLError object as current err.
<code>setXMLProperties()</code>	Set XML Properties for runtime properties
<code>setXMLProperty()</code>	Set a property.
<code>startElement()</code>	Receive notification of a beginning of the element

XSDValidator()

Description

XSDValidator constructor.

Syntax

```
public XSDValidator()
```

characters()

Description

Propagate notification of character data inside an element. See also `org.xml.sax.DocumentHandler`

Syntax

```
public void characters(char[] ch, int start, int length)
```

Parameters

Table 7–41 *Parameters of endElement*

Parameter	Description
ch	The characters
start	The start position in the character array.
length	The number of characters to use from the character array.

Throws

`org.xml.sax.SAXException` - Any SAX exception, possibly wrapping another exception.

endElement()

Description

Receive notification of the end of an element.

Syntax

```
public void endElement(String namespaceURI, String localName, String qName)
```

Parameters

Table 7–42 *Parameters of endElement*

Parameter	Description
uri	The Namespace URI, or the empty string if the element has no Namespace URI or if Namespace processing is not being performed
localName	The local name (without prefix), or the empty string if Namespace processing is not being performed.
qName	The qualified XML 1.0 name (with prefix), or the empty string if qualified names are not available.

Throws

`org.xml.sax.SAXException` - Any SAX exception, possibly wrapping another exception.

setDocumentLocator()

Description

Propagates Locator object for document events.

See also `org.xml.sax.DocumentHandler`, `org.xml.sax.Locator`

Syntax

```
public void setDocumentLocator(org.xml.sax.Locator locator)
```

Parameters

Table 7–43 Parameters of setDocumentLocator

Parameter	Description
locator	A locator for all SAX document events

setError()

Description

Sets an XMLError object as current err.

Syntax

```
public void setError(oracle.xml.parser.v2.XMLError he)
```

Parameters

Table 7–44 Parameters of setError

Parameter	Description
he	XMLError object

Throws

`SAXException` - A `SAXException` could be thrown.

setXMLProperties()

Description

Set XML Properties for runtime properties

Syntax

```
public void setXMLProperties(XMLProperties xmlProp)
```

Parameters**Table 7–45** *Parameters of setXMLProperties*

Parameter	Description
xmlProp	XMLProperties
value	value of the property

setXMLProperty()**Description**

Set a property. The value of the property is returned if successfully set. A null is returned if the property is read-only and cannot be set or is not supported.

Syntax

```
public Object setXMLProperty(java.lang.String name, java.lang.Object value)
```

Parameters**Table 7–46** *Parameters of setXMLProperty*

Parameter	Description
name	name of the property
value	value of the property

Returns

Object the set property

startElement()**Description**

Receive notification of the beginning of an element.

See also: `endElement(String, String, String)`,
`org.xml.sax.Attributes`

Syntax

```
public void startElement(String namespaceURI, String localName, String qName,
Attributes atts)
```

Parameters

Table 7–47 *Parameters of startElement*

Parameter	Description
uri	The Namespace URI, or the empty string if the element has no Namespace URI or if Namespace processing is not being performed
localName	The local name (without prefix), or the empty string if Namespace processing is not being performed.
qName	The qualified name (with prefix), or the empty string if qualified names are not available.
atts	The attributes attached to the element. If there are no attributes, it shall be an empty Attributes object.

Throws

`org.xml.sax.SAXException` - Any SAX exception, possibly wrapping another exception.

Package oracle.xml.sql.dml

This chapter documents the class contained in package `oracle.xml.sql.dml`, which handles data manipulation and modification for XML SQL Utility for Java (XSU). XSU is part of the Oracle XDK for Java. XML SQL Utility for Java generates and stores XML data to and from the database from SQL queries or result sets or tables. It achieves data transformation by mapping canonically any SQL query result to XML, and vice versa.

This chapter contains these sections:

- [Package oracle.xml.sql.dml Description](#)
- [OracleXMLSave Class](#)

Package oracle.xml.sql.dml Description

Package oracle.xml.sql.dml implements data manipulation and modification functions for Oracle XDK for Java. (DML is for Data Manipulation/Modification Language.) The methods for DML operations are provided in the OracleXMLSave class contained in this package. The OracleXMLSave class supports canonical mapping from XML to object-relational tables or views.

See Also: [Chapter 9, "Package oracle.xml.sql.query"](#)

OracleXMLSave Class

Description

OracleXMLSave class supports canonical mapping from XML to object-relational tables or views. It supports inserts, updates and deletes. The user first creates the class by passing in the table name on which these DML operations need to be done. After that, the user is free to use the insert/update/delete on this table.

Many useful functions are provided in this class to help in identifying the key columns for update or delete and to restrict the columns being updated.

Syntax

```
public class OracleXMLSave extends java.lang.Object
```

```
java.lang.Object
|
+--oracle.xml.sql.dml.OracleXMLSave
```

Fields

Table 8–1 Summary of Fields of OracleXMLSave

Field	Syntax	Description
DATE_FORMAT	public static final java.lang.String DATE_FORMAT	The date format for use in setDateFormat
DEFAULT_BATCH_SIZE	public static int DEFAULT_BATCH_SIZE	Default insert batch size is 17
xDocIsEsc	public boolean xDocIsEsc	Indicates whether or not the xml doc has undergone SQL to XML escaping

Methods

Table 8–2 Summary of Methods of OracleXMLSave

Method	Description
OracleXMLSave()	The public constructor for the Save class.
close()	It closes/deallocates all the context associated with this object.

Table 8–2 (Cont.) Summary of Methods of OracleXMLSave

Method	Description
<code>deleteXML()</code>	Deletes the rows in the table based on the XML document.
<code>getURL()</code>	Return a URL object given a file name or a URL.
<code>insertXML()</code>	Inserts an XML document into the specified table.
<code>removeXSLTParam()</code>	Removes the value of a top-level stylesheet parameter.
<code>setBatchSize()</code>	Changes the batch size used during DML operations.
<code>setCommitBatch()</code>	Sets the commit batch size.
<code>setDateFormat()</code>	Describes to the XSU the format of the dates in the XML document.
<code>setIgnoreCase()</code>	Instructs the XSU to perform a case-insensitive match of XML elements to database columns or attributes.
<code>setKeyColumnList()</code>	Sets the list of columns to be used for identifying a particular row in the database table during update or delete.
<code>setPreserveWhitespace()</code>	Instructs the XSU whether to preserve whitespaces.
<code>setRowTag()</code>	Names the tag used in the XML doc. to enclose the XML elements corresponding to each row value.
<code>setSQLToXMLNameEscaping()</code>	This turns on or off escaping of XML tags when an SQL object name would not make a valid XML identifier.
<code>setUpdateColumnList()</code>	Set the column values to be updated.
<code>setXSLT()</code>	Registers a XSL transform to be applied to generated XML.
<code>setXSLTParam()</code>	Sets the value of a top-level stylesheet parameter.
<code>updateXML()</code>	Updates the table given the XML document.

OracleXMLSave()

Description

The public constructor for the OracleXMLSave class.

Syntax

```
public OracleXMLSave( java.sql.Connection oconn,
                     java.lang.String tableName;
```

Parameters

oconn	Connection object (connection to the database)
tableName	The name of the table that should be updated

close()**Description**

Closes/deallocates all the context associated with this object.

Syntax

```
public void close();
```

deleteXML()**Description**

Deletes the rows in the table based on the XML document. Returns the number of XML ROW elements processed. This may or may not be equal to the number of database rows deleted based on whether the rows selected through the XML document uniquely identified the rows in the table.

By default, the delete processing matches all the element values with the corresponding column names. Each ROW element in the input document is taken as a separate delete statement on the table. By using the [setKeyColumnList\(\)](#), the list of columns that must be matched to identify the row to be deleted is set, and other elements are ignored. This is an efficient method for deleting more than one row in the table if matching is employed (since the delete statement is cached). Otherwise, a new delete statement has to be created for each ROW element in the input document. The syntax options are described in the table here.

Syntax	Description
public int deleteXML(org.w3c.dom.Document doc);	XML document is in DOM form.
public int deleteXML(java.io.InputStream xmlStream);	XML document is in Stream form.

(Cont.) Syntax	Description
<code>public int deleteXML(java.io.Reader xmlReader);</code>	XML document is in Reader form.
<code>public int deleteXML(java.lang.String xmlDoc);</code>	XML document is in String form.
<code>public int deleteXML (java.net.URL url);</code>	XML document is accessed via the URL.

Parameters

<code>doc</code>	The XML document in DOM form.
<code>xmlStream</code>	The XML document in Stream form.
<code>xmlReader</code>	The XML document in Reader form.
<code>xmlDoc</code>	The XML document in String form.
<code>url</code>	The URL to the document to use to delete the rows in the table.

getURL()

Description

Returns a URL object identifying the target entity, given a file name or a URL. If the argument passed is not in a valid URL format, such as "http://..." or "file://...", then this method attempts to correct the argument by pre-pending "file://". If a NULL or an empty string are passed to it, NULL is returned.

Syntax

```
public static java.net.URL getURL( java.lang.String target);
```

Parameters

<code>target</code>	File name or URL string.
---------------------	--------------------------

insertXML()

Description

Inserts an XML document into the specified table. Returns the number of rows inserted.

- Inserts the values into the table by matching the element name with the column name, and inserts a NULL value for all elements that are missing in the input document. By using the [setUpdateColumnList\(\)](#), no NULL values would be inserted for the rest of the columns; instead, default values would be used.
- To set the list of all key column, use [setKeyColumnList\(\)](#).
- To set the list of columns to update, use [setUpdateColumnList\(\)](#).

The options are described in the following table.

Syntax	Description
<code>public int insertXML(org.w3c.dom.Document doc);</code>	Inserts an XML document from a DOM.
<code>public int insertXML(java.io.InputStream xmlStream);</code>	Inserts an XML document from an InputStream.
<code>public int insertXML(java.io.Reader xmlStream);</code>	Inserts an XML document from a Reader.
<code>public int insertXML(java.lang.String xmlDoc);</code>	Inserts an XML document from a String.
<code>public int insertXML(java.net.URL url);</code>	Inserts an XML document from a URL.

Parameters

<code>doc</code>	DOM for inserting rows into the table.
<code>xmlStream</code>	Stream of data used for inserting rows into the table.
<code>xmlDOC</code>	String used for inserting rows into the table.
<code>url</code>	The URL to the document used for inserting rows into the table.

removeXSLTParam()

Description

Removes the value of a top-level stylesheet parameter. If no stylesheet is registered, this method is a no op.

Syntax

```
public void removeXSLTParam( java.lang.String name);
```

Parameters

name	Parameter name
------	----------------

setBatchSize()**Description**

Changes the batch size used during DML operations. When performing inserts, updates or deletes, it is recommended to batch the operations to minimize I/O cycles; however, this requires more cache for storing the bind values while the operations are executing. When batching is used, the commits occur only in terms of batches. If a single statement inside a batch fails, the entire batch is rolled back. If this behavior is undesirable, set batch size to 1. The default batch size is `DEFAULT_BATCH_SIZE`.

Syntax

```
public void setBatchSize(int size);
```

Parameters

size	The batch size to use for all DML.
------	------------------------------------

setCommitBatch()**Description**

Sets the commit batch size, which refers to the number of records inserted after which a commit must follow. If `size < 1`, or the session is in "auto-commit" mode, the XSU does not make any explicit commits. Default commit-batch size is 0.

Syntax

```
public void setCommitBatch( int size);
```

Parameters

size	Commit batch size.
------	--------------------

setDateFormat()**Description**

Describes to the XSU the format of the dates in the XML document. By default, `OracleXMLSave` assumes that the date is in format 'MM/dd/yyyy HH:mm:ss'. You can

override this default format by calling this function. The syntax of the date format pattern (i.e. the date mask), should conform to the requirements of the `java.text.SimpleDateFormat` class. Setting the mask to `NULL` or an empty string, causes the use of the default mask -- `OracleXMLSave.DATE_FORMAT`.

Syntax

```
public void setDateFormat( java.lang.String mask);
```

Parameters

mask The date mask.

setIgnoreCase()

Description

The XSU performs mapping of XML elements to database columns or attributes based on the element names (XML tags). This function instructs the XSU to perform a case-insensitive match. This may affect the metadata caching performed when creating the Save object.

Syntax

```
public void setIgnoreCase(boolean ignore);
```

Parameters

flag Should the tag case in the XML doc be ignored?

setKeyColumnList()

Description

Sets the list of columns to be used for identifying a particular row in the database table during update or delete. This call is ignored for the insert case. The key columns must be set before updates can be done. It is optional for deletes. When this key columns is set, then the values from these tags in the XML document is used to identify the database row for update or delete. Currently, there is no way to update the values of the key columns themselves, since there is no way in the XML document to specify that case.

Syntax

```
public void setKeyColumnList( java.lang.String[] keyColNames);
```

Parameters

keyColNames The names of the list of columns that are used as keys.

setPreserveWhitespace()

Description

Instructs the XSU whether to preserve whitespaces.

Syntax

```
public void setPreserveWhitespace( boolean flag);
```

Parameters

flag Should the whitespaces be preserved?

setRowTag()

Description

Names the tag used in the XML doc so to enclose the XML elements corresponding to each row value. Setting the value of this to NULL implies that there is no row tag present, and the top level elements of the document correspond to the rows themselves.

Syntax

```
public void setRowTag( java.lang.String rowTag);
```

Parameters

tag Tag name.

setSQLToXMLNameEscaping()

Description

This turns on or off escaping of XML tags when the SQL object name, which is mapped to a XML identifier, is not a valid XML identifier.

Syntax

```
public void setSQLToXMLNameEscaping( boolean flag);
```

Parameters

flag Should the SQL to XML escaping be turned on?

setUpdateColumnList()**Description**

Set the column values to be updated. Applies to inserts and updates, not deletes.

- In case of insert, the default is to insert values to all the columns in the table.
- In case of updates, the default is to only update the columns corresponding to the tags present in the ROW element of the XML document. When specified, these columns alone will get updated in the update or insert statement. All other elements in the document will be ignored.

Syntax

```
public void setUpdateColumnList( java.lang.String[] updColNames );
```

Parameters

updColNmaes The string list of columns to be updated.

setXSLT()**Description**

Registers a XSL transform to be applied to generated XML. If a stylesheet was already registered, it gets replaced by the new one. To un-register the stylesheet pass in a NULL for the `stylesheet` argument. The options are described in the following table.

Syntax	Description
<code>public void setXSLT(java.io.Reader stylesheet, java.lang.String ref);</code>	The stylesheet parameter is passed in as the data.
<code>public void setXSLT(java.lang.String stylesheet, java.lang.String ref);</code>	The stylesheet parameter is passed in as a URI to the document.

Parameters

stylesheet The stylesheet URI.

ref URL for include, import and external entities.

setXSLTParam()

Description

Sets the value of a top-level stylesheet parameter. The parameter value is expected to be a valid XPath expression (note that string literal values would therefore have to be explicitly quoted). If no stylesheet is registered, this method is a no op.

Syntax

```
public void setXSLTParam(java.lang.String name,  
                        java.lang.String value);
```

Parameters

name	Parameter name.
value	Parameter value as an XPATH expression.

updateXML()

Description

Updates the table given the XML document. Returns the number of XML elements processed. This may or may not be equal to the number of database rows modified, depending on whether the rows selected through the XML document uniquely identify the rows in the table.

- The update requires a list of key columns which are used to uniquely identify a row to update in the given table. By default, the update uses the list of key columns and matches the values of the corresponding elements in the XML document to identify a particular row, subsequently updating all the columns in the table for which there is an equivalent element present in the XML document. Each ROW element present in the input document is treated as a separate update to the table.
- A list of columns to update can be supplied to update only desired columns and ignore any other elements present in the XML document. This is a very efficient method, because if there are more than one row present in the input XML document, the update statement itself is cached and batched.
- To set the list of all key column, use [setKeyColumnList\(\)](#).
- To set the list of columns to update, use [setUpdateColumnList\(\)](#).

The options are described in the following table.

Syntax	Description
<code>public int updateXML(org.w3c.dom.Document doc);</code>	Updates the table given the XML document in a DOM tree form.
<code>public int updateXML(java.io.InputStream xmlStream);</code>	Updates the table given the XML document in a stream form.
<code>public int updateXML(java.io.Reader xmlStream);</code>	Updates the table given the XML document in a stream form.
<code>public int updateXML(java.lang.String xmlDoc);</code>	Updates the table given the XML document in a string form.
<code>public int updateXML(java.net.URL url);</code>	Updates the columns in a database table, based on the element values in the supplied XML document.

Parameters

<code>doc</code>	The DOM tree form of the XML document
<code>xmlStream</code>	The stream form of the XML document
<code>xmlDoc</code>	The string form of the XML document
<code>url</code>	The URL to the document to use to update the table

Package oracle.xml.sql.query

This chapter describes the Java classes for XML SQL Utility for Java, which are contained in the oracle.xml.sql.query package. XML SQL Utility for Java (XSU) generates and stores XML from SQL queries.

This chapter contains these sections:

- [Package oracle.xml.sql.query Description](#)
- [OracleXMLQuery Class](#)
- [OracleXMLSQLException Class](#)
- [OracleXMLSQLNoRowsException Class](#)

Package oracle.xml.sql.query Description

The Java classes for XML SQL Utility for Java are contained in the oracle.xml.sql.query package. XML SQL Utility for Java (XSU) generates and stores XML data to and from the database from SQL queries or result sets or tables. It achieves data transformation by mapping canonically any SQL query result to XML, and vice versa.

See Also: [Chapter 8, "Package oracle.xml.sql.dml"](#)

OracleXMLQuery Class

Description

The OracleXMLQuery class generates XML given an SQL query.

Syntax

```
public class OracleXMLQuery extends java.lang.Object
```

```
java.lang.Object
|
+--oracle.xml.sql.query.OracleXMLQuery
```

Fields

Table 9–1 Summary of Fields of OracleXMLQuery

Field	Syntax	Description
DTD	public static final int DTD	Specifies that the DTD is to be generated
ERROR_TAG	public static final java.lang.String ERROR_TAG	Specifies the default tag name for the ERROR document
MAXROWS_ALL	public static final int MAXROWS_ALL	Specifies that all rows be included in the result
NONE	public static final int NONE	Specifies that no DTD is to be generated
ROW_TAG	public static final java.lang.String ROW_TAG	Specifies the default tag name for the ROW elements
ROWIDATTR_TAG	public static final java.lang.String ROWIDATTR_TAG	Specifies the default tag name for the ROW elements
ROWSET_TAG	public static final java.lang.String ROWSET_ TAG	Specifies the default tag name for the document
SCHEMA	public static final int SCHEMA	Specifies that an XML schema is to be generated
SKIPROWS_ALL	public static final int SKIPROWS_ALL	Specifies that all rows be skipped in the result.

Methods

Table 9–2 Summary of Methods of OracleXMLQuery

Method	Description
OracleXMLQuery() , on page 9-5	Class constructor.
close() , on page 9-6	Closes open resources created by the Oracle XML engine.
getNumRowsProcessed() , on page 9-6	Returns the number of rows processed.
getXMLDOM() , on page 9-6	Transforms data into an XML document.
getXMLDOM() , on page 9-6	Transforms object-relational data, specified in the constructor, into an XML document.
getXMLMetaData() , on page 9-7	Returns the DTD or XMLSchema for the XML document.
getXMLSAX() , on page 9-8	Transforms object-relational data, specified in the constructor, into an XML document.
getXMLSchema() , on page 9-8	Generates XMLSchema(s) corresponding to the specified query.
getXMLString() , on page 9-8	Transforms object-relational data, specified in the constructor, into an XML document.
keepObjectOpen() , on page 9-9	Has the effect of turning on and off the persistency of objects from which XML data is retrieved.
removeXSLTParam() , on page 9-9	Removes the value of a top-level stylesheet parameter.
setCollIdAttrName() , on page 9-10	Sets the name of the id attribute of the collection element's separator tag.
setDataHeader() , on page 9-10	Sets the XML data header.
setDateFormat() , on page 9-11	Sets the format of the generated dates in the XML doc.
setEncoding() , on page 9-11	Sets the encoding processing instruction in the XML doc.
setErrorTag() , on page 9-11	Sets the tag to be used to enclose the XML error docs.
setException() , on page 9-12	Allows the user to pass in an exception to be handled by the XSU.
setMaxRows() , on page 9-12	Sets the maximum number of rows to be converted to XML.
setMetaHeader() , on page 9-12	Sets the XML meta header.
setRaiseException() , on page 9-13	Instructs the XSU whether to throw the raised exceptions.

Table 9–2 (Cont.) Summary of Methods of OracleXMLQuery

Method	Description
setRaiseNoRowsException() , on page 9-13	Instructs the XSU whether to throw an <code>OracleXMLNoRowsException</code> when the generated XML doc is empty.
setRowIdAttrName() , on page 9-13	Sets the name of the id attribute of the row enclosing tag.
setRowIdAttrValue() , on page 9-14	Specifies the scalar column whose value will be assigned to the id attribute of the row enclosing tag.
setRowsetTag() , on page 9-14	Sets the tag to be used to enclose the XML dataset.
setSkipRows() , on page 9-15	Sets the number of rows to skip.
setSQLToXMLNameEscaping() , on page 9-15	Has the effect of turning on and off the escaping of XML tags in cases where mapped SQL object name would not make a valid XML identifier.
setStylesheetHeader() , on page 9-15	Sets the stylesheet header.
setXSLT() , on page 9-16	Registers an XSL transform to be applied to the generated XML.
setXSLTParam() , on page 9-16	Sets the value of a top-level stylesheet parameter.
useLowerCaseTagNames() , on page 9-17	Sets the tag names to lower case.
useNullAttributeIndicator() , on page 9-17	Specifies if NULLness is indicated by a special XML attribute or by omitting the entity from the XML document.
useTypeForCollElemTag() , on page 9-17	Instructs the XSU to use the collection element's type name as the collection element's tag name.
useUpperCaseTagNames() , on page 9-18	Sets the tag names to upper case.

OracleXMLQuery()

Description

Class constructor for the `OracleXMLQueryObject`. The options are described in the following table.

Syntax	Description
<code>public OracleXMLQuery(java.sql.Connection conn, java.sql.ResultSet rset);</code>	Creates an OracleXMLQuery from a database connection and a jdbc result set object.
<code>public OracleXMLQuery(java.sql.Connection conn, java.lang.String query);</code>	Creates an OracleXMLQuery from a database connection and an SQL query string.
<code>public OracleXMLQuery(oracle.xml.sql.dataset.OracleXMLDataSet dset);</code>	Creates an OracleXMLQuery from a dataset.

Parameters

<code>conn</code>	database connection
<code>rset</code>	jdbc result set object
<code>query</code>	the SQL query string
<code>dset</code>	dataset

close()

Description

Closes any open resource, created by the OracleXML engine. This will not close for instance result set supplied by the user.

Syntax

```
public void close();
```

getNumRowsProcessed()

Description

Returns the number of rows processed.

Syntax

```
public long getNumRowsProcessed();
```

getXMLDOM()

Description

Transforms the object-relational data, specified in the constructor, into XML. Returns a representation of the XML document. The options are described in the following table.

Syntax	Description
<code>public org.w3c.dom.Document getXMLDOM()</code>	Returns a DOM representation of the XML document.
<code>public org.w3c.dom.Document getXMLDOM(int metaType)</code>	The argument is used to specify the type of XML metadata the XSU is to generate along with the XML. Currently this value is ignored, and no XML metadata is generated. Returns a string representation of the XML document.
<code>public org.w3c.dom.Document getXMLDOM(org.w3c.dom.Node root)</code>	If not <code>NULL</code> , the argument is considered the "root" element of the XML doc. Returns the string representation of the XML document.
<code>public org.w3c.dom.Document getXMLDOM(org.w3c.dom.Node root, int metaType)</code>	If not <code>NULL</code> , the <code>root</code> argument is considered the "root" element of the XML doc. The <code>metaType</code> argument is used to specify the type of XML metadata the XSU is to generate along with the XML. Currently this value is ignored, and no XML metadata is generated. Returns the string representation of the XML document.

Parameters

<code>metaType</code>	the type of XML metadata (NONE, SCHEMA)
<code>root</code>	root node to which to append the new XML

getXMLMetaData()

Description

This functions returns the DTD or the XMLSchema for the XML document which would have been generated by a `getXML*()` call, such as `getXMLDOM()`, `getXMLSAX()`, `getXMLSchema()`, or `getXMLString()`.

Syntax

```
public java.lang.String getXMLMetaData( int metaType,  
boolean withVer);
```

Parameters

<code>metaType</code>	Specifies the type of XML metadata to be generated (NONE or DTD).
<code>withVer</code>	Specifies whether to generate the version processing instruction

getXMLSAX()

Description

Transforms the object-relational data, specified in the constructor, into an XML document.

Syntax

```
public void getXMLSAX(org.xml.sax.ContentHandler sax);
```

Parameters

sax ContentHandler object to be registered.

getXMLSchema()

Description

This methods generates the XML Schema(s) corresponding to the specified query; returns the XML Schema(s).

Syntax

```
public org.w3c.dom.Document[] getXMLSchema();
```

getXMLString()

Description

Transforms the object-relational data, specified in the constructor, into a XML document. Returns the string representation of the XML document. The options are described in the following table.

Syntax	Description
<code>public java.lang.String getXMLString();</code>	Takes no arguments.
<code>public java.lang.String getXMLString(int metaType);</code>	The <code>metaType</code> argument is used to specify the type of XML metadata the XSU is to generate along with the XML.
<code>public java.lang.String getXMLString(org.w3c.dom.Node root);</code>	If not <code>NULL</code> , the <code>root</code> argument, is considered the root element of the XML doc.

(Cont.) Syntax	Description
<pre>public java.lang.String getXMLString(org.w3c.dom.Node root, int metaType)</pre>	<p>If not NULL, the <code>root</code> argument is considered the root element of the XML doc. The <code>metaType</code> argument is used to specify the type of XML metadata the XSU is to generate along with the XML. Note that if the <code>root</code> argument is non-NULL, no DTD is generated even if requested.</p>

Parameters

<code>metaType</code>	The type of XML metadata (NONE, DTD, or SCHEMA, static fields of this class)
<code>root</code>	root node to which to append the new XML

keepObjectOpen()

Description

The default behavior for all the `getXML*()` functions which DO NOT TAKE in a `ResultSet` object, such as `getXMLDOM()`, `getXMLSAX()`, `getXMLSchema()`, or `getXMLString()`, is to close the `ResultSet` object and `Statement` objects at the end of the call. If the persistent feature is needed, where by calling `getXML()` repeatedly the next set of rows is obtained, this behavior must be turned off by calling this function with value `TRUE`. `OracleXMLQuery` would not close the `ResultSet` and `Statement` objects after the `getXML()` calls. To close the cursor state, the `close()` function must be called explicitly.

Syntax

```
public void keepObjectOpen( boolean alive);
```

Parameters

<code>alive</code>	Should the object be kept open?
--------------------	---------------------------------

removeXSLTParam()

Description

Removes the value of a top-level stylesheet parameter. NOTE: if no stylesheet is registered, this method is a no op.

Syntax

```
public void removeXSLTParam( String name);
```

Parameters

name	Parameter name
------	----------------

setCollIdAttrName()**Description**

Sets the name of the `id` attribute of the collection element's separator tag. Passing `NULL` or an empty string causes the `row id` attribute to be omitted.

Syntax

```
public void setCollIdAttrName( String attrName);
```

Parameters

attrName	Attribute Name
----------	----------------

setDataHeader()**Description**

Sets the XML data header, the XML entity which is appended at the beginning of the query-generated XML entity (i.e., rowset). The two entities are enclosed by the tag specified via the `docTag` argument. The last data header specified is the one that is used. Passing in `NULL` for the `header` parameter unsets the data header.

Syntax

```
public void setDataHeader( java.io.Reader header,  
                          java.lang.String docTag);
```

Parameters

header	Header
docTag	Tag used to enclose the data header and the rowset

setDateFormat()

Description

Sets the format of the generated dates in the XML doc. The syntax of the date format pattern (i.e. the date mask), should conform to the requirements of the `java.text.SimpleDateFormat` class. Setting the mask to `NULL` or an empty string, unsets the date mask.

Syntax

```
public void setDateFormat( java.lang.String mask);
```

Parameters

mask The data mask

setEncoding()

Description

Sets the encoding processing instruction (PI) in the XML doc. If `NULL` or an empty string are specified as the encoding, then the default charset is specified in the encoding PI.

Syntax

```
public void setEncoding( java.lang.String enc)
```

Parameters

enc Encoding of the XML doc (IANA name of encoding)

setErrorTag()

Description

Sets the tag to be used to enclose the XML error docs.

Syntax

```
public void setErrorTag( java.lang.String tag);
```

Parameters

tag Tag name

setException()

Description

Allows the user to pass in an exception, and have the XSU handle it.

Syntax

```
public void setException( java.lang.Exception e);
```

Parameters

e The exception to be processed by XSU

setMaxRows()

Description

Sets the maximum number of rows to be converted to XML. By default there is no maximum set. To explicitly specify no max, see MAXROWS_ALL field.

Syntax

```
public void setMaxRows( int rows);
```

Parameters

rows Maximum number of rows to generate

setMetaHeader()

Description

Sets the XML meta header. When set, the header is inserted at the beginning of the metadata part (DTD or XMLSchema) of each XML document generated by this object. The last meta header specified is the one that is used. Setting the header to NULL or an empty string unsets the meta header.

Syntax

```
public void setMetaHeader( java.io.Reader header);
```

Parameters

header Header

setRaiseException()

Description

Instructs the XSU whether to throw the raised exceptions. If this call isn't made, or if `FALSE` is passed to the `flag` argument, the XSU catches the SQL exceptions and generates an XML doc from the exception message.

Syntax

```
public void setRaiseException(boolean flag);
```

Parameters

`flag` Should the raised exception be thrown?

setRaiseNoRowsException()

Description

Instructs the XSU whether to throw an `OracleXMLNoRowsException` when the generated XML doc is empty. By default, the exception is not thrown.

Syntax

```
public void setRaiseNoRowsException( boolean flag);
```

Parameters

`flag` Should the `OracleXMLNoRowsException` be thrown if no data found?

setRowIdAttrName()

Description

Sets the name of the id attribute of the row enclosing tag. Passing `NULL` or an empty string causes the row id attribute to be omitted.

Syntax

```
public void setRowIdAttrName( java.lang.String attrName);
```

Parameters

`attrName` Attribute name

setRowIdAttrValue()

Description

Specifies the scalar column whose value is to be assigned to the id attribute of the row enclosing tag. Passing NULL or an empty string causes the row id attribute to be assigned the row count value (i.e. 0, 1, 2, ...).

Syntax

```
public void setRowIdAttrValue( java.lang.String colName);
```

Parameters

colName Column whose value will be assigned to the row id attribute

setRowsetTag()

Description

Sets the tag to be used to enclose the XML dataset.

Syntax

```
public void setRowsetTag( java.lang.String tag);
```

Parameters

tag Tag name

setRowTag()

Description

Sets the tag to be used to enclose the XML element corresponding to a db. record.

Syntax

```
public void setRowTag( java.lang.String tag);
```

Parameters

tag Tag name

setSkipRows()

Description

Sets the number of rows to skip. By default 0 rows are skipped. To skip all the rows use SKIPROWS_ALL.

Syntax

```
public void setSkipRows(int rows);
```

Parameters

rows Number of rows to skip.

setSQLToXMLNameEscaping()

Description

This turns on or off escaping of XML tags in the case that the SQL object name, which is mapped to a XML identifier, is not a valid XML identifier.

Syntax

```
public void setSQLToXMLNameEscaping( boolean flag);
```

Parameters

flag Whether to turn on SQL to XML identifier escaping.

setStylesheetHeader()

Description

Sets the stylesheet header (i.e. stylesheet processing instructions) in the generated XML doc. Passing NULL in the argument will unset the stylesheet header and the stylesheet type. The options are described in the following table.

Syntax	Description
public void setStylesheetHeader(java.lang.String uri);	Sets stylesheet header using the stylesheet URI.
public void setStylesheetHeader(java.lang.String uri, java.lang.String type);	Sets the stylesheet header using the stylesheet URI and the stylesheet type.

Parameters

uri	Stylesheet URI
type	Stylesheet type; defaults to 'text/xsl'

setXSLT()**Description**

Registers a XSL transform to be applied to generated XML. If a stylesheet is already registered, it is replaced by the new one. To un-register the stylesheet, pass in `NULL` value for the argument. The options are described in the following table.

Syntax	Description
<code>public void setXSLT(java.io.Reader stylesheet, java.lang.String ref);</code>	The stylesheet parameter is passed in as the data.
<code>public void setXSLT(j ava.lang.String stylesheet, java.lang.String ref);</code>	The stylesheet parameter is passed in as a URI to the document.

Parameters

stylesheet	The stylesheet.
ref	URL for include, import and external entities.

setXSLTParam()**Description**

Sets the value of a top-level stylesheet parameter. The parameter value is expected to be a valid XPath expression; therefore the string literal values have to be explicitly quoted). If no stylesheet is registered, this method is a no op.

Syntax

```
public void setXSLTParam( java.lang.String name,
                        java.lang.String value);
```

Parameters

name	Parameter name
value	Parameter value as an XPATH expression

useLowerCaseTagNames()

Description

This will set the case to be lower for all tag names. Note, make this call after all the desired tags have been set.

Syntax

```
public void useLowerCaseTagNames();
```

useNullAttributeIndicator()

Description

Specifies if NULLness is indicated by a special XML attribute or by omitting the entity from the XML document.

Syntax

```
public void useNullAttributeIndicator(boolean flag);
```

Parameters

flag	Should the attribute be used to indicate NULL?
------	--

useTypeForCollElemTag()

Description

By default, the tag name for elements of a collection is the collection's tag name followed by "_item". This method, when called with argument value of `TRUE`, instructs the XSU to use the collection element's type name as the collection element's tag name.

Syntax

```
public void useTypeForCollElemTag( boolean flag);
```

Parameters

flag	Should the column element type be used to indicate its tag name?
------	--

useUpperCaseTagNames()

Description

Sets all tag names to upper case. This call should be made only after all the desired tags have been set.

Syntax

```
public void useUpperCaseTagNames();
```


OracleXMLSQLException Class

Description

Class for managing all exceptions thrown by the XSU.

Syntax

```
public class OracleXMLSQLException extends java.lang.RuntimeException
```

```

java.lang.Object
|
+-- java.lang.Throwable
    |
    +-- java.lang.Exception
        |
        +-- java.lang.RuntimeException
            |
            +-- oracle.xml.sql.OracleXMLSQLException

```

Direct Subclasses of OracleXMLSQLException

- [OracleXMLSQLNoRowsException Class](#)

Implemented Interfaces of OracleXMLSQLException

- `java.io.Serializable`

Methods of OracleXMLSQLException

Table 9–3 Summary of Methods of OracleXMLSQLException

Method	Description
OracleXMLSQLException()	Creates a new OracleXMLSQLException.
getErrorCode()	Returns the SQL error code thrown.
getParentException()	Returns the original exception, if there was one; otherwise, returns NULL.
getXMLErrorString()	Prints XML error string with given error tag name.
getXMLSQLExceptionString()	Prints the SQL parameters in the error message.
setErrorTag()	Sets error tag used to generate XML error reports.

OracleXMLSQLException()

Description

Creates a new OracleXMLSQLException. The options are described in the following table.

Syntax	Description
<code>public OracleXMLSQLException(java.lang.Exception e);</code>	Sets the parent exception as passed in.
<code>public OracleXMLSQLException(java.lang.Exception e, java.lang.String errorTagName);</code>	Sets the error tag name as passed in.
<code>public OracleXMLSQLException(java.lang.String message);</code>	Sets the error message to be returned.
<code>public OracleXMLSQLException(java.lang.String message, java.lang.Exception e);</code>	Sets the parent exception and the error message to be returned.
<code>public OracleXMLSQLException(java.lang.String message, java.lang.Exception e, java.lang.String errorTagName);</code>	Sets the error message, parent exception, and error tag to be used.
<code>public OracleXMLSQLException(java.lang.String message, int errorCode);</code>	Sets the error message and SQL error code.
<code>public OracleXMLSQLException(java.lang.String message, int errorCode, java.lang.String errorTagName);</code>	Sets the error message, SQL error code, and the error tag to be used.
<code>public OracleXMLSQLException(java.lang.String message, java.lang.String errorTagName);</code>	Sets the error message and the error tag to be used.

Parameters

<code>e</code>	The exception.
<code>errorTagName</code>	The error tag name.
<code>message</code>	The error message.
<code>errorCode</code>	the SQL error code.

getErrorCode()

Description

Returns the SQL error code thrown.

Syntax

```
public int getErrorCode();
```

getParentException()**Description**

Returns the original exception, if there was one; otherwise, returns NULL.

Syntax

```
public java.lang.Exception getParentException();
```

getXMLErrorString()**Description**

Prints the XML error string with the given error tag name.

Syntax

```
public java.lang.String getXMLErrorString();
```

getXMLSQLExceptionString()**Description**

Prints the SQL parameters as well in the error message.

Syntax

```
public java.lang.String getXMLSQLExceptionString();
```

setErrorTag()**Description**

Sets the error tag name, which is then used by [getXMLErrorString\(\)](#) and [getXMLSQLExceptionString\(\)](#) to generate XML error reports.

Syntax

```
public void setErrorTag(java.lang.String tagName);
```

Parameters

tagName	The tag name of the error
---------	---------------------------

OracleXMLSQLNoRowsException Class

Description

The exception that can be thrown when no rows are found.

Syntax

```
public class OracleXMLSQLNoRowsException extends OracleXMLSQLException
```

```
java.lang.Object
|
+-- java.lang.Throwable
    |
    +-- java.lang.Exception
        |
        +-- java.lang.RuntimeException
            |
            +-- OracleXMLSQLException
                |
                +-- oracle.xml.sql.OracleXMLSQLNoRowsException
```

Implemented Interfaces of OracleXMLSQLNoRowsException

```
java.io.Serializable
```

Methods

OracleXMLSQLNoRowsException()

Creates a new `OracleXMLSQLNoRowsException`. The options are described in the following table.

Syntax	Description
<code>public OracleXMLSQLNoRowsException();</code>	Default class constructor.
<code>public OracleXMLSQLNoRowsException(java.lang.String errorTag);</code>	Sets the error tag as the passed in argument.

Parameters

errorTag The error tag.

Inherited Members of OracleXMLSQLNoRowsException

Table 9–4 *Summary of Inherited Members of OracleXMLSQLNoRowsException*

Member	Inherited from
getErrorCode()	OracleXMLSQLException Class
getParentException()	OracleXMLSQLException Class
getXMLErrorMessageString()	OracleXMLSQLException Class
getXMLSQLExceptionString()	OracleXMLSQLException Class
setErrorTag()	OracleXMLSQLException Class

Package oracle.xml.util

This chapter documents package `oracle.xml.util`. This package contains the utility classes that provide error handling and support extensions to XSLT Processor for Java in the XDK for Java.

The full functionality of the XML Parser, DOM, and SAX APIs are contained in the `oracle.xml.parser.v2` package, which is documented in [Chapter 11, "Package oracle.xml.parser.v2"](#) in this manual.

This chapter contains these sections:

- [Package oracle.xml.util Description](#)
- [Package oracle.xml.util Summary](#)

Package oracle.xml.util Description

The classes contained in oracle.xml.util provide error handling and extensions to the XSLT Processor for Java, which is provided in the Oracle XDK for Java. The classes in this package support the W3C XML standard.

For descriptions of the classes that implement the Java DOM Parser and the XSLT Processor, refer to [Chapter 11, "Package oracle.xml.parser.v2"](#) in this manual.

See Also: For information about developing applications using the XDK, refer to *Oracle9i XML Developer's Kits Guide - XDK*.

Package oracle.xml.util Summary

Table 10–1 oracle.xml.util Package Interfaces and Classes

Interface or Class	Description
NSName	Public interface that provides Namespace support for Element and Attr names.
XMLError	Class that holds the error message and the line number where it occurred.
XMLException	Class that indicates that a parsing exception occurred while processing an XML document.

NSName

Description

Interface in package oracle.xml.util. This interface provides Namespace support for Element and Attr names.

Syntax

```
public interface NSName
```

Methods

getExpandedName()

Description

Get the fully resolved name for this name

Syntax

```
public java.lang.String getExpandedName()
```

Returns

The fully resolved name

getLocalName()

Description

Get the local name for this name

Syntax

```
public java.lang.String getLocalName()
```

Returns

The local name

getNamespace()

Description

Get the resolved Namespace for this name

Syntax

```
public java.lang.String getNamespace()
```

Returns

The resolved Namespace

getPrefix()**Description**

Get the prefix for this name

Syntax

```
public java.lang.String getPrefix()
```

Returns

The prefix

getQualifiedName()**Description**

Get the qualified name

Syntax

```
public java.lang.String getQualifiedName()
```

Returns

The qualified name

XMLERror

Description

This class holds the error message and the line number where it occurred

Syntax

```
public class XMLERror
```

```
oracle.xml.util.XMLERror
```

Fields

Table 10–2 *Fields of oracle.xml.util.XMLERror*

Field	Syntax
col	protected int[] col
errid	protected int[] errid
exp	protected java.lang.Exception[] exp
line	protected int[] line
mesg	protected java.lang.String[] mesg
publd	protected java.lang.String[] publd
sysld	protected java.lang.String[] sysld
types	protected int[] types

Constructors

XMLERror()

Description

Default constructor

Syntax

```
public XMLERror()
```

Methods

Table 10–3 Summary of Methods of oracle.xml.util.XML_Error

Method	Description
<code>error(int, int, String)</code>	Adds a new error to the vector
<code>error(int, int, String[])</code>	Adds a new error to the vector
<code>error0(int, int)</code>	Adds a new error to the vector
<code>error1(int, int, String)</code>	Adds a new error to the vector
<code>error2(int, int, String, String)</code>	Adds a new error to the vector
<code>error3(int, int, String, String, String)</code>	Adds a new error to the vector
<code>flushErrorStream()</code>	Flush all the error to the output stream output stream defaults or to error handler
<code>formatErrorMesg(int)</code>	
<code>getColumnNumber()</code>	Get the column number of error at specified index
<code>getException(int)</code>	Get the exception (if exists) that occurred in error at specified index
<code>getFirstError()</code>	Get first error
<code>getLineNumber(int)</code>	Get the line number of error at specified index
<code>getLocator()</code>	Return the registered locator
<code>getMessage(int)</code>	Get the error message at specified index
<code>getMessage(int, String[])</code>	Get error message with more than 5 arguments
<code>getMessage0(int)</code>	Get error message with no arguments
<code>getMessage1(int, String)</code>	Get error message with 1 argument
<code>getMessage3(int, String, String)</code>	Get error message with 2 arguments
<code>getMessage3(int, String, String, String)</code>	Get error message with 3 arguments
<code>getMessage4(int, String, String, String, String)</code>	Get error message with 4 arguments
<code>getMessage5(int, String, String, String, String, String)</code>	Get error message with 5 arguments
<code>getMessageType(int)</code>	Get the type of the error message at specified index

Table 10–3 (Cont.) Summary of Methods of *oracle.xml.util.XMLError*

Method	Description
<code>getNumMessages()</code>	Return the total number of errors/warnings found during parsing
<code>getPublicId(int)</code>	Get the public ID of input when error at specified index occurred
<code>getSystemId(int)</code>	Get the system ID of input when error at specified index occurred
<code>printErrorListener()</code>	Flush all the JAXP 1.1 errors to the <code>ErrorListener</code> If no <code>ErrorListener</code> was set, default to <code>System.err</code>
<code>reset()</code>	Rset the error class
<code>setErrorStream(OutputStream)</code>	Register an output stream
<code>setErrorStream(OutputStream, String)</code>	Register an output stream
<code>setErrorStream(PrintWriter)</code>	Register an output stream
<code>setException(Exception)</code>	Register a exception
<code>setLocale(Locale)</code>	Register a locale
<code>setLocator(Locator)</code>	Register a locator
<code>showWarnings(boolean)</code>	Turn reporting warning on/off

error(int, int, String)**Description**

Adds a new error to the vector

Syntax

```
public void error(int id, int type, java.lang.String msg)
```

Parameters

`id` - - id of the error message

`msg` - - error message (without paramters)

`type` - - type of the error

error(int, int, String[])

Description

Adds a new error to the vector

Syntax

```
public void error(int id, int type, java.lang.String[] p)
```

Parameters

`id` -- id of the error message

`type` -- type of the error

`p` -- parameter array

error0(int, int)

Description

Adds a new error to the vector

Syntax

```
public void error0(int id, int type)
```

Parameters

`id` -- id of the error message

`type` -- type of the error

error1(int, int, String)

Description

Adds a new error to the vector

Syntax

```
public void error1(int id, int type, java.lang.String p1)
```

Parameters

`id` -- id of the error message

`type` -- type of the error

`p1` -- parameter 1

error2(int, int, String, String)

Description

Adds a new error to the vector

Syntax

```
public void error2(int id, int type, java.lang.String p1, java.lang.String p2)
```

Parameters

`id` -- id of the error message

`type` -- type of the error

`p1` -- paramter 1

`p2` -- paramter 2

error3(int, int, String, String, String)

Description

Adds a new error to the vector

Syntax

```
public void error3(int id, int type, java.lang.String p1, java.lang.String p2,  
java.lang.String p3)
```

Parameters

`id` -- id of the error message

`type` -- type of the error

`p1` -- paramter 1

`p2` -- paramter 2

`p3` -- paramter 3

flushErrorStream()

Description

Flush all the error to the ouput stream output stream defaults or to error handler

Syntax

```
public void flushErrorStream()
```


formatErrorMesg(int)

Syntax

```
public java.lang.String formatErrorMesg(int index)
```

getColumnNumber(int)

Description

Get the column number of error at specified index

Syntax

```
public int getColumnNumber(int i)
```

Returns

The column number

getException(int)

Description

Get the exception (if exists) that occurred in error at specified index

Syntax

```
public java.lang.Exception getException(int i)
```

Returns

The exception

getFirstError()

Syntax

```
public int getFirstError()
```

getLineNumber(int)

Description

Get the line number of error at specified index

Syntax

```
public int getLineNumber(int i)
```

Returns

The line number

getLocator()

Description

Return the registered locator

Syntax

```
public org.xml.sax.Locator getLocator()
```

Returns

locator

getMessage(int)

Description

Get the error message at specified index

Syntax

```
public java.lang.String getMessage(int i)
```

Returns

The error message

getMessage(int, String[])

Description

Get error message with more than 5 arguments

Syntax

```
public java.lang.String getMessage(int errId, java.lang.String[] params)
```

getMessage0(int)

Description

Get error message with no arguments

Syntax

```
public java.lang.String getMessage0(int errId)
```

getMessage1(int, String)**Description**

Get error message with 1 arguments

Syntax

```
public java.lang.String getMessage1(int errId, java.lang.String a1)
```

getMessage2(int, String, String)**Description**

Get error message with 2 arguments

Syntax

```
public java.lang.String getMessage2(int errId, java.lang.String a1,  
java.lang.String a2)
```

getMessage3(int, String, String, String)**Description**

Get error message with 3 arguments

Syntax

```
public java.lang.String getMessage3(int errId, java.lang.String a1,  
java.lang.String a2, java.lang.String a3)
```

getMessage4(int, String, String, String, String)**Description**

Get error message with 4 arguments

Syntax

```
public java.lang.String getMessage4(int errId, java.lang.String a1,  
java.lang.String a2, java.lang.String a3, java.lang.String a4)
```

getMessage5(int, String, String, String, String, String)

Description

Get error message with 5 arguments

Syntax

```
public java.lang.String getMessage5(int errId, java.lang.String a1,  
java.lang.String a2, java.lang.String a3, java.lang.String a4, java.lang.String  
a5)
```

getMessageType(int)

Description

Get the type of the error message at specified index

Syntax

```
public int getMessageType(int i)
```

Returns

The error message type

getNumMessages()

Description

Return the total number of errors/warnings found during parsing

Syntax

```
public int getNumMessages()
```

Returns

The number of errors/warnings

getPublicId(int)

Description

Get the public ID of input when error at specified index occurred

Syntax

```
public java.lang.String getPublicId(int i)
```

Returns

The public ID

getSystemId(int)**Description**

Get the system ID of input when error at specified index occurred

Syntax

```
public java.lang.String getSystemId(int i)
```

Returns

The system ID

printErrorListener()**Description**

Flush all the JAXP 1.1 errors to the ErrorListener If no ErrorListener was set, default to System.err

Syntax

```
public void printErrorListener()
```

reset()**Description**

Reset the error class

Syntax

```
public void reset()
```

setErrorStream(OutputStream)**Description**

Register an output stream

Syntax

```
public void setErrorStream(java.io.OutputStream out)
```

Parameters

out - - OutputStream to output the error/warnings

setErrorStream(OutputStream, String)

Description

Register an output stream

Syntax

```
public void setErrorStream(java.io.OutputStream out, java.lang.String enc)
```

Parameters

out - - OutputStream to output the error/warnings

enc - - Encoding of the outputstream

Throws

IOException - - if an error occurs initializing the outputstream

setErrorStream(PrintWriter)

Description

Register an output stream

Syntax

```
public void setErrorStream(java.io.PrintWriter out)
```

Parameters

out - - PrintWriter to output the error/warnings

setException(Exception)

Description

Register an exception

Syntax

```
public void setException(java.lang.Exception exp)
```

Parameters

`exp` -- last exception occurred

setLocale(Locale)**Description**

Register a locale

Syntax

```
public void setLocale(java.util.Locale locale)
```

Parameters

`locale` -- locale for error reporting

setLocator(Locator)**Description**

Register a locator

Syntax

```
public void setLocator(org.xml.sax.Locator locator)
```

Parameters

`locator` -- locator to get lin/col/sysid/pubid info

showWarnings(boolean)**Description**

Turn reporting warning on/off

Syntax

```
public void showWarnings(boolean flag)
```

Parameters

`flag` -- controls reporting of warnings

XMLException

Description

In package oracle.xml.util. Indicates that a parsing exception occurred while processing an XML document

Syntax

```
public class XMLException extends java.lang.Exception
```

```
java.lang.Object
|
+--java.lang.Throwable
   |
   +--java.lang.Exception
      |
      +--oracle.xml.util.XMLException
```

Implemented Interfaces

java.io.Serializable

Fields

Table 10–4 *Fields of AttrDecl*

Field	Syntax	Description
ERROR	public static final int ERROR	Code for non-fatal error
FATAL_ERROR	public static final int FATAL_ERROR	Code for fatal error
WARNING	public static final int WARNING	Code for warning

Constructors

XMLException(String, String, String, int, int, int)

Syntax

```
public XMLException(java.lang.String msg, java.lang.String pubId,
java.lang.String sysId, int line, int col, int type)
```


XMLException(XMLError, Exception)

Syntax

```
public XMLException(XMLError err, java.lang.Exception e)
```

XMLException(XMLError, int)

Syntax

```
public XMLException(XMLError err, int firsterr)
```

XMLException(XMLError, int, Exception)

Syntax

```
public XMLException(XMLError err, int firsterr, java.lang.Exception e)
```

Methods

Table 10–5 Summary of Methods of oracle.xml.util.XMLException

Method	Description
formatErrorMessage(int)	Get the error message at specified index
getColumnNumber(int)	Get the column number of error at specified index
getException(int)	Get the exception (if exists) that occurred in error at specified index
getLineNumber(int)	Get the line number of error at specified index
getMessage(int)	Get the error message at specified index
getMessageType(int)	Get the type of the error message at specified index
getNumMessages(int)	Return the total number of errors/warnings found during parsing
getPublicId(int)	Get the public ID of input when error at specified index occurred
getSystemId(int)	Get the system ID of input when error at specified index occurred
getXMLError	Get XMLError object inside XMLException.
printStackTrace()	Prints this Throwable and its backtrace to the standard error stream.
printStackTrace(PrintStream)	Prints this Throwable and its backtrace to the specified print stream.
printStackTrace(PrintWriter)	Prints this Throwable and its backtrace to the specified print writer.
setException(Exception)	Set the underlying exception (if exists)

Table 10–5 (Cont.) Summary of Methods of *oracle.xml.util.XMLEException*

Method	Description
<code>toString()</code>	Override <code>toString</code> to pick up a <code>RuntimeException</code> embedded exception.

formatErrorMessage(int)

Description

Get the error message at specified index

Syntax

```
public java.lang.String formatErrorMessage(int i)
```

Returns

The error message

getColumnNumber(int)

Description

Get the column number of error at specified index

Syntax

```
public int getColumnNumber(int i)
```

Returns

The column number

getException(int)

Description

Get the exception (if exists) that occurred in error at specified index

Syntax

```
public java.lang.Exception getException(int i)
```

Returns

The exception

getLineNumber(int)**Description**

Get the line number of error at specified index

Syntax

```
public int getLineNumber(int i)
```

Returns

The line number

getMessage(int)

Get the error message at specified index

Description**Syntax**

```
public java.lang.String getMessage(int i)
```

Returns

The error message

getMessageType(int)**Description**

Get the type of the error message at specified index

Syntax

```
public int getMessageType(int i)
```

Returns

The error message type

getNumMessages()

Return the total number of errors/warnings found during parsing

Description

Syntax

```
public int getNumMessages()
```

Returns

The number of errors/warnings

getPublicId(int)

Description

Get the public ID of input when error at specified index occurred

Syntax

```
public java.lang.String getPublicId(int i)
```

Returns

The public ID

getSystemId(int)

Description

Get the system ID of input when error at specified index occurred

Syntax

```
public java.lang.String getSystemId(int i)
```

Returns

The system ID

getXMLError()

Description

Get XMLError object inside XMLException.

Syntax

```
public XMLError getXMLError()
```

Returns

the XMLError object.

printStackTrace()**Description**

Prints this Throwable and its backtrace to the standard error stream.

Syntax

```
public void printStackTrace()
```

Overrides

```
java.lang.Throwable.printStackTrace() in class  
java.lang.Throwable
```

printStackTrace(PrintStream)**Description**

Prints this Throwable and its backtrace to the specified print stream.

Syntax

```
public void printStackTrace(java.io.PrintStream s)
```

Overrides

```
java.lang.Throwable.printStackTrace(java.io.PrintStream) in class  
java.lang.Throwable
```

printStackTrace(PrintWriter)**Description**

Prints this Throwable and its backtrace to the specified print writer.

Syntax

```
public void printStackTrace(java.io.PrintWriter s)
```

Overrides

```
java.lang.Throwable.printStackTrace(java.io.PrintWriter) in  
class java.lang.Throwable
```

setException(Exception)

Description

Set the underlying exception (if exists)

Syntax

```
public void setException(java.lang.Exception ex)
```

Parameters

ex - exception

toString()

Description

Override toString to pick up aReturnsny embedded exception.

```
public java.lang.String toString()
```

Overrides

java.lang.Throwable.toString() in class java.lang.Throwable

Returns

A string representation of this exception

Package oracle.xml.parser.v2

This chapter describes the classes contained in package `oracle.xml.parser.v2`. These classes implement the XML Parser, the DOM, and the SAX APIs in the Oracle XDK for Java. (DOM is for Document Object Model, and SAX is for Simple API for XML.) The full functionality of XML Parser, DOM, and SAX APIs are contained in the `oracle.xml.parser.v2` package.

This chapter contains these sections:

- [Package oracle.xml.parser.v2 Description](#)
- [Package oracle.xml.parser.v2 Summary](#)
- [XSLT Processor Classes](#)

Note: The supporting utility classes are found in package `oracle.xml.util`, which is documented in [Chapter 10, "Package oracle.xml.util"](#) in this manual.

Package oracle.xml.parser.v2 Description

The classes contained in package `oracle.xml.parser.v2` implement the APIs for the XML Parser, DOM, and SAX in the Oracle9i XDK for Java. The classes that implement the XSLT Processor for Java are also contained in the `oracle.xml.parser.v2` package.

The Oracle implementations of the *DOM* and *SAX* APIs adhere to the World Wide Web Consortium (W3C) recommendations for the XML standard.

The supporting utility classes are found in package `oracle.xml.util`, which is documented in [Chapter 10, "Package oracle.xml.util"](#) in this manual.

See Also:

- [Chapter 10, "Package oracle.xml.util"](#) in this manual
- *Oracle9i XML Developer's Kits Guide - XDK*
- *Oracle9i XML API Reference - XDK and Oracle XML DB*

Package oracle.xml.parser.v2 Summary

The tables in this section summarize the oracle.xml.parser.v2 interfaces and classes documented in this chapter.

Table 11–1 *oracle.xml.parser.v2 Package Interfaces*

Interface	Description
NSResolver Interface	Provides support for resolving Namespaces.
PrintDriver Interface	The <code>PrintDriver</code> interface defines methods used to print XML documents represented as DOM trees.
XMLToken Interface	Basic interface for <code>XMLToken</code> .

Table 11–2 *oracle.xml.parser.v2 Package Classes*

Class	Description
AttrDecl	Holds information about each attribute declared in an attribute list in the Document Type Definition (DTD).
DefaultXMLDocumentHandler	Implements the default behaviour for the <code>XMLDocumentHandler</code> interface.
DOMParser	Implements an eXtensible Markup Language (XML) 1.0 parser according to the World Wide Web Consortium (W3C) recommendation.
DTD	Implements the DOM <code>DocumentType</code> interface and holds the Document Type Definition (DTD) information for an XML document.
ElementDecl	Represents an element declaration in a Document Type Definition DTD.
NodeFactory	Specifies methods to create various nodes of the DOM tree built during parsing.
oraxml	Implements the XML Decl Processing Instruction.
SAXAttrList	Implements the SAX <code>AttributeList</code> interface and also provides Namespace support.
SAXParser	Implements an eXtensible Markup Language (XML) 1.0 SAX parser according to the World Wide Web Consortium (W3C) recommendation.

Table 11–2 (Cont.) oracle.xml.parser.v2 Package Classes

Class	Description
XMLAttr	Implements the DOM Attr interface and holds information on each attribute of an element.
XMLCDATA	Implements the DOM CDATASection interface.
XMLComment	Implements the DOM Comment interface.
XMLDeclPI	Implements the XML Decl Processing Instruction.
XMLDocument	This class implements the DOM Document interface, represents an entire XML document and serves the root of the Document Object Model tree.
XMLDocumentFragment	This class implements the DOM DocumentFragment interface.
XMLDOMImplementation	Implements the DOMImplementation.
XMLElement	This class implements the DOM Element interface.
XMLEntityReference	Implements DOM EntityReference interface.
XMLNode	Implements the DOM Node interface and serves as the primary datatype for the entire Document Object Model.
XMLParser	This class serves as a base class for the DOMParser and SAXParser classes.
XMLPI	This class implements the DOM Processing Instruction interface.
XMLText	This class implements the DOM Text interface.
XMLTokenizer	This class implements an eXtensible Markup Language (XML) 1.0 parser according to the World Wide Web Consortium (W3C) recommendation.

Table 11–3 oracle.xml.parser.v2 Package Exceptions

Exception	Description
XMLDOMException	Indicates an exception in DOM operation.
XMLParseException	Indicates that a parsing exception occurred while processing an XML document
XMLRangeException	Indicates an exception in DOM Range operation.

The classes listed in [Table 11-4, " Summary of XSLT Processor Classes in package oracle.xml.parserv2"](#) summarize the XSLT Processor classes contained in the oracle.xml.parser.v2 package.

Table 11-4 Summary of XSLT Processor Classes in package oracle.xml.parserv2

Class	Description
oraxsl Class	Provides a command-line interface to applying stylesheets on multiple XML documents.
XPathException Class	Indicates that an exception occurred during XSL transformation.
XSLProcessor Class	Provides methods to transform an input XML document using a previously constructed XSLStylesheet.
XSLStylesheet Class	Holds XSL stylesheet information such as templates, keys, variables, and attribute sets.

NSResolver Interface

Syntax of NSResolver

```
public interface NSResolver
```

Description of NSResolver

This interface provides support for resolving Namespaces

Known Implementing Classes of NSResolver

`XML`Element

Methods

resolveNamespacePrefix(String)

Description

Find the namespace definition in scope for a given namespace prefix

Syntax

```
public java.lang.String resolveNamespacePrefix(java.lang.String prefix)
```

Parameters

`prefix` - Namespace prefix to be resolved

Returns

the resolved Namespace (null, if prefix could not be resolved)

PrintDriver Interface

Description of PrintDriver

The `PrintDriver` interface defines methods used to print XML documents represented as DOM trees.

Syntax of PrintDriver

```
public interface PrintDriver
```

Implementing Classes of PrintDriver

```
XMLPrintDriver
```

Methods

Table 11–5 Summary of Methods of PintDriver

Method	Description
<code>close()</code>	Closes the output stream or print writer
<code>flush()</code>	Flushes the output stream or print writer
<code>printAttribute(XMLAttr)</code>	Prints a <code>XMLAttr</code> node
<code>printAttributeNodes(XMLElement)</code>	Calls <code>print</code> method for each attribute of the <code>XMLElement</code>
<code>printCDATASection(XMLCDATA)</code>	Prints a <code>XMLCDATA</code> node
<code>printChildNodes(XMLNode)</code>	Calls <code>print</code> method for each child of the <code>XMLNode</code>
<code>printComment(XMLComment)</code>	Prints a <code>XMLComment</code> node
<code>printDoctype(DTD)</code>	Prints an <code>DTD</code> .
<code>printDocument(XMLDocument)</code>	Prints an <code>XMLDocument</code> .
<code>printDocumentFragment(XMLDocumentFragment)</code>	Prints an empty <code>XMLDocumentFragment</code> object.
<code>printElement(XMLElement)</code>	Prints an <code>XMLElement</code> .
<code>printEntityReference(XMLEntityReference)</code>	Prints a <code>XMLEntityReference</code> node
<code>printProcessingInstruction(XMLPI)</code>	Prints a <code>XMLPI</code> node
<code>printTextNode(XMLText)</code>	Prints a <code>XMLText</code> node

Table 11–5 (Cont.) Summary of Methods of *PintDriver*

Method	Description
<code>setEncoding(String)</code>	Sets the encoding of the print driver.

close()**Description**

Closes the output stream or print writer

Syntax

```
public void close()
```

flush()**Description**

Flushes the output stream or print writer

Syntax

```
public void flush()
```

printAttribute(XMLAttr)**Description**

Prints a `XMLAttr` node

Syntax

```
public void printAttribute(XMLAttr attr)
```

Parameters

`attr` - The `XMLAttr` node.

printAttributeNodes(XMLElement)**Description**

Calls `print` method for each attribute of the `XMLElement`

Syntax

```
public void printAttributeNodes(XMLElement elem)
```

Parameters

`elem` - The `elem` whose attributes are to be printed.

printCDATASection(XMLCDATA)**Description**

Prints a `XMLCDATA` node

Syntax

```
public void printCDATASection(XMLCDATA cdata)
```

Parameters

`cdata` - The `XMLCDATA` node.

printChildNodes(XMLNode)**Description**

Calls `print` method for each child of the `XMLNode`

Syntax

```
public void printChildNodes(XMLNode node)
```

Parameters

`node` - The node whose children are to be printed.

printComment(XMLComment)**Description**

Prints a `XMLComment` node

Syntax

```
public void printComment(XMLComment comment)
```

Parameters

`comment` - The comment node.

printDoctype(DTD)

Description

Prints an DTD.

Syntax

```
public void printDoctype(DTD dtd)
```

Parameters

dtd - The dtd to be printed.

printDocument(XMLDocument)

Description

Prints an XMLDocument.

Syntax

```
public void printDocument(XMLDocument doc)
```

Parameters

elem - The document to be printed.

printDocumentFragment(XMLDocumentFragment)

Syntax

```
public void printDocumentFragment(XMLDocumentFragment dfrag)
```

Description

Prints an empty XMLDocumentFragment object.

Parameters

dfrag - The document fragment to be printed.

printElement(XMLElement)

Syntax

```
public void printElement(XMLElement elem)
```


Description

Prints an `XMLElement`.

Parameters

`elem` - The element to be printed.

printEntityReference(XMLEntityReference)**Description**

Prints a `XMLEntityReference` node

Syntax

```
public void printEntityReference(XMLEntityReference en)
```

Parameters

`en` - The `XMLEntityReference` node.

printProcessingInstruction(XMLPI)**Description**

Prints a `XMLPI` node

Syntax

```
public void printProcessingInstruction(XMLPI pi)
```

Parameters

`pi` - The `XMLPI` node.

printTextNode(XMLText)**Description**

Prints a `XMLText` node

Syntax

```
public void printTextNode(XMLText text)
```

Parameters

`text` - The text node.

setEncoding(String)

Description

Sets the encoding of the print driver.

Syntax

```
public void setEncoding(java.lang.String enc)
```

Parameters

`enc` - The encoding of the document being printed.

NSName

Interface in package oracle.xml.util.

Description

Part of package oracle.xml.util. This interface provides Namespace support for Element and Attr names.

Syntax

```
public interface NSName
```

Methods

getExpandedName()

Description

Get the fully resolved name for this name

Syntax

```
public java.lang.String getExpandedName()
```

Returns

The fully resolved name

getLocalName()

Description

Get the local name for this name

Syntax

```
public java.lang.String getLocalName()
```

Returns

The local name

getNamespace()

Description

Get the resolved Namespace for this name

Syntax

```
public java.lang.String getNamespace()
```

Returns

The resolved Namespace

getPrefix()

Description

Get the prefix for this name

Syntax

```
public java.lang.String getPrefix()
```

Returns

The prefix

getQualifiedName()

Description

Get the qualified name

Syntax

```
public java.lang.String getQualifiedName()
```

Returns

The qualified name

AttrDecl

Description

This class hold information about each attribute declared in an attribute list in the Document Type Definition.

Syntax

```
public class AttrDecl implements java.io.Externalizable
```

```
oracle.xml.parser.v2.AttrDecl
```

Implemented Interfaces

```
java.io.Externalizable, java.io.Serializable
```

Fields

Table 11–6 *Fields of AttrDecl*

Field	Syntax	Description
CDATA	public static final int CDATA	AttType - StringType - CDATA
DEFAULT	public static final int DEFAULT	Attribute presence - Default
ENTITIES	public static final int ENTITIES	AttType - TokenizedType - Entities
ENTITY	public static final int ENTITY	AttType - TokenizedType - Entity
ENUMERATION	public static final int ENUMERATION	AttType - EnumeratedType - Enumeration
FIXED	public static final int FIXED	Attribute presence - Fixed
ID	public static final int ID	AttType - TokenizedType - ID
IDREF	public static final int IDREF	AttType - TokenizedType - ID reference
IDREFS	public static final int IDREFS	AttType - TokenizedType - ID references
IMPLIED	public static final int IMPLIED	Attribute presence - Implied
NMTOKEN	public static final int NMTOKEN	AttType - TokenizedType - Name token
NMTOKENS	public static final int NMTOKENS	AttType - TokenizedType - Name tokens
NOTATION	public static final int NOTATION	AttType - EnumeratedType - Notation

Table 11–6 *Fields of AttrDecl*

Field	Syntax	Description
REQUIRED	public static final int REQUIRED	Attribute presence - Required

Constructors

AttrDecl()

Description

Default constructor. Note that this constructor is used only during deserialization/decompression of this DOM node. In order to deserialize this node to construct the DOM node from the serialized/ compressed stream, it is required to create a handle of the object.

Syntax

```
public static final int REQUIRED public AttrDecl()
```

Methods

Table 11–7 *Summary of Methods of AttrDecl*

Methods	Description
getAttrPresence()	Gets attribute presence
getAttrType()	Gets attribute type
getDefaultValue	Gets attribute default value
getEnumerationValues()	Gets attribute values
getNodeName()	Gets the name of the Attr Decl
getNodeType()	Gets a code representing the type of the underlying object
getExternal(ObjectInput)	This method reads the information written in the compressed stream by writeExternal method and restores the object correspondingly
readExternal(ObjectInput)	Reads the information written in the compressed stream by writeExternal method and restores the object correspondingly.
typeToString (int)	Gets a string representation of the attribute type

Table 11–7 (Cont.) Summary of Methods of AttrDecl

Methods	Description
<code>writeExternal(ObjectOutput)</code>	This method saves the state of the object by creating a binary compressed stream with information about this object.

getAttrPresence()

Description

Gets attribute presence

Syntax

```
public int getAttrPresence()
```

Returns

The presence of the attribute

getAttrType()

Description

Gets attribute type

Syntax

```
public int getAttrType()
```

Returns

The type of the attribute

getDefaultValue()

Description

Gets attribute default value

Syntax

```
public java.lang.String getDefaultValue()
```

Returns

The default value of the attribute

getEnumerationValues()

Description

Gets attribute values

Syntax

```
public java.util.Vector getEnumerationValues()
```

Returns

The values of the attribute as an Enumeration

getNodeName()

Description

Gets the name of the Attr Decl.

Syntax

```
public java.lang.String getNodeName()
```

Returns

Name of the node

getNodeType()

Description

Gets a code representing the type of the underlying object

Syntax

```
public short getNodeType()
```

Returns

type of the node

readExternal(ObjectInput)

Description

This method reads the information written in the compressed stream by writeExternal method and restores the object correspondingly.

Syntax

```
public void readExternal(java.io.ObjectInput inArg)
```

Specified By

```
java.io.Externalizable.readExternal(java.io.ObjectInput) in  
interface java.io.Externalizable
```

Parameters

`inArg` - the `ObjectInput` stream used for reading the compressed stream.

Throws

`IOException` - is thrown when there is an error in reading the input stream.

`ClassNotFoundException` - is thrown when the class is not found

typeToString(int)**Description**

Gets a string representation of the attribute type

Syntax

```
public static java.lang.String typeToString(int type)
```

Returns

A string representing the attribute type

writeExternal(ObjectOutput)**Description**

This method saves the state of the object by creating a binary compressed stream with information about this object.

Syntax

```
public void writeExternal(java.io.ObjectOutput outArg)
```

Specified By

```
java.io.Externalizable.writeExternal(java.io.ObjectOutput) in  
interface java.io.Externalizable
```

Parameters

`outArg` - The `ObjectOutput` stream used to write the serialized/ compressed stream.

Throws

`IOException` - is thrown when there is an exception while writing the serialized/compressed stream.

DefaultXMLDocumentHandler

Description of DefaultXMLDocumentHandler

This class implements the default behaviour for the `XMLDocumentHandler` interface.

Application writers can extend this class when they need to implement only part of the interface

Syntax of DefaultXMLDocumentHandler

```
public class DefaultXMLDocumentHandler implements
oracle.xml.parser.v2.XMLDocumentHandler
```

```
oracle.xml.parser.v2.DefaultXMLDocumentHandler
```

Implemented Interfaces of DefaultXMLDocumentHandler

```
XMLDocumentHandler
```

Constructors

DefaultXMLDocumentHandler()

Description

Constructs a default document

Syntax

```
public DefaultXMLDocumentHandler()
```

Methods

Table 11–8 Summary of Methods of DefaultXMLDocumentHandler

Methods	Description
<code>cDATASection(char[], int, int)</code>	Receive notification of a CDATA Section. The Parser will invoke this method once for each CDATA Section found.
<code>comment(String)</code>	Receive notification of a comment. The Parser will invoke this method once for each comment found: note that comment may occur before or after the main document element.

Table 11–8 (Cont.) Summary of Methods of DefaultXMLDocumentHandler

Methods	Description
endDoctype()	Receive notification of end of the DTD.
endElement(NSName)	Receive notification of the end of an element.
endElement(String, String, String)	Receive notification of the end of an element.
endPrefixMapping(String)	End the scope of a prefix-URI mapping.
getHandler()	Get the next pipe-line node handler.
setDoctype(DTD)	Receive notification of DTD. Sets the DTD.
setError(XMLError)	Receive notification of a XMLError handler.
setHandler(XMLDocumentHandler)	Receive notification of a next pipe-line node handler.
setTextDecl(String, String)	Receive notification of a Text XML Declaration. The Parser will invoke this method once for each text XML Decl
setXMLDecl(String, String, String)	Receive notification of an XML Declaration. The Parser will invoke this method once for XML Decl.
setXMLSchema(Object)	Receive notification of a XMLSchema object.
skippedEntity(String)	Receive notification of a skipped entity.
startElement(NSName, SAXAttrList)	Receive notification of the beginning of an element.
startElement(String, String, String, Attributes)	Receive notification of the beginning of an element.
startPrefixMapping(String, String)	Begin the scope of a prefix-URI Namespace mapping.

cDATASection(char[], int, int)**Description**

Receive notification of a CDATA Section. The Parser will invoke this method once for each CDATA Section found.

Syntax

```
public void cDATASection(char[] ch, int start, int length)
```

Specified By

`XMLDocumentHandler.cDATASection(char[], int, int)` in interface `XMLDocumentHandler`

Parameters

`ch` - The CDATA section characters.

`start` - The start position in the character array.

`length` - The number of characters to use from the character array.

Throws

`org.xml.sax.SAXException` - Any SAX exception, possibly wrapping another exception.

comment(String)**Description**

Receive notification of a comment. The Parser will invoke this method once for each comment found: note that comment may occur before or after the main document element.

Syntax

```
public void comment(java.lang.String data)
```

Specified By

`XMLDocumentHandler.comment(String)` in interface `XMLDocumentHandler`

Parameters

`data` - The comment data, or null if none was supplied.

Throws

`org.xml.sax.SAXException` - Any SAX exception, possibly wrapping another exception.

endDoctype()**Description**

Receive notification of end of the DTD.

Syntax

```
public void endDoctype()
```

Specified By

`XMLDocumentHandler.endDoctype()` in interface `XMLDocumentHandler`

Throws

`org.xml.sax.SAXException` - Any SAX exception, possibly wrapping another exception.

endElement(NSName)

Description

Receive notification of the end of an element.

Syntax

```
public void endElement(NSName elem)
```

Specified By

`XMLDocumentHandler.endElement(NSName)` in interface `XMLDocumentHandler`

Parameters

`elem` - `NSName` object

Throws

`SAXException` - A `SAXException` could be thrown.

See Also

`org.xml.sax.DocumentHandler#endElement`

endElement(String, String, String)

Description

Receive notification of the end of an element.

Syntax

```
public void endElement(java.lang.String namespaceURI, java.lang.String
```

localName, java.lang.String qName)

Parameters

`uri` - The Namespace URI, or the empty string if the element has no Namespace URI or if Namespace processing is not being performed.

`localName` - The local name (without prefix), or the empty string if Namespace processing is not being performed.

`qName` - The qualified XML 1.0 name (with prefix), or the empty string if qualified names are not available.

Throws

`org.xml.sax.SAXException` - Any SAX exception, possibly wrapping another exception.

endPrefixMapping(String)

Description

End the scope of a prefix-URI mapping.

Syntax

```
public void endPrefixMapping(java.lang.String prefix)
```

Parameters

`prefix` - The prefix that was being mapping.

Throws

`org.xml.sax.SAXException` - The client may throw an exception during processing.

See Also

`startPrefixMapping(String, String)`, `endElement(NSName)`

getHandler()

Description

Get the next pipe-line node handler.

Syntax

```
public XMLDocumentHandler getHandler()
```

Specified By

`XMLDocumentHandler.getHandler()` in interface `XMLDocumentHandler`

Returns

The `XMLDocumentHandler` node

setDoctype(DTD)

Description

Receive notification of DTD. Sets the DTD.

Syntax

```
public void setDoctype(DTD dtd)
```

Specified By

`XMLDocumentHandler.setDoctype(DTD)` in interface `XMLDocumentHandler`

Throws

`org.xml.sax.SAXException` - Any SAX exception, possibly wrapping another exception.

setError(XMLError)

Description

Receive notification of a `XMLError` handler.

Syntax

```
public void setError(XMLError he)
```

Specified By

`XMLDocumentHandler.setError(XMLError)` in interface `XMLDocumentHandler`

Parameters

`err` - The `XMLError` object

Throws

`org.xml.sax.SAXException` - Any SAX exception, possibly wrapping another exception.

setHandler(XMLDocumentHandler)**Description**

Receive notification of a next pipe-line node handler.

Syntax

```
public void setHandler(XMLDocumentHandler h)
```

Specified By

`XMLDocumentHandler.setHandler(XMLDocumentHandler)` in interface `XMLDocumentHandler`

Parameters

`h` - The `XMLDocumentHandler` node

Throws

`org.xml.sax.SAXException` - Any SAX exception, possibly wrapping another exception.

setTextDecl(String, String)**Description**

Receive notification of a Text XML Declaration. The Parser will invoke this method once for each text XML Decl

Syntax

```
public void setTextDecl(java.lang.String version, java.lang.String encoding)
```

Specified By

`XMLDocumentHandler.setTextDecl(String, String)` in interface `XMLDocumentHandler`

Parameters

`version` - The version number (or null, if not specified)

`encoding` - The encoding name

Throws

`org.xml.sax.SAXException` - Any SAX exception, possibly wrapping another exception.

setXMLDecl(String, String, String)

Description

Receive notification of an XML Declaration. The Parser will invoke this method once for XML Decl.

Syntax

```
public void setXMLDecl(java.lang.String version, java.lang.String standalone,
java.lang.String encoding)
```

Specified By

`XMLDocumentHandler.setXMLDecl(String, String, String)` in interface `XMLDocumentHandler`

Parameters

`version` - The version number

`standalone` - The standalone value (or null, if not specified)

`encoding` - The encoding name (or null, if not specified)

Throws

`org.xml.sax.SAXException` - Any SAX exception, possibly wrapping another exception.

setXMLSchema(Object)

Description

Receive notification of a XMLSchema object.

Syntax

```
public void setXMLSchema(java.lang.Object s)
```

Specified By

`XMLDocumentHandler.setXMLSchema(Object)` in interface `XMLDocumentHandler`

Parameters

s - The XMLSchema object

Throws

org.xml.sax.SAXException - Any SAX exception, possibly wrapping another exception.

skippedEntity(String)**Description**

Receive notification of a skipped entity.

Syntax

```
public void skippedEntity(java.lang.String name)
```

Parameters

name - The name of the skipped entity. If it is a parameter entity, the name will begin with '%', and if it is the external DTD subset, it will be the string "[dtd]".

Throws

org.xml.sax.SAXException - Any SAX exception, possibly wrapping another exception.

startElement(NSName, SAXAttrList)**Description**

Receive notification of the beginning of an element.

Syntax

```
public void startElement(NSName elem, SAXAttrList attrlist)
```

Specified By

XMLDocumentHandler.startElement(NSName, SAXAttrList) in interface XMLDocumentHandler

Parameters

elem - NSName object

attrlist - SAXAttrList for the element

Throws

`SAXException` - A `SAXException` could be thrown.

See Also

`org.xml.sax.DocumentHandler#startElement`

startElement(String, String, String, Attributes)

Description

Receive notification of the beginning of an element.

Syntax

```
public void startElement(java.lang.String namespaceURI, java.lang.String  
localName, java.lang.String qName, org.xml.sax.Attributes atts)
```

Parameters

`uri` - The Namespace URI, or the empty string if the element has no Namespace URI or if Namespace processing is not being performed.

`localName` - The local name (without prefix), or the empty string if Namespace processing is not being performed.

`qName` - The qualified name (with prefix), or the empty string if qualified names are not available.

`atts` - The attributes attached to the element. If there are no attributes, it shall be an empty `Attributes` object.

Throws

`org.xml.sax.SAXException` - Any SAX exception, possibly wrapping another exception.

See Also

`endElement(NSName)`, `org.xml.sax.Attributes`

startPrefixMapping(String, String)

Description

Begin the scope of a prefix-URI Namespace mapping.

Syntax

```
public void startPrefixMapping(java.lang.String prefix, java.lang.String uri)
```

Parameters

`prefix` - The Namespace prefix being declared.

`uri` - The Namespace URI the prefix is mapped to.

Throws

`org.xml.sax.SAXException` - The client may throw an exception during processing.

See Also

```
endPrefixMapping(String), startElement(NSName, SAXAttrList)
```

DocumentBuilder

Syntax of DocumentBuilder

```
public class DocumentBuilder
```

```
oracle.xml.parser.v2.DocumentBuilder
```

Description of DocumentBuilder

This class implements XMLDocumentHandler (deprecated) and ContentHandler to build a DOM Tree from SAX 2.0 events. XMLDocumentHandler events are supported for backward compatibility

Constructors

DocumentBuilder()

Description

Default Constructor. Creates a document builder that can be used as XMLDocumentHandler

Syntax

```
public DocumentBuilder()
```

Methods

Table 11–9 Summary of Methods of DocumentBuilder

Methods	Description
attributeDecl(String, String, String, String, String)	Report an attribute type declaration.
cDATASection(char[], int, int)	Receive notification of CDATA Section data inside an element.
characters(char[], int, int)	Receive notification of character data inside an element.
comment(char[], int, int)	Report an XML comment anywhere in the document.
comment(String)	Receive notification of a comment.
elementDecl(String, String)	Report an element type declaration.

Table 11–9 (Cont.) Summary of Methods of DocumentBuilder

Methods	Description
endCDATA()	Report the end of a CDATA section.
endDoctype()	Receive notification of end of the DTD.
endDocument()	Receive notification of the end of the document.
endDTD()	Report the end of DTD declarations.
endElement(NSName)	Receive notification of the end of an element.
endElement(String, String, String)	Receive notification of the end of an element.
endEntity(String)	Report the end of an entity.
externalEntityDecl(String, String, String)	Report a parsed external entity declaration.
getCurrentNode()	Get the current node being build
getDocument()	Get the document being build
ignorableWhitespace(char[], int, int)	Receive notification of ignorable whitespace in element content.
internalEntityDecl(String, String)	Report an internal entity declaration
processingInstruction(String, String)	Receive notification of a processing instruction.
retainCDATASection(boolean)	Sets a flag to retain CDATA sections
setDebugMode(boolean)	Sets a flag to turn on debug information in the document
setDoctype(DTD)	Receive notification of DTD Sets the DTD
setDocumentLocator(Locator)	Receive a Locator object for document events. By default, do nothing. Application writers may override this method in a subclass if they wish to store the locator for use with other document events.
setNodeFactory(NodeFactory)	Set a optional NodeFactory to be used for creating custom DOM trees
setTextDecl(String, String)	Receive notification of a Text XML Declaration. The Parser will invoke this method once for each text XML Decl.
setXMLDecl(String, String, String)	Receive notification of a XML Declaration. The Parser will invoke this method once for XML Decl .
startCDATA()	Report the start of a CDATA section.

Table 11–9 (Cont.) Summary of Methods of DocumentBuilder

Methods	Description
startDocument()	Receive notification of the beginning of the document.
startDTD(String, String, String)	Report the start of DTD declarations, if any.
startElement(NSName, SAXAttrList)	Receive notification of the beginning of an element.
startElement(String, String, String, Attributes)	Receive notification of the beginning of an element.
startEntity(String)	Report the beginning of some internal and external XML entities. All start/endEntity events must be properly nested.

attributeDecl(String, String, String, String, String)

Description

Report an attribute type declaration.

Syntax

```
public void attributeDecl(java.lang.String eName, java.lang.String aName,
    java.lang.String type, java.lang.String valueDefault, java.lang.String value)
```

Parameters

eName - The name of the associated element.

aName - The name of the attribute.

type - A string representing the attribute type.

valueDefault - A string representing the attribute default ("#IMPLIED", "#REQUIRED", or "#FIXED") or null if none of these applies.

value - A string representing the attribute's default value, or null if there is none.

Throws

SAXException - The application may raise an exception.

cDATASection(char[], int, int)

Description

Receive notification of CDATA Section data inside an element.

Syntax

```
public void cDATASection(char[] ch, int start, int length)
```

Parameters

`ch` - The CDATA characters.

`start` - The start position in the character array.

`length` - The number of characters to use from the character array.

Throws

`org.xml.sax.SAXException` - Any SAX exception, possibly wrapping another exception.

See Also

`org.xml.sax.DocumentHandler#characters`

characters(char[], int, int)**Description**

Receive notification of character data inside an element.

Syntax

```
public void characters(char[] ch, int start, int length)
```

Parameters

`ch` - The characters.

`start` - The start position in the character array.

`length` - The number of characters to use from the character array.

Throws

`org.xml.sax.SAXException` - Any SAX exception, possibly wrapping another exception.

See Also

`org.xml.sax.DocumentHandler#characters`

comment(char[], int, int)

Description

Report an XML comment anywhere in the document.

Syntax

```
public void comment(char[] ch, int start, int length)
```

Parameters

ch - An array holding the characters in the comment.

start - The starting position in the array.

length - The number of characters to use from the array.

Throws

SAXException - The application may raise an exception.

comment(String)

Description

Receive notification of a comment.

Syntax

```
public void comment(java.lang.String data)
```

Parameters

data - The comment data, or null if none was supplied.

Throws

org.xml.sax.SAXException - Any SAX exception, possibly wrapping another exception.

elementDecl(String, String)

Description

Report an element type declaration.

Syntax

```
public void elementDecl(java.lang.String name, java.lang.String model)
```

Parameters

`name` - The element type name.

`model` - The content model as a normalized string.

Throws

`SAXException` - The application may raise an exception.

endCDATA()**Description**

Report the end of a CDATA section.

Syntax

```
public void endCDATA()
```

Throws

`SAXException` - The application may raise an exception.

See Also

```
startCDATA()
```

endDoctype()**Description**

Receive notification of end of the DTD.

Syntax

```
public void endDoctype()
```

Throws

`org.xml.sax.SAXException` - Any SAX exception, possibly wrapping another exception.

endDocument()**Description**

Receive notification of the end of the document.

Syntax

```
public void endDocument()
```

Throws

`org.xml.sax.SAXException` - Any SAX exception, possibly wrapping another exception.

See Also

`org.xml.sax.DocumentHandler#endDocument`

endDTD()

Description

Report the end of DTD declarations.

Syntax

```
public void endDTD()
```

Throws

`SAXException` - The application may raise an exception.

See Also

`startDTD(String, String, String)`

endElement(NSName)

Description

Receive notification of the end of an element.

Syntax

```
public void endElement(NSName elem)
```

Parameters

`elem` - `NSName` object

Throws

`SAXException` - A `SAXException` could be thrown.

See Also

`org.xml.sax.DocumentHandler#endElement`

endElement(String, String, String)**Description**

Receive notification of the end of an element.

Syntax

```
public void endElement(java.lang.String namespaceURI, java.lang.String  
localName, java.lang.String qName)
```

Parameters

`namespaceURI` - The Namespace URI, or the empty string if the element has no Namespace URI or if Namespace processing is not being performed.

`localName` - The local name (without prefix), or the empty string if Namespace processing is not being performed.

`qName` - The qualified XML 1.0 name (with prefix), or the empty string if qualified names are not available.

Throws

`org.xml.sax.SAXException` - Any SAX exception, possibly wrapping another exception.

endEntity(String)**Description**

Report the end of an entity.

Syntax

```
public void endEntity(java.lang.String name)
```

Parameters

`name` - The name of the entity that is ending.

Throws

`SAXException` - The application may raise an exception.

See Also

`startEntity(String)`

externalEntityDecl(String, String, String)

Description

Report a parsed external entity declaration.

Syntax

```
public void externalEntityDecl(java.lang.String name, java.lang.String publicId,  
java.lang.String systemId)
```

Parameters

`name` - The name of the entity. If it is a parameter entity, the name will begin with '%!'.
`publicId` - The declared public identifier of the entity, or `null` if none was declared.
`systemId` - The declared system identifier of the entity.

Throws

`SAXException` - The application may raise an exception.

See Also

`internalEntityDecl(String, String)`,
`org.xml.sax.DTDHandler#unparsedEntityDecl`

getCurrentNode()

Description

Get the current node being build

Syntax

```
public XMLNode getCurrentNode()
```

Returns

`XMLNode`

getDocument()

Description

Get the document being build

Syntax

```
public XMLDocument getDocument()
```

Returns

XMLDocument

ignorableWhitespace(char[], int, int)

Description

Receive notification of ignorable whitespace in element content.

Syntax

```
public void ignorableWhitespace(char[] ch, int start, int length)
```

Parameters

`ch` - The whitespace characters.

`start` - The start position in the character array.

`length` - The number of characters to use from the character array.

Throws

`org.xml.sax.SAXException` - Any SAX exception, possibly wrapping another exception.

See Also

`org.xml.sax.DocumentHandler#ignorableWhitespace`

internalEntityDecl(String, String)

Description

Report an internal entity declaration.

Syntax

```
public void internalEntityDecl(java.lang.String name, java.lang.String value)
```

Parameters

`name` - The name of the entity. If it is a parameter entity, the name will begin with '%'.
`value` - The replacement text of the entity.

Throws

`SAXException` - The application may raise an exception.

See Also

`externalEntityDecl(String, String, String)`,
`org.xml.sax.DTDHandler#unparsedEntityDecl`

processingInstruction(String, String)

Description

Receive notification of a processing instruction.

Syntax

```
public void processingInstruction(java.lang.String target, java.lang.String data)
```

Parameters

`target` - The processing instruction target.

`data` - The processing instruction data, or null if none is supplied.

Throws

`org.xml.sax.SAXException` - Any SAX exception, possibly wrapping another exception.

See Also

`org.xml.sax.DocumentHandler#processingInstruction`

retainCDATASection(boolean)

Description

Sets a flag to retain CDATA sections

Syntax

```
public void retainCDATASection(boolean flag)
```

Parameters

flag - determines whether CDATA sections are retained

setDebugMode(boolean)**Description**

Sets a flag to turn on debug information in the document

Syntax

```
public void setDebugMode(boolean flag)
```

Parameters

flag - determines whether debug info is stored

setDoctype(DTD)**Description**

Receive notification of DTD Sets the DTD

Syntax

```
public void setDoctype(DTD dtd)
```

Parameters

dtd - set the DTD for the document

Throws

`org.xml.sax.SAXException` - Any SAX exception, possibly wrapping another exception.

setDocumentLocator(Locator)**Description**

Receive a Locator object for document events. By default, do nothing. Application writers may override this method in a subclass if they wish to store the locator for use with other document events.

Syntax

```
public void setDocumentLocator(org.xml.sax.Locator locator)
```

Parameters

`locator` - A locator for all SAX document events.

See Also

`org.xml.sax.DocumentHandler#setDocumentLocator`,
`org.xml.sax.Locator`

setNodeFactory(NodeFactory)

Description

Set a optional NodeFactory to be used for creating custom DOM trees

Syntax

```
public void setNodeFactory(NodeFactory f)
```

Parameters

`f` - NodeFactory

setTextDecl(String, String)

Description

Receive notification of a Text XML Declaration. The Parser will invoke this method once for each text XML Decl

Syntax

```
public void setTextDecl(java.lang.String version, java.lang.String encoding)
```

Parameters

`version` - The version number (or null, if not specified)

`encoding` - The encoding name

Throws

`org.xml.sax.SAXException` - Any SAX exception, possibly wrapping another exception.

setXMLDecl(String, String, String)

Description

Receive notification of a XML Declaration. The Parser will invoke this method once for XML Decl .

Syntax

```
public void setXMLDecl(java.lang.String version, java.lang.String standalone,  
java.lang.String encoding)
```

Parameters

version - The version number

standalone - The standalone value (or null, if not specified)

encoding - The encoding name (or null, if not specified)

Throws

`org.xml.sax.SAXException` - Any SAX exception, possibly wrapping another exception.

startCDATA()

Description

Report the start of a CDATA section.

Syntax

```
public void startCDATA()
```

Throws

`SAXException` - The application may raise an exception.

See Also

`endCDATA()`

startDocument()

Description

Receive notification of the beginning of the document.

Syntax

```
public void startDocument()
```

Throws

`org.xml.sax.SAXException` - Any SAX exception, possibly wrapping another exception.

See Also

`org.xml.sax.DocumentHandler#startDocument`

startDTD(String, String, String)

Description

Report the start of DTD declarations, if any.

Syntax

```
public void startDTD(java.lang.String name, java.lang.String publicId,  
java.lang.String systemId)
```

Parameters

`name` - The document type name.

`publicId` - The declared public identifier for the external DTD subset, or null if none was declared.

`systemId` - The declared system identifier for the external DTD subset, or null if none was declared.

Throws

`SAXException` - The application may raise an exception.

See Also

`endDTD()`, `startEntity(String)`

startElement(NSName, SAXAttrList)

Description

Receive notification of the beginning of an element.

Syntax

```
public void startElement(NSName elem, SAXAttrList attrlist)
```

Parameters

`elem` - NSName object

`attrlist` - SAXAttrList for the element

Throws

SAXException - A SAXException could be thrown.

See Also

`org.xml.sax.DocumentHandler#startElement`

startElement(String, String, String, Attributes)**Description**

Receive notification of the beginning of an element.

Syntax

```
public void startElement(java.lang.String namespaceURI, java.lang.String  
localName, java.lang.String qName, org.xml.sax.Attributes atts)
```

Parameters

`namespaceURI` - The Namespace URI, or the empty string if the element has no Namespace URI or if Namespace processing is not being performed.

`localName` - The local name (without prefix), or the empty string if Namespace processing is not being performed.

`qName` - The qualified name (with prefix), or the empty string if qualified names are not available.

`atts` - The attributes attached to the element. If there are no attributes, it shall be an empty Attributes object.

Throws

`org.xml.sax.SAXException` - Any SAX exception, possibly wrapping another exception.

See Also

`endElement(NSName)`, `org.xml.sax.Attributes`

startEntity(String)

Description

Report the beginning of some internal and external XML entities. All start/endEntity events must be properly nested.

Syntax

```
public void startEntity(java.lang.String name)
```

Parameters

name - The name of the entity. If it is a parameter entity, the name will begin with '%', and if it is the external DTD subset, it will be "[dtd]".

Throws

SAXException - The application may raise an exception.

See Also

endEntity(String), org.xml.sax.ext.DeclHandler,
org.xml.sax.ext.DeclHandler

DOMParser

Syntax of DOMParser

```
public class DOMParser
```

```
oracle.xml.parser.v2.DOMParser
```

Description of DOMParser

This class implements an eXtensible Markup Language (XML) 1.0 parser according to the World Wide Web Consortium (W3C) recommendation, to parse a XML document and build a DOM tree.

Fields of DOMParser

Table 11–10 *Fields of DOMParser*

Field	Syntax	Description
DEBUG_MODE	public static final java.lang.String DEBUG_MODE	Sets Debug Mode Boolean.TRUE or Boolean.FALSE
ERROR_ENCODING	public static final java.lang.String ERROR_ENCODING	Encoding for errors report via error stream (only if ERROR_STREAM is set).
ERROR_STREAM	public static final java.lang.String ERROR_STREAM	Error stream for reporting errors. The object can be OutputStream or PrintWriter. This attribute is ignored if ErrorHandler is set.
NODE_FACTORY	public static final java.lang.String NODE_FACTORY	Set NodeFactory to build custom Nodes
SHOW_WARNINGS	public static final java.lang.String SHOW_WARNINGS	Boolean to ignore warnings Boolean.TRUE or Boolean.FALSE

Constructors of DOMParser

DOMParser()

Description

Creates a new parser object.

Syntax

```
public DOMParser()
```

Methods of DOMParser**Table 11–11 Summary of Methods of DOMParser**

Methods	Description
<code>getAttribute(String)</code>	Allows the user to retrieve specific attributes on the underlying implementation.
<code>getDoctype()</code>	Get the DTD
<code>getDocument()</code>	Gets the document
<code>parseDTD(InputSource, String)</code>	Parses the XML External DTD from given input source
<code>parseDTD(InputStream, String)</code>	Parses the XML External DTD from given input stream. The base URL should be set for resolving external entities and DTD.
<code>parseDTD(Reader, String)</code>	Parses the XML External DTD from given input stream. The base URL should be set for resolving external entities and DTD.
<code>parseDTD(String, String)</code>	Parses the XML External DTD from the URL indicated
<code>parseDTD(URL, String)</code>	Parses the XML External DTD document pointed to by the given URL and creates the corresponding XML document hierarchy.
<code>reset()</code>	Resets the parser state
<code>retainCDATASection(boolean)</code>	Switch to determine whether to retain CDATA sections
<code>setAttribute(String, Object)</code>	Allows the user to set specific attributes on the underlying implementation.
<code>setDebugMode(boolean)</code>	Sets a flag to turn on debug information in the document
<code>setErrorStream(OutputStream)</code>	Creates an output stream for the output of errors and warnings. If an output stream for errors is not specified, the parser will use the standard error output stream <code>System.err</code> for outputting errors and warnings.
<code>setErrorStream(OutputStream, String)</code>	Creates an output stream for the output of errors and warnings. If an output stream for errors is not specified, the parser will use the standard error output stream <code>System.err</code> for outputting errors and warnings. Additionally, an <code>.exception</code> is thrown if the encoding specified is unsupported.

Table 11–11 Summary of Methods of DOMParser

Methods	Description
setErrorStream(PrintWriter)	Creates an output stream for the output of errors and warnings. If an output stream for errors is not specified, the parser will use the standard error output stream <code>System.err</code> for outputting errors and warnings.
setNodeFactory(NodeFactory)	Set the node factory. Applications can extend the <code>NodeFactory</code> and register it through this method. The parser will then use the user supplied <code>NodeFactory</code> to create nodes of the DOM tree.
showWarnings(boolean)	Switch to determine whether to print warnings

getAttribute(String)

Description

Allows the user to retrieve specific attributes on the underlying implementation.

Syntax

```
public java.lang.Object getAttribute(java.lang.String name)
```

Parameters

name - The name of the attribute.

Returns

value The value of the attribute.

Throws

`IllegalArgumentException` - thrown if the underlying implementation doesn't recognize the attribute.

getDoctype()

Description

Get the DTD

Syntax

```
public DTD getDoctype()
```

Returns

The DTD

getDocument()

Description

Gets the document

Syntax

```
public XMLDocument getDocument()
```

Returns

The document being parsed

parseDTD(InputSource, String)

Description

Parses the XML External DTD from given input source

Syntax

```
public final void parseDTD(org.xml.sax.InputSource in, java.lang.String  
rootName)
```

Parameters

`in` - the `org.xml.sax.InputSource` to parse

`rootName` - the element to be used as root Element

Throws

`XMLParseException` - if syntax or other error encountered.

`SAXException` - Any SAX exception, possibly wrapping another exception.

`IOException` - IO Error.

parseDTD(InputStream, String)

Description

Parses the XML External DTD from given input stream. The base URL should be set for resolving external entities and DTD.

Syntax

```
public final void parseDTD(java.io.InputStream in, java.lang.String rootName)
```

Parameters

`in` - the `InputStream` containing XML data to parse.

`rootName` - the element to be used as root Element

Throws

`XMLParseException` - if syntax or other error encountered.

`SAXException` - Any SAX exception, possibly wrapping another exception.

`IOException` - IO Error.

See Also

`XMLParser.setBaseURL(URL)`

parseDTD(Reader, String)**Description**

Parses the XML External DTD from given input stream. The base URL should be set for resolving external entities and DTD.

Syntax

```
public final void parseDTD(java.io.Reader r, java.lang.String rootName)
```

Parameters

`r` - the `Reader` containing XML data to parse.

`rootName` - the element to be used as root Element

Throws

`XMLParseException` - if syntax or other error encountered.

`SAXException` - Any SAX exception, possibly wrapping another exception.

`IOException` - IO Error.

See Also

`XMLParser.setBaseURL(URL)`

parseDTD(String, String)

Description

Parses the XML External DTD from the URL indicated

Syntax

```
public final void parseDTD(java.lang.String in, java.lang.String rootName)
```

Parameters

in - the String containing the URL to parse from

rootName - the element to be used as root Element

Throws

XMLParseException - if syntax or other error encountered.

SAXException - Any SAX exception, possibly wrapping another exception.

IOException - IO Error.

parseDTD(URL, String)

Description

Parses the XML External DTD document pointed to by the given URL and creates the corresponding XML document hierarchy.

Syntax

```
public final void parseDTD(java.net.URL url, java.lang.String rootName)
```

Parameters

url - the url points to the XML document to parse.

rootName - the element to be used as root Element

Throws

XMLParseException - if syntax or other error encountered.

SAXException - Any SAX exception, possibly wrapping another exception.

IOException - IO Error.

reset()

Description

Resets the parser state.

Syntax

```
public void reset()
```

retainCDATASection(boolean)

Description

Switch to determine whether to retain CDATA sections

Syntax

```
public void retainCDATASection(boolean flag)
```

Parameters

`flag` - TRUE - keep CDATASections (default) FALSE - convert CDATASection to Text nodes

setAttribute(String, Object)

Description

Allows the user to set specific attributes on the underlying implementation.

Syntax

```
public void setAttribute(java.lang.String name, java.lang.Object value)
```

Parameters

`name` - The name of the attribute.

`value` - The value of the attribute.

Throws

`IllegalArgumentException` - thrown if the underlying implementation doesn't recognize the attribute.

setDebugMode(boolean)

Description

Sets a flag to turn on debug information in the document

Syntax

```
public void setDebugMode(boolean flag)
```

Parameters

`flag` - determines whether debug info is stored

setErrorStream(OutputStream)

Description

Creates an output stream for the output of errors and warnings. If an output stream for errors is not specified, the parser will use the standard error output stream `System.err` for outputting errors and warnings.

Syntax

```
public final void setErrorStream(java.io.OutputStream out)
```

Parameters

`out` - The output stream to use for errors and warnings

setErrorStream(OutputStream, String)

Description

Creates an output stream for the output of errors and warnings. If an output stream for errors is not specified, the parser will use the standard error output stream `System.err` for outputting errors and warnings. Additionally, an `.exception` is thrown if the encoding specified is unsupported.

Syntax

```
public final void setErrorStream(java.io.OutputStream out, java.lang.String enc)
```

Parameters

`out` - The output stream to use for errors and warnings

`enc` - the encoding to use

Throws

`IOException` - if an unsupported encoding is specified

setErrorStream(PrintWriter)**Description**

Creates an output stream for the output of errors and warnings. If an output stream for errors is not specified, the parser will use the standard error output stream `System.err` for outputting errors and warnings.

Syntax

```
public final void setErrorStream(java.io.PrintWriter out)
```

Parameters

`out` - The `PrintWriter` to use for errors and warnings

Throws

`IOException` - if I/O error occurs in setting the error stream.

setNodeFactory(NodeFactory)**Description**

Set the node factory. Applications can extend the `NodeFactory` and register it through this method. The parser will then use the user supplied `NodeFactory` to create nodes of the DOM tree.

Syntax

```
public void setNodeFactory(NodeFactory factory)
```

Parameters

`factory` - The `NodeFactory` to set

Throws

`XMLParseException` - if an invalid factory is set

See Also

`NodeFactory`

showWarnings(boolean)

Description

Switch to determine whether to print warnings.

Syntax

```
public void showWarnings(boolean flag)
```

Parameters

`flag` - determines whether warnings should be shown

DTD

Description of DTD

Implements the DOM DocumentType interface and holds the Document Type. Definition information for an XML document.

Syntax of DTD

```
public class DTD implements java.io.Externalizable
```

```
oracle.xml.parser.v2.DTD
```

Implemented Interfaces of DTD

```
java.io.Externalizable, java.io.Serializable
```

Constructors of DTD

DTD()

Description

Default constructor. Note that this constructor is used only during deserialization/decompression of this DOM node. In order to deserialize this node to construct the DOM node from the serialized/ compressed stream, it is required to create a handle of the object.

Syntax

```
public DTD()
```

Methods of DTD

Table 11–12 Summary of Methods of DTD

Method	Description
findElementDecl(String)	Finds an element declaration for the given tag name.
findEntity(String, boolean)	Finds a named entity in the DTD.
findNotation(String)	Retrieves the named notation from the DTD.

Table 11–12 Summary of Methods of DTD

Method	Description
<code>getChildNodes()</code>	A <code>NodeList</code> that contains all children of this node. If there are no children, this is a <code>NodeList</code> containing no nodes. The content of the returned <code>NodeList</code> is "live" in the sense that, for instance, changes to the children of the node object that it was created from are immediately reflected in the nodes returned by the <code>NodeList</code> accessors; it is not a static snapshot of the content of the node. This is true for every <code>NodeList</code> , including the ones returned by the <code>getElementsByTagName</code> method.
<code>getElementDecls()</code>	A <code>NamedNodeMap</code> containing the element declarations in the DTD. Every node in this map is an <code>ElementDecl</code> object.
<code>getEntities()</code>	A <code>NamedNodeMap</code> containing the general entities, both external and internal, declared in the DTD. Duplicates are discarded. For example in: <code><!DOCTYPE ex SYSTEM "ex.dtd" [<!ENTITY foo "foo"> <!ENTITY bar "bar"> <!ENTITY % baz "baz">]> <ex/></code> the interface provides access to <code>foo</code> and <code>bar</code> but not <code>baz</code> . Every node in this map also implements the <code>Entity</code> interface. The DOM Level 1 does not support editing entities, therefore <code>entities</code> cannot be altered in any way.
<code>getInternalSubset()</code>	Gets the internal subset of the DTD
<code>getName()</code>	Gets the name of the DTD; i.e., the name immediately following the <code>DOCTYPE</code> keyword.
<code>getNodeName()</code>	Gets the name of the DTD; i.e., the name immediately following the <code>DOCTYPE</code> keyword.
<code>getNodeTypeInfo()</code>	Gets a code representing the type of the underlying object
<code>getNotations()</code>	A <code>NamedNodeMap</code> containing the notations declared in the DTD. Duplicates are discarded. Every node in this map also implements the <code>Notation</code> interface. The DOM Level 1 does not support editing notations, therefore <code>notations</code> cannot be altered in any way.
<code>getOwnerImplementation()</code>	Gets the owner of the DTD implementation.
<code>getPublicId()</code>	Gets The public identifier associated with the DTD, if specified. If the public identifier was not specified, this is <code>null</code> .
<code>getRootTag()</code>	Get the root tag for the DTD
<code>getSystemId()</code>	Gets the system identifier associated with the DTD, if specified. If the system identifier was not specified, this is <code>null</code> .

Table 11–12 Summary of Methods of DTD

Method	Description
hasChildNodes()	This is a convenience method to allow easy determination of whether a node has any children. return false always, as DTD cannot have any overrides method in XMLNode
normalize()	Normalize the DTD.
printExternalDTD(OutputStream)	Writes the contents of this document to the given output stream.
printExternalDTD(OutputStream, String)	Writes the contents of the external DTD to the given output stream.
printExternalDTD(PrintWriter)	Writes the contents of this document to the given output stream.
readExternal(ObjectInput)	This method reads the information written in the compressed stream by writeExternal method and restores the object correspondingly.
setRootTag(String)	Set the root tag for the DTD
writeExternal(ObjectOutput)	This method saves the state of the object by creating a binary compressed stream with information about this object.

findElementDecl(String)

Description

Finds an element declaration for the given tag name.

Syntax

```
public final ElementDecl findElementDecl(java.lang.String name)
```

Parameters

name - The tag name.

Returns

the element declaration object.

findEntity(String, boolean)

Description

Finds a named entity in the DTD.

Syntax

```
public final org.w3c.dom.Entity findEntity(java.lang.String n, boolean par)
```

Parameters

n - The name of the entity.

par - Boolean indicating if the entity is parameter `Entity`.

Returns

The specified `Entity` object; returns null if it is not found.

findNotation(String)

Description

Retrieves the named notation from the DTD.

Syntax

```
public final org.w3c.dom.Notation findNotation(java.lang.String name)
```

RParameters

name - The name of the notation.

Returns

the `Notation` object; returns null if it is not found.

getChildNodes()

Description

A `NodeList` that contains all children of this node. If there are no children, this is a `NodeList` containing no nodes. The content of the returned `NodeList` is "live" in the sense that, for instance, changes to the children of the node object that it was created from are immediately reflected in the nodes returned by the `NodeList` accessors; it is not a static snapshot of the content of the node. This is true for every `NodeList`, including the ones returned by the `getElementsByTagName` method.

Syntax

```
public org.w3c.dom.NodeList getChildNodes()
```

Returns

The children of this node

getElementDecls()**Description**

A `NamedNodeMap` containing the element declarations in the DTD. Every node in this map is an `ElementDecl` object.

Syntax

```
public org.w3c.dom.NamedNodeMap getElementDecls()
```

Returns

The element declarations in the DTD. The DOM Level 1 does not support editing `elementDecls`, therefore `elementDecls` cannot be altered in any way.

getEntities()**Description**

A `NamedNodeMap` containing the general entities, both external and internal, declared in the DTD. Duplicates are discarded. For example in: `<!DOCTYPE ex SYSTEM "ex.dtd" [<!ENTITY foo "foo"> <!ENTITY bar "bar"> <!ENTITY % baz "baz">]> <ex/>` the interface provides access to `foo` and `bar` but not `baz`. Every node in this map also implements the `Entity` interface. The DOM Level 1 does not support editing entities, therefore `entities` cannot be altered in any way.

Syntax

```
public org.w3c.dom.NamedNodeMap getEntities()
```

Returns

The entities declared in the DTD

getInternalSubset()**Description**

Gets the internal subset of the DTD

Syntax

```
public java.lang.String getInternalSubset()
```

Returns

The internal subset declarations as a string

getName()

Description

Gets the name of the DTD; i.e., the name immediately following the DOCTYPE keyword.

Syntax

```
public java.lang.String getName()
```

Returns

Name of the DTD

getNodeName()

Description

Gets the name of the DTD; i.e., the name immediately following the DOCTYPE keyword.

Syntax

```
public java.lang.String getNodeName()
```

Returns

Name of the DTD

getNodeType()

Description

Gets a code representing the type of the underlying object

Syntax

```
public short getNodeType()
```

Returns

type of the node

getNotations()

Description

A `NamedNodeMap` containing the notations declared in the DTD. Duplicates are discarded. Every node in this map also implements the `Notation` interface. The DOM Level 1 does not support editing notations, therefore notations cannot be altered in any way.

Syntax

```
public org.w3c.dom.NamedNodeMap getNotations()
```

Returns

The notations declared in the DTD

getOwnerImplementation()

Description

Gets the owner of the DTD implementation.

Syntax

```
public XMLDOMImplementation getOwnerImplementation()
```

Returns

the `Implementation` this DTD was created from

Since

DOM 2

getPublicId()

Description

Gets The public identifier associated with the DTD, if specified. If the public identifier was not specified, this is `null`.

Syntax

```
public java.lang.String getPublicId()
```

Returns

the public identifier associated with the DTD

getRootTag()

Description

Get the root tag for the DTD

Syntax

```
public java.lang.String getRootTag()
```

Returns

root tag

getSystemId()

Description

Gets the system identifier associated with the DTD, if specified. If the system identifier was not specified, this is `null`.

Syntax

```
public java.lang.String getSystemId()
```

Returns

the system identifier associated with the DTD

hasChildNodes()

Description

This is a convenience method to allow easy determination of whether a node has any children. return false always, as DTD cannot have any overrides method in XMLNode

Syntax

```
public boolean hasChildNodes()
```

Returns

false as DTD node can not have any children,

normalize()

Description

Normalize the DTD.

Syntax

```
public void normalize()
```

Since

DOM 2

printExternalDTD(OutputStream)

Description

Writes the contents of this document to the given output stream.

Syntax

```
public void printExternalDTD(java.io.OutputStream out)
```

Parameters

out - `OutputStream` to write to

Throws

`IOException` - if an error occurs

printExternalDTD(OutputStream, String)

Description

Writes the contents of the external DTD to the given output stream.

Syntax

```
public void printExternalDTD(java.io.OutputStream out, java.lang.String enc)
```

Writes the contents of the external DTD to the given output stream.

Parameters

out - `OutputStream` to write to

enc - Encoding to use for the output

Throws

`IOException` - if an invalid encoding was specified or if any other error occurs

printExternalDTD(PrintWriter)

Description

Writes the contents of this document to the given output stream.

Syntax

```
public void printExternalDTD(java.io.PrintWriter out)
```

Parameters

`out` - `PrintWriter` to write to

Throws

`IOException` - if an error occurs

readExternal(ObjectInput)

Description

This method reads the information written in the compressed stream by `writeExternal` method and restores the object correspondingly.

Syntax

```
public void readExternal(java.io.ObjectInput inArg)
```

Specified By

`java.io.Externalizable.readExternal(java.io.ObjectInput)` in interface `java.io.Externalizable`

Parameters

`inArg` - the `ObjectInput` stream used for reading the compressed stream.

Throws

`IOException` - is thrown when there is an error in reading the input stream.

`ClassNotFoundException` - is thrown when the class is not found.

setRootTag(String)

Description

Set the root tag for the DTD

Syntax

```
public void setRootTag(java.lang.String root)
```

Parameters

`root` - the root tag

writeExternal(ObjectOutput)

Description

This method saves the state of the object by creating a binary compressed stream with information about this object.

Syntax

```
public void writeExternal(java.io.ObjectOutput outArg)
```

Specified By

`java.io.Externalizable.writeExternal(java.io.ObjectOutput)` in interface `java.io.Externalizable`

Parameters

`outArg` - The `ObjectOutput` stream used to write the serialized/ compressed stream.

Throws

`IOException` - is thrown when there is an exception while writing the serialized/compressed stream.

ElementDecl

Description of ElementDecl

This class represents an element declaration in a DTD.

Syntax of ElementDecl

```
public class ElementDecl implements java.io.Serializable, java.io.Externalizable
    oracle.xml.parser.v2.ElementDecl
```

Implemented Interfaces of ElementDecl

java.io.Externalizable, java.io.Serializable

Fields of ElementDecl

Table 11–13 *Fields of ElementDecl*

Field	Syntax	Description
ANY	public static final byte ANY	Element content type - Children can be any element
ASTERISK	public static final int ASTERISK	ContentModelParseTreeNode type - "*" node (has one children)
COMMA	public static final int COMMA	ContentModelParseTreeNode type - "," node (has two children)
ELEMENT	public static final int ELEMENT	ContentModelParseTreeNode type - 'leaf' node (has no children)
ELEMENT_DECLARED	public static final int ELEMENT_DECLARED	
ELEMENTS	public static final byte ELEMENTS	Element content type - Children can be elements as per Content Model
EMPTY	public static final byte EMPTY	Element content type - No Children
ID_ATTR_DECL	public static final int ID_ATTR_DECL	
MIXED	public static final byte MIXED	Element content type - Children can be PCDATA & elements as per Content Model

Table 11–13 *Fields of ElementDecl*

Field	Syntax	Description
OR	public static final int OR	ContentModelParseTreeNode type - " " node (has two children)
PLUS	public static final int PLUS	ContentModelParseTreeNode type - "+" node (has one children)
QMARK	public static final int QMARK	ContentModelParseTreeNode type - "?" node (has one children)

Constructors of ElementDecl

ElementDecl()

Description

Default constructor. Note that this constructor is used only during deserialization/decompression of this DOM node. In order to deserialize this node to construct the DOM node from the serialized/ compressed stream, it is required to create a handle of the object.

Syntax

```
public ElementDecl()
```

Methods of ElementDecl

Table 11–14 *Summary of Methods of ElementDecl*

Method	Description
cloneNode(boolean)	Returns a duplicate of this node, i.e., serves as a generic copy constructor for nodes. The duplicate node has no parent (<code>parentNode</code> returns <code>null</code>). Cloning an <code>Element</code> copies all attributes and their values, including those generated by the XML processor to represent defaulted attributes, but this method does not copy any text it contains unless it is a deep clone, since the text is contained in a child <code>Text</code> node. Cloning any other type of node simply returns a copy of this node.
expectedElements(Element)	Returns vector of element names that can be appended to the element.
findAttrDecl(String)	Gets an attribute declaration object or null if not found

Table 11–14 Summary of Methods of ElementDecl

Method	Description
<code>getAttrDecls()</code>	Gets an enumeration of attribute declarations
<code>getContentElements()</code>	Returns Vector of elements that can be appended to this element
<code>getContentType()</code>	Returns content model of element
<code>getNodeName()</code>	Gets the name of the Element Decl.
<code>getNodeType()</code>	Gets a code representing the type of the underlying object
<code>getParseTree()</code>	Returns the root node of Content Model Parse Tree. <code>Node.getFirstChild()</code> and <code>Node.getLastChild()</code> return the the parse tree branches. <code>Node.getNodeType()</code> and <code>Node.getNodeName()</code> return the the parse tree node type and name.
<code>readExternal(ObjectInput)</code>	This method reads the information written in the compressed stream by <code>writeExternal</code> method and restores the object correspondingly.
<code>validateContent(Element)</code>	Validates the content of a element node.
<code>writeExternal(ObjectOutput)</code>	This method saves the state of the object by creating a binary compressed stream with information about this object.

cloneNode(boolean)

Description

Returns a duplicate of this node, i.e., serves as a generic copy constructor for nodes. The duplicate node has no parent (`parentNode` returns `null`). Cloning an `Element` copies all attributes and their values, including those generated by the XML processor to represent defaulted attributes, but this method does not copy any text it contains unless it is a deep clone, since the text is contained in a child `Text` node. Cloning any other type of node simply returns a copy of this node.

Syntax

```
public org.w3c.dom.Node cloneNode(boolean deep)
```

Parameters

`deep` - If `true`, recursively clone the subtree under the specified node; if `false`, clone only the node itself (and its attributes, if it is an `Element`).

Returns

The duplicate node.

expectedElements(Element)**Description**

Returns vector of element names that can be appended to the element.

Syntax

```
public java.util.Vector expectedElements(org.w3c.dom.Element e)
```

Parameters

e - Element

Returns

Vector of names

findAttrDecl(String)**Description**

Gets an attribute declaration object or null if not found

Syntax

```
public final AttrDecl findAttrDecl(java.lang.String name)
```

Parameters

name - Attribute declaration to find

Returns

The `AttrDecl` object, or null, if it was not found

getAttrDecls()**Description**

Gets an enumeration of attribute declarations

Syntax

```
public org.w3c.dom.NamedNodeMap getAttrDecls()
```

Returns

An enumeration of attribute declarations

getContentElements()

Description

Returns Vector of elements that can be appended to this element

Syntax

```
public final java.util.Vector getContentElements()
```

Returns

The `Vector` containing the element names.

getContentType()

Description

Returns content model of element

Syntax

```
public int getContentType()
```

Returns

The `type` of the element declaration.

getNodeName()

Description

Gets the name of the Element Decl.

Syntax

```
public java.lang.String getNodeName()
```

Returns

name of the node

getNodeTypes()

Description

Gets a code representing the type of the underlying object

Syntax

```
public short getNodeTypes()
```

Returns

Type of the node

getParseTree()

Description

Returns the root node of Content Model Parse Tree. `Node.getFirstChild()` and `Node.getLastChild()` return the the parse tree branches. `Node.getNodeTypes()` and `Node.getNodeName()` return the the parse tree node type and name.

Syntax

```
public final org.w3c.dom.Node getParseTree()
```

Returns

The `Node` containing the Content Model parse tree root node.

readExternal(ObjectInput)

Description

This method reads the information written in the compressed stream by `writeExternal` method and restores the object correspondingly.

Syntax

```
public void readExternal(java.io.ObjectInput inArg)
```

Specified By

`java.io.Externalizable.readExternal(java.io.ObjectInput)` in interface `java.io.Externalizable`

Parameters

`in` - the `ObjectInput` stream used for reading the compressed stream.

Throws

`IOException` - is thrown when there is an error in reading the input stream.

`ClassNotFoundException` - is thrown when the class is not found

validateContent(Element)

Description

Validates the content of a element node.

Syntax

```
public boolean validateContent(org.w3c.dom.Element e)
```

Returns

True if valid, else false

writeExternal(ObjectOutput)

Description

This method saves the state of the object by creating a binary compressed stream with information about this object.

Syntax

```
public void writeExternal(java.io.ObjectOutput outArg)
```

Specified By

`java.io.Externalizable.writeExternal(java.io.ObjectOutput)` in interface `java.io.Externalizable`

Parameters

`out` - The `ObjectOutput` stream used to write the serialized/ compressed stream.

Throws

`IOException` - is thrown when there is an exception while writing the serialized/compressed stream.

NodeFactory

Description of NodeFactory

This class specifies methods to create various nodes of the DOM tree built during parsing. Applications can override these methods to create their own custom classes to be added to the DOM tree while parsing. Applications have to register their own NodeFactory using the XMLParser's `setNodeFactory()` method. If a null pointer is returned by these methods, then the node will not be added to the DOM tree.

Syntax of NodeFactory

```
public class NodeFactory extends java.lang.Object implements  
java.io.Serializable
```

```
java.lang.Object  
|  
+--oracle.xml.parser.v2.NodeFactory
```

Implemented Interfaces of NodeFactory

```
java.io.Serializable
```

See Also

```
DOMParser.setNodeFactory(NodeFactory)
```

Constructors of NodeFactory

NodeFactory()

Syntax

```
public NodeFactory()
```

Methods of NodeFactory

Table 11–15 Summary of Methods of NodeFactory

Method	Description
<code>createAttribute(String, String)</code>	Creates an attribute node with the specified tag, and text.
<code>createAttribute(String, String, String, String)</code>	Creates an attribute node with the specified tag, and text.
<code>createCDATASection(String)</code>	Creates a CDATA node with the specified text.
<code>createComment(String)</code>	Creates a comment node with the specified text.
<code>createDocument()</code>	Creates a document node. This method cannot return a null pointer.
<code>createDocumentFragment()</code>	Creates a document fragment node with the specified tag.
<code>createElement(String)</code>	Creates an Element node with the specified tag.
<code>createElementNS(String, String, String)</code>	Creates an Element node with the specified local name, prefix , namespaceURI
<code>createEntityReference(String)</code>	Creates an entity reference node with the specified tag.
<code>createProcessingInstruction(String, String)</code>	Creates a PI node with the specified tag, and text.
<code>createTextNode(String)</code>	Creates a text node with the specified text.

createAttribute(String, String)

Description

Creates an attribute node with the specified tag, and text.

Syntax

```
public XMLAttr createAttribute(java.lang.String tag, java.lang.String text)
```

Parameters

`tag` - The name of the node.

`text` - The text associated with the node.

Returns

The created attribute node.

createAttribute(String, String, String, String)

Description

Creates an attribute node with the specified tag, and text.

Syntax

```
public XMLAttr createAttribute(java.lang.String localName, java.lang.String prefix, java.lang.String namespaceURI, java.lang.String value)
```

Parameters

`localName` - the name of the node.

`prefix` - the prefix of the node.

`namespaceURI` - the namespace of the node

`value` - The value associated with the node.

Returns

The created attribute node.

createCDATASection(String)

Description

Creates a CDATA node with the specified text.

Syntax

```
public XMLCDATA createCDATASection(java.lang.String text)
```

Parameters

`text` - The text associated with the node.

Returns

The created CDATA node.

createComment(String)

Description

Creates a comment node with the specified text.

Syntax

```
public XMLComment createComment(java.lang.String text)
```

Parameters

text - The text associated with the node.

Returns

The created comment node.

createDocument()

Description

Creates a document node. This method cannot return a null pointer.

Syntax

```
public XMLDocument createDocument()
```

Returns

The created element.

createDocumentFragment()

Description

Creates a document fragment node with the specified tag.

Syntax

```
public XMLDocumentFragment createDocumentFragment()
```

Returns

The created document fragment node.

createElement(String)

Description

Creates an Element node with the specified tag.

Syntax

```
public XMLElement createElement(java.lang.String tag)
```

Parameters

`tag` - The name of the element.

Returns

The created element.

createElementNS(String, String, String)**Description**

Creates an Element node with the specified local name, prefix , namespaceURI

Syntax

```
public XMLElement createElementNS(java.lang.String localName, java.lang.String  
prefix, java.lang.String namespaceURI)
```

Parameters

`localName` - the name of the element

`prefix` - the prefix of the element,

`namespaceURI` - the namespace of the element

Returns

The created element.

createEntityReference(String)**Description**

Creates an entity reference node with the specified tag.

Syntax

```
public XMLEntityReference createEntityReference(java.lang.String tag)
```

Parameters

`tag` - The name of the node.

Returns

The created entity reference node.

createProcessingInstruction(String, String)

Description

Creates a PI node with the specified tag, and text.

Syntax

```
public XMLPI createProcessingInstruction(java.lang.String tag, java.lang.String text)
```

Parameters

`tag` - The name of the node.

`text` - The text associated with the node.

Returns

The created PI node.

createTextNode(String)

Description

Creates a text node with the specified text.

Syntax

```
public XMLText createTextNode(java.lang.String text)
```

Parameters

`text` - The text associated with the node.

Returns

The created text node.

oraxml

Description of oraxml

The oraxml class provides a command-line interface to validate XML files

Table 11–16 *Command-line interface of oraxml*

command	description
-help	Prints the help message
-version	Prints the release version
-novalidate	Parses the input file to check for well-formedness
-dtd	Validates the input file with DTD validation
-schema	Validates the input file with Schema validation
-log <logfile>	Writes the errors/logs to the output file
-comp	Compresses the input xml file
-decomp	Decompresses the input compressed file
-enc	Prints the encoding of the input file
-warning	Show warnings

Syntax of oraxml

```
public class oraxml extends java.lang.Object
```

```
java.lang.Object
|
+--oracle.xml.parser.v2.oraxml
```

Constructors of oraxml

oraxml()

Syntax

```
public oraxml()
```

Methods of oraxml

main(String[])

syntax

```
public static void main(java.lang.String[] args)
```

SAXAttrList

Description of SAXAttrList

This class implements the `SAX AttributeList` interface and also provides Namespace support. Applications that require Namespace support can explicitly cast any attribute list returned by an Oracle parser class to `SAXAttrList` and use the methods described here. It also implements `Attributes (SAX 2.0)` interface

Syntax of SAXAttrList

```
public class SAXAttrList
```

```
oracle.xml.parser.v2.SAXAttrList
```

Comments on SAXAttrList

This interface allows access to a list of attributes in three different ways:

- by attribute index;
- by Namespace-qualified name; or
- by qualified (prefixed) name.

The list will not contain attributes that were declared `#IMPLIED` but not specified in the start tag. It will also not contain attributes used as Namespace declarations (`xmlns*`) unless the `http://xml.org/sax/features/namespace-prefixes` feature is set to true (it is false by default).

If the `namespace-prefixes` feature (see above) is false, access by qualified name may not be available; if the `http://xml.org/sax/features/namespaces` feature is false, access by Namespace-qualified names may not be available.

This interface replaces the now-deprecated `SAX1` interface, which does not contain Namespace support. In addition to Namespace support, it adds the `getIndex` methods (below).

The order of attributes in the list is unspecified, and will vary from implementation to implementation.

Constructors of SAXAttrList

SAXAttrList(int)

Syntax

```
public SAXAttrList(int elems)
```

Methods of SAXAttrList

addAttr(String, String, String, String, boolean, int)

Description

Add an attribute to the parent element node.

Syntax

```
public void addAttr(java.lang.String pfx, java.lang.String lname,  
java.lang.String tag, java.lang.String value, boolean spec, int type)
```

Parameters

`pfx` - the prefix of the attribute

`lname` - the local name of the attribute

`tag` - the qname of the attribute

`value` - the attribute value

`spec` - the specified flag

`type` - the attribute type

addAttr(String, String, String, String, boolean, int, String)

Description

Add an attribute to the parent element node.

Syntax

```
public void addAttr(java.lang.String pfx, java.lang.String lname,  
java.lang.String tag, java.lang.String value, boolean spec, int type,  
java.lang.String nmsp)
```

Parameters

`pfx` - the prefix of the attribute

`lname` - the local name of the attribute

`tag` - the qname of the attribute

`value` - the attribute value

`spec` - the specified flag

`type` - the attribute type #param nmsp the namespace of the attribute

getExpandedName(int)

Description

Get the expanded name for an attribute in the list (by position)

Syntax

```
public java.lang.String getExpandedName(int i)
```

Parameters

`i` - The index of the attribute in the list.

Returns

The expanded name for the attribute

getIndex(String)

Description

Look up the index of an attribute by XML 1.0 qualified name.

Syntax

```
public int getIndex(java.lang.String qName)
```

Parameters

`qName` - The qualified (prefixed) name.

Returns

The index of the attribute, or -1 if it does not appear in the list.

getIndex(String, String)

Description

Look up the index of an attribute by Namespace name.

Syntax

```
public int getIndex(java.lang.String uri, java.lang.String localPart)
```

Parameters

`uri` - The Namespace URI, or the empty string if the name has no Namespace URI.

`localName` - The attribute's local name.

Returns

The index of the attribute, or -1 if it does not appear in the list.

Since

SAX2

getLength()

Description

Return the number of attributes in this list.

Syntax

```
public int getLength()
```

Comments

The SAX parser may provide attributes in any arbitrary order, regardless of the order in which they were declared or specified. The number of attributes may be zero.

Returns

The number of attributes in the list.

getLocalName(int)

Look up an attribute's local name by index.

Description

Syntax

```
public java.lang.String getLocalName(int index)
```

Parameters

`index` - The attribute index (zero-based).

Returns

The local name, or the empty string if Namespace processing is not being performed, or null if the index is out of range.

Since

SAX2

See Also

`getLength()`

getPrefix(int)**Description**

Get the namespace prefix for an attribute in the list (by position)

Syntax

```
public java.lang.String getPrefix(int i)
```

Parameters

`i` - The index of the attribute in the list.

Returns

The namespace prefix for the attribute

getQName(int)**Description**

Look up an attribute's XML 1.0 qualified name by index.

Syntax

```
public java.lang.String getQName(int index)
```

Parameters

`index` - The attribute index (zero-based).

Returns

The XML 1.0 qualified name, or the empty string if none is available, or null if the index is out of range.

Since

SAX2

See Also

`getLength()`

getType(int)

Description

Look up an attribute's type by index.

Syntax

```
public java.lang.String getType(int index)
```

Comments

The attribute type is one of the strings "CDATA", "ID", "IDREF", "IDREFS", "NMTOKEN", "NMTOKENS", "ENTITY", "ENTITIES", or "NOTATION" (always in upper case).

If the parser has not read a declaration for the attribute, or if the parser does not report attribute types, then it must return the value "CDATA" as stated in the XML 1.0 Recommendation (clause 3.3.3, "Attribute-Value Normalization").

For an enumerated attribute that is not a notation, the parser will report the type as "NMTOKEN".

Parameters

`index` - The attribute index (zero-based).

Returns

The attribute's type as a string, or null if the `index` is out of range.

See Also

`getLength()`

getType(String)

Description

Look up an attribute's type by XML 1.0 qualified name.

Syntax

```
public java.lang.String getType(java.lang.String qName)
```

Comments

See `getType(int)` for a description of the possible types.

Parameters

`qName` - The XML 1.0 qualified name.

Returns

The attribute type as a string, or null if the attribute is not in the list or if qualified names are not available.

getType(String, String)

Description

Look up an attribute's type by Namespace name.

Syntax

```
public java.lang.String getType(java.lang.String uri, java.lang.String  
localName)
```

Comments

See `getType(int)` for a description of the possible types.

Parameters

`uri` - The Namespace URI, or the empty String if the name has no Namespace URI.

`localName` - The local name of the attribute.

Returns

The attribute type as a string, or null if the attribute is not in the list or if Namespace processing is not being performed.

Since

SAX2

getURI(int)

Description

Look up an attribute's Namespace URI by index.

Syntax

```
public java.lang.String getURI(int index)
```

Parameters

`index` - The attribute index (zero-based).

Returns

The Namespace URI, or the empty string if none is available, or null if the index is out of range.

Since

SAX2

See Also

`getLength()`

getValue(int)

Description

Look up an attribute's value by index.

Syntax

```
public java.lang.String getValue(int index)
```

Description

If the attribute value is a list of tokens (IDREFS, ENTITIES, or NMTOKENS), the tokens will be concatenated into a single string with each token separated by a single space.

Parameters

`index` - The attribute index (zero-based).

Returns

The attribute's value as a string, or null if the `index` is out of range.

See Also

`getLength()`

getValue(String)**Description**

Look up an attribute's value by XML 1.0 qualified name.

Syntax

```
public java.lang.String getValue(java.lang.String qName)
```

Comments

See `getValue(int)` for a description of the possible values.

Parameters

`qName` - The XML 1.0 qualified name.

Returns

The attribute value as a string, or null if the attribute is not in the list or if qualified names are not available.

getValue(String, String)**Description**

Look up an attribute's value by Namespace name.

Syntax

```
public java.lang.String getValue(java.lang.String uri, java.lang.String  
localName)
```

Comments

See `getValue(int)` for a description of the possible values.

Parameters

`uri` - The Namespace URI, or the empty String if the name has no Namespace URI.
`localName` - The local name of the attribute.

Returns

The attribute value as a string, or null if the attribute is not in the list.

Since

SAX2

reset()

Description

Resets the SAXAttrList

Syntax

```
public void reset()
```

SAXParser

Syntax

```
public class SAXParser
```

```
oracle.xml.parser.v2.SAXParser
```

Description

This class implements an eXtensible Markup Language (XML) 1.0 SAX parser according to the World Wide Web Consortium (W3C) recommendation. Applications can register a SAX handler to receive notification of various parser events.

XMLReader is the interface that an XML parser's SAX2 driver must implement. This interface allows an application to set and query features and properties in the parser, to register event handlers for document processing, and to initiate a document parse.

All SAX interfaces are assumed to be synchronous: the parse methods must not return until parsing is complete, and readers must wait for an event-handler callback to return before reporting the next event.

This interface replaces the (now deprecated) SAX 1.0 Parser interface. The XMLReader interface contains two important enhancements over the old Parser interface:

- it adds a standard way to query and set features and properties; and
- it adds Namespace support, which is required for many higher-level XML standards.

Constructors

SAXParser()

Description

Creates a new parser object.

Syntax

```
public SAXParser()
```

Methods

getContentHandler()

Description

Return the current content handler.

Syntax

```
public org.xml.sax.ContentHandler getContentHandler()
```

Returns

The current content handler, or null if none has been registered.

Since

SAX 2.0

See Also

```
setContentHandler(ContentHandler)
```

getDTDHandler()

Description

Return the current DTD handler.

Syntax

```
public org.xml.sax.DTDHandler getDTDHandler()
```

Returns

The current DTD handler, or null if none has been registered.

Since

SAX 2.0

See Also

```
setDTDHandler(DTDHandler)
```

getFeature(String)

Description

Look up the value of a feature.

Syntax

```
public boolean getFeature(java.lang.String name)
```

Comments

The feature name is any fully-qualified URI. It is possible for an XMLReader to recognize a feature name but to be unable to return its value; this is especially true in the case of an adapter for a SAX1 Parser, which has no way of knowing whether the underlying parser is performing validation or expanding external entities.

All XMLReaders are required to recognize the

<http://xml.org/sax/features/namespaces> and the

<http://xml.org/sax/features/namespace-prefixes> feature names.

Some feature values may be available only in specific contexts, such as before, during, or after a parse.

Typical usage is something like this:

```
XMLReader r = new MySAXDriver();

                                // try to activate validation
try {
    r.setFeature("http://xml.org/sax/features/validation", true);
} catch (SAXException e) {
    System.err.println("Cannot activate validation.");
}

                                // register event handlers
r.setContentHandler(new MyContentHandler());
r.setErrorHandler(new MyErrorHandler());

                                // parse the first document
try {
    r.parse("http://www.foo.com/mydoc.xml");
} catch (IOException e) {
    System.err.println("I/O exception reading XML document");
} catch (SAXException e) {
    System.err.println("XML exception reading document.");
}
```

The feature "http://xml.org/sax/features/validation" due to its binary input value only controls DTD validation. The value true sets DTD validation to TRUE. This feature cannot be used to control XML Schema based validation.

Implementors are free (and encouraged) to invent their own features, using names built on their own URIs.

Parameters

feature - Name of the feature, which is a fully-qualified URI.

version - Version of the feature.

Returns

The current state of the feature (true or false).

Throws

`org.xml.sax.SAXNotRecognizedException` - When the XMLReader does not recognize the feature name.

`org.xml.sax.SAXNotSupportedException` - When the XMLReader recognizes the feature name but cannot determine its value at this time.

See Also

`setFeature(String, boolean)`

getProperty(String)

Description

Look up the value of a property.

Syntax

```
public java.lang.Object getProperty(java.lang.String name)
```

Comments

The property name is any fully-qualified URI. It is possible for an XMLReader to recognize a property name but to be unable to return its state; this is especially true in the case of an adapter for a SAX1 Parser.

XMLReaders are not required to recognize any specific property names, though an initial core set is documented for SAX2.

Some property values may be available only in specific contexts, such as before, during, or after a parse.

Implementors are free (and encouraged) to invent their own properties, using names built on their own URIs.

Parameters

`name` - The property name, which is a fully-qualified URI.

Returns

The current value of the property.

Throws

`org.xml.sax.SAXNotRecognizedException` - When the XMLReader does not recognize the property name.

`org.xml.sax.SAXNotSupportedException` - When the XMLReader recognizes the property name but cannot determine its value at this time.

See Also

`setProperty(String, Object)`

setContentHandler(ContentHandler)**Description**

Allow an application to register a content event handler.

Syntax

```
public void setContentHandler(org.xml.sax.ContentHandler handler)
```

Comments

If the application does not register a content handler, all content events reported by the SAX parser will be silently ignored.

Applications may register a new or different handler in the middle of a parse, and the SAX parser must begin using the new handler immediately.

Parameters

`handler` - The content handler.

Throws

`java.lang.NullPointerException` - If the handler argument is null.

Since

SAX 2.0

See Also

`getContentHandler()`

setDTDHandler(DTDHandler)

Description

Allow an application to register a DTD event handler.

Syntax

```
public void setDTDHandler(org.xml.sax.DTDHandler handler)
```

Comments

If the application does not register a DTD handler, all DTD events reported by the SAX parser will be silently ignored.

Applications may register a new or different handler in the middle of a parse, and the SAX parser must begin using the new handler immediately.

Parameters

`handler` - The DTD handler.

Throws

`java.lang.NullPointerException` - If the `handler` argument is null.

See Also

`getDTDHandler()`

setFeature(String, boolean)

Description

Set the state of a feature.

Syntax

```
public void setFeature(java.lang.String name, boolean value)
```

Comments

The feature name is any fully-qualified URI. It is possible for an XMLReader to recognize a feature name but to be unable to set its value; this is especially true in the case of an adapter for a SAX1 Parser, which has no way of affecting whether the underlying parser is validating, for example.

All XMLReaders are required to support setting `http://xml.org/sax/features/namespace` to true and `http://xml.org/sax/features/namespace-prefixes` to false.

Some feature values may be immutable or mutable only in specific contexts, such as before, during, or after a parse.

The feature "`http://xml.org/sax/features/validation`" due to its binary input value only controls DTD validation. The value true sets DTD validation to TRUE. This feature cannot be used to control XML Schema based validation

Parameters

`name` - The feature name, which is a fully-qualified URI.

`state` - The requested state of the feature (true or false).

Throws

`org.xml.sax.SAXNotRecognizedException` - When the XMLReader does not recognize the feature name.

`org.xml.sax.SAXNotSupportedException` - When the XMLReader recognizes the feature name but cannot set the requested value.

See Also

`getFeature(String)`

setProperty(String, Object)

Description

Set the value of a property.

Syntax

```
public void setProperty(java.lang.String name, java.lang.Object value)
```

Comments

The property name is any fully-qualified URI. It is possible for an XMLReader to recognize a property name but to be unable to set its value; this is especially true in the case of an adapter for a SAX1 Parser .

XMLReaders are not required to recognize setting any specific property names, though a core set is provided with SAX2.

Some property values may be immutable or mutable only in specific contexts, such as before, during, or after a parse.

This method is also the standard mechanism for setting extended handlers.

Parameters

`name` - The property name, which is a fully-qualified URI.

`state` - The requested value for the property.

Throws

`org.xml.sax.SAXNotRecognizedException` - When the XMLReader does not recognize the property name.

`org.xml.sax.SAXNotSupportedException` - When the XMLReader recognizes the property name but cannot set the requested value.

XMLAttr

Syntax of XMLAttr

```
public class XMLAttr implements oracle.xml.parser.v2.NSName,  
java.io.Externalizable
```

```
oracle.xml.parser.v2.XMLAttr
```

Implemented Interfaces of XMLAttr

```
java.io.Externalizable, NSName, oracle.xml.util.NSName, java.io.Serializable
```

Description

This class implements the DOM Attr interface and holds information on each attribute of an element.

See Also

Attr, NodeFactory, DOMParser.setNodeFactory(NodeFactory)

XMLAttr()

Description

Default constructor.

Syntax

```
public XMLAttr()
```

Comments

Deprecated. Use createAttribute(String) or createAttributeNS(String, String) of XMLDocument

Note that this constructor is used only during deserialization/decompression of this DOM node. In order to deserialize this node to construct the DOM node from the serialized/ compressed stream, it is required to create a handle of the object. For all normal XMLAttr creation use createAttribute(String) or createAttributeNS(String, String) of XMLDocument

XMLAttr(String, String)

Description

Construct attribute with given name and value.

Syntax

```
public XMLAttr(java.lang.String n, java.lang.String v)
```

Comments

Deprecated; use createAttribute(String) method of XMLDocument

Parameters

n - Name of the attribute

v - Value of the attribute

XMLAttr(String, String, String, String)

Description

Namespace support

Syntax

```
public XMLAttr(java.lang.String name, java.lang.String prefix, java.lang.String namespace, java.lang.String v)
```

Comments

Deprecated; use createAttributeNS(String, String) method of XMLDocument

Parameters

name - Local name of the attribute

prefix - Prefix of the attribute

namespace - Namespace of the attribute

v - Value of the attribute

XMLAttr(String, String, String, String, String)

Description

private constructor (doesn't intern names)

Syntax

```
public XMLAttr(java.lang.String name, java.lang.String prefix, java.lang.String qname, java.lang.String namespace, java.lang.String v)
```

Comments

Deprecated; use `createAttribute(String)` or `createAttributeNS(String, String)` of `XMLDocument`

Parameters

`name` - Local name of the attribute

`prefix` - Prefix of the attribute

`qname` - Qname of the attribute

`namespace` - Namespace of the attribute

`v` - Value of the attribute

Methods

addText(String)

Description

Syntax

```
public XMLNode addText(java.lang.String str)
```

cloneNode(boolean)

Description

Returns a duplicate of this node, i.e., serves as a generic copy constructor for nodes.

Syntax

```
public org.w3c.dom.Node cloneNode(boolean deep)
```

Comments

The duplicate node has no parent (`parentNode` returns `null`). Cloning an `Element` copies all attributes and their values, including those generated by the XML processor to represent defaulted attributes, but this method does not copy any text it contains unless it is a deep clone, since the text is contained in a child `Text` node. Cloning any other type of node simply returns a copy of this node.

Parameters

`deep` - If `true`, recursively clone the subtree under the specified node; if `false`, clone only the node itself (and its attributes, if it is an `Element`).

Returns

The duplicate node.

getExpandedName()

Description

Gets the fully resolved Name for this attribute

Syntax

```
public java.lang.String getExpandedName()
```

Specified By

`oracle.xml.util.NSName.getExpandedName()` in interface `oracle.xml.util.NSName`

Returns

the fully resolved Name

getLocalName()

Description

Gets the local name of this attribute

Syntax

```
public java.lang.String getLocalName()
```

Specified By

`oracle.xml.util.NSName.getLocalName()` in interface `oracle.xml.util.NSName`

Returns

the local Name for this attribute

Since

DOM 2

getName()

Description

Gets the attribute name.

Syntax

```
public java.lang.String getName()
```

Returns

attribute name

getNamespaceURI()

Description

Gets the namespace of the attribute

Syntax

```
public java.lang.String getNamespaceURI()
```

Returns

the namespace URI associated with this attribute

Since

DOM 2

getNextAttribute()

Description

Gets the next attribute if any

Syntax

```
public XMLAttr getNextAttribute()
```

Returns

the next attribute

getNextSibling()

Description

Syntax

```
public org.w3c.dom.Node getNextSibling()
```

getNodeType()

Description

Gets a code representing the type of the underlying object

Syntax

```
public short getNodeType()
```

Returns

type of the node

getNodeValue()

Description

Gets the value of this node, depending on its type

Syntax

```
public java.lang.String getNodeValue()
```

Returns

Value of this node

Throws

`DOMException - NO_MODIFICATION_ALLOWED_ERR`: Raised when the node is readonly. `DOMSTRING_SIZE_ERR`: Raised when it would return more characters than fit in a `DOMString` variable on the implementation platform.

getOwnerElement()

Description

Gets the element which owns this attribute

Syntax

```
public org.w3c.dom.Element getOwnerElement()
```

Returns

the element node that owns this attribute

Since

DOM 2

getParentNode()

Description

Gets the parent of this node.

Syntax

```
public org.w3c.dom.Node getParentNode()
```

Comments

All nodes, except `Document`, `DocumentFragment`, and `Attr` may have a parent. However, if a node has just been created and not yet added to the tree, or if it has been removed from the tree, this is `null`.

Returns

The parent of this node

getPrefix()

Description

Gets the name space prefix of the element

Syntax

```
public java.lang.String getPrefix()
```

Specified By

oracle.xml.util.NSName.getPrefix() in interface oracle.xml.util.NSName

Returns

the namespace prefix for this attribute

Since

DOM 2

getPreviousSibling()

Description

Syntax

```
public org.w3c.dom.Node getPreviousSibling()
```

getSpecified()

Description

Returns true if the attribute was specified explicitly in the element

Syntax

```
public boolean getSpecified()
```

Returns

true, if the attribute was specified explicitly, false, if it was not

getValue()

Description

Gets the attribute value.

Syntax

```
public java.lang.String getValue()
```

Returns

attribute value

readExternal(ObjectInput)

Description

This method restores the information written by writeExternal.

Syntax

```
public void readExternal(java.io.ObjectInput inArg)
```

Specified By

java.io.Externalizable.readExternal(java.io.ObjectInput) in interface java.io.Externalizable

Parameters

inArg - is the ObjectInput stream used to read the compressed stream

Throws

IOException - is thrown when there is an exception while reading the compressed stream.

ClassNotFoundException - is thrown when the class is not found

setNodeValue(String)

Description

Sets the value of this node, depending on its type

Syntax

```
public void setNodeValue(java.lang.String nodeValue)
```

Parameters

nodeValue - the value of the node to be set

Throws

DOMException - NO_MODIFICATION_ALLOWED_ERR: Raised when the node is readonly. DOMSTRING_SIZE_ERR: Raised when it would return more characters than fit in a DOMString variable on the implementation platform.

setValue(String)

Description

Sets the value.

Syntax

```
public void setValue(java.lang.String arg)
```

Parameters

`arg` - Value to set

writeExternal(ObjectOutput)

Description

This method saves the state of the object. The object information is saved in a binary compressed stream.

Syntax

```
public void writeExternal(java.io.ObjectOutput outArg)
```

Specified By

`java.io.Externalizable.writeExternal(java.io.ObjectOutput)` in interface `java.io.Externalizable`

Parameters

`outArg` - The `ObjectOutput` stream used to write the compressed stream

Throws

`IOException` - is thrown when there is an exception while writing the compressed stream.

XMLCDATA

Description

This class implements the DOM CDATASection interface.

Syntax

```
public class XMLCDATA implements java.io.Externalizable
```

```
oracle.xml.parser.v2.XMLCDATA
```

Implemented Interfaces

```
java.io.Externalizable, java.io.Serializable
```

See Also

```
CDATASection, NodeFactory, DOMParser.setNodeFactory(NodeFactory)
```

Constructors

XMLCDATA()

Description

Default constructor.

Syntax

```
public XMLCDATA()
```

Comments

Note that this constructor is used only during deserialization/decompression of this DOM node. In order to deserialize this node to construct the DOM node from the serialized/ compressed stream, it is required to create a handle of the object.

Methods

getNodeName()

Description

Gets a name of the node

Syntax

```
public java.lang.String getNodeName()
```

Returns

name of the node

getNodeType()

Description

Gets a code representing the type of the underlying object

Syntax

```
public short getNodeType()
```

Returns

type of the node

readExternal(ObjectInput)

Description

This method reads the information written in the compressed stream by writeExternal method and restores the object correspondingly.

Syntax

```
public void readExternal(java.io.ObjectInput inArg)
```

Specified By

java.io.Externalizable.readExternal(java.io.ObjectInput) in interface java.io.Externalizable

Parameters

`inArg` - The `ObjectInputStream` used for reading the compressed stream

Throws

`IOException` - is thrown when there is an error in reading the input stream.

`ClassNotFoundException` - is thrown when the class is not found

writeExternal(ObjectOutput)**Description**

This method saves the state of the object by creating a binary compressed stream with information about this object.

Syntax

```
public void writeExternal(java.io.ObjectOutput outArg)
```

Specified By

`java.io.Externalizable.writeExternal(java.io.ObjectOutput)` in interface `java.io.Externalizable`

Parameters

`outArg` - is the `ObjectOutput` stream used to write the compressed stream.

Throws

`IOException` - is thrown when there is an exception while writing the compressed stream.

XMLComment

Description

This class implements the DOM Comment interface.

Syntax

```
public class XMLComment implements java.io.Externalizable
```

```
oracle.xml.parser.v2.XMLComment
```

Implemented Interfaces

java.io.Externalizable, java.io.Serializable

See Also

Comment, NodeFactory, DOMParser.setNodeFactory(NodeFactory)

Constructors

XMLComment()

Description

Default constructor.

Syntax

```
public XMLComment()
```

Comments

Note that this constructor is used only during deserialization/decompression of this DOM node. In order to deserialize this node to construct the DOM node from the serialized/ compressed stream, it is required to create a handle of the object.

Methods

addText(String)

Description

Adds the comment text

Syntax

```
public XMLNode addText(java.lang.String str)
```

Parameters

`str` - the comment text

getNodeName()**Description**

Gets a name of the node

Syntax

```
public java.lang.String getNodeName()
```

Returns

name of the node

getNodeType()**Description**

Gets a code representing the type of the underlying object

Syntax

```
public short getNodeType()
```

Returns

type of the node

readExternal(ObjectInput)**Description**

This method reads the information written in the compressed stream by `writeExternal` method and restores the object correspondingly.

Syntax

```
public void readExternal(java.io.ObjectInput inArg)
```

Specified By

`java.io.Externalizable.readExternal(java.io.ObjectInput)` in interface `java.io.Externalizable`

Parameters

`inArg` - The `ObjectInput` stream used for reading the compressed stream

Throws

`IOException` - is thrown when there is an error in reading the input stream.
`ClassNotFoundException` - is thrown when the class is not found

reportSAXEvents(ContentHandler)

Description

Report SAX Events from a DOM Tree

Syntax

```
public void reportSAXEvents(org.xml.sax.ContentHandler cntHandler)
```

Parameters

`ContentHandler` - `cntHandler`

Throws

`SAXException` - thrown by SAX Callback functions

writeExternal(ObjectOutput)

Description

This method saves the state of the object by creating a binary compressed stream with information about this object.

Syntax

```
public void writeExternal(java.io.ObjectOutput outArg)
```

Specified By

`java.io.Externalizable.writeExternal(java.io.ObjectOutput)` in interface `java.io.Externalizable`

Parameters

`outArg` - is the `ObjectOutput` stream used to write the compressed stream.

Throws

`IOException` - is thrown when there is an exception while writing the compressed stream.

XMLDeclPI

Description

This class implements the XML Decl Processing Instruction.

Syntax

```
public class XMLDeclPI extends oracle.xml.parser.v2.XMLPI implements  
java.io.Externalizable
```

```
oracle.xml.parser.v2.XMLPI  
|  
+--oracle.xml.parser.v2.XMLDeclPI
```

Implemented Interfaces

```
java.io.Externalizable, java.io.Serializable
```

See Also

ProcessingInstruction

Constructors

XMLDeclPI()

Description

Syntax

```
public XMLDeclPI()
```

XMLDeclPI(String, String, String, boolean)

Description

Syntax

```
public XMLDeclPI(java.lang.String version, java.lang.String encoding,  
java.lang.String standalone, boolean textDecl)
```

Methods

cloneNode(boolean)

Description

Returns a duplicate of this node, i.e., serves as a generic copy

Syntax

```
public org.w3c.dom.Node cloneNode(boolean deep)
```

Returns

The duplicate node.

getData()

Description

Returns the fully constructed string 'version=1.0 ...'

Syntax

```
public java.lang.String getData()
```

Returns

the nodevalue

Throws

`DOMException - DOMSTRING_SIZE_ERR`: Raised when it would return more characters than fit in a `DOMString` variable on the implementation platform.

getEncoding()

Description

Retrieves the character encoding information.

Syntax

```
public final java.lang.String getEncoding()
```

Returns

the encoding information stored in the `<?xml ...?>` tag or the user-defined output encoding if it has been more recently set.

getNodeValue()

Description

Gets the value of this node

Syntax

```
public java.lang.String getNodeValue()
```

Returns

the nodevalue

Throws

`DOMException - DOMSTRING_SIZE_ERR`: Raised when it would return more characters than fit in a `DOMString` variable on the implementation

getStandalone()

Description

Retrieves the standalone information.

Syntax

```
public final java.lang.String getStandalone()
```

Returns

the standalone attribute stored in the `<?xml ...?>` tag.

getVersion()

Description

Retrieves the version information.

Syntax

```
public final java.lang.String getVersion()
```


Returns

the version number stored in the `<?xml ...?>` tag.

readExternal(ObjectInput)**Description**

This method reads the information written in the compressed stream by `writeExternal` method and restores the object correspondingly.

Syntax

```
public void readExternal(java.io.ObjectInput inArg)
```

Specified By

`java.io.Externalizable.readExternal(java.io.ObjectInput)` in interface `java.io.Externalizable`

Specified By

`java.io.Externalizable.readExternal(java.io.ObjectInput)` in interface `java.io.Externalizable`

Overrides

`XMLPI.readExternal(ObjectInput)` in class `XMLPI`

Parameters

`inArg` - The `ObjectInput` stream used for reading the compressed stream

Throws

`IOException` - is thrown when there is an error in reading the input stream.

`ClassNotFoundException` - is thrown when the class is not found

setEncoding(String)**Description**

Sets the character encoding for output.

Syntax

```
public final void setEncoding(java.lang.String encoding)
```

Comments

Eventually it sets the ENCODING stored in the `<?xml ...?>` tag, but not until the document is saved. You should not call this method until the Document has been loaded.

Parameters

`encoding` - The character encoding to set

setStandalone(String)

Description

Sets the standalone information stored in the `<?xml ...?>` tag.

Syntax

```
public final boolean setStandalone(java.lang.String value)
```

Parameters

`value` - The attribute value ('yes' or 'no').

setVersion(String)

Description

Sets the version number stored in the `<?xml ...?>` tag.

Syntax

```
public final void setVersion(java.lang.String version)
```

Parameters

`version` - The version information to set.

writeExternal(ObjectOutput)

Description

This method saves the state of the object by creating a binary compressed stream with information about this object.

Syntax

```
public void writeExternal(java.io.ObjectOutput outArg)
```

Specified By

`java.io.Externalizable.writeExternal(java.io.ObjectOutput)` in interface `java.io.Externalizable`

Overrides

`XMLPI.writeExternal(ObjectOutput)` in class `XMLPI`

Parameters

`outArg` - is the `ObjectOutput` stream used to write the compressed stream.

Throws

`IOException` - is thrown when there is an exception while writing the compressed stream.

XMLDocument

Description

This class implements the DOM Document interface, represents an entire XML document and serves the root of the Document Object Model tree. Each XML tag can either represent a node or a leaf of this tree.

Syntax

```
public class XMLDocument implements java.io.Externalizable
```

```
oracle.xml.parser.v2.XMLDocument
```

Implemented Interfaces

```
java.io.Externalizable, java.io.Serializable
```

Comments

According to the XML specification, the root of the tree consists of any combination of comments and processing instructions, but only one root element. A helper method `getDocumentElement` is provided as a short cut to finding the root element.

Constructors

XMLDocument()

Description

Creates an empty document.

Syntax

```
public XMLDocument()
```

Methods

addID(String, XMLElement)

Description

Add a ID Element associated with this document

Syntax

```
public void addID(java.lang.String name, XMLElement e)
```

Parameters

id - String - id value

e - XMLElement associated with id

adoptNode(Node)**Description**

Adopts a node from another document to this document.

Syntax

```
public org.w3c.dom.Node adoptNode(org.w3c.dom.Node srcNode)
```

Comments

The returned node has no parent; (parentNode is null). The source node is removed from the original document;

Parameters

Node - to be adopted

Returns

Node with document association updated

Throws

DOMException - NOT_SUPPORTED_ERR: Raised if the type of the node being adopted is not supported.

Since

DOM 2

appendChild(Node)**Description**

Appends a new node to the document

Syntax

```
public org.w3c.dom.Node appendChild(org.w3c.dom.Node elem)
```

Parameters

the - new node to be added

Returns

the node after appending it to the document

Throws

`DOMException - HIERARCHY_REQUEST_ERR`: Raised if this node is of a type that does not allow children of the type of the `elem` node. `WRONG_DOCUMENT_ERR`: Raised if `elem` was created from a different document than this one.

cloneNode(boolean)

Description

Returns a duplicate of this node, i.e., serves as a generic copy constructor for nodes.

Syntax

```
public org.w3c.dom.Node cloneNode(boolean deep)
```

Comments

The duplicate node has no parent (`parentNode` returns `null`). Cloning an `Element` copies all attributes and their values, including those generated by the XML processor to represent defaulted attributes, but this method does not copy any text it contains unless it is a deep clone, since the text is contained in a child `Text` node. Cloning any other type of node simply returns a copy of this node.

Parameters

`deep` - If `true`, recursively clone the subtree under the specified node; if `false`, clone only the node itself (and its attributes, if it is an `Element`).

Returns

The duplicate node.

createAttribute(String)

Description

Creates an `Attr` of the given name.

Syntax

```
public org.w3c.dom.Attr createAttribute(java.lang.String name)
```

Comments

Note that the `Attr` instance can then be set on an `Element` using the `setAttribute` method.

Parameters

`name` - The name of the attribute.

Returns

A new `Attr` object.

Throws

`DOMException - INVALID_CHARACTER_ERR`: Raised if the specified name contains an invalid character.

createAttributeNS(String, String)

Description

Creates an attribute with the given qualified name and namespace URI.

Syntax

```
public org.w3c.dom.Attr createAttributeNS(java.lang.String namespaceURI,  
java.lang.String qualifiedName)
```

Parameters

`namespaceURI` - namespace of the attribute/element to be created

`qualifiedName` - qualified name of the attribute/element to be created

Returns

Element node with given qualified name and namespace URI

Throws

`DOMException - INVALID_CHARACTER_ERR`: Raised if the specified qualified name contains illegal Characters
`DOMException - NAMESPACE_ERR`: Raised if the qualified name is malformed, if the qualified name has a prefix and the namespace URI is null or an empty string, or if the qualified name has a prefix that is "xml" and namespace URI is different from "http://www.w3.org/2000/xmlns/"

Since

DOM 2

createCDATASection(String)

Description

Creates a `CDATASection` node whose value is the specified string.

Syntax

```
public org.w3c.dom.CDATASection createCDATASection(java.lang.String data)
```

Parameters

`data` - The data for the `CDATASection` contents.

Returns

The new `CDATASection` object.

Throws

`DOMException` - A `DOMException` could be thrown.

createComment(String)

Description

Creates a `Comment` node given the specified string.

Syntax

```
public org.w3c.dom.Comment createComment(java.lang.String data)
```

Parameters

`data` - The data for the node.

Returns

The new `Comment` object.

createDocumentFragment()**Description**

Creates an empty `DocumentFragment` object.

Syntax

```
public org.w3c.dom.DocumentFragment createDocumentFragment()
```

Returns

A new `DocumentFragment`.

createElement(String)**Description**

Creates an element of the type specified.

Syntax

```
public org.w3c.dom.Element createElement(java.lang.String tagName)
```

Comments

Note that the instance returned implements the `Element` interface, so attributes can be specified directly on the returned object.

Parameters

`tagName` - The name of the element type to instantiate. The name is treated as case-sensitive.

Returns

A new `Element` object.

Throws

`DOMException - INVALID_CHARACTER_ERR`: Raised if the specified name contains an invalid character.

createElementNS(String, String)

Description

Creates an element of the given qualified name and namespace URI.

Syntax

```
public org.w3c.dom.Element createElementNS(java.lang.String namespaceURI,  
java.lang.String qualifiedName)
```

Parameters

namespaceURI - namespace of the attribute/element to be created

qualifiedName - qualified name of the attribute/element to be created

Returns

Element node with given qualified name and namespace URI

Throws

`DOMException - INVALID_CHARACTER_ERR`: Raised if the specified qualified name contains illegal Characters
`DOMException - NAMESPACE_ERR`: Raised if the qualified name is malformed, if the qualified name has a prefix and the namespace URI is null or an empty string, or if the qualifiedName has a prefix that is "xml" and namespace URI is different from "http://www.w3.org/XML/1998/namespace"

Since

DOM 2

createEntityReference(String)

Description

Creates an EntityReference object.

Syntax

```
public org.w3c.dom.EntityReference createEntityReference(java.lang.String name)
```

Parameters

name - The name of the entity to reference.

Returns

The new EntityReference object.

Throws

`DOMException - INVALID_CHARACTER_ERR`: Raised if the specified name contains an invalid character.

createEvent(String)**Description**

Creates an event object of the specified type

Syntax

```
public org.w3c.dom.events.Event createEvent(java.lang.String type)
```

Parameters

`type` - the type of the event

Returns

Event object of the specified type

createMutationEvent(String)**Description**

Creates a Mutation Event object of specified type

Syntax

```
public org.w3c.dom.events.MutationEvent createMutationEvent(java.lang.String type)
```

Parameters

`type` - the type of the mutation event

Returns

Event object of the specified type

createNodeIterator(Node, int, NodeFilter, boolean)**Description**

Creates a Node Iterator with specified root, flag which governs what type of nodes it should include in logical view, filter for filtering nodes, flag determining whether entity references and its descendants could be included

Syntax

```
public org.w3c.dom.traversal.NodeIterator createNodeIterator(org.w3c.dom.Node
root, int whatToShow, org.w3c.dom.traversal.NodeFilter filter, boolean
expandEntityReferences)
```

Parameters

root - root node of the iterator.

whatToShow - flag indicating what type of nodes will be included in the iterator/tree walker.

filter - NodeFilter to filter unwanted nodes from the iterator/tree walker.

expandEntityReference - flag to indicate traversal of entity references.

Returns

object that implements NodeIterator interface, created on this document

Throws

DOMException - NOT_SUPPORTED_ERR: if the NodeIterator could not be created with specified root

createProcessingInstruction(String, String)

Description

Creates a ProcessingInstruction node given the specified name and data strings.

Syntax

```
public org.w3c.dom.ProcessingInstruction
createProcessingInstruction(java.lang.String target, java.lang.String data)
```

Parameters

target - The target part of the processing instruction.

data - The data for the node.

Returns

The new ProcessingInstruction object.

Throws

`DOMException - INVALID_CHARACTER_ERR`: Raised if an invalid character is specified.

createRange()**Description**

Create a new Document Range Object ,with Start and End Boundary points at the begining of the document.

Syntax

```
public org.w3c.dom.ranges.Range createRange()
```

Returns

new Range object with start and end boundary points at the beginning of the document

createRangeEvent(String)**Description**

Creates a Range Event object of specified type

Syntax

```
public org.w3c.dom.events.Event createRangeEvent(java.lang.String type)
```

Parameters

`type` - the type of the event

Returns

Event object of the specified type

createTextNode(String)**Description**

Creates a `Text` node given the specified string.

Syntax

```
public org.w3c.dom.Text createTextNode(java.lang.String data)
```

Parameters

`data` - The data for the node.

Returns

The new `Text` object.

createTraversalEvent(String)

Description

Creates a Traversal Event object of specified type

Syntax

```
public org.w3c.dom.events.Event createTraversalEvent(java.lang.String type)
```

Parameters

`type` - the type of the event

Returns

Event object of the specified type

createTreeWalker(Node, int, NodeFilter, boolean)

Description

Creates a Node Iterator with specified `root`, `flag` which governs what type of nodes it should include in logical view, `filter` for filtering nodes, `flag` determining whether entity references and its descendants could be included

Syntax

```
public org.w3c.dom.traversal.TreeWalker createTreeWalker(org.w3c.dom.Node root,  
int whatToShow, org.w3c.dom.traversal.NodeFilter filter, boolean  
expandEntityReferences)
```

Parameters

`root` - root node of the iterator.

`whatToShow` - flag indicating what type of nodes will be included in the iterator/tree walker.

`filter` - `NodeFilter` to filter unwanted nodes from the iterator/tree walker

`expandEntityReference` - flag to indicate traversal of entity references

Returns

an object that implements TreeWalker interface, created on this document

Throws

`DOMException - NOT_SUPPORTED_ERR`: if the NodeIterator could not be created with specified root

expectedElements(Element)**Description**

Returns vector of element names that can be appended to the element.

Syntax

```
public java.util.Vector expectedElements(org.w3c.dom.Element e)
```

Parameters

e - Element

Returns

Vector of names

getColumnNumber()**Description**

Get column number debug information

Syntax

```
public int getColumnNumber()
```

Returns

column the column number

getDebugMode()**Description**

Get the debug flag

Syntax

```
public boolean getDebugMode()
```

Returns

boolean flag

getDoctype()

Description

The Document Type Declaration (DTD) associated with this document. For XML documents without a DTD, this returns `null`.

Syntax

```
public org.w3c.dom.DocumentType getDoctype()
```

Comments

Note that the DOM Level 1 specification does not support editing the DTD.

Returns

The associated DTD

See Also

`org.w3c.dom.DocumentType`

getDocumentElement()

Description

This is a convenience attribute that allows direct access to the child node that is the root element of the document.

Syntax

```
public org.w3c.dom.Element getDocumentElement()
```

Returns

The root element

getElementById(String)

Description

Returns the Element whose ID is given by elementId. If no such element exists, returns null. Behavior is not defined if more than one element has this ID.

Syntax

```
public org.w3c.dom.Element getElementById(java.lang.String elementId)
```

Parameters

String - elementId used to get the matching Id Element

Returns

The matching Id Element if one exists or null if none exists

Since

DOM 2

getElementsByTagName(String)

Description

Returns a NodeList of all the Elements with a given tag name in the order in which they would be encountered in a preorder traversal of the Document tree.

Syntax

```
public org.w3c.dom.NodeList getElementsByTagName(java.lang.String tagname)
```

Parameters

tagname - The name of the tag to match on. The special value "*" matches all tags.

Returns

A new NodeList object containing all the matched Elements.

getElementsByTagNameNS(String, String)

Description

Returns a NodeList of all the Elements with a given local name and namespace URI in the order in which they are encountered in a preorder traversal of the Document tree.

Syntax

```
public org.w3c.dom.NodeList getElementsByTagNameNS( java.lang.String  
namespaceURI, java.lang.String localName)
```

Parameters

namespaceURI - namespace of the elements requested.

localName - local name of the element requested.

Returns

nodelist of matching elements

Since

DOM 2

getEncoding()

Description

Retrieves the character encoding information.

Syntax

```
public final java.lang.String getEncoding()
```

Returns

the encoding information stored in the `<?xml ...?>` tag or the user-defined output encoding if it has been more recently set.

getIDHashtable()

Description

Get the ID element hashtable in the XML DOM Tree

Syntax

```
public java.util.Hashtable getIDHashtable()
```

Returns

Hashtable - associated with XMLDocument

getImplementation()

Description

The `DOMImplementation` object that handles this document. A DOM application may use objects from multiple implementations.

Syntax

```
public org.w3c.dom.DOMImplementation getImplementation()
```

Returns

The associated DOM implementation.

getLineNumber()

Description

Get line number debug information

Syntax

```
public int getLineNumber()
```

Returns

line the line number

getNodeTypes()

Description

Gets a code representing the type of the underlying object

Syntax

```
public short getNodeTypes()
```

Returns

type of the node

getOwnerDocument()

Description

The `Document` object associated with this node. Since this node is a `Document` this is `null`.

Syntax

```
public org.w3c.dom.Document getOwnerDocument()
```

Returns

null

getStandalone()

Description

Retrieves the standalone information.

Syntax

```
public final java.lang.String getStandalone()
```

Returns

the standalone attribute stored in the `<?xml ...?>` tag.

getSystemId()

Description

Get the system id of the entity contain this node

Syntax

```
public java.lang.String getSystemId()
```

Returns

sysid the system id

getText()

Description

Returns the non-marked-up text contained by this element.

Syntax

```
public java.lang.String getText()
```

Comments

For text elements, this is the raw data. For elements with child nodes, this method traverses the entire subtree and appends the text for each terminal text element, effectively stripping out the XML markup for the subtree. For example, if the XML document contains the following: William Shakespeare

`XMLDocument.getText` returns "William Shakespeare".

Returns

Non-marked-up text contained by the element.

getVersion()**Description**

Retrieves the version information.

Syntax

```
public final java.lang.String getVersion()
```

Returns

the version number stored in the `<?xml ...?>` tag.

importNode(Node, boolean)**Description**

Imports a node from another document to this document.

Syntax

```
public org.w3c.dom.Node importNode(org.w3c.dom.Node importedNode, boolean deep)
```

Comments

The returned node has no parent; (`parentNode` is null). The source node is not altered or removed from the original document; this method creates a new copy of the source node. For all nodes, importing a node creates a node object owned by the importing document, with attribute values identical to the source node's `nodeName` and `nodeType`, plus the attributes related to namespaces (`prefix`, `localName`, and `namespaceURI`). As in the `cloneNode` operation on a `Node`, the source node is not altered.

Parameters

`Node` - to be imported, boolean variable indicating whether the descendants of this node are to be imported

Returns

A copy of the `importedNode` which is attached to the current document but not a part of the document tree

Throws

`DOMException - NOT_SUPPORTED_ERR`: Raised if the type of the node being imported is not supported.

Since

DOM 2

insertBefore(Node, Node)

Description

Inserts the node `newChild` before the existing child node `refChild`.

Syntax

```
public org.w3c.dom.Node insertBefore(org.w3c.dom.Node newChild, org.w3c.dom.Node refChild)
```

Comments

If `refChild` is `null`, insert `newChild` at the end of the list of children. If `newChild` is a `DocumentFragment` object, all of its children are inserted, in the same order, before `refChild`. If the `newChild` is already in the tree, it is first removed.

Parameters

`newChild` - The node to insert.

`refChild` - The reference node, i.e., the node before which the new node must be inserted.

Returns

The node being inserted.

Throws

`DOMException - HIERARCHY_REQUEST_ERR`: Raised if this node is of a type that does not allow children of the type of the `newChild` node, or if the node to insert is one of this node's ancestors. `WRONG_DOCUMENT_ERR`: Raised if `newChild` was created from a different document than the one that created this node. `NO_MODIFICATION_ALLOWED_ERR`: Raised if this node is readonly. `NOT_FOUND_ERR`: Raised if `refChild` is not a child of this node.

print(OutputStream)**Description**

Writes the contents of this document to the given output stream.

Syntax

```
public void print(java.io.OutputStream out)
```

Parameters

`out` - `OutputStream` to write to

Throws

`IOException` - if an error occurs

print(OutputStream, String)**Description**

Writes the contents of this document to the given output stream.

Syntax

```
public void print(java.io.OutputStream out, java.lang.String enc)
```

Parameters

`out` - `OutputStream` to write to

`enc` - Encoding to use for the output

Throws

`IOException` - if an invalid encoding was specified or if any other error occurs

print(PrintDriver)

Description

Writes the contents of this document to the given output stream.

Syntax

```
public void print(PrintDriver pd)
```

Parameters

`pd` - `PrintDriver` used to write each node

Throws

`IOException` - if an error occurs

print(PrintWriter)

Description

Writes the contents of this document to the given output stream.

Syntax

```
public void print(java.io.PrintWriter out)
```

Parameters

`out` - `PrintWriter` to write to

Throws

`IOException` - if an error occurs

printExternalDTD(OutputStream)

Description

Writes the contents of this document to the given output stream.

Syntax

```
public void printExternalDTD(java.io.OutputStream out)
```

Parameters

`out` - `OutputStream` to write to

Throws

`IOException` - if an error occurs

printExternalDTD(OutputStream, String)**Description**

Writes the contents of the external DTD to the given output stream.

Syntax

```
public void printExternalDTD(java.io.OutputStream out, java.lang.String enc)
```

Parameters

`out` - `OutputStream` to write to

`enc` - Encoding to use for the output

Throws

`IOException` - if an invalid encoding was specified or if any other error occurs

printExternalDTD(PrintWriter)**Description**

Writes the contents of this document to the given output stream.

Syntax

```
public void printExternalDTD(java.io.PrintWriter out)
```

Parameters

`out` - `PrintWriter` to write to

Throws

`IOException` - if an error occurs

readExternal(ObjectInput)**Description**

This method reads the information written in the compressed stream by `writeExternal` method and restores the object correspondingly.

Syntax

```
public void readExternal(java.io.ObjectInput inArg)
```

Specified By

`java.io.Externalizable.readExternal(java.io.ObjectInput)` in interface `java.io.Externalizable`

Parameters

`inArg` - the `ObjectInput` stream used for reading the compressed stream

Throws

`IOException` - is thrown when there is an error in reading the input stream.
`ClassNotFoundException` - is thrown when the class is not found

removeChild(Node)

Description

removes the elem from this documents list of child nodes

Syntax

```
public org.w3c.dom.Node removeChild(org.w3c.dom.Node elem)
```

Parameters

`the` - node to be removed

Returns

the node after its removed from the document

Throws

`DOMException` - `NO_MODIFICATION_ALLOWED_ERR`: Raised if this document is readonly. `NOT_FOUND_ERR`: Raised if `oldChild` is not a child of this node.

replaceChild(Node, Node)

Description

Replaces the child node `oldChild` with `newChild` in the list of children, and returns the `oldChild` node.

Syntax

```
public org.w3c.dom.Node replaceChild(org.w3c.dom.Node newChild, org.w3c.dom.Node oldChild)
```

Comments

If the `newChild` is already in the tree, it is first removed. This is an override of the `XMLNode.removeChild` method

Parameters

`newChild` - The new node to put in the child list.

`oldChild` - The node being replaced in the list.

Returns

The node replaced.

Throws

`DOMException - HIERARCHY_REQUEST_ERR`: Raised if this node is of a type that does not allow children of the type of the `newChild` node. `WRONG_DOCUMENT_ERR`: Raised if `newChild` was created from a different document than this one. `NOT_FOUND_ERR`: Raised if `oldChild` is not a child of this node.

reportSAXEvents(ContentHandler)

Description

Report SAX Events from a DOM Tree

Syntax

```
public void reportSAXEvents(org.xml.sax.ContentHandler cntHandler)
```

Parameters

`ContentHandler` - `cntHandler`

Throws

`SAXException` - thrown by SAX Callback functions

setDoctype(String, String, String)

Description

Sets the doctype URI for the document

Syntax

```
public void setDoctype(java.lang.String rootname, java.lang.String sysid,  
java.lang.String pubid)
```

Parameters

`root` - The name of the root element

`sysid` - The system id of the doctype

`pubid` - The public id of the doctype (can be null)

setEncoding(String)

Description

Sets the character encoding for output. Eventually it sets the `ENCODING` stored in the `<?xml ...?>` tag, but not until the document is saved.

Syntax

```
public final void setEncoding(java.lang.String encoding)
```

Comments

You should not call this method until the Document has been loaded.

Parameters

`encoding` - The character encoding to set

setLocale(Locale)

Description

Sets the locale for error reporting

Syntax

```
public final void setLocale(java.util.Locale locale)
```

Parameters

`locale` - Locale for error reporting.

setNodeContext(NodeContext)

Description

Syntax

```
public void setNodeContext(oracle.xml.util.NodeContext nctx)
```

setParsedDoctype(String, String, String)

Description

Sets the doctype object by parsing sysid

Syntax

```
public void setParsedDoctype(java.lang.String rootname, java.lang.String sysid,
java.lang.String pubid)
```

Parameters

root - The name of the root element

sysid - The system id of the doctype

pubid - The public id of the doctype (can be null)

setStandalone(String)

Description

Sets the standalone information stored in the <?xml ...?> tag.

Syntax

```
public final void setStandalone(java.lang.String value)
```

Parameters

value - The attribute value ('yes' or 'no').

setVersion(String)

Description

Sets the version number stored in the <?xml ...?> tag.

Syntax

```
public final void setVersion(java.lang.String version)
```

Parameters

`version` - The version information to set.

validateElementContent(Element)

Description

Validates the content of a element node.

Syntax

```
public boolean validateElementContent(org.w3c.dom.Element e)
```

Parameters

`e` - Element to be validated

Returns

True if valid, else false

writeExternal(ObjectOutput)

Description

This method saves the state of the object by creating a binary compressed stream with information about this object.

Syntax

```
public void writeExternal(java.io.ObjectOutput outArg)
```

Specified By

`java.io.Externalizable.writeExternal(java.io.ObjectOutput)` in interface `java.io.Externalizable`

Parameters

`outArg` - The `ObjectOutput` stream used to write the serialized/ compressed stream.

Throws

`IOException` - is thrown when there is an exception while writing the serialized/compressed stream.

XMLDocumentFragment

Description

This class implements the DOM DocumentFragment interface.

Syntax

```
public class XMLDocumentFragment implements java.io.Serializable
```

```
oracle.xml.parser.v2.XMLDocumentFragment
```

Implemented Interfaces

```
java.io.Serializable
```

Comments

Extends XMLElement rather than XMLNode so it can be handled as an element. This is convenient in processing

See Also

DocumentFragment, NodeFactory, DOMParser.setNodeFactory(NodeFactory)

Constructors

XMLDocumentFragment()

Description

Creates an empty document fragment

Syntax

```
public XMLDocumentFragment()
```

Comments

Deprecated; use createElement(String) method of XMLDocument

Note that this constructor is used only during deserialization/decompression of this DOM node. In order to deserialize this node to construct the DOM node from the serialized/ compressed stream, it is required to create a handle of the object. For all normal XMLElement creation use createElement(String) of XMLDocument.

Methods

getAttributes()

Description

Syntax

```
public org.w3c.dom.NamedNodeMap getAttributes()
```

Returns

An empty NamedNodeMap.

getNodeTypes()

Description

Gets a code representing the type of the underlying object

Syntax

```
public short getNodeType()
```

Returns

type of the node

getParentNode()

Description

Gets the parent of this node

Syntax

```
public org.w3c.dom.Node getParentNode()
```

Returns

The parent of this node (always null)

XMLDOMException

Description

This class is used to throw DOM exceptions.

Syntax

```
public class XMLDOMException  
  
oracle.xml.parser.v2.XMLDOMException
```

Constructors

XMLDOMException(short)

Description

Constructs a `XMLDOMException` exception with a specified message and a code.

Syntax

```
public XMLDOMException(short code)
```

Parameters

`code` - Code indicated in DOM interface, uses default message
`err` - `XMLError` used for locale information

XMLDOMException(short, String)

Description

Constructs a `XMLDOMException` exception with a specified message and a code.

Syntax

```
public XMLDOMException(short code, java.lang.String s)
```

Parameters

`code` - Code indicated in DOM interface, uses default message

XMLDOMImplementation

Description

This class implements the DOMImplementation

Syntax

```
public class XMLDOMImplementation implements java.io.Serializable
```

```
oracle.xml.parser.v2.XMLDOMImplementation
```

Implemented Interfaces

```
java.io.Serializable
```

Constructors

XMLDOMImplementation()

Description

Syntax

```
public XMLDOMImplementation()
```

Methods

createDocument(String, String, DocumentType)

Description

Creates an XMLDocument object containing the specified DocumentType Node and a root element with the specified names and the empty DocumentType node.

Syntax

```
public org.w3c.dom.Document createDocument(java.lang.String namespaceURI,  
java.lang.String qualifiedName, org.w3c.dom.DocumentType doctype)
```

Parameters

namespaceURI - Namespace of the root element in the document.

qualifiedName - Qualified name of the root element in the document.

doctype - DocumentType (DTD) associated with the document.

Returns

The Document object created.

Throws

INVALID_CHARACTER_ERR: Raised if the specified qualified name contains an illegal character
NAMESPACE_ERR: Raised if the qualifiedName is malformed, if the qualifiedName has a prefix and the namespaceURI is null or an empty String, or if the qualifiedName has a prefix that is "xml" and namespaceURI is different from "http://www.w3.org/XML/1998/namespace"
WRONG_DOCUMENT_ERR: Raised if doctype has already been used with a different document or was created from a different implementation.

createDocumentType(String, String, String)

Description

Creates an empty DocumentType node with root element name, and system/public identifier.

Syntax

```
public org.w3c.dom.DocumentType createDocumentType(java.lang.String  
qualifiedName, java.lang.String publicId, java.lang.String systemId)
```

Parameters

qualifiedName - Qualified name of the root element.

systemid - System identifier of the DocumentType node.

publicid - Public identifier of the DocumentType node.

Returns

the DocumentType object created.

Throws

DOMException - INVALID_CHARACTER_ERR: Raised if the specified qualified name contains an illegal character
NAMESPACE_ERR: Raised if the qualifiedName is malformed.

hasFeature(String, String)

Description

Test if the DOM implementation implements a specific feature.

Syntax

```
public boolean hasFeature(java.lang.String feature, java.lang.String version)
```

Returns

TRUE if the feature is implemented, FALSE otherwise

setFeature(String)

Description

Sets a specified feature.

Syntax

```
public void setFeature(java.lang.String feature)
```

Parameters

feature - the DOM feature

Throws

`DOMException` - if the feature could not be set.

XMLElement

Syntax

```
public class XMLElement implements oracle.xml.parser.v2.NSName,  
oracle.xml.parser.v2.NSResolver, java.io.Externalizable
```

```
oracle.xml.parser.v2.XMLElement
```

Implemented Interfaces:

```
java.io.Externalizable, NSName, oracle.xml.util.NSName, NSResolver,  
java.io.Serializable
```

Description

This class implements the DOM `Element` Interface

Constructors

XMLElement()

Description

Default constructor.

Syntax

```
public XMLElement()
```

Comments

Note that this constructor is used only during deserialization/decompression of this DOM node. In order to deserialize this node to construct the DOM node from the serialized/ compressed stream, it is required to create a handle of the object. For all normal XMLElement creation use `createElement(String)` of `XMLDocument`.

Methods

cloneNode(boolean)

Description

Returns a duplicate of this node, i.e., serves as a generic copy constructor for nodes.

Syntax

```
public org.w3c.dom.Node cloneNode(boolean deep)
```

Comments

The duplicate node has no parent (`parentNode` returns `null`). Cloning an `Element` copies all attributes and their values, including those generated by the XML processor to represent defaulted attributes, but this method does not copy any text it contains unless it is a deep clone, since the text is contained in a child `Text` node. Cloning any other type of node simply returns a copy of this node.

Parameters

`deep` - If `true`, recursively clone the subtree under the specified node; if `false`, clone only the node itself (and its attributes, if it is an `Element`).

Returns

The duplicate node.

getAttribute(String)

Description

Retrieves an attribute value by name.

Syntax

```
public java.lang.String getAttribute(java.lang.String name)
```

Parameters

`name` - The name of the attribute to retrieve.

Returns

The `Attr` value as a string, or the empty string if that attribute does not have a specified or default value.

getAttributeNode(String)

Description

Retrieves an `Attr` node by name.

Syntax

```
public org.w3c.dom.Attr getAttributeNode(java.lang.String name)
```

Parameters

`name` - The name of the attribute to retrieve.

Returns

The `Attr` node with the specified attribute name or `null` if there is no such attribute.

getAttributeNodeNS(String, String)**Description****Syntax**

```
public org.w3c.dom.Attr getAttributeNodeNS(java.lang.String namespaceURI,  
java.lang.String localName)
```

Parameters

`String` - `namespaceURI`, `String` `localName` of the attribute to be retrieved

Returns

Attribute with the given `namespaceURI` and `localName` if it exists, else `null`.

Since

DOM 2 Retrieves an `Attr` node by local name and namespace URI.

getAttributeNS(String, String)**Description****Syntax**

```
public java.lang.String getAttributeNS(java.lang.String namespaceURI,  
java.lang.String localName)
```

Parameters

`namespaceURI` - namespace of the attribute requested.

`localName` - local name of the attribute requested.

Returns

the value of the attribute with the above mentioned namespace URI and localName if it exists ,else null.

Since

DOM 2 Retrieves an attribute value by local name and namespace URI.

getAttributes()

Description

A `NamedNodeMap` containing the attributes of this node (if it is an `Element`) or `null` otherwise.

Syntax

```
public org.w3c.dom.NamedNodeMap getAttributes()
```

Returns

The list of attributes of this element

getChildrenByTagName(String)

Description

Returns a `NodeList` of all immediate children with a given tag name,

Syntax

```
public org.w3c.dom.NodeList getChildrenByTagName(java.lang.String name)
```

Parameters

name - The name of the tag to match on.

Returns

A list of matching children

getChildrenByTagName(String, String)

Description

Returns a `NodeList` of all immediate children with a given tag name and namespace

Syntax

```
public org.w3c.dom.NodeList getChildrenByTagName(java.lang.String name,  
java.lang.String ns)
```

Parameters

name - The name of the tag to match on. (should be local name)

ns - The name space

Returns

A list of matching children

getElementsByTagName(String)**Description**

Returns a `NodeList` of all the `Elements` with a given tag name in the order in which they would be encountered in a preorder traversal of the `Document` tree.

Syntax

```
public org.w3c.dom.NodeList getElementsByTagName(java.lang.String tagname)
```

Parameters

tagname - The name of the tag to match on. The special value "*" matches all tags.

Returns

A new `NodeList` object containing all the matched `Elements`.

getElementsByTagNameNS(String, String)**Description**

Returns a `NodeList` of all the descendant `Elements` with a given local name and namespace URI in the order in which they are encountered in a preorder traversal of this `Element` tree.

Syntax

```
public org.w3c.dom.NodeList getElementsByTagNameNS(java.lang.String  
namespaceURI, java.lang.String localName)
```

Parameters

`namespaceURI` - the namespace of the element

`localName` - the local name of the element

Since

DOM 2

getExpandedName()

Description

Get the fully resolved name for this element.

Syntax

```
public java.lang.String getExpandedName()
```

Specified By

`oracle.xml.util.NSName.getExpandedName()` in interface `oracle.xml.util.NSName`

Returns

the fully resolved name

getFirstAttribute()

Description

Retrieves the first `Attr`.

Syntax

```
public XMLNode getFirstAttribute()
```

Returns

The first `Attr` node `null` if there is no attribute.

getLocalName()

Description

Gets the local `Name` for this element.

Syntax

```
public java.lang.String getLocalName()
```

Specified By

oracle.xml.util.NSName.getLocalName() in interface oracle.xml.util.NSName

Returns

the local Name

getNamespaceURI()**Description**

Gets the name space URI of this element

Syntax

```
public java.lang.String getNamespaceURI()
```

Returns

the namespace URI of this element

Since

DOM 2

getNodeTypes()**Description**

Gets a code representing the type of the underlying object

Syntax

```
public short getNodeTypes()
```

Returns

type of the node

getPrefix()**Description**

Get the namespace prefix for this element.

Syntax

```
public java.lang.String getPrefix()
```

Specified By

oracle.xml.util.NSName.getPrefix() in interface oracle.xml.util.NSName

Returns

the namespace prefix

getQualifiedName()

Description

Get the qualified name for this element.

Syntax

```
public java.lang.String getQualifiedName()
```

Specified By

oracle.xml.util.NSName.getQualifiedName() in interface oracle.xml.util.NSName

Returns

the qualified name

getTagName()

Description

Gets the name of the element.

Syntax

```
public java.lang.String getTagName()
```

Comments

For example, in: `<elementExample id="demo"> ... </elementExample>`, `tagName` has the value `"elementExample"`. Note that this is case-preserving in XML, as are all of the operations of the DOM. The HTML DOM returns the `tagName` of an HTML element in the canonical uppercase form, regardless of the case in the source HTML document.

Returns

The element name

hasAttribute(String)**Description**

Returns true when an attribute with a given name is specified on this element or has a default value, false otherwise.

Syntax

```
public boolean hasAttribute(java.lang.String name)
```

Parameters

`String` - name of the attribute whose presence is checked

Returns

true if the attribute with the specified name is present, else null

Since

DOM 2

hasAttributeNS(String, String)**Description**

Returns true when an attribute with a given local name and namespace URI is specified on this element or has a default value, false otherwise

Syntax

```
public boolean hasAttributeNS(java.lang.String namespaceURI, java.lang.String  
localName)
```

Parameters

`namespaceURI` - namespace of the attribute whose presence is checked

`localName` - local name of the attribute whose presence is checked

Returns

true when an attribute with a given local name and namespace URI is specified on this element or has a default value, false otherwise

Since

DOM 2

hasAttributes()

Description

Returns whether this node (if it is an element) has any attributes.

Syntax

```
public boolean hasAttributes()
```

Returns

true if this node has any attributes, false otherwise.

Since

DOM Level 2

normalize()

Description

Puts all `Text` nodes in the full depth of the sub-tree underneath this `Element` into a "normal" form where only markup (e.g., tags, comments, processing instructions, CDATA sections, and entity references) separates `Text` nodes, i.e., there are no adjacent `Text` nodes. This can be used to ensure that the DOM view of a document is the same as if it were saved and re-loaded, and is useful when operations (such as `XPointer` lookups) that depend on a particular document tree structure are to be used.

Syntax

```
public void normalize()
```

Comments

Deprecated as of DOM 2

See Also

```
XMLNode.normalize()
```

readExternal(ObjectInput)

Description

This method restores the information written by writeExternal by reading the input stream and regenerating the objects as per the information of the input stream.

Syntax

```
public void readExternal(java.io.ObjectInput inArg)
```

Specified By

java.io.Externalizable.readExternal(java.io.ObjectInput) in interface
java.io.Externalizable

Parameters

inArg - is the ObjectInput stream used to read the compressed stream.

Throws

IOException - is thrown when there is an exception reading the compressed stream.

ClassNotFoundException - is thrown when the class is not found

removeAttribute(String)

Description

Removes an attribute by name. If the removed attribute has a default value it is immediately replaced.

Syntax

```
public void removeAttribute(java.lang.String name)
```

Parameters

name - The name of the attribute to remove.

Throws

DOMException - NO_MODIFICATION_ALLOWED_ERR: Raised if this node is readonly.

removeAttributeNode(Attr)

Description

Removes the specified attribute.

Syntax

```
public org.w3c.dom.Attr removeAttributeNode(org.w3c.dom.Attr oldAttr)
```

Parameters

`oldAttr` - The `Attr` node to remove from the attribute list. If the removed `Attr` has a default value it is immediately replaced.

Returns

The `Attr` node that was removed.

Throws

`DOMException - NO_MODIFICATION_ALLOWED_ERR`: Raised if this node is `readonly`. `NOT_FOUND_ERR`: Raised if `oldAttr` is not an attribute of the element.

removeAttributeNS(String, String)

Description

Removes an attribute by local name and namespace URI.

Syntax

```
public void removeAttributeNS(java.lang.String namespaceURI, java.lang.String localName)
```

Parameters

`namespaceURI` - namespace of the attribute to be removed

`localName` - local name of the attribute to be removed

Throws

`DOMException - NO_MODIFICATIONS_ALLOWED_ERR` : if this element is `readonly`

Since

DOM 2

reportSAXEvents(ContentHandler)

Description

Report SAX Events from a DOM Tree

Syntax

```
public void reportSAXEvents(org.xml.sax.ContentHandler cntHandler)
```

Parameters

ContentHandler - cntHandler

Throws

SAXException - thrown by SAX Callback functions

resolveNamespacePrefix(String)

Description

Given a namespace prefix, find the namespace definition in scope in this element.

Syntax

```
public java.lang.String resolveNamespacePrefix(java.lang.String prefix)
```

Specified By

NSResolver.resolveNamespacePrefix(String) in interface NSResolver

Parameters

prefix - Namespace prefix to be resolved if the prefix == #default, return the default namespace

Returns

the resolved Namespace (null, if prefix could not be resolved)

setAttribute(String, String)

Description

Adds a new attribute. If an attribute with that name is already present in the element, its value is changed to be that of the value parameter.

Syntax

```
public void setAttribute(java.lang.String name, java.lang.String value)
```

Comments

This value is a simple string, it is not parsed as it is being set. So any markup (such as syntax to be recognized as an entity reference) is treated as literal text, and needs to be appropriately escaped by the implementation when it is written out. In order to assign an attribute value that contains entity references, the user must create an `Attr` node plus any `Text` and `EntityReference` nodes, build the appropriate subtree, and use `setAttributeNode` to assign it as the value of an attribute. This method is namespace unaware and hence wont result in update of namespace table if a new attr is added thru this method

Parameters

`name` - The name of the attribute to create or alter.

`value` - Value to set in string form.

Throws

`DOMException` - `INVALID_CHARACTER_ERR`: Raised if the specified name contains an invalid character. `NO_MODIFICATION_ALLOWED_ERR`: Raised if this node is readonly.

setAttributeNode(Attr)

Description

Adds a new attribute. If an attribute with that name is already present in the element, it is replaced by the new one.

Syntax

```
public org.w3c.dom.Attr setAttributeNode(org.w3c.dom.Attr newAttr)
```

Parameters

`newAttr` - The `Attr` node to add to the attribute list.

Returns

If the `newAttr` attribute replaces an existing attribute with the same name, the previously existing `Attr` node is returned, otherwise `null` is returned.

Throws

`DOMException - WRONG_DOCUMENT_ERR`: Raised if `newAttr` was created from a different document than the one that created the element. `NO_MODIFICATION_ALLOWED_ERR`: Raised if this node is readonly. `INUSE_ATTRIBUTE_ERR`: Raised if `newAttr` is already an attribute of another `Element` object. The DOM user must explicitly clone `Attr` nodes to re-use them in other elements.

setAttributeNodeNS(Attr)**Description**

Adds a new attribute. If an attribute with that local name and that namespace URI is already present in the element, it is replaced by the new one.

Syntax

```
public org.w3c.dom.Attr setAttributeNodeNS(org.w3c.dom.Attr newAttr)
```

Parameters

`Attribute` - node to be added

Returns

the `Attribute Node` added

Throws

`DOMException - WRONG_DOCUMENT_ERR` : raised if the `newAttr` was created from a document different from the one that created the document `NO_MODIFICATIONS_ALLOWED_ERR` :Raised if this element is readonly `INUSE_ATTRIBUTE_ERR` : Raised if `newAttr` is already an attribute of another `Element` object

Since

DOM 2

setAttributeNS(String, String, String)**Description**

Adds a new attribute. If an attribute with the same local name and namespace URI is already present on the element, its prefix is changed to be the prefix part of the `qualifiedName`, and its value is changed to be the `value` parameter. This value is a

simple string; it is not parsed as it is being set. So any markup (such as syntax to be recognized as an entity reference) is treated as literal text, and needs to be appropriately escaped by the implementation when it is written out.

Syntax

```
public void setAttributeNS(java.lang.String namespaceURI, java.lang.String
qualifiedName, java.lang.String value)
```

Parameters

namespaceURI - namespace of the attribute to be added

localName - local name of the attribute to be added

value - value of the attribute to be added

Throws

DOMException - INVALID_CHARACTER_ERR: Raised if the specified qualified name contains illegal Characters NAMESPACE_ERR : Raised if the qualified name is malformed ,if the qualified name has a prefix and the namespace URI is null or an empty string,or if the qualifiedName is "xmlns" and namespace URI is different from "http://www.w3.org/2000/xmlns/" ,or if qualifiedName has a prefix that is "xml" and the namespaceURI is different from http://www.w3.org/XML/1998/namespace NO_MODIFICATION_ALLOWED_ERR: Raised if this node is readonly

Since

DOM 2

validateContent(DTD)

Description

Validates the content of a element node.

Syntax

```
public boolean validateContent(DTD dtd)
```

Parameters

dtd - The DTD object used to validate the element.

schema - The XMLSchema object used to validate the element.

Returns

True if valid, else false

validateContent(XMLSchema)**Description**

Validates the content of the element against given XML Schema param schema - schema used to validate

Syntax

```
public boolean validateContent(oracle.xml.parser.schema.XMLSchema schema)
```

Returns

True if valid, else false

validateContent(XMLSchema, String)**Description**

Validates the content of the element against given XML Schema in the given mode.

Syntax

```
public boolean validateContent(oracle.xml.parser.schema.XMLSchema schema,  
java.lang.String mode)
```

Parameters

schema - - schema used to validate the content

mode - - the validation mode

Returns

True if valid, else false

writeExternal(ObjectOutput)**Description**

This method saves the state of the object by creating a binary compressed stream with information about this object.

Syntax

```
public void writeExternal(java.io.ObjectOutput outArg)
```

Parameters

`outArg`: The `ObjectOutput` stream used to write the serialized/compressed stream.

Specified By

`java.io.Externalizable.writeExternal(java.io.ObjectOutput)` in interface `java.io.Externalizable`

`cxmlContext` - The context of the output compressed stream

XMLEntity

Description

This class implements the DOM `Entity` interface and represents an XML internal or external entity as defined in the XML Document Type Definition (DTD).

Syntax

```
public class XMLEntity implements java.io.Externalizable
```

```
oracle.xml.parser.v2.XMLEntity
```

Implemented Interfaces

```
java.io.Externalizable, java.io.Serializable
```

Constructors

XMLEntity()

Description

Default constructor.

Syntax

```
public XMLEntity()
```

Comments

Note that this constructor is used only during deserialization/decompression of this DOM node. In order to deserialize this node to construct the DOM node from the serialized/ compressed stream, it is required to create a handle of the object.

Methods

cloneNode(boolean)

Description

Returns a duplicate of this node, i.e., serves as a generic copy constructor for nodes.

Syntax

```
public org.w3c.dom.Node cloneNode(boolean deep)
```

Comments

The duplicate node has no parent (`parentNode` returns `null`). Cloning an `Element` copies all attributes and their values, including those generated by the XML processor to represent defaulted attributes, but this method does not copy any text it contains unless it is a deep clone, since the text is contained in a child `Text` node. Cloning any other type of node simply returns a copy of this node.

Parameters

`deep` - If `true`, recursively clone the subtree under the specified node; if `false`, clone only the node itself (and its attributes, if it is an `Element`).

Returns

The duplicate node.

getNodeTypes()**Description**

Gets a code representing the type of the underlying object

Syntax

```
public short getNodeTypes()
```

Returns

type of the node

getNodeValue()**Description**

Gets the value of this node, depending on its type

Syntax

```
public java.lang.String getNodeValue()
```

Returns

Value of this node

Throws

`DOMException - NO_MODIFICATION_ALLOWED_ERR`: Raised when the node is readonly. `DOMSTRING_SIZE_ERR`: Raised when it would return more characters than fit in a `DOMString` variable on the implementation platform.

getNotationName()**Description**

For unparsed entities, gets the name of the notation for the entity. For parsed entities, this is `null`.

Syntax

```
public java.lang.String getNotationName()
```

Returns

The notation name

getPublicId()**Description**

Gets the public identifier associated with the entity, if specified. If the public identifier was not specified, this is `null`.

Syntax

```
public java.lang.String getPublicId()
```

Returns

The public identifier

getSystemId()**Description**

Gets the system identifier associated with the entity, if specified. If the system identifier was not specified, this is `null`.

Syntax

```
public java.lang.String getSystemId()
```

Returns

The system identifier

readExternal(ObjectInput)

Description

This method reads the information written in the compressed stream by writeExternal method and restores the object correspondingly.

Syntax

```
public void readExternal(java.io.ObjectInput inArg)
```

Specified By

java.io.Externalizable.readExternal(java.io.ObjectInput) in interface java.io.Externalizable

Parameters

inArg - the ObjectInput stream used for reading the compressed stream.

Throws

IOException - is thrown when there is an error in reading the input stream.

ClassNotFoundException - is thrown when the class is not found.

setNodeValue(String)

Description

Sets the value of entity.

Syntax

```
public void setNodeValue(java.lang.String arg)
```

Parameters

arg - The new value of the entity.

writeExternal(ObjectOutput)

Description

This method saves the state of the object by creating a binary compressed stream with information about this object.

Syntax

```
public void writeExternal(java.io.ObjectOutput outArg)
```

Specified By

`java.io.Externalizable.writeExternal(java.io.ObjectOutput)` in interface `java.io.Externalizable`

Parameters

`outArg` - The `ObjectOutput` stream used to write the serialized/ compressed stream.

Throws

`IOException` - is thrown when there is an exception while writing the serialized/compressed stream.

XMLEntityReference

Description

This class implements DOM `EntityReference` interface.

Syntax

```
public class XMLEntityReference implements java.lang.Cloneable,  
java.io.Externalizable
```

```
oracle.xml.parser.v2.XMLEntityReference
```

Implemented Interfaces

`java.lang.Cloneable`, `java.io.Externalizable`, `java.io.Serializable`

Constructors

XMLEntityReference()

Description

Default constructor.

Syntax

```
public XMLEntityReference()
```

Comments

Note that this constructor is used only during deserialization/decompression of this DOM node. In order to deserialize this node to construct the DOM node from the serialized/ compressed stream, it is required to create a handle of the object.

Methods

getNodeType()

Description

Gets a code representing the type of the underlying object

Syntax

```
public short getNodeType()
```

Returns

type of the node

readExternal(ObjectInput)**Description**

This method reads the information written in the compressed stream by writeExternal method and restores the object correspondingly.

Syntax

```
public void readExternal(java.io.ObjectInput inArg)
```

Specified By

java.io.Externalizable.readExternal(java.io.ObjectInput) in interface java.io.Externalizable

Parameters

inArg - The ObjectInput stream used for reading the compressed stream

Throws

IOException - is thrown when there is an error in reading the input stream.

ClassNotFoundException - is thrown when the class is not found

writeExternal(ObjectOutput)**Description**

This method saves the state of the object by creating a binary compressed stream with information about this object.

Syntax

```
public void writeExternal(java.io.ObjectOutput outArg)
```

Specified By

java.io.Externalizable.writeExternal(java.io.ObjectOutput) in interface java.io.Externalizable

Parameters

`outArg` - is the `ObjectOutput` stream used to write the compressed stream.

Throws

`IOException` - is thrown when there is an exception while writing the compressed stream.

XMLERror

Description

This class hold error message and the line number where it occurred

Syntax

```
public class XMLERror extends oracle.xml.util.XMLERror
```

```
oracle.xml.util.XMLERror
|
+--oracle.xml.parser.v2.XMLERror
```

Constructors

XMLERror()

Description

Default constructor

Syntax

```
public XMLERror()
```

Methods

error(int, int, String, String, String, int, int, boolean)

Description

Adds a new error to the vector

Syntax

```
public void error(int line, int col, java.lang.String pubId, java.lang.String
sysId, java.lang.String mesg, int id, int type, boolean stop)
```

Parameters

line - the line number where error occurs

col - the column number where the error occurs

`pubId` - the public Identifier
`sysId` - the system identifier
`mesg` - the error message
`id` - the error id
`type` - the error type
`stop` - boolean to indicate if the processing needs to be stopped.

Throws

`throws` - `ParseException` in case of a fatal error

flushErrorListener(DOMLocator)

Description

Flush all the error to the error listener

Syntax

```
public void flushErrorListener(oracle.xml.parser.v2.DOMLocator locator)
```

Parameters

`locator` - the DOM Locator object

flushErrorListenerStream(DOMLocator)

Description

Flush all the error to the error listener

Syntax

```
public void flushErrorListenerStream(oracle.xml.parser.v2.DOMLocator locator)
```

Parameters

`locator` - the DOM Locator object

flushErrors()

Description

Flush all the error to the output stream output stream defaults or to error handler

Syntax

```
public void flushErrors()
```

Throws

throws - ParseException in case of a fatal error

getErrorHandler()**Description**

Return register error handler

Syntax

```
public org.xml.sax.ErrorHandler getErrorHandler()
```

Returns

ErrorHandler

getErrorListener()**Description**

Return register error listener

Syntax

```
public javax.xml.transform.ErrorListener getErrorListener()
```

Returns

ErrorListener

setErrorHandler(ErrorHandler)**Description**

Register error handler

Syntax

```
public void setErrorHandler(org.xml.sax.ErrorHandler err)
```

Parameters

err -- ErrorHandler

setErrorListener(ErrorListener)

Description

Register error listener

Syntax

```
public void setErrorListener(javax.xml.transform.ErrorListener el)
```

Parameters

`err` -- ErrorListener

XMLNode

Description

Implements the DOM `Node` interface and serves as the primary datatype for the entire Document Object Model. It represents a single node in the document tree.

Syntax

```
public abstract class XMLNode implements java.lang.Cloneable,
    java.io.Externalizable
```

```
oracle.xml.parser.v2.XMLNode
```

Direct Known Subclasses

```
XMLNSNode
```

Implemented Interfaces

```
java.lang.Cloneable, java.io.Externalizable, java.io.Serializable
```

Comments

The attributes `nodeName`, `nodeValue` and `attributes` are included as a mechanism to get at node information without casting down to the specific derived instance. In cases where there is no obvious mapping of these attributes for a specific `nodeType` (e.g., `nodeValue` for an `Element` or `attributes` for a `Comment`), this returns null. Note that the derived classes may contain additional and more convenient mechanisms to get and set the relevant information. This DOM Nodes extending `XMLNode` instead of `XMLNSNode` have fixed `Nodename` defined by DOM specification. Also only node that cannot have child nodes extend this class.

Fields of XMLNode

Table 11–17 *Fields of XMLNode*

Field	Syntax	Description
ATTRDECL	public static final short ATTRDECL	A attribute declaration node
Auto_Events	public static final java.lang.String Auto_Events	Flag to set Auto EVENTS

Table 11–17 Fields of XMLNode

Field	Syntax	Description
capturing	public static final java.lang.String capturing	
DOMAttrModified	public static final java.lang.String DOMAttrModified	
DOMCharacterDataModified	public static final java.lang.String DOMCharacterDataModified	
DOMNodeInserted	public static final java.lang.String DOMNodeInserted	
DOMNodeInsertedIntoDocument	public static final java.lang.String DOMNodeInsertedIntoDocument	
DOMNodeRemoved	public static final java.lang.String DOMNodeRemoved	
DOMNodeRemovedFromDocument	public static final java.lang.String DOMNodeRemovedFromDocument	
DOMSubtreeModified	public static final java.lang.String DOMSubtreeModified	
ELEMENTDECL	public static final short ELEMENTDECL	An element declaration node
noncapturing	public static final java.lang.String noncapturing	
RANGE_DELETE_EVENT	public static final java.lang.String RANGE_DELETE_EVENT	Flag to delete range event
RANGE_DELETETEXT_EVENT	public static final java.lang.String RANGE_DELETETEXT_EVENT	Flag to set range delete text event
RANGE_INSERT_EVENT	public static final java.lang.String RANGE_INSERT_EVENT	Flag to set range event
RANGE_INSERTTEXT_EVENT	public static final java.lang.String RANGE_INSERTTEXT_EVENT	Flag to set range insert text event
RANGE_REPLACE_EVENT	public static final java.lang.String RANGE_REPLACE_EVENT	Flag to replace range event
RANGE_SETTEXT_EVENT	public static final java.lang.String RANGE_SETTEXT_EVENT	Flag to set range text event
TRAVERSAL_DELETE_EVENT	public static final java.lang.String TRAVERSAL_DELETE_EVENT	Flag to set traversal delete event

Table 11–17 Fields of XMLNode

Field	Syntax	Description
TRaversal_REPLACE_EVENT	public static final java.lang.String Traversal_REPLACE_EVENT	Flag to set traversal replace event
XMLDECL_NODE	public static final short XMLDECL_ NODE	A attribute declaration node

Methods

XMLNode()

Description

Constructs a new XMLNode having the given name and type

Syntax

```
protected XMLNode()
```

Parameters

tag - Name of the node

addEventListener(String, EventListener, boolean)

Description

This method allows the registration of event listeners on the event target (node).

Syntax

```
public void addEventListener(java.lang.String type,  
org.w3c.dom.events.EventListener listener, boolean useCapture)  
DOMEvents
```

Parameters

type - Type of event for which the listener is registered

listener - The listener object

useCapture - flag to indicate if the listener wants to initiate capture

appendChild(Node)

Description

Adds the node `newChild` to the end of the list of children of this node. If the `newChild` is already in the tree, it is first removed.

Syntax

```
public org.w3c.dom.Node appendChild(org.w3c.dom.Node newChild)
```

Parameters

`newChild` - The node to add. If it is a `DocumentFragment` object, the entire contents of the document fragment are moved into the child list of this node

Returns

The node added.

Throws

`DOMException - HIERARCHY_REQUEST_ERR`: Raised if this node is of a type that does not allow children of the type of the `newChild` node, or if the node to append is one of this node's ancestors. `WRONG_DOCUMENT_ERR`: Raised if `newChild` was created from a different document than the one that created this node. `NO_MODIFICATION_ALLOWED_ERR`: Raised if this node is readonly.

cloneNode(boolean)

Description

Returns a duplicate of this node, i.e., serves as a generic copy constructor for nodes.

Syntax

```
public org.w3c.dom.Node cloneNode(boolean deep)
```

Comments

The duplicate node has no parent (`parentNode` returns `null`). Cloning an `Element` copies all attributes and their values, including those generated by the XML processor to represent defaulted attributes, but this method does not copy any text it contains unless it is a deep clone, since the text is contained in a child `Text` node. Cloning any other type of node simply returns a copy of this node.

Parameters

`deep` - If `true`, recursively clone the subtree under the specified node; if `false`, clone only the node itself (and its attributes, if it is an `Element`).

Returns

The duplicate node

dispatchEvent(Event)**Description**

This method allows the dispatch of events into the implementations event model

Syntax

```
public boolean dispatchEvent(org.w3c.dom.events.Event evt)
```

Returns

boolean value that indicates whether `preventDefault` or `stopPropagation` was called

Throws

`UNSPECIFIED_EVENT_TYPE` : - Raised if the Event's type was not specified by initializing the event before `dispatchEvent` was called.

getAttributes()**Description**

Gets a `NamedNodeMap` containing the attributes of this node (if it is an `Element`) or `null` otherwise.

Syntax

```
public org.w3c.dom.NamedNodeMap getAttributes()
```

Returns

the attributes of this node

getChildNodes()

Description

Gets a `NodeList` that contains all children of this node. If there are no children, this is a `NodeList` containing no nodes.

Syntax

```
public org.w3c.dom.NodeList getChildNodes()
```

Comments

The content of the returned `NodeList` is "live" in the sense that, for instance, changes to the children of the node object that it was created from are immediately reflected in the nodes returned by the `NodeList` accessors; it is not a static snapshot of the content of the node. This is true for every `NodeList`, including the ones returned by the `getElementsByTagName` method.

Returns

The children of this node

getColumnNumber()

Description

Get column number debug information

Syntax

```
public int getColumnNumber()
```

Returns

column the column number

getDebugMode()

Description

Get debug information mode

Syntax

```
public boolean getDebugMode()
```


Returns

flag debugging mode

getFirstChild()**Description**

Gets the first child of this node. If there is no such node, this returns `null`.

Syntax

```
public org.w3c.dom.Node getFirstChild()
```

Returns

The first child of this node

getLastChild()**Description**

Gets the last child of this node. If there is no such node, this returns `null`.

Syntax

```
public org.w3c.dom.Node getLastChild()
```

Returns

The last child of this node

getLineNumber()**Description**

Get line number debug information

Syntax

```
public int getLineNumber()
```

Returns

line the line number

getLocalName()

Description

Gets the Local Name of this node overridden by node types for which namespace is meaningful.

Syntax

```
public java.lang.String getLocalName()
```

Since

DOM 2

getNamespaceURI()

Description

Gets the namespace URI of this node. overridden by node types for which namespace is meaningful.

Syntax

```
public java.lang.String getNamespaceURI()
```

Returns

the namespace

Since

DOM 2

getNextSibling()

Description

Gets The node immediately following this node. If there is no such node, this returns null.

Syntax

```
public org.w3c.dom.Node getNextSibling()
```

Returns

The next node

getNodeName()

Description

Gets the name of the node

Syntax

```
public java.lang.String getNodeName()
```

Returns

name of the node

getNodeType()

Description

Gets the type of the node

Syntax

```
public short getNodeType()
```

Returns

type of the node

getNodeValue()

Description

Gets the value of this node, depending on its type

Syntax

```
public java.lang.String getNodeValue()
```

Returns

Value of this node

Throws

`DOMException - NO_MODIFICATION_ALLOWED_ERR`: Raised when the node is readonly. `DOMSTRING_SIZE_ERR`: Raised when it would return more characters than fit in a `DOMString` variable on the implementation platform.

getOwnerDocument()

Description

Gets the `Document` object associated with this node. This is also the `Document` object used to create new nodes. When this node is a `Document` this is `null`.

Syntax

```
public org.w3c.dom.Document getOwnerDocument()
```

Returns

The document associated with this node

getParentNode()

Description

Gets the parent of this node. All nodes, except `Document`, `DocumentFragment`, and `Attr` may have a parent. However, if a node has just been created and not yet added to the tree, or if it has been removed from the tree, this is `null`.

Syntax

```
public org.w3c.dom.Node getParentNode()
```

Returns

The parent of this node

getPrefix()

Description

Gets the prefix of this node overridden by node types for which namespace is meaningful.

Syntax

```
public java.lang.String getPrefix()
```

Returns

the prefix

Since

DOM 2

getPreviousSibling()

Description

Gets the node immediately preceding this node. If there is no such node, this returns null.

Syntax

```
public org.w3c.dom.Node getPreviousSibling()
```

Returns

the previous node

getProperty(String)

Description

Get a property of the node

Syntax

```
public java.lang.Object getProperty(java.lang.String propName)
```

Parameters

propName - - Name of the property

Returns

Object propValue - Value of the property

getSystemId()

Description

Get the system id of the entity contain this node

Syntax

```
public java.lang.String getSystemId()
```

Returns

sysid the system id

getText()

Description

Returns the non-marked-up text contained by this element. For text elements, this is the raw data. For elements with child nodes, this method traverses the entire subtree and appends the text for each terminal text element, effectively stripping out the XML markup for the subtree.

Syntax

```
public java.lang.String getText()
```

Comments

For example, if the XML document contains the following: William Shakespeare
`XMLDocument.getText` returns "William Shakespeare".

Returns

Non-marked-up text contained by the element

hasAttributes()

Description

Returns whether this node (if it is an element) has any attributes.

Syntax

```
public boolean hasAttributes()
```

Returns

`true` if this node has any attributes, `false` otherwise.

Since

DOM Level 2

hasChildNodes()

Description

This is a convenience method to allow easy determination of whether a node has any children.

Syntax

```
public boolean hasChildNodes()
```

Returns

true if the node has any children, false if the node has no children.

insertBefore(Node, Node)**Description**

Inserts the node `newChild` before the existing child node `refChild`.

Syntax

```
public org.w3c.dom.Node insertBefore(org.w3c.dom.Node newChild, org.w3c.dom.Node  
refChild)
```

Comments

If `refChild` is null, insert `newChild` at the end of the list of children. If `newChild` is a `DocumentFragment` object, all of its children are inserted, in the same order, before `refChild`. If the `newChild` is already in the tree, it is first removed.

Parameters

`newChild` - The node to insert.

`refChild` - The reference node, i.e., the node before which the new node must be inserted.

Returns

The node being inserted.

Throws

`DOMException - HIERARCHY_REQUEST_ERR`: Raised if this node is of a type that does not allow children of the type of the `newChild` node, or if the node to insert is one of this node's ancestors. `WRONG_DOCUMENT_ERR`: Raised if `newChild` was created from a different document than the one that created this node. `NO_MODIFICATION_ALLOWED_ERR`: Raised if this node is readonly. `NOT_FOUND_ERR`: Raised if `refChild` is not a child of this node.

isNodeFlag(int)

Description

Returns the node flag information

Syntax

```
public boolean isNodeFlag(int flag)
```

Returns

true if the flag is set

isSupported(String, String)

Description

Tests whether the DOM implementation implements a specific feature and that feature is supported by this node.

Syntax

```
public boolean isSupported(java.lang.String feature, java.lang.String version)
```

Parameters

String - feature ,*String* version

Returns

true,if the feature is supported else false

print(OutputStream)

Description

Writes the contents of this node to the given output stream.

Syntax

```
public void print(java.io.OutputStream out)
```

Parameters

out - *OutputStream* to write to

Throws

IOException - if an error occurs

print(OutputStream, String)

Description

Writes the contents of this node to the given output stream.

Syntax

```
public void print(java.io.OutputStream out, java.lang.String enc)
```

Parameters

out - OutputStream to write to

enc - Encoding to use for the output

Throws

IOException - if an invalid encoding was specified or if any other error occurs

print(PrintWriter)

Description

Writes the contents of this node using the given print writer.

Syntax

```
public void print(java.io.PrintWriter out)
```

Parameters

out - PrintWriter to use

Throws

IOException - if an error occurs

readExternal(ObjectInput)

Description

This method reads the information written in the compressed stream by writeExternal method and restores the object correspondingly.

Syntax

```
public void readExternal(java.io.ObjectInput inArg)
```

Specified By

`java.io.Externalizable.readExternal(java.io.ObjectInput)` in interface `java.io.Externalizable`

Parameters

`inArg` - the `ObjectInput` stream used for reading the compressed stream

Throws

`IOException` - is thrown when there is an error in reading the input stream.
`ClassNotFoundException` - is thrown when the class is not found

removeChild(Node)

Description

Removes the child node indicated by `oldChild` from the list of children, and returns it.

Syntax

```
public org.w3c.dom.Node removeChild(org.w3c.dom.Node oldChild)
```

Parameters

`oldChild` - The node being removed.

Returns

The node removed.

Throws

`DOMException - NO_MODIFICATION_ALLOWED_ERR`: Raised if this node is `readonly`. `NOT_FOUND_ERR`: Raised if `oldChild` is not a child of this node.

removeEventListener(String, EventListener, boolean)

Description

This method allows the removal of event listeners from the event target (node).

Syntax

```
public void removeEventListener(java.lang.String type,  
org.w3c.dom.events.EventListener listener, boolean useCapture)
```

Parameters

`type` - Type of event for which the listener is registered

`listener` - The listener object

`useCapture` - flag to indicate if the listener wants to initiate capture

replaceChild(Node, Node)**Description**

Replaces the child node `oldChild` with `newChild` in the list of children, and returns the `oldChild` node. If the `newChild` is already in the tree, it is first removed.

Syntax

```
public org.w3c.dom.Node replaceChild(org.w3c.dom.Node newChild, org.w3c.dom.Node oldChild)
```

Parameters

`newChild` - The new node to put in the child list.

`oldChild` - The node being replaced in the list.

Returns

The node replaced.

Throws

`DOMException - HIERARCHY_REQUEST_ERR`: Raised if this node is of a type that does not allow children of the type of the `newChild` node, or it the node to put in is one of this node's ancestors. `WRONG_DOCUMENT_ERR`: Raised if `newChild` was created from a different document than the one that created this node. `NO_MODIFICATION_ALLOWED_ERR`: Raised if this node is readonly. `NOT_FOUND_ERR`: Raised if `oldChild` is not a child of this node.

reportSAXEvents(ContentHandler)**Description**

Report SAX Events from a DOM Tree

Syntax

```
public void reportSAXEvents(org.xml.sax.ContentHandler cntHandler)
```

Parameters

ContentHandler - cntHandler

Throws

SAXException - thrown by SAX Callback functions

resetNodeFlag(int)

Description

Resets the node flag information

Syntax

```
public void resetNodeFlag(int flag)
```

Parameters

flag - the node flag

selectNodes(String)

Description

Selects nodes from the tree which match the given pattern. This method assumes that the pattern does not contain namespace prefixes.

Syntax

```
public org.w3c.dom.NodeList selectNodes(java.lang.String pattern)
```

Parameters

pattern - XSL pattern to match

Returns

a list of matching nodes

Throws

XSLException - Raised if there is an error while doing the match

selectNodes(String, NSResolver)

Description

Selects nodes from the tree which match the given pattern

Syntax

```
public org.w3c.dom.NodeList selectNodes(java.lang.String pattern, NSResolver  
nsr)
```

Parameters

`pattern` - XSL pattern to match

`nsr` - NSResolver to resolve any namespace prefixes that occur in the given pattern

Returns

a list of matching nodes

Throws

`XSLException` - Raised if there is an error while doing the match

selectSingleNode(String)

Description

Selects the first node from the tree that matches the given pattern

Syntax

```
public org.w3c.dom.Node selectSingleNode(java.lang.String pattern)
```

Parameters

`pattern` - XSL pattern to match

Returns

matching node

Throws

`XSLException` - Raised if there is an error while doing the match

selectSingleNode(String, NSResolver)

Description

Selects the first node from the tree that matches the given pattern

Syntax

```
public org.w3c.dom.Node selectSingleNode(java.lang.String pattern, NSResolver nsr)
```

Parameters

`pattern` - XSL pattern to match

`nsr` - NSResolver to resolve any prefixes that occur in given pattern

Returns

matching node

Throws

`XSLException` - Raised if there is an error while doing the match

setDebugInfo(int, int, String)

Description

Set debug information in the node

Syntax

```
public void setDebugInfo(int line, int col, java.lang.String sysid)
```

Parameters

`line` - the line number

`col` - the col number

`sysid` - the system id

setNodeFlag(int)

Description

Sets the node flag information

Syntax

```
public void setNodeFlag(int flag)
```

Parameters

flag - the node flag

setNodeValue(String)**Description**

Sets the value of this node, depending on its type

Syntax

```
public void setNodeValue(java.lang.String nodeValue)
```

Throws

`DOMException - NO_MODIFICATION_ALLOWED_ERR`: Raised when the node is readonly. `DOMSTRING_SIZE_ERR`: Raised when it would return more characters than fit in a `DOMString` variable on the implementation platform.

setPrefix(String)**Description**

Sets the prefix of this node overridden by node types for which namespace is meaningful.

Syntax

```
public void setPrefix(java.lang.String prefix)
```

Parameters

prefix - set the prefix

Throws

`DOMException` - if any DOM Exception occurs

Since

DOM 2

setProperty(String, Object)

Description

Set a property of the node

Syntax

```
public void setProperty(java.lang.String propName, java.lang.Object propValue)
```

Parameters

`propName` -- Name of the property

`propValue` -- Value of the property

supports(String, String)

Description

Syntax

```
public boolean supports(java.lang.String feature, java.lang.String version)
```

Comments

Deprecated; use `isSupported` instead

Parameters

`String` - `feature` ,`String` `version`

Returns

true,if the feature is supported else false

Since

DOM 2 Tests whether the DOM implementation implements a specific feature and that feature is supported by this node.

transformNode(XSLStylesheet)

Description

Transforms a node in the tree using the given stylesheet

Syntax

```
public org.w3c.dom.DocumentFragment transformNode(XSLStylesheet xsl)
```


Parameters

`xsl` - XSLStylesheet to be used for transformation

Returns

a document fragment

Throws

`XSLException` - Raised if there is an error while doing the XSL transformation.

valueOf(String)**Description**

Selects the value of the first node from tree that matches the pattern

Syntax

```
public java.lang.String valueOf(java.lang.String pattern)
```

Parameters

`pattern` - XSL pattern to match

Returns

value of the matching node

Throws

`XSLException` - Raised if there is an error while doing the match

valueOf(String, NSResolver)**Description**

Selects the value of the first node from tree that matches the pattern

Syntax

```
public java.lang.String valueOf(java.lang.String pattern, NSResolver nsr)
```

Parameters

`pattern` - XSL pattern to match

`nsr` - NSResolver to resolve any prefixes that occur in given pattern

Returns

value of the matching node

Throws

`XSLException` - Raised if there is an error while doing the match

writeExternal(ObjectOutput)

Description

This method saves the state of the object by creating a binary compressed stream with information about this object.

Syntax

```
public void writeExternal(java.io.ObjectOutput out)
```

Specified By

`java.io.Externalizable.writeExternal(java.io.ObjectOutput)` in interface `java.io.Externalizable`

Parameters

`out` - The `ObjectOutput` stream used to write the serialized/ compressed stream.

Throws

`IOException` - is thrown when there is an exception while writing the serialized/compressed stream.

XMLNotation

Description

This class implements the DOM `Notation` interface and represents a notation declared in the Document Type Definition.

Syntax

```
public class XMLNotation implements java.io.Externalizable
```

```
oracle.xml.parser.v2.XMLNotation
```

Implemented Interfaces

```
java.io.Externalizable, java.io.Serializable
```

Constructors

XMLNotation()

Description

Default constructor

Syntax

```
public XMLNotation()
```

Comments

Note that this constructor is used only during deserialization/decompression of this DOM node. In order to deserialize this node to construct the DOM node from the serialized/ compressed stream, it is required to create a handle of the object. For all normal `XMLElement` creation use `XMLNotation(String)`

Methods

cloneNode(boolean)

Description

Returns a duplicate of this node, i.e., serves as a generic copy constructor for nodes.

Syntax

```
public org.w3c.dom.Node cloneNode(boolean deep)
```

Comments

The duplicate node has no parent (`parentNode` returns `null`). Cloning an `Element` copies all attributes and their values, including those generated by the XML processor to represent defaulted attributes, but this method does not copy any text it contains unless it is a deep clone, since the text is contained in a child `Text` node. Cloning any other type of node simply returns a copy of this node.

Parameters

`deep` - If `true`, recursively clone the subtree under the specified node; if `false`, clone only the node itself (and its attributes, if it is an `Element`).

Returns

The duplicate node.

getNodeName()

Description

Gets the name of the Notation

Syntax

```
public java.lang.String getNodeName()
```

Returns

name of the node

getNodeType()

Description

Gets a code representing the type of the underlying object

Syntax

```
public short getNodeType()
```

Returns

type of the node

getPublicId()

Description

Gets the Public identifier; if not specified, then null

Syntax

```
public java.lang.String getPublicId()
```

Returns

The public identifier

getSystemId()

Description

Gets the System identifier; if not specified, then null

Syntax

```
public java.lang.String getSystemId()
```

Returns

The System identifier

readExternal(ObjectInput)

Description

This method reads the information written in the compressed stream by writeExternal method and restores the object correspondingly.

Syntax

```
public void readExternal(java.io.ObjectInput inArg)
```

Specified By

java.io.Externalizable.readExternal(java.io.ObjectInput) in interface java.io.Externalizable

Parameters

in - the ObjectInput stream used for reading the compressed stream.

Throws

`IOException` - is thrown when there is an error in reading the input stream.

`ClassNotFoundException` - is thrown when the class is not found.

setPublicId(String)

Description

Sets the Public Identifier

Syntax

```
public void setPublicId(java.lang.String pubid)
```

Parameters

`pubid` - Public Identifier to set

setSystemId(String)

Description

Sets the System Identifier

Syntax

```
public void setSystemId(java.lang.String url)
```

Parameters

`url` - System Identifier to set

writeExternal(ObjectOutput)

Description

This method saves the state of the object by creating a binary compressed stream with information about this object.

Syntax

```
public void writeExternal(java.io.ObjectOutput outArg)
```

Specified By

`java.io.Externalizable.writeExternal(java.io.ObjectOutput)` in interface

`java.io.Externalizable`

Parameters

`out` - The `ObjectOutput` stream used to write the serialized/ compressed stream.

Throws

`IOException` - is thrown when there is an exception while writing the serialized/compressed stream.

XMLNSNode

Syntax

```
public class XMLNSNode extends oracle.xml.parser.v2.XMLNode
```

```
oracle.xml.parser.v2.XMLNode  
|  
+--oracle.xml.parser.v2.XMLNSNode
```

Implemented Interfaces

```
java.lang.Cloneable, java.io.Externalizable, java.io.Serializable
```

Description

Extends XMLNode to add support for Namespace names and children

Constructors

XMLNSNode(String)

Description

Constructs a new XMLNSNode having the given name and type

Syntax

```
protected XMLNSNode(java.lang.String tag)
```

Parameters

tag - Name of the node

Methods

addText(char[], int, int)

Description

Adds text to this node, or appends str to the last child if the last child is a text node.

Syntax

```
public void addText(char[] ch, int start, int length)
```


Parameters

ch - char array to add
start - start index in the char array
length - no of chars to be added

Throws

`XMLDOMException` - if text can't be added to this node

addText(String)**Description**

Adds text to this node, or appends str to the last child if the last child is a text node.

Syntax

```
public XMLNode addText(java.lang.String str)
```

Parameters

str - text to add

Throws

`XMLDOMException` - if text can't be added to this node

appendChild(Node)**Description**

Adds the node `newChild` to the end of the list of children of this node. If the `newChild` is already in the tree, it is first removed.

Syntax

```
public org.w3c.dom.Node appendChild(org.w3c.dom.Node newChildArg)
```

Overrides

`XMLNode.appendChild(Node)` in class `XMLNode`

Parameters

`newChild` - The node to add. If it is a `DocumentFragment` object, the entire contents of the document fragment are moved into the child list of this node

Returns

The node added.

Throws

`DOMException - HIERARCHY_REQUEST_ERR`: Raised if this node is of a type that does not allow children of the type of the `newChild` node, or if the node to append is one of this node's ancestors. `WRONG_DOCUMENT_ERR`: Raised if `newChild` was created from a different document than the one that created this node. `NO_MODIFICATION_ALLOWED_ERR`: Raised if this node is readonly.

getChildNodes()**Description**

Gets a `NodeList` that contains all children of this node. If there are no children, this is a `NodeList` containing no nodes. The content of the returned `NodeList` is "live" in the sense that, for instance, changes to the children of the node object that it was created from are immediately reflected in the nodes returned by the `NodeList` accessors; it is not a static snapshot of the content of the node. This is true for every `NodeList`, including the ones returned by the `getElementsByTagName` method.

Syntax

```
public org.w3c.dom.NodeList getChildNodes()
```

Overrides

`XMLNode.getChildNodes()` in class `XMLNode`

Returns

The children of this node

getFirstChild()**Description**

Gets the first child of this node. If there is no such node, this returns `null`.

Syntax

```
public org.w3c.dom.Node getFirstChild()
```

Overrides

`XMLNode.getFirstChild()` in class `XMLNode`

Returns

The first child of this node

getLastChild()**Description**

Gets the last child of this node. If there is no such node, this returns null.

Syntax

```
public org.w3c.dom.Node getLastChild()
```

Overrides

`XMLNode.getLastChild()` in class `XMLNode`

Returns

The last child of this node

getLocalName()**Description**

Gets the Local Name of this node overridden by node types for which namespace is meaningful.

Syntax

```
public java.lang.String getLocalName()
```

Overrides

`XMLNode.getLocalName()` in class `XMLNode`

Returns

localname of the node

Since

DOM 2

getNamespaceURI()

Description

Gets the namespace URI of this node. overridden by node types for which namespace is meaningful.

Syntax

```
public java.lang.String getNamespaceURI()
```

Overrides

`XMLNode.getNamespaceURI()` in class `XMLNode`

Returns

namespace of the node

Since

DOM 2

getNodeName()

Description

Gets the name of this node, depending on its type

Syntax

```
public java.lang.String getNodeName()
```

Overrides

`XMLNode.getNodeName()` in class `XMLNode`

Returns

Name of this node

getPrefix()

Description

Gets the prefix of this node overridden by node types for which namespace is meaningful.

Syntax

```
public java.lang.String getPrefix()
```

Overrides

```
XMLNode.getPrefix() in class XMLNode
```

Returns

prefix of the node

Since

DOM 2

getText()Description**Description**

Returns the non-marked-up text contained by this element.

Syntax

```
public java.lang.String getText()
```

Comments

For text elements, this is the raw data. For elements with child nodes, this method traverses the entire subtree and appends the text for each terminal text element, effectively stripping out the XML markup for the subtree. For example, if the XML document contains the following: William Shakespeare

`XMLDocument.getText` returns "William Shakespeare".

Overrides

```
XMLNode.getText() in class XMLNode
```

Returns

non-marked-up text contained bu this element

hasChildNodes()**Description**

This is a convenience method to allow easy determination of whether a node has any children.

Syntax

```
public boolean hasChildNodes()
```

Overrides

```
XMLNode.hasChildNodes() in class XMLNode
```

Returns

true if the node has any children, false if the node has no children.

insertBefore(Node, Node)

Description

Inserts the node `newChild` before the existing child node `refChild`.

Syntax

```
public org.w3c.dom.Node insertBefore(org.w3c.dom.Node newChildArg,  
org.w3c.dom.Node refChildArg)
```

Comments

If `refChild` is null, insert `newChild` at the end of the list of children. If `newChild` is a `DocumentFragment` object, all of its children are inserted, in the same order, before `refChild`. If the `newChild` is already in the tree, it is first removed.

Overrides

```
XMLNode.insertBefore(Node, Node) in class XMLNode
```

Parameters

`newChild` - The node to insert.

`refChild` - The reference node, i.e., the node before which the new node must be inserted.

Returns

The node being inserted.

Throws

`DOMException - HIERARCHY_REQUEST_ERR`: Raised if this node is of a type that does not allow children of the type of the `newChild` node, or if the node to

`insert` is one of this node's ancestors. `WRONG_DOCUMENT_ERR`: Raised if `newChild` was created from a different document than the one that created this node. `NO_MODIFICATION_ALLOWED_ERR`: Raised if this node is readonly. `NOT_FOUND_ERR`: Raised if `refChild` is not a child of this node.

normalize()

Description

Puts all Text nodes in the full depth of the sub-tree underneath this Node, including attribute nodes, into "normal" form where only structure (e.g., elements, comments, processing instructions, CDATA sections, and entity references) separates Text nodes, i.e., there are neither adjacent Text nodes nor empty Text nodes.

Syntax

```
public void normalize()
```

Comments

This can be used to ensure that the DOM view of a document is the same as if it were saved and re-loaded, and is useful when operations (such as XPointer lookups) that depend on a particular document tree structure are to be used.

Overrides

`XMLNode.normalize()` in class `XMLNode`

Since

DOM 2

removeChild(Node)

Description

Removes the child node indicated by `oldChild` from the list of children, and returns it.

Syntax

```
public org.w3c.dom.Node removeChild(org.w3c.dom.Node oldChildArg)
```

Overrides

`XMLNode.removeChild(Node)` in class `XMLNode`

Parameters

`oldChild` - The node being removed.

Returns

The node removed.

Throws

`DOMException - NO_MODIFICATION_ALLOWED_ERR`: Raised if this node is `readonly`. `NOT_FOUND_ERR`: Raised if `oldChild` is not a child of this node.

replaceChild(Node, Node)**Description**

Replaces the child node `oldChild` with `newChild` in the list of children, and returns the `oldChild` node. If the `newChild` is already in the tree, it is first removed.

Syntax

```
public org.w3c.dom.Node replaceChild(org.w3c.dom.Node newChildArg,  
org.w3c.dom.Node oldChildArg)
```

Overrides

`XMLNode.replaceChild(Node, Node)` in class `XMLNode`

Parameters

`newChild` - The new node to put in the child list.

`oldChild` - The node being replaced in the list.

Returns

The node replaced.

Throws

`DOMException - HIERARCHY_REQUEST_ERR`: Raised if this node is of a type that does not allow children of the type of the `newChild` node, or if the node to put in is one of this node's ancestors. `WRONG_DOCUMENT_ERR`: Raised if `newChild` was created from a different document than the one that created this node. `NO_MODIFICATION_ALLOWED_ERR`: Raised if this node is `readonly`. `NOT_FOUND_ERR`: Raised if `oldChild` is not a child of this node.

setPrefix(String)

Description

Sets the prefix of this node overridden by node types for which namespace is meaningful.

Syntax

```
public void setPrefix(java.lang.String prefix)
```

Overrides

`XMLNode.setPrefix(String)` in class `XMLNode`

Parameters

`prefix` - Sets the prefix of the node

Since

DOM 2

XMLOutputStream

Description

XMLOutputStream -- writes output stream, can handle XML encoding

Syntax

```
public class XMLOutputStream extends java.lang.Object
```

```
java.lang.Object  
|  
+--oracle.xml.parser.v2.XMLOutputStream
```

Fields of XMLOutputStream

Table 11–18 *Fields of XMLOutputStream*

Field	Syntax	Description
COMPACT	public static int COMPACT	
DEFAULT	public static int DEFAULT	
PRETTY	public static int PRETTY	

Constructors

XMLOutputStream(OutputStream)

Description

Builds the ASCII output stream

Syntax

```
public XMLOutputStream(java.io.OutputStream out)
```

Parameters

out - the output stream

XMLOutputStream(PrintWriter)

Description

Builds the output stream from PrintWriter

Syntax

```
public XMLOutputStream(java.io.PrintWriter out)
```

Parameters

out - the PrintWriter stream

Methods

addIndent(int)

Description

Set indenting level for output

Syntax

```
public void addIndent(int offset)
```

Parameters

offset - the indenting level

close()

Description

Closes the output stream

Syntax

```
public void close()
```

Throws

IOException - if there is any error in closing the output stream

flush()

Description

Flushes the output stream

Syntax

```
public void flush()
```

Throws

`IOException` - if there is any error in flushing the output stream

getOutputStyle()

Description

Gets the Current output style

Syntax

```
public int getOutputStyle()
```

setEncoding(String, boolean, boolean)

Description

Sets the output character encoding

Syntax

```
public void setEncoding(java.lang.String encoding, boolean lendian, boolean  
byteOrderMark)
```

Parameters

`encoding` - the encoding of the stream

`lendian` - flag to indicate if the encoding is of type little endian

`byteOrderMark` - flag to indicate if byte order mark is set

Throws

`IOException` - if error is thrown in setting the encoding type

setOutputStyle(int)

Description

Sets the Output the style

Syntax

```
public void setOutputStyle(int style)
```

Parameters

`style` - the output style

write(int)

Description

Outputs character according to type of the output stream

Syntax

```
public void write(int c)
```

Parameters

`c` - the character that needs to be written

Throws

`IOException` - if there is any error in writing the character

writeChars(String)

Description

Write string to the output

Syntax

```
public void writeChars(java.lang.String str)
```

Parameters

`str` - the string that is written to the output stream

Throws

`IOException` - error thrown if there is any error in writing the string

writeIndent()

Description

Output the indentation

Syntax

```
public void writeIndent()
```

Throws

`IOException` - error thrown if there is any error in writing the string

writeNewLine()

Description

Newline writer

Syntax

```
public void writeNewLine()
```

Throws

`IOException` - error thrown if there is any error in writing the string

writeQuotedString(String)

Description

Write string with surrounding quotes

Syntax

```
public void writeQuotedString(java.lang.String str)
```

Parameters

`str` - the string that is written to the output stream

Throws

`IOException` - error thrown if there is any error in writing the string

XMLParseException

Description

Indicates that a parsing exception occurred while processing an XML document

Syntax

```
public class XMLParseException
```

```
oracle.xml.parser.v2.XMLParseException
```

Fields of XMLParseException

Table 11–19 *Fields of XMLParseException*

Field	Syntax	Description
ERROR	public static final int ERROR	Code for non-fatal error
FATAL_ERROR	public static final int FATAL_ERROR	Code for fatal error
WARNING	public static final int WARNING	Code for warning

Constructors

XMLParseException(String, String, String, int, int, int)

Description

Syntax

```
public XMLParseException(java.lang.String msg, java.lang.String pubId,
    java.lang.String sysId, int line, int col, int type)
```

Methods

formatErrorMessage(int)

Description

Get the error message at specified index

Syntax

```
public java.lang.String formatErrorMessage(int i)
```

Returns

The error message

getColumnNumber(int)

Description

Get the column number of error at specified index

Syntax

```
public int getColumnNumber(int i)
```

Returns

The column number

getException(int)

Description

Get the exception (if exists) that occurred in error at specified index

Syntax

```
public java.lang.Exception getException(int i)
```

Returns

The exception

getLineNumber(int)

Description

Get the line number of error at specified index

Syntax

```
public int getLineNumber(int i)
```

Returns

The line number

getMessage(int)

Description

Get the error message at specified index

Syntax

```
public java.lang.String getMessage(int i)
```

Returns

The error message

getMessageType(int)

Description

Get the type of the error message at specified index

Syntax

```
public int getMessageType(int i)
```

Returns

The error message type

getNumMessages()

Description

Return the total number of errors/warnings found during parsing

Syntax

```
public int getNumMessages()
```

Returns

The number of errors/warnings

getPublicId(int)

Description

Get the public ID of input when error at specified index occurred

Syntax

```
public java.lang.String getPublicId(int i)
```

Returns

The public ID

getSystemId(int)

Description

Get the system ID of input when error at specified index occurred

Syntax

```
public java.lang.String getSystemId(int i)
```

Returns

The system ID

XMLParser

Description

This class serves as a base class for the `DOMParser` and `SAXParser` classes. It contains methods to parse eXtensible Markup Language (XML) 1.0 documents according to the World Wide Web Consortium (W3C) recommendation. This class can not be instantiated (applications may use the DOM or SAX parser depending on their requirements).

Syntax

```
public abstract class XMLParser
```

```
oracle.xml.parser.v2.XMLParser
```

Fields of XMLParser

Table 11–20 *Fields of XMLParser*

Field	Syntax	Description
<code>BASE_URL</code>	<code>public static final java.lang.String BASE_URL</code>	Base URL used in parsing entities. Replaces <code>setBaseURL()</code> ; Should be URL object
<code>DTD_OBJECT</code>	<code>public static final java.lang.String DTD_OBJECT</code>	DTD Object to be used for validation. Replaces <code>XMLParser.setDoctype()</code>
<code>SCHEMA_OBJECT</code>	<code>public static final java.lang.String SCHEMA_OBJECT</code>	Schema Object to be used for validation. Replaces <code>XMLParser.setXMLSchema()</code>
<code>STANDALONE</code>	<code>public static final java.lang.String STANDALONE</code>	Sets the standalone property of the input files. If true the DTDs are not retrieved.
<code>USE_DTD_ONLY_FOR_VALIDATION</code>	<code>public static final java.lang.String USE_DTD_ONLY_FOR_VALIDATION</code>	If true, DTD Object is used only for validation and is not added to the parser document (Boolean.TRUE or Boolean.FALSE) This property/attribute is only if <code>setDoctype</code> is called to use a fixed DTD.

Methods

getAttribute(String)

Description

Allows the user to retrieve specific attributes on the underlying implementation.

Syntax

```
public java.lang.Object getAttribute(java.lang.String name)
```

Parameters

name - The name of the attribute.

Returns

value The value of the attribute.

Throws

`IllegalArgumentException` - thrown if the underlying implementation doesn't recognize the attribute.

getBaseUrl()

Description

Gets the base URL

Syntax

```
public java.net.URL getBaseUrl()
```

Returns

the base URL

getEntityResolver()

Description

Return the current entity resolver.

Syntax

```
public org.xml.sax.EntityResolver getEntityResolver()
```

Returns

The current entity resolver, or null if none has been registered.

Since

SAX 2.0

See Also

`setEntityResolver(EntityResolver)`

getErrorHandler()**Description**

Return the current error handler.

Syntax

```
public org.xml.sax.ErrorHandler getErrorHandler()
```

Returns

The current error handler, or null if none has been registered.

Since

SAX 2.0

See Also

`setErrorHandler(ErrorHandler)`

getReleaseVersion()Description**Description**

Returns the release version of the Oracle XML Parser

Syntax

```
public static java.lang.String getReleaseVersion()
```

Returns

the release version string

getValidationModeValue()

Description

Returns the validation mode

Syntax

```
public int getValidationModeValue()
```

Returns

0 if the XML parser is NONVALIDATING 1 if the XML parser is PARTIAL_VALIDATION 2 if the XML parser is DTD_VALIDATION 3 if the XML parser is SCHEMA_VALIDATION

getXMLProperty(String)

Description

Get a property. The property is returned if present and supported, else null is returned

Syntax

```
public java.lang.Object getXMLProperty(java.lang.String name)
```

Parameters

name - - name of the property

Returns

Object - value of the property

isXMLPropertyReadOnly(String)

Description

Check is the property is read-only Returns true if the property is not supported

Syntax

```
public boolean isXMLPropertyReadOnly(java.lang.String name)
```

Parameters

name - - name of the property

Returns

boolean - TRUE is read-only

isXMLPropertySupported(String)

Description

Check is the property is supported

Syntax

```
public boolean isXMLPropertySupported(java.lang.String name)
```

Parameters

name - - name of the property

Returns

boolean - TRUE is supported

parse(InputSource)

Description

Parses the XML from given input source

Syntax

```
public void parse(org.xml.sax.InputSource in)
```

Parameters

in - the org.xml.sax.InputSouce to parse

Throws

XMLParseException - if syntax or other error encountered.

SAXException - Any SAX exception, possibly wrapping another exception.

IOException - IO Error.

parse(InputStream)

Description

Parses the XML from given input stream. The base URL should be set for resolving external entities and DTD.

Syntax

```
public final void parse(java.io.InputStream in)
```

Parameters

in - the `InputStream` containing XML data to parse.

Throws

`XMLParseException` - if syntax or other error encountered.

`SAXException` - Any SAX exception, possibly wrapping another exception.

`IOException` - IO Error.

See Also

```
setBaseURL(URL)
```

parse(Reader)

Description

Parses the XML from given input stream. The base URL should be set for resolving external entities and DTD.

Syntax

```
public final void parse(java.io.Reader r)
```

Parameters

r - the `Reader` containing XML data to parse.

Throws

`XMLParseException` - if syntax or other error encountered.

`SAXException` - Any SAX exception, possibly wrapping another exception.

`IOException` - IO Error.

See Also

```
setBaseURL(URL)
```


parse(String)

Description

Parses the XML from the URL indicated

Syntax

```
public void parse(java.lang.String in)
```

Parameters

in - the String containing the URL to parse from

Throws

XMLParseException - if syntax or other error encountered.

SAXException - Any SAX exception, possibly wrapping another exception.

IOException - IO Error.

parse(URL)

Description

Parses the XML document pointed to by the given URL and creates the corresponding XML document hierarchy.

Syntax

```
public final void parse(java.net.URL url)
```

Parameters

url - the url points to the XML document to parse.

Throws

XMLParseException - if syntax or other error encountered.

SAXException - Any SAX exception, possibly wrapping another exception.

IOException - IO Error.

reset()

Description

Resets the parser state

Syntax

```
public void reset()
```

setAttribute(String, Object)

Description

Allows the user to set specific attributes on the underlying implementation.

Syntax

```
public void setAttribute(java.lang.String name, java.lang.Object value)
```

Parameters

`name` - The name of the attribute.

`value` - The value of the attribute.

Throws

`IllegalArgumentException` - thrown if the underlying implementation doesn't recognize the attribute.

setBaseURL(URL)

Description

Set the base URL for loading external entities and DTDs. This method should be called if the `parse(InputStream)` is used to parse the XML Document

Syntax

```
public void setBaseURL(java.net.URL url)
```

Parameters

`url` - The base URL

setDoctype(DTD)

Description

Set the DTD

Syntax

```
public void setDoctype(DTD dtd)
```

Parameters

dtd - DTD to set and used while parsing

setEntityResolver(EntityResolver)**Description**

Allow an application to register an entity resolver.

Syntax

```
public void setEntityResolver(org.xml.sax.EntityResolver resolver)
```

Comments

If the application does not register an entity resolver, the XMLReader will perform its own default resolution.

Applications may register a new or different resolver in the middle of a parse, and the SAX parser must begin using the new resolver immediately.

Parameters

resolver - The entity resolver.

Throws

java.lang.NullPointerException - If the resolver argument is null.

See Also

```
getEntityResolver()
```

setErrorHandler(ErrorHandler)**Description**

Allow an application to register an error event handler.

Syntax

```
public void setErrorHandler(org.xml.sax.ErrorHandler handler)
```

Comments

If the application does not register an error handler, all error events reported by the SAX parser will be silently ignored; however, normal processing may not continue.

It is highly recommended that all SAX applications implement an error handler to avoid unexpected bugs.

Applications may register a new or different handler in the middle of a parse, and the SAX parser must begin using the new handler immediately.

Parameters

`handler` - The error handler.

Throws

`java.lang.NullPointerException` - If the handler argument is null.

See Also

`getErrorHandler()`

setLocale(Locale)

Description

Applications can use this to set the locale for error reporting.

Syntax

```
public void setLocale(java.util.Locale locale)
```

Parameters

`locale` - Locale to set

Throws

`SAXException` - A `SAXException` could be thrown.

See Also

`org.xml.sax.Parser`

setPreserveWhitespace(boolean)

Description

Set the white space preserving mode

Syntax

```
public void setPreserveWhitespace(boolean flag)
```

Parameters

flag - preserving mode

setValidationMode(int)**Description**

Set the validation mode

Syntax

```
public void setValidationMode(int valMode)
```

Comments

This method sets the validation mode of the parser to one of the 4 types: NONVALIDATING, PARTIAL_VALIDATION, DTD_VALIDATION and SCHEMA_VALIDATION.

Parameters

valMode - determines the type of the validation mode which the parser is set to

setXMLProperty(String, Object)**Description**

Set a property

Syntax

```
public java.lang.Object setXMLProperty(java.lang.String name, java.lang.Object value)
```

Comments

The value of the property set is returned if successfully set, a null is returned if the property is read-only and cannot be set or is not supported.

Parameters

name - - name of the property

value - - value of the property

Returns

Object - the set property

setXMLSchema(Object)

Description

Set an XMLSchema for validating the instance document

Syntax

```
public void setXMLSchema(java.lang.Object schema)
```

Parameters

schema - The XMLSchema object

XMLPI

Description

This class implements the DOM Processing Instruction interface.

Syntax

```
public class XMLPI implements java.io.Externalizable
```

```
oracle.xml.parser.v2.XMLPI
```

Direct Known Subclasses

```
XMLDeclPI
```

All Implemented Interfaces

```
java.io.Externalizable, java.io.Serializable
```

See Also

```
ProcessingInstruction, NodeFactory, DOMParser.setNodeFactory(NodeFactory)
```

Constructors

XMLPI()

Description

Default constructor.

Syntax

```
public XMLPI()
```

Comments

Note that this constructor is used only during deserialization/decompression of this DOM node. In order to deserialize this node to construct the DOM node from the serialized/ compressed stream, it is required to create a handle of the object.

Methods

addText(String)

Description

Adds text string to the node

Syntax

```
public XMLNode addText(java.lang.String str)
```

Parameters

`str` - the text string to be added

Returns

the node

getNodeName()

Description

Gets the name of the PI

Syntax

```
public java.lang.String getNodeName()
```

Returns

name of the node

getNodeType()

Description

Gets a code representing the type of the underlying object

Syntax

```
public short getNodeType()
```

Returns

type of the node

getTarget()

Description

Returns the target of this PI. XML defines this as the first token following markup that begins the processing instruction.

Syntax

```
public java.lang.String getTarget()
```

Returns

The target of the PI.

readExternal(ObjectInput)

Description

This method reads the information written in the compressed stream by writeExternal method and restores the object correspondingly.

Syntax

```
public void readExternal(java.io.ObjectInput inArg)
```

Specified By

java.io.Externalizable.readExternal(java.io.ObjectInput) in interface java.io.Externalizable

Parameters

in - The ObjectInput stream used for reading the compressed stream

Throws

IOException - is thrown when there is an error in reading the input stream.

ClassNotFoundException - is thrown when the class is not found

reportSAXEvents(ContentHandler)

Description

Report SAX Events from a DOM Tree

Syntax

```
public void reportSAXEvents(org.xml.sax.ContentHandler cntHandler)
```

Parameters

ContentHandler - cntHandler

Throws

SAXException - thrown by SAX Callback functions

writeExternal(ObjectOutput)

Description

This method saves the state of the object by creating a binary compressed stream with information about this object.

Syntax

```
public void writeExternal(java.io.ObjectOutput outArg)
```

Specified By

java.io.Externalizable.writeExternal(java.io.ObjectOutput) in interface java.io.Externalizable

Parameters

out - is the ObjectOutput stream used to write the compressed stream.

Throws

IOException - is thrown when there is an exception while writing the compressed stream.

XMLPrintDriver

Description

The XMLPrintDriver implements PrintDriver interface.

Syntax

```
public class XMLPrintDriver extends java.lang.Object implements
oracle.xml.parser.v2.PrintDriver
```

```
java.lang.Object
|
+--oracle.xml.parser.v2.XMLPrintDriver
```

Implemented Interfaces

PrintDriver

Fields of XMLPrintDriver

Table 11–21 Fields of XMLPrintDriver

Field	Syntax	Description
out	protected XMLOutputStream out	

Constructors

XMLPrintDriver(OutputStream)

Description

Syntax

```
public XMLPrintDriver(java.io.OutputStream os)
```

XMLPrintDriver(PrintWriter)

Description

Syntax

```
public XMLPrintDriver(java.io.PrintWriter pw)
```

Methods

close()

Description

Closes the output stream or print writer

Syntax

```
public void close()
```

Specified By

`PrintDriver.close()` in interface `PrintDriver`

flush()

Description

Flushes the output stream or print writer

Syntax

```
public void flush()
```

Specified By

`PrintDriver.flush()` in interface `PrintDriver`

printAttribute(XMLAttr)

Description

Prints a `XMLAttr` node

Syntax

```
public void printAttribute(XMLAttr attr)
```

Specified By

`PrintDriver.printAttribute(XMLAttr)` in interface `PrintDriver`

Parameters

`attr` - The `XMLAttr` node.

printAttributeNodes(XMLElement)**Description**

Calls `print` method for each attribute of the `XMLElement`

Syntax

```
public final void printAttributeNodes(XMLElement elem)
```

Specified By

`PrintDriver.printAttributeNodes(XMLElement)` in interface `PrintDriver`

Parameters

`elem` - The `elem` whose attributes are to be printed.

printCDATASection(XMLCDATA)**Description**

Prints a `XMLCDATA` node

Syntax

```
public void printCDATASection(XMLCDATA cdata)
```

Specified By

`PrintDriver.printCDATASection(XMLCDATA)` in interface `PrintDriver`

Parameters

`cdata` - The `XMLCDATA` node.

printChildNodes(XMLNode)

Description

Calls print method for each child of the XMLNode

Syntax

```
public final void printChildNodes(XMLNode node)
```

Specified By

PrintDriver.printChildNodes(XMLNode) in interface PrintDriver

Parameters

node - The node whose children are to be printed.

printComment(XMLComment)

Description

Prints a XMLComment node

Syntax

```
public void printComment(XMLComment comment)
```

Specified By

PrintDriver.printComment(XMLComment) in interface PrintDriver

Parameters

comment - The comment node.

printDoctype(DTD)

Description

Prints an DTD.

Syntax

```
public void printDoctype(DTD dtd)
```

Specified By

PrintDriver.printDoctype(DTD) in interface PrintDriver

Parameters

dtd - The dtd to be printed.

printDocument(XMLDocument)**Description**

Prints an XMLDocument.

Syntax

```
public void printDocument(XMLDocument doc)
```

Specified By

`PrintDriver.printDocument(XMLDocument)` in interface `PrintDriver`

Parameters

elem - The document to be printed.

printDocumentFragment(XMLDocumentFragment)**Description**

Prints an empty XMLDocumentFragment object.

Syntax

```
public void printDocumentFragment(XMLDocumentFragment dfrag)
```

Specified By

`PrintDriver.printDocumentFragment(XMLDocumentFragment)` in interface `PrintDriver`

Parameters

dfrag - The document fragment to be printed.

printElement(XMLElement)**Description**

Prints an XMLElement.

Syntax

```
public void printElement(XMLElement elem)
```

Specified By

`PrintDriver.printElement(XMLElement)` in interface `PrintDriver`

Parameters

`elem` - The element to be printed.

printEntityReference(XMLEntityReference)

Description

Prints a `XMLEntityReference` node

Syntax

```
public void printEntityReference(XMLEntityReference en)
```

Specified By

`PrintDriver.printEntityReference(XMLEntityReference)` in interface `PrintDriver`

Parameters

`en` - The `XMLEntityReference` node.

printProcessingInstruction(XMLPI)

Description

Prints a `XMLPI` node

Syntax

```
public void printProcessingInstruction(XMLPI pi)
```

Specified By

`PrintDriver.printProcessingInstruction(XMLPI)` in interface `PrintDriver`

Parameters

`pi` - The `XMLPI` node.

printTextNode(XMLText)

Description

Prints a XMLText node

Syntax

```
public void printTextNode(XMLText text)
```

Specified By

`PrintDriver.printTextNode(XMLText)` in interface `PrintDriver`

Parameters

`text` - The text node.

setEncoding(String)

Description

Sets the encoding of the print driver.

Syntax

```
public void setEncoding(java.lang.String enc)
```

Specified By

`PrintDriver.setEncoding(String)` in interface `PrintDriver`

Parameters

`enc` - The encoding of the document being printed.

XMLRangeException

Description

This class customizes the RangeException

Syntax

```
public class XMLRangeException
```

```
oracle.xml.parser.v2.XMLRangeException
```

Constructors

XMLRangeException(short)

Description

Syntax

```
public XMLRangeException(short code)
```

XMLText

Description

This class implements the DOM Text interface.

Syntax

```
public class XMLText implements java.io.Serializable, java.io.Externalizable
```

```
oracle.xml.parser.v2.XMLText
```

Implemented Interfaces

```
java.io.Externalizable, java.io.Serializable
```

See Also

[Text](#), [NodeFactory](#), [DOMParser.setNodeFactory\(NodeFactory\)](#)

Constructors

XMLText()

Description

Default constructor.

Syntax

```
public XMLText()
```

Comments

Note that this constructor is used only during deserialization/decompression of this DOM node. In order to deserialize this node to construct the DOM node from the serialized/ compressed stream, it is required to create a handle of the object.

Methods

addText(char[], int, int)

Description

Adds text to the data of the text node, similar to `appendData`

Syntax

```
public void addText(char[] ch, int start, int length)
```

Parameters

`ch` - char array to be appended

`start` - start index

`length` - length of the char array

getData()

Description

The character data of the node that implements this interface.

Syntax

```
public java.lang.String getData()
```

Comments

The DOM implementation may not put arbitrary limits on the amount of data that may be stored in a `Text` node. However, implementation limits may mean that the entirety of a node's data may not fit into a single `DOMString`. In such cases, the user may call `substringData` to retrieve the data in appropriately sized pieces.

Throws

`DOMException - NO_MODIFICATION_ALLOWED_ERR`: Raised when the node is `readonly`.

`DOMException - DOMSTRING_SIZE_ERR`: Raised when it would return more characters than fit in a `DOMString` variable on the implementation platform.

getNodeName()

Description

Gets the name of the XMLText

Syntax

```
public java.lang.String getNodeName()
```

Returns

name of the node

getNodeType()

Description

Gets a code representing the type of the underlying object

Syntax

```
public short getNodeType()
```

Returns

type of the node

getNodeValue()

Description

Gets a value of this text node

Syntax

```
public java.lang.String getNodeValue()
```

Returns

String value of the node

Throws

`DOMException` - in any error occurs when retrieving the value

isWhiteSpaceNode()

Description

Check is the text node is a whitespace node

Syntax

```
public boolean isWhiteSpaceNode()
```

Returns

boolean

readExternal(ObjectInput)

Description

This method reads the information written in the compressed stream by writeExternal method and restores the object correspondingly.

Syntax

```
public void readExternal(java.io.ObjectInput inArg)
```

Comments

This method is called if XMLText object is deserialized (or read) as an independent node and not called from some other DOM node.

Specified By

java.io.Externalizable.readExternal(java.io.ObjectInput) in interface java.io.Externalizable

Parameters

in - The ObjectInput stream used for reading the compressed stream

Throws

IOException - is thrown when there is an error in reading the input stream.

ClassNotFoundException - is thrown when the class is not found

reportSAXEvents(ContentHandler)

Description

Report SAX Events from a DOM Tree

Syntax

```
public void reportSAXEvents(org.xml.sax.ContentHandler cntHandler)
```

Parameters

ContentHandler - cntHandler

Throws

SAXException - thrown by SAX Callback functions

splitText(int)**Description**

Breaks `Text` node into two `Text` nodes at specified offset, so they are both siblings, and the node only contains content up to the offset. New node inserted as next sibling contains all content at and after the offset point.

Syntax

```
public org.w3c.dom.Text splitText(int offset)
```

Parameters

offset - Offset at which to split, starting from 0

Returns

New `Text` node

Throws

DOMException - `INDEX_SIZE_ERR`: Raised if specified offset is negative or greater than number of characters in data. `NO_MODIFICATION_ALLOWED_ERR`: Raised if this node is readonly.

writeExternal(ObjectOutput)**Description**

This method saves the state of the object by creating a binary compressed stream with information about this object.

Syntax

```
public void writeExternal(java.io.ObjectOutput outArg)
```

Specified By

`java.io.Externalizable.writeExternal(java.io.ObjectOutput)` in interface `java.io.Externalizable`

Parameters

`out` - The `ObjectOutput` stream used to write the compressed stream.

Throws

`IOException` - is thrown when there is an exception while writing the compressed stream.

XMLToken Interface

Description

Basic interface for XMLToken

Syntax

```
public interface XMLToken
```

Comments

All XMLParser applications with Tokenizer feature must implement this interface. The interface has to be registered using XMLParser method `setTokenHandler(XMLToken handler)`.

If XMLtoken handler != null then for each registered and found token the parser calls the XMLToken call-back method `token(int token, String value)`. During tokenizing the parser doesn't validate the document and doesn't include/read internal/external entities. If XMLtoken handler == null then the parser parses as usual.

A request for XML token is registered (on/off) using XMLParser method `setToken(int token, boolean set)`. The requests could be registered during the parsing (from inside the call-back method) as well.

The XML tokens are defined as public constants in XMLToken interface. They correspond to the XML syntax variables from W3C XML Syntax Specification.

Fields of XMLToken

Table 11–22 *Fields of XMLToken*

Field	Syntax	Description
AttListDecl	public static final int AttListDecl	AttListDecl ::= '<' '!' 'ATTLIST' S Name AttDef* S? '>'
AttName	public static final int AttName	AttName ::= Name
Attribute	public static final int Attribute	Attribute ::= AttName Eq AttValue
AttValue	public static final int AttValue	AttValue ::= '"' ([^<"] Reference)* '"' "'" ([^<&'] Reference)* "'"

Table 11–22 Fields of XMLToken

Field	Syntax	Description
CDSect	public static final int CDSect	CDSect ::= CDStart CData CDEnd CDStart ::= '<' '!' '[CDATA[' CData ::= (Char* - (Char* '])>' Char*) CDEnd ::= ']>'
CharData	public static final int CharData	CharData ::= [<&]* - ([<&]* ' ')] [<&]*
Comment	public static final int Comment	Comment ::= '<' '!' '--' ((Char - '-') ('-' (Char - '-')))* '-->'
DTDName	public static final int DTDName	DTDName ::= name
ElemDeclName	public static final int ElemDeclName	ElemDeclName ::= name
elementdecl	public static final int elementdecl	elementdecl ::= '<' '!ELEMENT' S ElemDeclName S contentspec S? '>'
EmptyElemTag	public static final int EmptyElemTag	EmptyElemTag ::= '<' STagName (S Attribute)* S? '/' '>'
EntityDecl	public static final int EntityDecl	EntityDecl ::= '<' '!' ENTITY' S EntityDeclName S EntityDef S? '>' '<' '!' ENTITY' S '%' S EntityDeclName S PDef S? '>' EntityDef ::= EntityValue (ExternalID NDataDecl?) PDef ::= EntityValue ExternalID
EntityDeclName	public static final int EntityDeclName	EntityValue ::= "" ([^%&"] PEReference Reference)* "" "" ([^%&"] PEReference Reference)* ""
EntityValue	public static final int EntityValue	EntityDeclName ::= Name
ETag	public static final int ETag	ETag ::= '<' '/' ETagName S? '>'
ETagName	public static final int ETagName	ETagName ::= Name
ExternalID	public static final int ExternalID	ExternalID ::= 'SYSTEM' S SystemLiteral 'PUBLIC' S PubidLiteral S SystemLiteral
NotationDecl	public static final int NotationDecl	NotationDecl ::= '<' '!NOTATION' S Name S (ExternalID PublicID) S? '>'
PI	public static final int PI	PI ::= '<' '?' PITarget (S (Char* - (Char* '?>' Char*)))? '?' '>'

Table 11–22 Fields of XMLToken

Field	Syntax	Description
PITarget	public static final int PITarget	PITarget ::= Name - (('X' 'x') ('M' 'm') ('L' 'l'))
Reference	public static final int Reference	Reference ::= EntityRef CharRef PEReference EntityRef ::= '&' Name ';' ; PEReference ::= '%' Name ';' ; CharRef ::= '&#' [0-9]+ ';' '&#x' [0-9a-fA-F]+ ';' ;
STag	public static final int STag	STag ::= '<' STagName (S Attribute)* S? '>'
STagName	public static final int STagName	STagName ::= Name
TextDecl	public static final int TextDecl	TextDecl ::= '<' '?' 'xml' VersionInfo? EncodingDecl S? '?>'
XMLDecl	public static final int XMLDecl	XMLDecl ::= '<' '?' 'xml' VersionInfo EncodingDecl? SDDDecl? S? '?>'

Methods

token(int, String)

Description

The interface call-back method. Receives an XML token and it's corresponding value

Syntax

```
public void token(int token, java.lang.String value)
```

Parameters

`token` - The XML token constant as specified in the interface.

`value` - The corresponding substring from the parsed text.

XMLTokenizer

Description

This class implements an eXtensible Markup Language (XML) 1.0 parser according to the World Wide Web Consortium (W3C) recommendation.

Syntax

```
public class XMLTokenizer
```

```
oracle.xml.parser.v2.XMLTokenizer
```

Constructors

XMLTokenizer()

Description

Creates a new Tokenizer object.

Syntax

```
public XMLTokenizer()
```

XMLTokenizer(XMLToken)

Description

Creates a new Tokenizer object.

Syntax

```
public XMLTokenizer(XMLToken handler)
```

Methods

parseDocument()

Description

Document ::= Prolog Element Misc*

Syntax

```
public void parseDocument()
```

setErrorHandler(ErrorHandler)**Description**

Applications can use this to register a new error event handler. This replaces any previous setting for error handling.

Syntax

```
public void setErrorHandler(org.xml.sax.ErrorHandler handler)
```

Parameters

handler - ErrorHandler being registered

setErrorStream(OutputStream)**Description**

Register a output stream for errors

Syntax

```
public void setErrorStream(java.io.OutputStream out)
```

setToken(int, boolean)**Description**

Applications can use this to register a new token for XML tokenizer.

Syntax

```
public void setToken(int token, boolean val)
```

Parameters

token - XMLToken being set

setTokenHandler(XMLToken)**Description**

Applications can use this to register a new XML tokenizer event handler.

Syntax

```
public void setTokenHandler(XMLToken handler)
```

Parameters

handler - XMLToken being registered

tokenize(InputSource)

Description

Tokenizes the XML from given input source

Syntax

```
public final void tokenize(org.xml.sax.InputSource in)
```

Parameters

in - the org.xml.sax.InputSouce to parse

Throws

XMLParseException - if syntax or other error encountered.

SAXException - Any SAX exception, possibly wrapping another exception.

IOException - IO Error.

tokenize(InputStream)

Description

Tokenizes the XML from given input stream.

Syntax

```
public final void tokenize(java.io.InputStream in)
```

Parameters

in - the InputStream containing XML data to parse.

Throws

XMLParseException - if syntax or other error encountered.

SAXException - Any SAX exception, possibly wrapping another exception.

IOException - IO Error.

See Also`XMLParser.setBaseURL(URL)`**tokenize(Reader)****Description**

Tokenizes the XML from given input stream.

Syntax

```
public final void tokenize(java.io.Reader r)
```

Parameters

`r` - the Reader containing XML data to parse.

Throws

`XMLParseException` - if syntax or other error encountered.

`SAXException` - Any SAX exception, possibly wrapping another exception.

`IOException` - IO Error.

See Also`XMLParser.setBaseURL(URL)`**tokenize(String)****Description**

Tokenizes the XML from the URL indicated

Syntax

```
public final void tokenize(java.lang.String in)
```

Parameters

`in` - the String containing the URL to parse from

Throws

`XMLParseException` - if syntax or other error encountered.

`SAXException` - Any SAX exception, possibly wrapping another exception.

`IOException` - IO Error.

tokenize(URL)

Description

Tokenizes the XML document pointed to by the given URL and creates the corresponding XML document hierarchy.

Syntax

```
public final void tokenize(java.net.URL url)
```

Parameters

`url` - the url points to the XML document to parse.

Throws

`XMLParseException` - if syntax or other error encountered.

`SAXException` - Any SAX exception, possibly wrapping another exception.

`IOException` - IO Error.

JXDocumentBuilder

Description

Defines the API to obtain DOM Document instances from an XML document. Using this class, an application programmer can obtain a `org.w3c.dom.Document` from XML.

Syntax

```
public class JXDocumentBuilder
```

```
oracle.xml.jaxp.JXDocumentBuilder
```

Comments

An instance of this class can be obtained from the `DocumentBuilderFactory.newDocumentBuilder` method. Once an instance of this class is obtained, XML can be parsed from a variety of input sources. These input sources are `InputStreams`, `Files`, `URLs`, and `SAX InputSources`.

Note that this class reuses several classes from the SAX API. This does not require that the implementor of the underlying DOM implementation use a SAX parser to parse XML document into a `Document`. It merely requires that the implementation communicate with the application using these existing APIs.

Since

JAXP 1.0

Methods

getDOMImplementation()

Description

The `DOMImplementation` object that handles this document. A DOM application may use objects from multiple implementations.

Syntax

```
public org.w3c.dom.DOMImplementation getDOMImplementation()
```

Returns

The associated DOM implementation.

isNamespaceAware()

Description

Indicates whether or not this parser is configured to understand namespaces.

Syntax

```
public boolean isNamespaceAware()
```

isValidating()

Description

Indicates whether or not this parser is configured to validate XML documents.

Syntax

```
public boolean isValidating()
```

newDocument()

Description

Obtain a new instance of a DOM Document object to build a DOM tree with.

Syntax

```
public org.w3c.dom.Document newDocument()
```

parse(InputSource)

Description

Parse the content of the given input source as an XML document and return a new DOM Document object.

Syntax

```
public org.w3c.dom.Document parse(org.xml.sax.InputSource is)
```

Parameters

is - InputSource containing the content to be parsed.

Throws

`IOException` - If any IO errors occur.

`SAXException` - If any parse errors occur.

`IllegalArgumentException` - If the `InputSource` is null

See Also

`org.xml.sax.DocumentHandler`

setEntityResolver(EntityResolver)**Description**

Specify the `EntityResolver` to be used to resolve entities present in the XML document to be parsed. Setting this to `null` will result in the underlying implementation using it's own default implementation and behavior.

Syntax

```
public void setEntityResolver(org.xml.sax.EntityResolver er)
```

setErrorHandler(ErrorHandler)**Description**

Specify the `ErrorHandler` to be used to resolve entities present in the XML document to be parsed. Setting this to `null` will result in the underlying implementation using it's own default implementation and behavior.

Syntax

```
public void setErrorHandler(org.xml.sax.ErrorHandler eh)
```

JXDocumentBuilderFactory

Description

Defines a factory API that enables applications to obtain a parser that produces DOM object trees from XML documents.

Syntax

```
public class JXDocumentBuilderFactory
```

```
oracle.xml.jaxp.JXDocumentBuilderFactory
```

Since

JAXP 1.0

Fields of JXDocumentBuilderFactory

Table 11–23 *Fields of JXDocumentBuilderFactory*

Field	Syntax	Description
BASE_URL	public static final java.lang.String BASE_URL	Base URL used in parsing entities.
DEBUG_MODE	public static final java.lang.String DEBUG_ MODE	Sets Debug Mode - Boolean.TRUE or Boolean.FALSE
DTD_OBJECT	public static final java.lang.String DTD_OBJECT	DTD Object to be used for validation.
ERROR_ENCODING	public static final java.lang.String ERROR_ ENCODING	Encoding for errors report via error stream (only if ERROR_STREAM is set)
ERROR_STREAM	public static final java.lang.String ERROR_ STREAM	Error stream for reporting errors. The object can be OutputStream or PrintWriter. This attribute is ignored if ErrorHandler is set.
NODE_FACTORY	public static final java.lang.String NODE_ FACTORY	Set NodeFactory to build custom Nodes
SCHEMA_OBJECT	public static final java.lang.String SCHEMA_ OBJECT	Schema Object to be used for validation.

Table 11–23 Fields of JXDocumentBuilderFactory

Field	Syntax	Description
SHOW_WARNINGS	public static final java.lang.String SHOW_WARNINGS	Boolean to ignore warnings - Boolean.TRUE or Boolean.FALSE
USE_DTD_ONLY_FOR_VALIDATION	public static final java.lang.String USE_DTD_ONLY_FOR_VALIDATION	If true, DTD Object is used only for validation and is not added to the parser document

Constructors

JXDocumentBuilderFactory()

Description

Default Constructor

Syntax

```
public JXDocumentBuilderFactory()
```

Methods

getAttribute(String)

Description

Allows the user to retrieve specific attributes on the underlying implementation.

Syntax

```
public java.lang.Object getAttribute(java.lang.String name)
```

Parameters

name - The name of the attribute.

Returns

value The value of the attribute.

Throws

`IllegalArgumentException` - thrown if the underlying implementation doesn't recognize the attribute.

isExpandEntityReferences()

Description

Indicates whether or not the factory is configured to produce parsers which expand entity reference nodes. Always return TRUE - Currently there is no way to prevent entity references expansions

Syntax

```
public boolean isExpandEntityReferences()
```

Returns

boolean

isIgnoringComments()

Description

Indicates whether or not the factory is configured to produce parsers which ignores comments. Always returns false - currently ignoring comments is not configurable

Syntax

```
public boolean isIgnoringComments()
```

Returns

boolean

isNamespaceAware()

Description

Indicates whether or not the factory is configured to produce parsers which are namespace aware. Always returns TRUE - Currently there is no way to turn of Namespaces

Syntax

```
public boolean isNamespaceAware()
```

Returns

boolean namespace awareness.

newDocumentBuilder()**Description**

Creates a new instance of a `DocumentBuilder` using the currently configured parameters.

Syntax

```
public javax.xml.parsers.DocumentBuilder newDocumentBuilder()
```

Throws

`ParserConfigurationException` - if a `DocumentBuilder` cannot be created which satisfies the configuration requested

setAttribute(String, Object)**Description**

Allows the user to set specific attributes on the underlying implementation.

Syntax

```
public void setAttribute(java.lang.String name, java.lang.Object value)
```

Parameters

`name` - The name of the attribute.

`value` - The value of the attribute.

Throws

`IllegalArgumentException` - thrown if the underlying implementation doesn't recognize the attribute.

JXSAXParser

Description

Defines the API that wraps an `org.xml.sax.XMLReader` implementation class. In JAXP 1.0, this class wrapped the `org.xml.sax.Parser` interface, however this interface was replaced by the `XMLReader`.

Syntax

```
public class JXSAXParser
```

```
oracle.xml.jaxp.JXSAXParser
```

Comments

For ease of transition, this class continues to support the same name and interface as well as supporting new methods. An instance of this class can be obtained from the `SAXParserFactory.newSAXParser` method. Once an instance of this class is obtained, XML can be parsed from a variety of input sources. These input sources are `InputStreams`, `Files`, `URLs`, and `SAX InputSources`.

This static method creates a new factory instance based on a system property setting or uses the platform default if no property has been defined.

The system property that controls which Factory implementation to create is named "javax.xml.style.TransformFactory". This property names a class that is a concrete subclass of this abstract class. If no property is defined, a platform default will be used.

As the content is parsed by the underlying parser, methods of the given `HandlerBase` are called.

Since

JAXP 1.0

Methods

getParser()

Description

Returns the SAX parser that is encapsulated by the implementation of this class.

Syntax

```
public org.xml.sax.Parser getParser()
```

Comments

Deprecated, use `getXMLReader()`

getProperty(String)**Description**

returns the particular property requested for in the underlying implementation of `org.xml.sax.XMLReader`.

Syntax

```
public java.lang.Object getProperty(java.lang.String name)
```

Parameters

`name` - The name of the property to be retrieved.

Returns

Value of the requested property.

Throws

`SAXNotRecognizedException` - When the underlying `XMLReader` does not recognize the property name.

`SAXNotSupportedException` - When the underlying `XMLReader` recognizes the property name but doesn't support the property.

See Also

`org.xml.sax.XMLReader#getProperty`

getXMLReader()**Description**

Returns the `XMLReader` that is encapsulated by the implementation of this class.

Syntax

```
public org.xml.sax.XMLReader getXMLReader()
```

isNamespaceAware()

Description

Indicates whether or not this parser is configured to understand namespaces.

Syntax

```
public boolean isNamespaceAware()
```

isValidating()

Description

Indicates whether or not this parser is configured to validate XML documents.

Syntax

```
public boolean isValidating()
```

setProperty(String, Object)

Description

Sets the particular property in the underlying implementation of `org.xml.sax.XMLReader`.

Syntax

```
public void setProperty(java.lang.String name, java.lang.Object value)
```

Parameters

`name` - The name of the property to be set.

`value` - The value of the property to be set.

Throws

`SAXNotRecognizedException` - When the underlying `XMLReader` does not recognize the property name.

`SAXNotSupportedException` - When the underlying `XMLReader` recognizes the property name but doesn't support the property.

See Also

`org.xml.sax.XMLReader#setProperty`

JXSAXParserFactory

Description

Defines a factory API that enables applications to configure and obtain a SAX based parser to parse XML documents.

Syntax

```
public class JXSAXParserFactory
```

```
oracle.xml.jaxp.JXSAXParserFactory
```

Since

JAXP 1.0

Constructors

JXSAXParserFactory()

Description

Default Constructor

Syntax

```
public JXSAXParserFactory()
```

Methods

getFeature(String)

Description

Returns the particular property requested for in the underlying implementation of `org.xml.sax.XMLReader`.

Syntax

```
public boolean getFeature(java.lang.String name)
```

Parameters

name - The name of the property to be retrieved.

Returns

Value of the requested property.

Throws

`SAXNotRecognizedException` - When the underlying `XMLReader` does not recognize the property name.

`SAXNotSupportedException` - When the underlying `XMLReader` recognizes the property name but doesn't support the property.

See Also

`org.xml.sax.XMLReader#getProperty`

isNamespaceAware()

Description

Indicates whether or not the factory is configured to produce parsers which are namespace aware.

Syntax

```
public boolean isNamespaceAware()
```

newSAXParser()

Description

Creates a new instance of a `SAXParser` using the currently configured factory parameters.

Syntax

```
public javax.xml.parsers.SAXParser newSAXParser()
```

Throws

`ParserConfigurationException` - if a parser cannot be created which satisfies the requested configuration.

setFeature(String, boolean)

Description

Sets the particular feature in the underlying implementation of `org.xml.sax.XMLReader`.

Syntax

```
public void setFeature(java.lang.String name, boolean value)
```

Parameters

`name` - The name of the feature to be set.

`value` - The value of the feature to be set.

Throws

`SAXNotRecognizedException` - When the underlying `XMLReader` does not recognize the property name.

`SAXNotSupportedException` - When the underlying `XMLReader` recognizes the property name but doesn't support the property.

See Also

`org.xml.sax.XMLReader#setFeature`

JXSAXTransformerFactory

Description

A JXTransformerFactory instance can be used to create Transformer and Templates objects.

Syntax

```
public class JXSAXTransformerFactory
```

```
oracle.xml.jaxp.JXSAXTransformerFactory
```

Comments

The system property that determines which Factory implementation to create is named "javax.xml.transform.TransformerFactory". This property names a concrete subclass of the TransformerFactory abstract class (in our case, it is JXSAXTransformerFactory). If the property is not defined, a platform default is be used.

This class also provides SAX-specific factory methods. It provides two types of ContentHandlers, one for creating Transformers, the other for creating Templates objects.

If an application wants to set the ErrorHandler or EntityResolver for an XMLReader used during a transformation, it should use a URIResolver to return the SAXSource which provides (with getXMLReader) a reference to the XMLReader.

Constructors

JXSAXTransformerFactory()

Description

The default constructor

Syntax

```
public JXSAXTransformerFactory()
```

Methods

getAssociatedStylesheet(Source, String, String, String)

Description

Get the stylesheet specification(s) associated via the xml-stylesheet processing instruction (see <http://www.w3.org/TR/xml-stylesheet/>) with the document document specified in the source parameter, and that match the given criteria.

Syntax

```
public javax.xml.transform.Source  
getAssociatedStylesheet(javax.xml.transform.Source source, java.lang.String  
media, java.lang.String title, java.lang.String charset)
```

Comments

Note that it is possible to return several stylesheets, in which case they are applied as if they were a list of imports or cascades in a single stylesheet.

Parameters

`source` - The XML source document.

`media` - The media attribute to be matched. May be null, in which case the preferred templates will be used (i.e. `alternate = no`).

`title` - The value of the title attribute to match. May be null.

`charset` - The value of the charset attribute to match. May be null.

Returns

A Source object suitable for passing to the TransformerFactory.

getAttribute(String)

Description

Allows the user to retrieve specific attributes on the underlying implementation.

Syntax

```
public java.lang.Object getAttribute(java.lang.String name)
```

Parameters

`name` - The name of the attribute.

Returns

value The value of the attribute.

Throws

`IllegalArgumentException` - thrown if the underlying implementation doesn't recognize the attribute.

getErrorListener()

Description

Get the error event handler for the `TransformerFactory`.

Syntax

```
public javax.xml.transform.ErrorListener getErrorListener()
```

Returns

The current error handler, which should never be null.

getFeature(String)

Description

Look up the value of a feature. The feature name is any absolute URI.

Syntax

```
public boolean getFeature(java.lang.String name)
```

Parameters

name - The feature name, which is an absolute URI.

Returns

The current state of the feature (true or false).

getURIResolver()

Description

Get the object that is used by default during the transformation to resolve URIs used in `document()`, `xsl:import`, or `xsl:include`.

Syntax

```
public javax.xml.transform.URIResolver getURIResolver()
```

Returns

The URIResolver that was set with setURIResolver.

newTemplates(Source)**Description**

Process the Source into a Templates object, which is a compiled representation of the source. This Templates object may then be used concurrently across multiple threads.

Syntax

```
public javax.xml.transform.Templates newTemplates(javax.xml.transform.Source source)
```

Comments

Creating a Templates object allows the TransformerFactory to do detailed performance optimization of transformation instructions, without penalizing runtime transformation.

Parameters

`source` - An object that holds a URL, input stream, etc.

Returns

A Templates object capable of being used for transformation purposes, never null.

Throws

`TransformerConfigurationException` - May throw this during the parse when it is constructing the Templates object and fails.

newTemplatesHandler()**Description**

Get a TemplatesHandler object that can process SAX ContentHandler events into a Templates object.

Syntax

```
public javax.xml.transform.sax.TemplatesHandler newTemplatesHandler()
```

Returns

A non-null reference to a TransformerHandler, that may be used as a ContentHandler for SAX parse events.

Throws

TransformerConfigurationException - If for some reason the TemplatesHandler cannot be created.

newTransformer()**Description**

Create a new Transformer object that performs a copy of the source to the result.

Syntax

```
public javax.xml.transform.Transformer newTransformer()
```

Parameters

source - An object that holds a URI, input stream, etc.

Returns

A Transformer object that may be used to perform a transformation in a single thread, never null.

Throws

TransformerConfigurationException - May throw this during the parse when it is constructing the Templates object and fails.

newTransformer(Source)**Description**

Process the Source into a Transformer object. Care must be given not to use this object in multiple threads running concurrently. Different TransformerFactories can be used concurrently by different threads.

Syntax

```
public javax.xml.transform.Transformer newTransformer(javax.xml.transform.Source
```

source)

Parameters

`source` - An object that holds a URI, input stream, etc.

Returns

A Transformer object that may be used to perform a transformation in a single thread, never null.

Throws

`TransformerConfigurationException` - May throw this during the parse when it is constructing the Templates object and fails.

newTransformerHandler()

Description

Get a TransformerHandler object that can process SAX ContentHandler events into a Result. The transformation is defined as an identity (or copy) transformation, for example to copy a series of SAX parse events into a DOM tree.

Syntax

```
public javax.xml.transform.sax.TransformerHandler newTransformerHandler()
```

Returns

A non-null reference to a TransformerHandler, that may be used as a ContentHandler for SAX parse events.

Throws

`TransformerConfigurationException` - If for some reason the TransformerHandler cannot be created.

newTransformerHandler(Source)

Description

Get a TransformerHandler object that can process SAX ContentHandler events into a Result, based on the transformation instructions specified by the argument.

Syntax

```
public javax.xml.transform.sax.TransformerHandler
```

```
newTransformerHandler( javax.xml.transform.Source src)
```

Parameters

`src` - The Source of the transformation instructions.

Returns

TransformerHandler ready to transform SAX events.

Throws

TransformerConfigurationException - If for some reason the TransformerHandler can not be created.

newTransformerHandler(Templates)

Description

Get a TransformerHandler object that can process SAX ContentHandler events into a Result, based on the Templates argument.

Syntax

```
public javax.xml.transform.sax.TransformerHandler  
newTransformerHandler( javax.xml.transform.Templates templates)
```

Parameters

`templates` - The compiled transformation instructions.

Returns

TransformerHandler ready to transform SAX events.

Throws

TransformerConfigurationException - If for some reason the TransformerHandler can not be created.

newXMLFilter(Source)

Description

Create an XMLFilter that uses the given Source as the transformation instructions.

Syntax

```
public org.xml.sax.XMLFilter newXMLFilter( javax.xml.transform.Source src)
```

Parameters

`src` - The Source of the transformation instructions.

Returns

An XMLFilter object, or null if this feature is not supported.

Throws

`TransformerConfigurationException` - If for some reason the `TemplatesHandler` cannot be created.

newXMLFilter(Templates)**Description**

Create an XMLFilter, based on the `Templates` argument.

Syntax

```
public org.xml.sax.XMLFilter newXMLFilter(javax.xml.transform.Templates templates)
```

Parameters

`templates` - The compiled transformation instructions.

Returns

An XMLFilter object, or null if this feature is not supported.

Throws

`TransformerConfigurationException` - If for some reason the `TemplatesHandler` cannot be created.

setAttribute(String, Object)**Description**

Allows the user to set specific attributes on the underlying implementation. An attribute in this context is defined to be an option that the implementation provides.

Syntax

```
public void setAttribute(java.lang.String name, java.lang.Object value)
```

Parameters

`name` - The name of the attribute.

`value` - The value of the attribute.

Throws

`IllegalArgumentException` - thrown if the underlying implementation doesn't recognize the attribute.

setErrorListener(ErrorListener)

Description

Set the error event listener for the `TransformerFactory`, which is used for the processing of transformation instructions, and not for the transformation itself.

Syntax

```
public void setErrorListener(javax.xml.transform.ErrorListener listener)
```

Parameters

`listener` - The new error listener.

Throws

`IllegalArgumentException` - if `listener` is null.

setURIResolver(URIResolver)

Description

Set an object that is used by default during the transformation to resolve URIs used in `xsl:import`, or `xsl:include`.

Syntax

```
public void setURIResolver(javax.xml.transform.URIResolver resolver)
```

Parameters

`resolver` - An object that implements the `URIResolver` interface, or null.

JXTransformer

Description

An instance of this class can transform a source tree into a result tree.

Syntax

```
public class JXTransformer
```

```
oracle.xml.jaxp.JXTransformer
```

Comments

An instance of this class can be obtained with the `TransformerFactory.newTransformer` method. This instance may then be used to process XML from a variety of sources and write the transformation output to a variety of sinks.

An object of this class may not be used in multiple threads running concurrently. Different Transformers may be used concurrently by different threads.

A Transformer may be used multiple times. Parameters and output properties are preserved across transformations.

Constructors

JXTransformer()

Description

Constructs a JXTransformer object that uses the XSLStylesheet to transform the source.

Syntax

```
public JXTransformer()
```

JXTransformer(XSLStylesheet)

Description

Constructor is based on

Syntax

```
public JXTransformer(oracle.xml.parser.v2.XSLStyleSheet templates)
```

Parameters

templates - a XSLStyleSheet or Templates

Methods

clearParameters()

Description

Clear all parameters set with setParameter.

Syntax

```
public void clearParameters()
```

getErrorListener()

Description

Get the error event handler in effect for the transformation.

Syntax

```
public javax.xml.transform.ErrorListener getErrorListener()
```

Returns

The current error handler, which should never be null.

getOutputProperties()

Description

Get a copy of the output properties for the transformation.

Syntax

```
public java.util.Properties getOutputProperties()
```

Comments

The properties returned should contain properties set by the user, and properties set by the stylesheet, and these properties are "defaulted" by default properties

specified by section 16 of the XSL Transformations (XSLT) W3C Recommendation. The properties that were specifically set by the user or the stylesheet should be in the base Properties list, while the XSLT default properties that were not specifically set should be the default Properties list. Thus, `getOutputProperties().getProperty(String key)` will obtain any property in that was set by `setOutputProperty(String, String)`, `setOutputProperties(Properties)`, in the stylesheet, or the default properties, while `getOutputProperties().get(String key)` will only retrieve properties that were explicitly set by `setOutputProperty(String, String)`, `setOutputProperties(Properties)`, or in the stylesheet.

Note that mutation of the Properties object returned will not effect the properties that the transformation contains.

If any of the argument keys are not recognized and are not namespace qualified, the property will be ignored. In other words the behaviour is not orthogonal with `setOutputProperties`.

See Also

`javax.xml.transform.OutputKeys`, `java.util.Properties`

getOutputProperty(String)

Description

Get an output property that is in effect for the transformation.

Syntax

```
public java.lang.String getOutputProperty(java.lang.String name)
```

Comments

The property specified may be a property that was set with `setOutputProperty`, or it may be a property specified in the stylesheet.

Parameters

`name` - A non-null String that specifies an output property name, which may be namespace qualified.

Returns

The string value of the output property, or null if no property was found.

Throws

`IllegalArgumentException` - If the property is not supported.

See Also

`javax.xml.transform.OutputKeys`

getParameter(String)

Description

Get a parameter that was explicitly set with `setParameter` or `setParameters`.

Syntax

```
public java.lang.Object getParameter(java.lang.String name)
```

Parameters

`name` - A parameter name

This method does not return a default parameter value, which cannot be determined until the node context is evaluated during the transformation process.

Returns

A parameter that has been set with `setParameter`, or null if a parameter with the given name was not found.

getURIResolver()

Description

Get an object that will be used to resolve URIs used in `document()`, etc.

Syntax

```
public javax.xml.transform.URIResolver getURIResolver()
```

Returns

An object that implements the `URIResolver` interface, or null.

setErrorListener(ErrorListener)

Description

Set the error event listener in effect for the transformation.

Syntax

```
public void setErrorListener(javax.xml.transform.ErrorListener listener)
```

Parameters

`listener` - The new error listener.

Throws

`IllegalArgumentException` - if listener is null.

setOutputProperties(Properties)**Description**

Set the output properties for the transformation. These properties will override properties set in the Templates with `xsl:output`.

Syntax

```
public void setOutputProperties(java.util.Properties oformat)
```

Comments

If argument to this function is null, any properties previously set are removed, and the value will revert to the value defined in the templates object.

Pass a qualified property key name as a two-part string, the namespace URI enclosed in curly braces (`{}`), followed by the local name. If the name has a null URL, the String only contain the local name. An application can safely check for a non-null URI by testing to see if the first character of the name is a `'{'` character.

For example, if a URI and local name were obtained from an element defined with `<xyz:foo xmlns:xyz="http://xyz.foo.com/yada/baz.html"/>`, then the qualified name would be `"{http://xyz.foo.com/yada/baz.html}foo"`. Note that no prefix is used.

Parameters

`oformat` - A set of output properties that will be used to override any of the same properties in affect for the transformation.

Throws

`IllegalArgumentException` - if any of the argument keys are not recognized and are not namespace qualified.

See Also

`javax.xml.transform.OutputKeys`, `java.util.Properties`

setOutputProperty(String, String)**Description**

Set an output property that will be in effect for the transformation.

Syntax

```
public void setOutputProperty(java.lang.String name, java.lang.String value)
```

Comments

Pass a qualified property name as a two-part string, the namespace URI enclosed in curly braces ({}), followed by the local name. If the name has a null URL, the String only contain the local name. An application can safely check for a non-null URI by testing to see if the first character of the name is a '{' character.

For example, if a URI and local name were obtained from an element defined with `<xyz:foo xmlns:xyz="http://xyz.foo.com/yada/baz.html"/>`, then the qualified name would be `"{http://xyz.foo.com/yada/baz.html}foo"`. Note that no prefix is used.

The `Properties` object that was passed to `setOutputProperties(Properties)` won't be effected by calling this method.

Parameters

`name` - A non-null String that specifies an output property name, which may be namespace qualified.

`value` - The non-null string value of the output property.

Throws

`IllegalArgumentException` - If the property is not supported, and is not qualified with a namespace.

See Also

`javax.xml.transform.OutputKeys`

setParameter(String, Object)

Description

Add a parameter for the transformation.

Syntax

```
public void setParameter(java.lang.String name, java.lang.Object value)
```

Comments

Pass a qualified name as a two-part string, the namespace URI enclosed in curly braces ({}), followed by the local name. If the name has a null URL, the String only contain the local name. An application can safely check for a non-null URI by testing to see if the first character of the name is a '{' character.

For example, if a URI and local name were obtained from an element defined with `<xyz:foo xmlns:xyz="http://xyz.foo.com/yada/baz.html"/>`, then the qualified name would be "{http://xyz.foo.com/yada/baz.html}foo". Note that no prefix is used.

Parameters

`name` - The name of the parameter, which may begin with a namespace URI in curly braces ({}).

`value` - The value object. This can be any valid Java object. It is up to the processor to provide the proper object coercion or to simply pass the object on for use in an extension.

setURIResolver(URIResolver)

Description

Set an object that will be used to resolve URIs used in `document()`. currently, we do not support URIResolver in `document()` fuction

Syntax

```
public void setURIResolver(javax.xml.transform.URIResolver resolver)
```

Comments

If the resolver argument is null, the URIResolver value will be cleared, and the default behavior will be used.

Parameters

`resolver` - An object that implements the `URIResolver` interface, or null.

transform(Source, Result)

Description

Process the source tree to the output result.

Syntax

```
public void transform(javax.xml.transform.Source xmlSource,  
    javax.xml.transform.Result outputTarget)
```

Parameters

`xmlSource` - The input for the source tree.

`outputTarget` - The output target.

Throws

`TransformerException` - If an unrecoverable error occurs during the course of the transformation.

XSLT Processor Classes

The classes listed in [Table 11-24, "Summary of XSLT Processor Classes"](#) summarizes the XSLT Processor classes contained in the *oracle.xml.parser.v2* package.

Table 11-24 Summary of XSLT Processor Classes

Class	Description
oraxsl Class	Provides a command-line interface to applying stylesheets on multiple XML documents.
XPathException Class	Indicates that an exception occurred during XSL transformation.
XSLProcessor Class	Provides methods to transform an input XML document using a previously constructed XSLStylesheet.
XSLStylesheet Class	Holds XSL stylesheet information such as templates, keys, variables, and attribute sets.

oraxsl Class

Description of oraxsl

The oraxsl class provides a command-line interface to applying stylesheets on multiple XML documents. It accepts a number of command-line options that dictate how it should behave.

Syntax of oraxsl

```
public class oraxsl extends java.lang.Object
```

```
java.lang.Object
```

```
|
```

```
+--oracle.xml.parser.v2.oraxsl
```

Usage of oraxsl

```
java oraxsl options* source? stylesheet? result?
```

Table 11–25 *Command-line options of oraxsl*

command	description
-w	Show warnings
-e <error log>	A file to write errors to
-l <xml file list>	List of files to transform
-d <directory>	Directory with files to transform
-x <source extension>	Extensions to exclude
-i <source extension>	Extensions to include
-s <stylesheet>	Stylesheet to use
-r <result extension>	Extension to use for results
-o <result extension>	Directory to place results
-p <param list>	List of Params
-t <# of threads>	Number of threads to use
-v	Verbose mode

Methods of oraxsl

Table 11–26 Summary of Methods of oraxsl

Method	Description
oraxsl() , on page 11-307	Class constructor
main() , on page 11-307	Invokes the oraxsl driver

oraxsl()

Description

Class constructor

Syntax

```
public oraxsl()
```

main()

Description

Invokes the oraxsl driver

Syntax

```
public static void main(java.lang.String[] args)
```

Parameters

Table 11–27 Parameters of main()

Parameter	Description
args	Command line arguments

XPathException Class

Description

Indicates that an exception occurred during XPath processing.

Syntax

```
public class XPathException extends oracle.xml.parser.v2.XSLException
```

```
java.lang.Object
|
+--java.lang.Throwable
    |
    +--java.lang.Exception
        |
        +--oracle.xml.util.XMLException
            |
            +--oracle.xml.parser.v2.XSLException
                |
                +--oracle.xml.parser.v2.XPathException
```

Implemented Interfaces

```
java.io.Serializable
```

Methods

getErrorID()

Description

Syntax

```
public int getErrorID()
```

getMessage()

Description

Override getMessage, in order to construct error message from error id, and error params

Syntax

```
public java.lang.String getMessage()
```

Overrides

java.lang.Throwable.getMessage() in class java.lang.Throwable

getMessage(XMLError)**Description**

Get localized message based on the XMLError sent as parameter

Syntax

```
public java.lang.String getMessage(XMLError err)
```

Parameters

err - XMLError class used to get the error message

XSLEException Class

Description of XSLEException

Indicates that an exception occurred during XSL transformation

Syntax of XSLEException

```
public class XSLEException extends oracle.xml.util.XMLException
```

```
java.lang.Object
|
+--java.lang.Throwable
    |
    +--java.lang.Exception
        |
        +--oracle.xml.util.XMLException
            |
            +--oracle.xml.parser.v2.XSLEException
```

Direct Subclasses of XSLEException

XPathException

Implemented Interfaces of XSLEException

java.io.Serializable

Constructors of XSLEException

XSLEException()

Description

Syntax

```
public XSLEException(String mesg);
```

XSLExtensionElement Class

Description of XSLExtensionElement

Base element for extension elements

Syntax of XSLExtensionElement

```
public class XSLExtensionElement
```

```
oracle.xml.parser.v2.XSLExtensionElement
```

Methods of XSLExtensionElement

Table 11–28 Summary of Methods of XSLExtensionElement

Method	Description
XSLExtensionElement()	Default Constructor
getAttributeTemplateValue()	Get an attribute value as template
getAttributeValue()	Get an attribute value
getChildNodes()	Gets the childNodes of the extension elements
processAction()	Function called to execute the body of the extension elements
processContent()	Process contents of the extension element

XSLExtensionElement()

Description

Default Constructor

Syntax

```
public XSLExtensionElement()
```

getAttributeTemplateValue()

Description

Get an attribute value as template

Syntax

```
protected final String getAttributeTemplateValue(  
    XSLTContext context, String namespace, String name)
```

Parameters**Table 11–29 Parameters of GetAttributeTempateValue**

Parameter	Description
context	XSLTContext
namespace	namespace of the attribute
name	name of the attribute

Returns

value of the attribute

getAttributeValue()**Description**

Get an attribute value

Syntax

```
protected final String getAttributeValue(String namespace, String name);
```

Parameters**Table 11–30 Parameters of GetAttributeValue**

Parameter	Description
namespace	namespace of the attribute
name	name of the attribute

Returns

value of the attribute

getChildNodes()

Description

Gets the childNodes of the extension elements

Syntax

```
protected final java.util.Vector getChildNodes();
```

Returns

nodelist

processAction()

Description

Function called to execute the body of the extension elements

Syntax

```
public void processAction(XSLTContext context);
```

Parameters

Table 11–31 Parameters of ProcessAction

Parameter	Description
context	XSLTContext

processContent()

Description

Process contents of the extension element

Syntax

```
protected final void processContent(XSLTContext context);
```

Parameters

Table 11–32 Parameters of ProcessContent

Parameter	Description
context	XSLTContext

XSLProcessor Class

Description of XSLProcessor

This class provides methods to transform an input XML document using a previously constructed `XSLStyleSheet`. The transformation effected is as specified by the XSLT 1.0 specification.

Syntax of XSLProcessor

```
public class XSLProcessor
```

```
oracle.xml.parser.v2.XSLProcessor
```

Methods of XSLProcessor

Table 11–33 Summary of Methods of XSLProcessor

Method	Description
<code>XSLProcessor()</code>	Default constructor
<code>getParam()</code>	
<code>newXSLStyleSheet()</code>	Constructs an <code>XSLStyleSheet</code> .
<code>processXSL()</code>	Transforms input XML document
<code>removeParam()</code>	Removes the value of a top-level stylesheet parameter.
<code>resetParams()</code>	Resets all the params set.
<code>setBaseURL()</code>	Set base url to resolve include/import hrefs.
<code>setEntityResolver()</code>	Set entity resolver to resolve include/import hrefs.
<code>setErrorStream()</code>	Creates an output stream for the output of warnings.
<code>setLocale()</code>	Applications can use this to set the locale for error reporting.
<code>setParam()</code>	Sets the value of a top-level stylesheet parameter.
<code>showWarnings()</code>	Set the overriding <code>XSLOutput</code> object.
<code>showWarnings()</code>	witch to determine whether to output warnings

XSLProcessor()

Description

Default Constructor

Syntax

```
public XSLProcessor();
```

getParam()

Description

Gets the value of top-level stylesheet parameter.

Syntax

```
public java.lang.Object getParam(String uri, String name)
```

Parameters

`uri` - namespace URI of the parameter

`name` - local name of the parameter

Returns

The value of the parameter.

newXSLStylesheet()

Description

Constructs an XSLStylesheet. XSL function document("") is not supported as there is no way to re-access the input Stylesheet as XMLDocument.

Table 11–34 *Versions of newXSLStylesheet()*

Syntax	Description
<code>public XSLStylesheet newXSLStylesheet(InputStream xsl);</code>	Constructs an XSLStylesheet using the given Inputstream XSL
<code>public XSLStylesheet newXSLStylesheet(Reader xsl);</code>	Constructs an XSLStylesheet using the given Reader
<code>public XSLStylesheet newXSLStylesheet(java.net.URL xsl);</code>	Constructs an XSLStylesheet using the given URL

Table 11–34 *Versions of newXSLStylesheet() (Cont.)*

Syntax	Description
public XSLStylesheet newXSLStylesheet(XMLDocument xsl);	Constructs an XSLStylesheet using the given XMLDocument

Parameters**Table 11–35** *Parameters of newXSLStylesheet()*

Parameter	Description
xsl	XSL input

Returns

New XSL Stylesheet

Throws

XSLException on error.

processXSL()**Description**Transforms input XML document. The options are described in [Table 11–36](#):**Table 11–36** *Versions of processXSL()*

Syntax	Description
public XMLDocumentFragment processXSL(XSLStylesheet xsl, InputStream xml, URL ref);	Transform input XML document using given InputStream and stylesheet.
public XMLDocumentFragment processXSL(XSLStylesheet xsl, Reader xml, URL ref);	Transform input XML document using given Reader and stylesheet.
public XMLDocumentFragment processXSL(XSLStylesheet xsl, URL xml, URL ref)	Transform input XML document using given URL and stylesheet.
public XMLDocumentFragment processXSL(XSLStylesheet xsl, XMLDocument xml)	Transform input XML document using given XMLDocument and stylesheet.

Table 11–36 Versions of processXSL() (Cont.)

Syntax	Description
public void processXSL(XSLStylesheet xsl, XMLDocument xml, org.xml.sax.ContentHandler handler)	Transform input XML document using given XMLDocument and stylesheet.
public XMLDocumentFragment processXSL(XSLStylesheet xsl, XMLDocumentFragment xml)	Transform input XML document using given XMLDocument and stylesheet.
public void processXSL(XSLStylesheet xsl, XMLDocumentFragment xml, OutputStream os)	Transform input XML using given XMLDocumentFragment and stylesheet.
public void processXSL(XSLStylesheet xsl, XMLDocumentFragment xml, PrintWriter pw)	Transform input XML using given XMLDocumentFragment and stylesheet.
public void processXSL(XSLStylesheet xsl, XMLDocumentFragment xml, XMLDocumentHandler handlerXML)	Transform input XML document using given XMLDocument and stylesheet. As the result of XSLT is a document fragment, the following functions in XMLDocumentHandler will not be called: - setDocumentLocator, startDocument, endDocument, - setDoctype, endDoctype, setXMLDecl, setTextDecl
public void processXSL(XSLStylesheet xsl, XMLDocument xml, OutputStream out)	Transform input XML document using given XMLDocument and stylesheet.
public void processXSL(XSLStylesheet xsl, XMLDocument xml, java.io.PrintWriter pw)	Transform input XML document using given XMLDocument and stylesheet.
public void processXSL(XSLStylesheet xsl, XMLDocument xml, XMLDocumentHandler handlerXML)	Transform input XML document using given XMLDocument and stylesheet. The output of the transformation is reported through XMLDocumentHandler As the result of XSLT is a document fragment, the following functions in XMLDocumentHandler will not be called: - setDocumentLocator, startDocument, endDocument, - setDoctype, endDoctype, setXMLDecl, setTextDecl
public XMLDocumentFragment processXSL(XSLStylesheet xsl, XMLElement inp)	Transform input XML document using given XMLDocument and stylesheet.

Table 11–36 *Versions of processXSL() (Cont.)*

Syntax	Description
public void processXSL(XSLStyleSheet xsl, XMLElement inp, org.xml.sax.ContentHandler handler)	Transform input XML document using given XMLElement and stylesheet. The output of the transformation is reported through ContentHandler. As the result of XSLT is a document fragment, the following functions in ContentHandler will not be called: - setDocumentLocator, startDocument, endDocument,
public void processXSL(XSLStyleSheet xsl, XMLElement xml, OutputStream out)	Transform input XML using given XMLElement and stylesheet.
public void processXSL(XSLStyleSheet xsl, XMLElement xml, PrintWriter pw)	Transform input XML using given XMLElement and stylesheet
public void processXSL(XSLStyleSheet xsl, XMLElement xml, XMLDocumentHandler handlerXML)	Transform input XML document using given XMLElement and stylesheet. As the result of XSLT is a document fragment, the following functions in XMLDocumentHandler will not be called: - setDocumentLocator, startDocument, endDocument, - setDoctype, endDoctype, setXMLDecl, setTextDecl

Parameters

Table 11–37 *Parameters of processXSL*

Parameter	Description
xsl	XSLStyleSheet to be used for transformation
xml	XML input to be transformed
ref	Reference URL to resolve external entities in input xml file
handler	Content handler
out	Output stream to which the result is printed
pw	PrintWriter to which the result is printed
handlerXML	XMLDocument handler

Returns

XMLDocumentFragment or void, depending on form of the function.

Throws

XSLException on error.

removeParam()**Description**

Removes the value of a top-level stylesheet parameter.

Syntax

```
public void removeParam(String uri, String name)
```

Parameters

Table 11–38 Parameters of removeParam

Parameter	Description
uri	URI of parameter
name	parameter name

Throws

XSLException on error

resetParams()**Description**

Resets all the params set.

Syntax

```
public void resetParams()
```

Throws

XSLException on error

setBaseURL()**Description**

Set base url to resolve include/import hrefs. EntityResolver if set is used before using the base url. See also [setEntityResolver\(\)](#).

Syntax

```
public void setBaseURL(java.net.URL url)
```

Parameters**Table 11–39 Parameters of setBaseURL**

Parameter	Description
url	Base URL to be set

setEntityResolver()**Description**

Set entity resolver to resolve include/import hrefs. If not set, base url (if set) is used.

Syntax

```
public void setEntityResolver(org.xml.sax.EntityResolver eResolver)
```

Parameters**Table 11–40 Parameters of setEntityResolver**

Parameter	Description
eResolver	Entity resolver

setErrorStream()**Description**

Creates an output stream for the output of warnings. If an output stream for warnings is not specified, the processor will not output any warnings.

Syntax

```
public final void setErrorStream(java.io.OutputStream out)
```

Parameters

Table 11–41 *Parameters of setErrorStream*

Parameter	Description
out	The output stream to use for erros and warnings

setLocale()

Description

Applications can use this to set the locale for error reporting.

Syntax

```
public void setLocale(java.util.Locale locale)
```

Parameters

Table 11–42 *Parameters of setLocale*

Parameter	Description
locale	Locale to set

setParam()

Description

Sets the value of a top-level stylesheet parameter.

The parameter value is expected to be a valid XPath expression (note that string literal values would therefore have to be explicitly quoted). The param functions CANNOT be used along with param functions in XSLStylesheet. If the param functions in XSLProcessor are used, any parameters set using XSLStylesheet functions will be ignored.

Syntax

```
public void setParam(String uri, String name, Object value)
```

Parameters

Table 11–43 *Parameters of setParam*

Parameter	Description
uri	URI of parameter
name	Parameter name
value	Parameter value; Strings are treated as XPath Expr for backward compatibility)

Throws

XSLException on error

showWarnings()

Description

Switch to determine whether to output warnings.

Syntax

```
public final void showWarnings(boolean flag)
```

Parameters

Table 11–44 *Parameters of showWarnings*

Parameter	Description
flag	Determines whether warning should be shown; default: warnings not output

XSLStylesheet Class

Description of XSLStylesheet

The class holds XSL stylesheet information such as templates, keys, variables, and attribute sets. The same stylesheet, once constructed, can be used to transform multiple XML documents.

Syntax of XSLStylesheet

```
public class XSLStylesheet

oracle.xml.parser.v2.XSLStylesheet
```

Fields of XSLStylesheet

Table 11–45 *Fields of XSLStylesheet*

Field	Syntax	Description
output	public oracle.xml.parser.v2.XSLOutput	output

Methods of XSLStylesheet

Table 11–46 *Summary of Methods of XSLStylesheet*

Method	Description
getDecimalFormat()	Get the decimal format symbols specified in the stylesheet
getOutputEncoding()	Get the value of the encoding specified in <code>xsl:output</code>
getOutputMediaType()	Get the value of the media-type specified in <code>xsl:output</code>
getOutputProperties()	
newTransformer()	
XSLTContext Class	Removes the value of a top-level stylesheet parameter.
XSLTContext Class	Resets all the params set.
XSLTContext Class	Sets the value of a top-level stylesheet parameter.

getDecimalFormat()

Description

Get the decimal format symbols specified in the stylesheet

Syntax

```
public java.text.DecimalFormatSymbols getDecimalFormat(NSName nsname)
```

Parameters

Table 11–47 Parameters of *getDecimalFormat*

Parameter	Description
nsname	Qualified name from xsl:decimal-format

Returns

DecimalFormatSymbols

getOutputEncoding()

Description

Get the value of the encoding specified in `xsl:output`

Syntax

```
public java.lang.String getOutputEncoding()
```

Returns

The encoding

getOutputMediaType()

Description

Get the value of the media-type specified in `xsl:output`

Syntax

```
public java.lang.String getOutputMediaType()
```

Returns

the media type

getOutputProperties()

Description

Returns the output properties specified in `xsl:output` as `java.util.Properties`.

Syntax

```
public java.util.Properties getOutputProperties()
```

newTransformer()

Description

Returns a JAXP Transformer object that uses this stylesheet for transformation.

Syntax

```
public javax.xml.transform.Transformer newTransformer()
```

XSLTContext Class

Syntax of XSLTContext

```
public class XSLTContext extends java.lang.Object
```

```
java.lang.Object  
|  
+--oracle.xml.parser.v2.XSLTContext
```

Description of XSLTContext

Class for Xpath processing Context

Methods of XSLTContext

Table 11–48 Summary of Methods of XSLTContext

Method	Description
getContextNode()	Get the current context node
getContextPosition()	Get the current context node position
getContextSize()	Get the current context size
getError()	Get the XMLError instance for reporting errors
getVariable()	Retrieve variable at the given stack offset
reportCharacters()	Report characters to the current output handler
reportNode()	Report a XMLNode to the current output handler
setError()	Set the XMLError

getContextNode()

Description

Get the current context node

Syntax

```
public XMLNode getContextNode()
```

Returns

XMLNode current context node

getContextPosition()**Description**

Get the current context node position

Syntax

```
public int getContextPosition()
```

Returns

int current context node position

getContextSize()**Description**

Get the current context size

Syntax

```
public int getContextSize()
```

Returns

int current context size

getError()**Description**

Get the XMLError instance for reporting errors

Syntax

```
public XMLError getError()
```

Returns

XMLError

getVariable()

Description

Retrieve variable at the given stack offset

Syntax

```
public getVariable(NSName name, int offset)
```

Parameters

Table 11–49 Parameters of *getVariable*

Parameter	Description
name	name of the variable
offset	offset of the variable

reportCharacters()

Description

Report characters to the current output handler

Syntax

```
public void reportCharacters(String data, boolean disableoutesc)
```

Parameters

Table 11–50 Parameters of *reportCharcters*

Parameter	Description
chars	String to be printed
disableoutesc	Boolean to disable or enable escaping of characters as defined in the W3C.org XML 1.0 specification. TRUE means <i>disabled</i> , FALSE means <i>enabled</i> .

reportNode()

Description

Report a XMLNode to the current output handler

Syntax

```
public void reportNode(XMLNode node)
```

Parameters**Table 11–51** *Parameters of reportNode*

Parameter	Description
node	node to be output

setError()**Description**

Set the XMLError

Syntax

```
public void setError(XMLError err)
```

Parameters**Table 11–52** *Parameters of setError*

Parameter	Description
err	instance of XMLError

Part III

Java Packages for Oracle9i XDK for Java Beans

This part describes the Java packages that comprise the Oracle9i XDK for Java Beans. The Oracle XML Transviewer beans are provided as part of XDK for Java Beans with the Oracle9i Enterprise and Standard Editions. These are used for Java applications or applets to view and transform XML documents. As standard Java Beans, they can be used in any graphical Java development environment.

This part contains these chapters:

- Chapter 12, "Package oracle.xml.async"
- Chapter 13, "Package oracle.xml.dbviewer"
- Chapter 14, "Package oracle.xml.differ"
- Chapter 15, "Package oracle.xml.srcviewer"
- Chapter 16, "Package oracle.xml.transviewer"
- Chapter 17, "Package oracle.xml.treeviewer"

The production Oracle XDKs are fully supported and come with a commercial redistribution license. The production libraries are regularly updated on the OTN Web site. For more information, refer to the XDK for Java Beans on the Oracle Technology Network (OTN) Web site as follows:

- Oracle XDK Home at <http://otn.oracle.com/tech/xml/xdkhome.html>
- Oracle XML Developer's Kit for Java at http://otn.oracle.com/tech/xml/xdk_java/content.html
- Oracle XML Developer's Kit for Java Beans at http://otn.oracle.com/tech/xml/xdk_jbeans/content.html

Package oracle.xml.async

This chapter describes package `oracle.xml.async`, which encapsulates the XML Parser for Java `DOMParser` class with a bean interface and extends its functionality to enable asynchronous parsing. It is part of Oracle XML Transviewer Beans that are provided as part of XDK for Java Beans. XML Transviewer Beans facilitate the addition of graphical or visual interfaces to your XML applications.

This chapter contains these sections:

- [Package oracle.xml.async Description](#)
- [Package oracle.xml.async Summary](#)

Package oracle.xml.async Description

Package oracle.xml.async implements a non-visual bean, DOMBuilder bean, which builds a DOM tree from an XML document.

The DOMBuilder bean encapsulates the XML Parser for Java `DOMParser` class with a bean interface and extends its functionality to enable asynchronous parsing. By registering a listener, Java applications can parse large documents or successive instances of documents and then allow control to return immediately to the caller.

It enables asynchronous DOM parsing in separate threads in the background. It utilizes the `EventHandler` interface to notify the calling class when the job is complete.

Information about developing applications using Oracle XML Transviewer Java Beans is available in *Oracle9i XML Developer's Kits Guide - XDK*.

Package oracle.xml.async Summary

Table 12–1 Class summary for oracle.xml.async

Class	Description
DOMBuilder	Encapsulates an eXtensible Markup Language (XML) 1.0 parser to parse an XML document and build a DOM tree.
DOMBuilderBeanInfo	Provides information about the DOMBuilder Bean.
DOMBuilderErrorEvent	Defines the error event which is sent when parse exception occurs.
DOMBuilderErrorListener	Implemented in order to receive notifications when error is found during parsing.
DOMBuilderEvent	Event object that DOMBuilder uses to notify all registered listeners about parse events.
DOMBuilderListener	Implemented in order to receive notifications about events during the asynchronous parsing.
ResourceManager	No description.
XSLTransformer	Applies XSL transformation in a background thread.
XSLTransformerBeanInfo	Provides information about the XSLTransformer Bean.
XSLTransformerErrorEvent	Error event object that XSLTransformer uses to notify all registered listeners about transformation error events.
XSLTransformerErrorListener	This interface must be implemented in order to receive notifications when an error is found during parsing.
XSLTransformerEvent	Provides information about the XSLTransformer Bean.
XSLTransformerListener	Implemented in order to receive notifications about events during the asynchronous transformation.

DOMBuilder

Syntax

```
public class DOMBuilder extends java.lang.Object implements  
java.io.Serializable, oracle.xml.async.DOMBuilderConstants, java.lang.Runnable
```

```
java.lang.Object  
|  
+--oracle.xml.async.DOMBuilder
```

All Implemented Interfaces

```
oracle.xml.async.DOMBuilderConstants, java.lang.Runnable, java.io.Serializable
```

Description

This class encapsulates an eXtensible Markup Language (XML) 1.0 parser to parse an XML document and build a DOM tree. The parsing is done in a separate thread and DOMBuilderInterface interface must be used for notification when the tree is built.

Fields

inSource

```
protected org.xml.sax.InputSource inSource  
InputSource containing XML data to parse
```

inStream

```
protected java.io.InputStream inStream  
InputStream containing XML data to parse
```

inString

```
protected java.lang.String inString  
String containing the URL to parse XML data from
```

methodToCall

```
protected int methodToCall  
XML Parser method to call based on input types
```

reader

protected java.io.Reader reader
java.io.Reader containing XML data to be parsed

result

protected oracle.xml.async.XMLDocument result
XML Document being parsed

rootName

protected java.lang.String rootName
Name of the XML element to be treated as root

url

protected java.net.URL url
URL to parse XML data from

Constructors**DOMBuilder()**

public DOMBuilder()
Creates a new parser object.

DOMBuilder(int)

public DOMBuilder(int id)
Creates a new parser object with a given id.

Parameters

id - The DOMBuilder id.

Methods**addDOMBuilderErrorListener(DOMBuilderErrorListener)**

public void addDOMBuilderErrorListener(DOMBuilderErrorListener p0)
Adds DOMBuilderErrorListener

Parameters

p1 - The DOMBuilderErrorListener to add

addDOMBuilderListener(DOMBuilderListener)

```
public void addDOMBuilderListener(DOMBuilderListener p0)
```

Adds DOMBuilderListener

Parameters

p1 - The DOMBuilderListener to add

getDoctype()

```
public synchronized oracle.xml.async.DTD getDoctype()
```

Get the DTD

Returns

The DTD

getDocument()

```
public synchronized oracle.xml.async.XMLDocument getDocument()
```

Gets the document

Returns

The document being parsed

getId()

```
public int getId()
```

Returns the parser object id.

Returns

The DOMBuilder id

getReleaseVersion()

```
public synchronized java.lang.String getReleaseVersion()
```

Returns the release version of the Oracle XML Parser

Returns

the release version string

getResult()

```
public synchronized org.w3c.dom.Document getResult()  
Gets the document
```

Returns

The document being parsed

getValidationMode()

```
public synchronized boolean getValidationMode()  
Returns the validation mode
```

Returns

true if the XML parser is validating false if not

parse(InputSource)

```
public final synchronized void parse(org.xml.sax.InputSource in)  
Parses the XML from given input source
```

Parameters

in - the org.xml.sax.InputSource to parse

Throws

XMLParseException - if syntax or other error encountered.

SAXException - Any SAX exception, possibly wrapping another exception.

IOException - IO Error.

parse(InputStream)

```
public final synchronized void parse(java.io.InputStream in)  
Parses the XML from given input stream. The base URL should be set for resolving  
external entities and DTD.
```

Parameters

in - the InputStream containing XML data to parse.

Throws

XMLParseException - if syntax or other error encountered.

SAXException - Any SAX exception, possibly wrapping another exception.

`IOException` - IO Error.

See Also

`oracle.xml.parser.v2.XMLParser`

parse(Reader)

```
public final synchronized void parse(java.io.Reader r)
```

Parses the XML from given input stream. The base URL should be set for resolving external entities and DTD.

Parameters

`r` - the `Reader` containing XML data to parse.

Throws

`XMLParseException` - if syntax or other error encountered.

`SAXException` - Any SAX exception, possibly wrapping another exception.

`IOException` - IO Error.

See Also

`oracle.xml.parser.v2.XMLParser`

parse(String)

```
public final synchronized void parse(java.lang.String in)
```

Parses the XML from the URL indicated

Parameters

`in` - the `String` containing the URL to parse from

Throws

`XMLParseException` - if syntax or other error encountered.

`SAXException` - Any SAX exception, possibly wrapping another exception.

`IOException` - IO Error.

parse(URL)

```
public final synchronized void parse(java.net.URL url)
```

Parses the XML document pointed to by the given URL and creates the corresponding XML document hierarchy.

Parameters

`url` - the url points to the XML document to parse.

Throws

`XMLParseException` - if syntax or other error encountered.

`SAXException` - Any SAX exception, possibly wrapping another exception.

`IOException` - IO Error.

parseDTD(InputSource, String)

```
public final synchronized void parseDTD(org.xml.sax.InputSource in,  
    java.lang.String rootName)
```

Parses the XML External DTD from given input source

Parameters

`in` - the `org.xml.sax.InputSource` to parse

`rootName` - the element to be used as root Element

Throws

`XMLParseException` - if syntax or other error encountered.

`SAXException` - Any SAX exception, possibly wrapping another exception.

`IOException` - IO Error.

parseDTD(InputStream, String)

```
public final synchronized void parseDTD(java.io.InputStream in, java.lang.String  
    rootName)
```

Parses the XML External DTD from given input stream. The base URL should be set for resolving external entities and DTD.

Parameters

`in` - the `InputStream` containing XML data to parse.

`rootName` - the element to be used as root Element

Throws

`XMLParseException` - if syntax or other error encountered.

`SAXException` - Any SAX exception, possibly wrapping another exception.

`IOException` - IO Error.

See Also

`oracle.xml.parser.v2.XMLParser`

parseDTD(Reader, String)

```
public final synchronized void parseDTD(java.io.Reader r, java.lang.String  
rootName)
```

Parses the XML External DTD from given input stream. The base URL should be set for resolving external entities and DTD.

Parameters

`r` - the Reader containing XML data to parse.

`rootName` - the element to be used as root Element

Throws

`XMLParseException` - if syntax or other error encountered.

`SAXException` - Any SAX exception, possibly wrapping another exception.

`IOException` - IO Error.

See Also

`oracle.xml.parser.v2.XMLParser`

parseDTD(String, String)

```
public final synchronized void parseDTD(java.lang.String in, java.lang.String  
rootName)
```

Parses the XML External DTD from the URL indicated

Parameters

`in` - the String containing the URL to parse from

`rootName` - the element to be used as root Element

Throws

`XMLParseException` - if syntax or other error encountered.

`SAXException` - Any SAX exception, possibly wrapping another exception.

`IOException` - IO Error.

parseDTD(URL, String)

```
public final synchronized void parseDTD(java.net.URL url, java.lang.String rootName)
```

Parses the XML External DTD document pointed to by the given URL and creates the corresponding XML document hierarchy.

Parameters

url - the url points to the XML document to parse.

rootName - the element to be used as root Element

Throws

XMLParseException - if syntax or other error encountered.

SAXException - Any SAX exception, possibly wrapping another exception.

IOException - IO Error.

removeDOMBuilderErrorListener(DOMBuilderErrorListener)

```
public synchronized void removeDOMBuilderErrorListener(DOMBuilderErrorListener p0)
```

Remove DOMBuilderErrorListener

Parameters

p1 - The DOMBuilderErrorListener to remove

removeDOMBuilderListener(DOMBuilderListener)

```
public synchronized void removeDOMBuilderListener(DOMBuilderListener p0)
```

Remove DOMBuilderListener

Parameters

p1 - The DOMBuilderListener to remove

run()

```
public void run()
```

This method runs in a thread

Specified By

java.lang.Runnable.run() in interface java.lang.Runnable

setBaseURL(URL)

```
public synchronized void setBaseURL(java.net.URL url)
```

Set the base URL for loading external entities and DTDs. This method should be called if the `parse(InputStream)` is used to parse the XML Document

Parameters

`url` - The base URL

setDebugMode(boolean)

```
public void setDebugMode(boolean flag)
```

Sets a flag to turn on debug information in the document

Parameters

`flag` - determines whether debug info is stored

setDoctype(DTD)

```
public synchronized void setDoctype(oracle.xml.async.DTD dtd)
```

Set the DTD

Parameters

`dtd` - DTD to set and used while parsing

setErrorStream(OutputStream)

```
public final synchronized void setErrorStream(java.io.OutputStream out)
```

Creates an output stream for the output of errors and warnings. If an output stream for errors is not specified, the parser will use the standard error output stream `System.err` for outputting errors and warnings.

Parameters

`out` - The output stream to use for errors and warnings

setErrorStream(OutputStream, String)

```
public final synchronized void setErrorStream(java.io.OutputStream out,  
java.lang.String enc)
```

Creates an output stream for the output of errors and warnings. If an output stream for errors is not specified, the parser will use the standard error output stream `System.err` for outputting errors and warnings. Additionally, an `.exception` is thrown if the encoding specified is unsupported.

Parameters

`out` - The output stream to use for errors and warnings
`enc` - the encoding to use

Throws

`IOException` - if an unsupported encoding is specified

setErrorStream(PrintWriter)

```
public final synchronized void setErrorStream(java.io.PrintWriter out)
```

Creates an output stream for the output of errors and warnings. If an output stream for errors is not specified, the parser will use the standard error output stream `System.err` for outputting errors and warnings.

Parameters

`out` - The `PrintWriter` to use for errors and warnings

setNodeFactory(NodeFactory)

```
public synchronized void setNodeFactory(oracle.xml.async.NodeFactory factory)
```

Set the node factory. Applications can extend the `NodeFactory` and register it through this method. The parser will then use the user supplied `NodeFactory` to create nodes of the DOM tree.

Parameters

`factory` - The `NodeFactory` to set

Throws

`XMLParseException` - if an invalid factory is set

See Also

`NodeFactory`

setPreserveWhitespace(boolean)

```
public synchronized void setPreserveWhitespace(boolean flag)
```

Set the white space preserving mode

Parameters

`flag` - preserving mode

setValidationMode(boolean)

```
public synchronized void setValidationMode(boolean yes)  
Set the validation mode
```

Parameters

yes - determines whether the XML parser should be validating

showWarnings(boolean)

```
public synchronized void showWarnings(boolean yes)  
Switch to determine whether to print warnings
```

Parameters

yes - determines whether warnings should be shown

DOMBuilderBeanInfo

Syntax

```
public class DOMBuilderBeanInfo extends java.beans.SimpleBeanInfo
```

```
java.lang.Object
|
+--java.beans.SimpleBeanInfo
    |
    +--oracle.xml.async.DOMBuilderBeanInfo
```

All Implemented Interfaces

```
java.beans.BeanInfo
```

Description

This class provides information about the DOMBuilder Bean.

Constructors

DOMBuilderBeanInfo()

```
public DOMBuilderBeanInfo()
The default Constructor
```

Methods

getIcon(int)

```
public java.awt.Image getIcon(int iconKind)
Gets an image object that can be used to represent DOMBuilder bean in toolbars,
toolboxes, etc.
```

Overrides:

```
java.beans.SimpleBeanInfo.getIcon(int) in class java.beans.SimpleBeanInfo
```

Parameters

```
iconKind - The kind of icon requested.
```

Returns

An image object representing the requested icon type for DOMBuilder bean.

getPropertyDescriptors()

```
public java.beans.PropertyDescriptor[] getPropertyDescriptors()  
Gets the DOMBuilder bean's PropertyDescriptors
```

Overrides:

java.beans.SimpleBeanInfo.getPropertyDescriptors() in class
java.beans.SimpleBeanInfo

Returns

An array of PropertyDescriptors describing the editable properties supported by DOMBuilder bean.

DOMBuilderErrorEvent

Syntax

```
public class DOMBuilderErrorEvent extends java.util.EventObject
```

```
java.lang.Object
|
+--java.util.EventObject
   |
   +--oracle.xml.async.DOMBuilderErrorEvent
```

All Implemented Interfaces

```
java.io.Serializable
```

Description

This class defines the error event which is sent when parse exception occurs.

Fields

```
protected java.lang.Exception e
The exception being raised.
```

Constructors

DOMBuilderErrorEvent(Object, Exception)

```
public DOMBuilderErrorEvent(java.lang.Object p0, java.lang.Exception e)
Constructor for DOMBuilderErrorEvent.
```

Parameters

p0 - The Object that created this event.
e - The Exception raised.

Methods

getException()

```
public java.lang.Exception getException()
Gets the Exception
```

Returns

The Exception beind raised

getMessage()

```
public java.lang.String getMessage()
```

Returns the error message generated by the parser

Returns

The error message string

DOMBuilderErrorListener

Syntax

```
public interface DOMBuilderErrorListener extends java.util.EventListener
```

All Superinterfaces

```
java.util.EventListener
```

Description

This interface must be implemented in order to receive notifications when error is found during parsing. The class implementing this interface must be added to the DOMBuilder using `addDOMBuilderErrorListener` method.

Methods

domBuilderErrorCalled(DOMBuilderErrorEvent)

```
public void domBuilderErrorCalled(DOMBuilderErrorEvent p0)
```

This method is called when a parse error occurs.

Parameters

`p0` - The `DOMBuilderErrorEvent` object produced by the `DOMBuilder` as result of parsing error

DOMBuilderEvent

Syntax

```
public class DOMBuilderEvent extends java.util.EventObject
```

```
java.lang.Object  
|  
+--java.util.EventObject  
|  
+--oracle.xml.async.DOMBuilderEvent
```

All Implemented Interfaces

```
java.io.Serializable
```

Description

The event object that DOMBuilder uses to notify all registered listeners about parse events.

Fields

id

```
protected int id  
ID of the source DOMBuilder object
```

Constructors

DOMBuilderEvent(Object, int)

```
public DOMBuilderEvent(java.lang.Object p0, int p1)  
Creates a new DOMBuilderEvent
```

Parameters

p0 - The Object creating this event.
p1 - Id of the DOMBuilder creating this event.

Methods

getID()

```
public int getID()
```

Returns unique id of the DOMBuilder object which can be used to identify which instance of the DOMBuilder generated this event in cases where multiple instances of DOMBuilder may be working in background.

Returns

The unique id of the source DOMBuilder for this event.

DOMBuilderListener

Syntax

```
public interface DOMBuilderListener extends java.util.EventListener
```

All Superinterfaces

```
java.util.EventListener
```

Description

This interface must be implemented in order to receive notifications about events during the asynchronous parsing. The class implementing this interface must be added to the DOMBuilder using addDOMBuilderListener method.

Methods

domBuilderError(DOMBuilderEvent)

```
public void domBuilderError(DOMBuilderEvent p0)
```

This method is called when parse error occur.

Parameters

p0 - - The DOMBuilderEvent object produced by the DOMBuilder

domBuilderOver(DOMBuilderEvent)

```
public void domBuilderOver(DOMBuilderEvent p0)
```

This method is called when the parse is complete

Parameters

p0 - - The DOMBuilderEvent object produced by the DOMBuilder

domBuilderStarted(DOMBuilderEvent)

```
public void domBuilderStarted(DOMBuilderEvent p0)
```

This method is called when parse starts

Parameters

p0 -- The DOMBuilderEvent object produced by the DOMBuilder

ResourceManager

Syntax

```
public class ResourceManager extends java.lang.Object
```

```
java.lang.Object  
|  
+--oracle.xml.async.ResourceManager
```

Constructors

ResourceManager(int)

```
public ResourceManager(int i)  
The ResourceManager constructor
```

Parameters

`i` -- the number of resources to manage

Methods

activeFound()

```
public boolean activeFound()  
Checks if any of the logical resources being managed are in active use
```

Returns

`true` - if one or more resource is in use `false` - if none of the resources are in use

getResource()

```
public synchronized void getResource()  
If the number of resources available for use is nonzero, the method decreases the number of resources by one. Otherwise, it waits until a resource is released & it becomes available for use.
```

releaseResource()

```
public void releaseResource()
```

Releases a resource. When this method is called, the number of resources available is increased by one.

sleep(int)

```
public void sleep(int i)
```

Allows usage of `Thread.sleep()` without try/catch

XSLTransformer

Syntax

```
public class XSLTransformer extends java.lang.Object implements  
java.io.Serializable, oracle.xml.async.XSLTransformerConstants,  
java.lang.Runnable
```

```
java.lang.Object  
|  
+--oracle.xml.async.XSLTransformer
```

All Implemented Interfaces

```
java.lang.Runnable, java.io.Serializable,  
oracle.xml.async.XSLTransformerConstants
```

Description

Applies XSL transformation in a background thread.

Fields

methodToCall

```
protected int methodToCall  
The XSL transformation method to call based on input types.
```

result

```
protected oracle.xml.async.DocumentFragment result  
Transformation result document.
```

Constructors

XSLTransformer()

```
public XSLTransformer()  
XSLTransformer constructor
```

XSLTransformer(int)

```
public XSLTransformer(int id)  
XSLTransformer constructor accepting an identifier
```

Parameters

`id` - - A unique integer that can be used to identify the XSLTransformer instance during event processing

Methods**addXSLTransformerErrorListener(XSLTransformerErrorListener)**

```
public void addXSLTransformerErrorListener(XSLTransformerErrorListener p0)
```

Adds an XSLTransformer error event listener

Parameters

`p0` - XSLTransformerErrorListener to be added

addXSLTransformerListener(XSLTransformerListener)

```
public void addXSLTransformerListener(XSLTransformerListener p0)
```

Adds a XSLTransformer listener

Parameters

`p0` - XSLTransformerListener to be added

getId()

```
public int getId()
```

Returns the unique XSLTransformer id

Returns

The id of this XSLTransformer.

getResult()

```
public synchronized oracle.xml.async.DocumentFragment getResult()
```

Returns the document fragment for the resulting document. Call this method only after receiving notification that the transformation is complete. Since the transformation occurs in background and asynchronously, calling this method immediately after processXSL will result in holding the control until the result is available.

Returns

The resulting document fragment of the XSL transformation.

processXSL(XSLStylesheet, InputStream, URL)

```
public void processXSL(oracle.xml.async.XSLStylesheet xsl, java.io.InputStream  
xml, java.net.URL ref)
```

Initiates XSL Transformation in the background. The control is returned immediately.

Parameters

`xsl` - The stylesheet to be used for XSL transformation

`xml` - The XML document to be used (as a `java.io.InputStream`)

`ref` - Reference URL to resolve external entities in input XML

Throws

`XSLException` - if an error occurs during XSL transformation

processXSL(XSLStylesheet, Reader, URL)

```
public void processXSL(oracle.xml.async.XSLStylesheet xsl, java.io.Reader xml,  
java.net.URL ref)
```

Initiates XSL Transformation in the background. The control is returned immediately.

Parameters

`xsl` - The stylesheet to be used for XSL transformation

`xml` - The XML document to be used (as a `java.io.Reader`)

`ref` - Reference URL to resolve external entities in input XML

Throws

`XSLException` - if an error occurs during XSL transformation

processXSL(XSLStylesheet, URL, URL)

```
public void processXSL(oracle.xml.async.XSLStylesheet xsl, java.net.URL xml,  
java.net.URL ref)
```

Initiates XSL Transformation in the background. The control is returned immediately.

Parameters

`xsl` - The stylesheet to be used for XSL transformation

`xml` - The XML document to be used (as a `java.net.URL`)

`ref` - Reference URL to resolve external entities in input XML

Throws

`XSLException` - if an error occurs during XSL transformation

processXSL(XSLStylesheet, XMLDocument)

```
public void processXSL(oracle.xml.async.XSLStylesheet xsl,  
oracle.xml.async.XMLDocument xml)
```

Initiates XSL Transformation in the background. The control is returned immediately.

Parameters

`xsl` - The stylesheet to be used for XSL transformation

`xml` - The XML document to be used (as a DOM Tree)

Throws

`XSLException` - if an error occurs during XSL transformation

processXSL(XSLStylesheet, XMLDocument, OutputStream)

```
public void processXSL(oracle.xml.async.XSLStylesheet xsl,  
oracle.xml.async.XMLDocument xml, java.io.OutputStream os)
```

Initiates XSL Transformation in the background. The control is returned immediately.

Parameters

`xsl` - The stylesheet to be used for XSL transformation

`xml` - The XML document to be used (as a DOM Tree)

`os` - Outputstream to which the XSL transformation result is written

Throws

`XSLException` - if an error occurs during XSL transformation

removeDOMTransformerErrorListener(XSLTransformerErrorListener)

```
public synchronized void  
removeDOMTransformerErrorListener(XSLTransformerErrorListener p0)
```

Removes an XSLTransformer error event listener

Parameters

p0 - XSLTransformerErrorListener to be removed

removeXSLTransformerListener(XSLTransformerListener)

```
public synchronized void removeXSLTransformerListener(XSLTransformerListener p0)
```

Removes a XSLTransformer listener

Parameters

p0 - XSLTransformerListener to be removed

run()

```
public void run()
```

Starts a separate thread to do the XSL Transformation.

Specified By

java.lang.Runnable.run() in interface java.lang.Runnable

setErrorStream(OutputStream)

```
public final void setErrorStream(java.io.OutputStream out)
```

Sets the error stream used by the XSL processor

Parameters

out - The error output stream for the XSL processor

showWarnings(boolean)

```
public final void showWarnings(boolean yes)
```

Sets the showWarnings flag used by the XSL processor

Parameters

yes - Boolean indicating if XSL processor warnings to be shown or not.

XSLTransformerBeanInfo

Syntax

```
public class XSLTransformerBeanInfo extends java.beans.SimpleBeanInfo
```

```
java.lang.Object
|
+--java.beans.SimpleBeanInfo
   |
   +--oracle.xml.async.XSLTransformerBeanInfo
```

All Implemented Interfaces

```
java.beans.BeanInfo
```

Description

This class provides information about the XSLTransformer Bean.

Constructors

XSLTransformerBeanInfo()

```
public XSLTransformerBeanInfo()
```

The default Constructor

Methods

getIcon(int)

```
public java.awt.Image getIcon(int iconKind)
```

Gets an image object that can be used to represent XSLTransformer bean in toolbars, toolboxes, etc.

Overrides:

java.beans.SimpleBeanInfo.getIcon(int) in class java.beans.SimpleBeanInfo

Parameters

iconKind - The kind of icon requested.

Returns

An image object representing the requested icon type for `XSLTransformer` bean.

getPropertyDescriptors()

```
public java.beans.PropertyDescriptor[] getPropertyDescriptors()
```

Gets the `XSLTransformer` bean's `PropertyDescriptors`

Overrides:

`java.beans.SimpleBeanInfo.getPropertyDescriptors()` in class

`java.beans.SimpleBeanInfo`

Returns

An array of `PropertyDescriptors` describing the editable properties supported by `XSLTransformer` bean.

XSLTransformerErrorEvent

Syntax

```
public class XSLTransformerErrorEvent extends java.util.EventObject
```

```
java.lang.Object
|
+--java.util.EventObject
   |
   +--oracle.xml.async.XSLTransformerErrorEvent
```

All Implemented Interfaces

```
java.io.Serializable
```

Description

The error event object that XSLTransformer uses to notify all registered listeners about transformation error events.

Fields

```
protected java.lang.Exception e
The exception being raised.
```

Constructors

XSLTransformerErrorEvent(Object, Exception)

```
public XSLTransformerErrorEvent(java.lang.Object p0, java.lang.Exception e)
Constructor for XSLTransformerErrorEvent.
```

Parameters

p0 - The Object that created this event
e - The Exception raised.

Methods

getException()

```
public java.lang.Exception getException()
```

Returns the exception that XSLTransformer encountered object unique id. Can be used to

Returns

The transformation exception

getMessage()

```
public java.lang.String getMessage()
```

Returns the error message that describes the error that XSLTransformer encountered

Returns

The error message

XSLTransformerErrorListener

Syntax

```
public interface XSLTransformerErrorListener extends java.util.EventListener
```

All Superinterfaces

```
java.util.EventListener
```

Description

This interface must be implemented in order to receive notifications about error events during the asynchronous transformation. The class implementing this interface must be added to the XSLTransformer using `addXSLTransformerListener` method.

Methods

`xslTransformerErrorCalled(XSLTransformerErrorEvent)`

```
public void xslTransformerErrorCalled(XSLTransformerErrorEvent p0)
```

This method is called when parse or transformation error occurs.

Parameters

`p0` -- The XSLTransformerErrorEvent object produced by the XSLTransformer

XSLTransformerEvent

Syntax

```
public class XSLTransformerEvent extends java.util.EventObject
```

```
java.lang.Object
|
+--java.util.EventObject
|
+--oracle.xml.async.XSLTransformerEvent
```

All Implemented Interfaces

```
java.io.Serializable
```

Fields

id

```
protected int id
ID of the source XSLTransformer object
```

Constructors

XSLTransformerEvent(Object, int)

```
public XSLTransformerEvent(java.lang.Object p0, int p1)
```

Constructs the XSLTransformerEvent object using the XSLTransformer source object and its unique id.

Parameters

`<code>p0</code>`

 - The source XSLTransformer object that will fire the events

`<code>p1</code>`

 - Unique id identifying the source object

Methods

getID()

```
public int getID()
```

Returns unique id of the XSLTransformer object which can be used to identify which instance of the XSLTransformer generated this event in cases where multiple instances of XSLTransformer may be working in background.

Returns

The unique `id` of the source XSLTransformer object for this event object.

XSLTransformerListener

Syntax

```
public interface XSLTransformerListener extends java.util.EventListener
```

All Superinterfaces

```
java.util.EventListener
```

Description

This interface must be implemented in order to receive notifications about events during the asynchronous transformation. The class implementing this interface must be added to the XSLTransformer using addXSLTransformerListener method.

Methods

xslTransformerError(XSLTransformerEvent)

```
public void xslTransformerError(XSLTransformerEvent p0)
```

This method is called when parse or transformation error occur.

Parameters

p0 -- The XSLTransformerEvent object produced by the XSLTransformer

xslTransformerOver(XSLTransformerEvent)

```
public void xslTransformerOver(XSLTransformerEvent p0)
```

This method is called when the transformation is complete

Parameters

p0 -- The XSLTransformerEvent object produced by the XSLTransformer

xslTransformerStarted(XSLTransformerEvent)

```
public void xslTransformerStarted(XSLTransformerEvent p0)
```

This method is called when the transformation starts

Parameters

p0 -- The XSLTransformerEvent object produced by the XSLTransformer.

Package oracle.xml.dbviewer

This chapter describes package `oracle.xml.dbviewer`, which contains the `DBViewer` bean for Oracle XML Transviewer Beans.

The Oracle XML Transviewer Beans are provided as part of Oracle9i XDK for Java Beans. XML Transviewer Beans facilitate the addition of graphical or visual interfaces to your XML applications.

The `DBViewer` bean displays database queries or any XML by applying XSL stylesheets and visually presenting the resulting HTML in a scrollable swing panel.

This chapter contains these sections:

- [Package `oracle.xml.dbviewer` Description](#)
- [Package `oracle.xml.dbviewer` Summary](#)

Package oracle.xml.dbviewer Description

Package oracle.xml.dbviewer implements the Java bean that can be used to display database queries or any XML by applying XSL stylesheets and presenting the resulting HTML in a scrollable swing panel.

This bean has three buffers:

- XML
- XSL
- Result

The bean API allows the calling program to load and save the buffers from various sources and to apply stylesheet transformations to the XML buffer using the stylesheet in the XSL buffer. The result can be stored in the result buffer. The XML and XSL buffer content can be shown as source or as a tree structure. The result buffer content can be rendered as HTML and also shown as source or a tree structure. The XML buffer can be loaded from a database query. All buffers can load and save files from CLOB tables in the Oracle database or from the file system. This way, the bean can be used to move files between the file system and the user schema in the database.

Package oracle.xml.dbviewer Summary

Table 13–1 Class Summary for oracle.xml.dbviewer

Class	Description
DBViewer	Java bean used to display database queries or any XML by applying XSL stylesheets and displaying the resulting HTML.
DBViewerBeanInfo	Provides information about the bean.

DBViewer

Syntax

```
public class DBViewer extends javax.swing.JPanel implements java.io.Serializable
```

```
java.lang.Object
|
+--java.awt.Component
|
+--java.awt.Container
|
+--javax.swing.JComponent
|
+--javax.swing.JPanel
|
+--oracle.xml.dbviewer.DBViewer
```

All Implemented Interfaces

```
javax.accessibility.Accessible, java.awt.image.ImageObserver,
java.awt.MenuContainer, java.io.Serializable
```

Description

Java bean that can be used to display database queries or any XML by applying XSL stylesheets and visualizing the resulting HTML in a scrollable swing panel. This bean has three buffers: XML, XSL, and result buffer. The bean API enables the calling program to load/save the buffers from various sources and to apply stylesheet transformation to the XML buffer using the stylesheet in the XSL buffer. The result can be stored in the result buffer. The XML and XSL buffers content can be shown as source or as a tree structure. The result buffer content can be rendered as HTML and also shown as source or tree structure. The XML buffer can be loaded from database query. All buffers can load and save files from CLOB tables in Oracle database and from the file system as well. Therefore, the control can be also used to move files between the file system and the user schema in the database.

Constructors

DBViewer()

```
public DBViewer()
Constructs a new instance.
```

Methods

getHostname()

```
public java.lang.String getHostname()  
Get database host name
```

Returns

host name

getInstancename()

```
public java.lang.String getInstancename()  
Get database instance name
```

Returns

database instance name

getPassword()

```
public java.lang.String getPassword()  
Get user password
```

Returns

user password

getPort()

```
public java.lang.String getPort()  
Get database port number
```

Returns

String with the database port number

getResBuffer()

```
public java.lang.String getResBuffer()  
Get the content of the result buffer
```

Returns

the buffer content

getResCLOBFileName()

```
public java.lang.String getResCLOBFileName()  
Get result CLOB file name
```

Returns

result CLOB file name

getResCLOBTableName()

```
public java.lang.String getResCLOBTableName()  
Get result CLOB table name
```

Returns

result CLOB table name

getResFileName()

```
public java.lang.String getResFileName()  
Get Result file name
```

Returns

XSL file name

getUsername()

```
public java.lang.String getUsername()  
Get user name
```

Returns

user name

getXmlBuffer()

```
public java.lang.String getXmlBuffer()  
Get the content of the XML buffer
```

Returns

the buffer content

getXmlCLOBFileName()

```
public java.lang.String getXmlCLOBFileName()
```

Get XML CLOB file name

Returns

XML CLOB file name

getXmlCLOBTableName()

public java.lang.String getXmlCLOBTableName()
Get XML CLOB table name

Returns

XML CLOB table name

getXmlFileName()

public java.lang.String getXmlFileName()
Get XML file name

Returns

XML file name

getXMLStringFromSQL(String)

public java.lang.String getXMLStringFromSQL(java.lang.String sqlText)
Get XML presentation of result set from SQL query

Returns

the query result set as XML string

getXslBuffer()

public java.lang.String getXslBuffer()
Get the content of the XSL buffer

Returns

the buffer content

getXslCLOBFileName()

public java.lang.String getXslCLOBFileName()
Get the XSL CLOB file name

Returns

XSL CLOB file name

getXsICLOBTableName()

```
public java.lang.String getXsICLOBTableName()
```

Get XSL CLOB table name

Returns

XSL CLOB table name

getXsIFilename()

```
public java.lang.String getXsIFilename()
```

Get XSL file name

Returns

XSL file name

loadResBuffer(String)

```
public void loadResBuffer(java.lang.String filename)
```

Load the result buffer from file

Parameters

filename - file name

loadResBuffer(String, String)

```
public void loadResBuffer(java.lang.String tablename, java.lang.String filename)
```

Load the result buffer from CLOB file

Parameters

tablename - CLOB table name

filename - CLOB file name

loadResBuffer(XMLDocument)

```
public void loadResBuffer(oracle.xml.parser.v2.XMLDocument resdoc)
```

Load the result buffer from XMLDocument

Parameters

resdoc -- the XMLDocument

loadResBufferFromClob()

```
public void loadResBufferFromClob()  
Load the result buffer from CLOB file
```

loadResBufferFromFile()

```
public void loadResBufferFromFile()  
Load the result buffer from file
```

loadXmlBuffer(String)

```
public void loadXmlBuffer(java.lang.String filename)  
Load the XML buffer from file
```

Parameters

filename - file name

loadXmlBuffer(String, String)

```
public void loadXmlBuffer(java.lang.String tablename, java.lang.String filename)  
Load the XML buffer from CLOB file
```

Parameters

tablename - CLOB table name

filename - CLOB file name

loadXmlBuffer(XMLDocument)

```
public void loadXmlBuffer(oracle.xml.parser.v2.XMLDocument xmldoc)  
Load the XML buffer from XMLDocument
```

Parameters

filename - file name

loadXmlBufferFromClob()

```
public void loadXmlBufferFromClob()  
Load the XML buffer from CLOB file
```

loadXmlBufferFromFile()

```
public void loadXmlBufferFromFile()  
Load the XML buffer from file
```

loadXMLBufferFromSQL(String)

```
public void loadXMLBufferFromSQL(java.lang.String sqltext)  
Load the XML buffer from SQL result set
```

Parameters

sqltext - SQL text

loadXslBuffer(String)

```
public void loadXslBuffer(java.lang.String filename)  
Load the XSL buffer from file
```

Parameters

filename - file name

loadXslBuffer(String, String)

```
public void loadXslBuffer(java.lang.String tablename, java.lang.String filename)  
Load the XSL buffer from CLOB file
```

Parameters

tablename - CLOB table name

filename - CLOB file name

loadXslBuffer(XMLDocument)

```
public void loadXslBuffer(oracle.xml.parser.v2.XMLDocument xsldoc)  
Load the XSL buffer from XMLDocument
```

Parameters

xsldoc -- the XML Document

loadXslBufferFromClob()

```
public void loadXslBufferFromClob()  
Load the XSL buffer from CLOB file
```

loadXslBufferFromFile()

```
public void loadXslBufferFromFile()  
Load the XSL buffer from file
```

parseResBuffer()

```
public oracle.xml.parser.v2.XMLDocument parseResBuffer()  
Parse the result buffer and refresh the tree view and source view
```

Returns

XMLDocument

parseXmlBuffer()

```
public oracle.xml.parser.v2.XMLDocument parseXmlBuffer()  
Parse the XML buffer and refresh the tree view and source view
```

Returns

XMLDocument

parseXslBuffer()

```
public oracle.xml.parser.v2.XMLDocument parseXslBuffer()  
Parse the XSL buffer and refresh the tree view and source view
```

Returns

XMLDocument

saveResBuffer(String)

```
public void saveResBuffer(java.lang.String filename)  
Save the result buffer to file
```

Parameters

filename - CLOB file name

saveResBuffer(String, String)

```
public void saveResBuffer(java.lang.String tablename, java.lang.String filename)  
Save the result buffer to CLOB file
```

Parameters

tablename - CLOB table name

filename - CLOB file name

saveResBufferToClob()

```
public void saveResBufferToClob()
```

Save the result buffer to CLOB file

saveResBufferToFile()

```
public void saveResBufferToFile()
```

Save the result buffer to file

saveXmlBuffer(String)

```
public void saveXmlBuffer(java.lang.String filename)
```

Save the XML buffer to file

Parameters

filename - file name

saveXmlBuffer(String, String)

```
public void saveXmlBuffer(java.lang.String tablename, java.lang.String filename)
```

Save the XML buffer to CLOB file

Parameters

tablename - CLOB table name

filename - CLOB file name

saveXmlBufferToClob()

```
public void saveXmlBufferToClob()
```

Save the XML buffer to CLOB file

saveXmlBufferToFile()

```
public void saveXmlBufferToFile()
```

Save the XML buffer to file

saveXslBuffer(String)

```
public void saveXslBuffer(java.lang.String filename)
```

Save the XSL buffer to file

Parameters

filename - file name

saveXslBuffer(String, String)

```
public void saveXslBuffer(java.lang.String tablename, java.lang.String filename)
```

Save the XSL buffer to CLOB file

Parameters

tablename - CLOB table name

filename - CLOB file name

saveXslBufferToClob()

```
public void saveXslBufferToClob()
```

Save the XSL buffer to CLOB file

saveXslBufferToFile()

```
public void saveXslBufferToFile()
```

Save the XSL buffer to file

setHostname(String)

```
public void setHostname(java.lang.String hostname)
```

Set database host name

Parameters

hostname - the host name

setInstancename(String)

```
public void setInstancename(java.lang.String instancename)
```

Set database instance name

Parameters

instancename - the database instance name

setPassword(String)

```
public void setPassword(java.lang.String password)
Set user password
```

Parameters

password - the user password

setPort(String)

```
public void setPort(java.lang.String port)
Set database port number
```

Parameters

port - String containing the port number

setResBuffer(String)

```
public void setResBuffer(java.lang.String text)
Set new text in the result buffer
```

Parameters

text - the new text

setResCLOBFileName(String)

```
public void setResCLOBFileName(java.lang.String name)
Set Result CLOB file name
```

Parameters

name - Result CLOB file name

setResCLOBTableName(String)

```
public void setResCLOBTableName(java.lang.String name)
Set Result CLOB table name
```

Parameters

name - Result CLOB table name

setResFileName(String)

```
public void setResFileName(java.lang.String name)
```

Set Result file name

Parameters

name - Result file name

setResHtmlView(boolean)

public void setResHtmlView(boolean on)
Show the result buffer as rendered HTML

setResSourceEditView(boolean)

public void setResSourceEditView(boolean on)
Show the result buffer as XML source and enter edit mode

setResSourceView(boolean)

public void setResSourceView(boolean on)
Show the result buffer as XML source

setResTreeView(boolean)

public void setResTreeView(boolean on)
Show the result buffer as XML tree view

setUsername(String)

public void setUsername(java.lang.String username)
Set user name

Parameters

username - the user name

setXmlBuffer(String)

public void setXmlBuffer(java.lang.String text)
Set new text in the XML buffer

Parameters

text - XML text

setXmlCLOBFileName(String)

public void setXmlCLOBFileName(java.lang.String name)

Set XML CLOB table name

Parameters

name - XML CLOB table name

setXmlCLOBTableName(String)

```
public void setXmlCLOBTableName(java.lang.String name)
Set XML CLOB table name
```

Parameters

name - XML CLOB table name

setXmlFileName(String)

```
public void setXmlFileName(java.lang.String name)
Set XML file name
```

Parameters

name - XML file name

setXmlSourceEditView(boolean)

```
public void setXmlSourceEditView(boolean on)
Show the XML buffer as XML source and enter edit mode
```

setXmlSourceView(boolean)

```
public void setXmlSourceView(boolean on)
Show the XML buffer as XML source
```

setXmlTreeView(boolean)

```
public void setXmlTreeView(boolean on)
Show the XML buffer as tree
```

setXslBuffer(String)

```
public void setXslBuffer(java.lang.String text)
Set new text in the XSL buffer
```

Parameters

text - XSL text

setXslCLOBFileName(String)

```
public void setXslCLOBFileName(java.lang.String name)
Set XSL CLOB file name
```

Parameters

name - XSL CLOB file name

setXslCLOBTableName(String)

```
public void setXslCLOBTableName(java.lang.String name)
Set XSL CLOB table name
```

Parameters

name - XSL CLOB table name

setXslFileName(String)

```
public void setXslFileName(java.lang.String name)
Set XSL file name
```

Parameters

name - XSL file name

setXslSourceEditView(boolean)

```
public void setXslSourceEditView(boolean on)
Show the XSL buffer as XML source and enter edit mode
```

setXslSourceView(boolean)

```
public void setXslSourceView(boolean on)
Show the XSL buffer as XML source
```

setXslTreeView(boolean)

```
public void setXslTreeView(boolean on)
Show the XSL buffer as tree
```

transformToDoc()

```
public oracle.xml.parser.v2.XMLDocument transformToDoc()
Transforms the content of the XML buffer by applying the stylesheet from the XSL
buffer.
```

transformToRes()

```
public void transformToRes()
```

Apply the stylesheet transformation from the XSL buffer to the XML in the XML buffer and stores the result into the result buffer

transformToString()

```
public java.lang.String transformToString()
```

Transforms the content of the XML buffer by applying the stylesheet from the XSL buffer.

DBViewerBeanInfo

Syntax

```
public class DBViewerBeanInfo extends java.beans.SimpleBeanInfo
```

```
java.lang.Object
|
+--java.beans.SimpleBeanInfo
   |
   +--oracle.xml.dbviewer.DBViewerBeanInfo
```

All Implemented Interfaces

```
java.beans.BeanInfo
```

Constructors

DBViewerBeanInfo()

```
public DBViewerBeanInfo()
```

Constructor

Methods

getIcon(int)

```
public java.awt.Image getIcon(int iconKind)
```

Overrides:

java.beans.SimpleBeanInfo.getIcon(int) in class java.beans.SimpleBeanInfo

getPropertyDescriptors()

```
public java.beans.PropertyDescriptor[] getPropertyDescriptors()
```

Overrides:

java.beans.SimpleBeanInfo.getPropertyDescriptors() in class
java.beans.SimpleBeanInfo

Package oracle.xml.differ

This chapter documents package oracle.xml.differ for XML Diff Bean. This package is part of the Oracle XDK for Java Beans.

This chapter contains these sections:

- [Package oracle.xml.differ Description](#)
- [Package oracle.xml.differ class Summary](#)
- [XMLDiff Class](#)
- [XMLDiffBeanInfo Class](#)

Package oracle.xml.differ Description

The classes contained in oracle.xml.differ implement the XML Diff Bean, which performs a comparison on two XML DOM trees using a graphical user interface. A node can be inserted, deleted, moved, or modified.

You can generate the differences between the two XML trees in the form of XSL code. The first XML file can be transformed into the second XML file by using the XSL code generated.

Package oracle.xml.differ class Summary

```
class java.lang.Object
    class java.beans.SimpleBeanInfo (implements java.beans.BeanInfo)
        class oracle.xml.differ.XMLDiffBeanInfo
    class oracle.xml.differ.XMLDiff (implements
        oracle.xml.async.DOMBuilderErrorListener,
        oracle.xml.async.DOMBuilderListener, java.io.Serializable)
```

Table 14–1 oracle.xml.differ classes

Class	Description
XMLDiff Class	Defines an interface for comparing two XML documents.
XMLDiffBeanInfo Class	Extends java.beans.SimpleBeanInfo.

XMLDiff Class

Description

Public class XMLDiff

Extends `java.lang.Object`

Implements `DOMBuilderListener`, `DOMBuilderErrorListener`, `java.io.Serializable`

Defines an interface for comparing two XML files. It enables two XML files to be compared to check for their equivalence. It provides the objects to display the differences, if any, in a graphical format. The differences can also be represented as XSL. The corresponding XSL stylesheet with the differences can be generated as a file or an `XMLDocument` object. The first XML file can be transformed into the second XML file by using the XSL stylesheet generated.

```
java.lang.Object
```

```
|
```

```
+-oracle.xml.differ.XMLDiff
```

Constructors

XMLDiff

Syntax

```
public XMLDiff()
```

Methods

setFiles

Description

Sets the XML files which need to be compared. Both files are parsed into DOM trees for comparison. This is faster than calling `setInput1()` and `setInput2()`

Syntax

```
public void setFiles(java.io.File file1,  
                    java.io.File file2)
```

Parameters

File - first XML file (separate File with comma for each increment)

File - second XML file

Throws

java.io.IOException - thrown when an I/O error occurs

XMLParseException - caused when parsing XML document

SAXException - caused when parsing XML document

java.lang.InterruptedExcepion - if a sleeping thread is interrupted

setDocuments

Description

Sets the XML documents which need to be compared.

Syntax

```
public void setDocuments(XMLDocument doc1,  
                        XMLDocument doc2)
```

Parameters

XMLDocument - first XML document

XMLDocument - second XML document

setInput1

Description

Sets the first XML file that needs to be compared. The input file is parsed into a DOM tree for comparison.

Syntax

```
public void setInput1(java.io.File file1)
```

Parameters

File - first XML file

Throws

java.io.IOException - thrown when an I/O error occurs

XMLParseException - caused when parsing XML document

SAXException - caused when parsing XML document

java.lang.InterruptedExcepTion - if a sleeping thread is interrupted

setInput2

Description

Sets the second XML file that needs to be compared. The input file is parsed into a DOM tree for comparison.

Syntax

```
public void setInput2(java.io.File file2)
```

Parameters

file - second XML file

Throws

java.io.IOException - thrown when an I/O error occurs

XMLParseException - caused when parsing XML document

SAXException - caused when parsing XML document

java.lang.InterruptedExcepTion - if a sleeping thread is interrupted

setInput1

Description

Sets the first XML document which need to be compared.

Syntax

```
public void setInput1(XMLDocument doc1)
```

Parameters

XMLDocument - first XML document

setInput2

Description

Sets the second XML document which need to be compared.

Syntax

```
public void setInput2(XMLDocument doc2)
```

Parameters

XML - Document second XML document

getDocument1

Description

Gets the document root as an XMLDocument object of the first XML tree

Syntax

```
public XMLDocument getDocument1()
```

Returns

the document root of the first XML file

getDocument2

```
public XMLDocument getDocument2()
```

Gets the document root as an XMLDocument object of the second XML tree

Returns:

the document root of the second XML file

diff

Description

Finds the differences between the two XML files or the two XMLDocument objects.

Syntax

```
public boolean diff()
```

Returns

false if the XML files or docs are same, true if they are different

Throws

java.lang.NullPointerException - occurs when xml files were not parsed successfully and this function was called. Or if xml documents have not been set.

getDiffPane1

Description

Gets the text panel as JTextPane object which visually shows the diffs in the first XML file.

Syntax

```
public javax.swing.JTextPane getDiffPanel()
```

Returns

visual text panel showing diffs in first XML file

getDiffPane2

Description

Gets the text panel as JTextPane object which visually shows the diffs in the second XML file or document

Syntax

```
public javax.swing.JTextPane getDiffPane2()
```

Returns

visual text panel showing diffs in second XML file

setIndentIncr

Description

Sets the indentation for the XSL generation. This should be called before the generateXSLFile() or generateXSLDoc(). The indentation will be applied to all attributes only. For indenting newly inserted nodes besides attributes see setNewNodeIndentIncr()

Syntax

```
public void setIndentIncr(int spaces)
```

Parameters

int - indentation increment in number of spaces for attributes

setNewNodeIndentIncr

Description

Sets the indentation for the XSL generation. This should be called before the `generateXSLFile()` or `generateXSLDoc()`. The indentation will be applied to all newly inserted nodes only (except attributes). For indentation attributes supported, see `setIndentIncr()`.

Syntax

```
public void setNewNodeIndentIncr(int spaces)
```

Parameters

int - indentation increment in number of spaces for new nodes

generateXSLFile

Description

Generates an XSL file of input filename which represents the differences between the 2 XML files which were set initially. If the input filename is null a default XSL file named `XMLDiff.xsl` will be generated. The first XML file can be transformed into the second XML file using the XSL stylesheet generated. If the XML files were the same then the XSL generated will transform the first XML file into the second XML file, where the first and second files will be equivalent.

Syntax

```
public void generateXSLFile(java.lang.String filename)
```

Parameters

String - output XSL file name

Throws

`java.io.IOException` - could occur if the xsl file was not created successfully

generateXSLDoc

Description

Generates an XSL stylesheet as an `XMLDocument` which represents the differences between the 2 XML documents set initially. The first XML file can be transformed into the second XML file using the XSL stylesheet generated. If the XML files were

the same then the XSL generated will transform the first XML file into the second XML file, where the first and second files will equivalent.

Syntax

```
public XMLDocument generateXSLDoc()
```

Returns

XSL stylesheet as an XML document

Throws

java.io.IOException - could occur if the xsl file was not created successfully

java.io.FileNotFoundException - if xsl file generated could not be found

SAXException - caused when parsing xml document

XMLParseException - caused when parsing xml document

equals

Description

Method that performs the comparison of two (2) nodes. It is called by the *differ* algorithm. If needed, this function can be overwritten for customized comparisons.

Syntax

```
protected boolean equals(Node node1,  
                          Node node2)
```

Parameters

Node - node1 first node to compare

Node - node2 second node to compare

domBuilderErrorCalled

Description

Method implementing the DOMBuilderErrorListener interface called only by the DOM parser when there is an error while parsing.

Syntax

```
public void domBuilderErrorCalled(DOMBuilderErrorEvent p0)
```

Specified By

domBuilderErrorCalled in interface DOMBuilderErrorListener

Parameters

DOMBuilderErrorEvent - error object thrown by parser

domBuilderError

Description

Method implementing the DOMBuilderErrorListener interface called only by the DOM parser.

Syntax

```
public void domBuilderError(DOMBuilderEvent p0)
```

Specified By

domBuilderError in interface DOMBuilderListener

Parameters

DOMBuilderEvent - parser event Parser errors handled by domBuilderErrorCalled

domBuilderOver

Description

Method implementing DOMBuilderListener interface called only by a DOM parser thread when the parsing is done.

Syntax

```
public void domBuilderOver(DOMBuilderEvent p0)
```

Specified By

domBuilderOver in interface DOMBuilderListener

Parameters

DOMBuilderEvent - parser event

domBuilderStarted

Description

Method implementing DOMBuilderInterface interface called only by the DOM parser when the parsing starts.

Syntax

```
public void domBuilderStarted(DOMBuilderInterface p0)
```

Specified By

domBuilderStarted in interface DOMBuilderInterface

Parameters

DOMBuilderInterface - parser event

printDiffTree

Description

Prints the diff tree which contains the node names and values which have been identified as diffs by the algorithm. Useful for debugging.

Syntax

```
public void printDiffTree(int tree,  
                           java.io.BufferedWriter out)
```

Parameters

int - the tree to print, i.e. one or two

BufferedWriter - containing the printed diff tree

Throws

java.io.IOException - could occur if the xsl file was not created successfully

setNoMoves

Description

Assume that there are no moves to be detected by the diff algorithm. This function should be called before the diff() function. It will result in a performance gain.

Syntax

```
public void setNoMoves()
```

XMLDiffBeanInfo Class

Description

Public class XMLDiffBeanInfo

Extends java.beans.SimpleBeanInfo

java.lang.Object

|

+--java.beans.SimpleBeanInfo

|

+--oracle.xml.differ.XMLDiffBeanInfo

Constructor

XMLDiffBeanInfo

Syntax

```
public XMLDiffBeanInfo()
```

Methods

getPropertyDescriptors

Syntax

```
public java.beans.PropertyDescriptor[] getPropertyDescriptors()
```

Overrides

getPropertyDescriptors in class java.beans.SimpleBeanInfo

getIcon

Syntax

```
public java.awt.Image getIcon(int iconKind)
```

Overrides

getIcon in class java.beans.SimpleBeanInfo

Package oracle.xml.srcviewer

This chapter describes package oracle.xml.srcviewer for, which contains the XMLSourceView bean.

The Oracle XML Transviewer Beans are provided as part of Oracle9i XDK for Java Beans. XML Transviewer Beans facilitate the addition of graphical or visual interfaces to your XML applications.

This chapter contains these sections:

- [Package oracle.xml.srcviewer Description](#)
- [Package oracle.xml.srcviewer Summary](#)
- [XMLSourceView Class](#)
- [XMLSourceViewBeanInfo Class](#)

Package oracle.xml.srcviewer Description

The classes in oracle.xml.srcviewer implement the XMLSourceView bean for displaying the attributes and source for an XML document. The XMLSourceView bean presents a visual interface for XML documents for easier editing. It enables the display of XML and XSL formatted files with color syntax highlighting when modifying an XML document with a text editor. This helps view and edit the files. It can be integrated with DOMBuilder bean, and allows pre- or post-parsing visualization and validation against a specified DTD. Information about developing applications using Oracle XML Transviewer Java Beans can be found in the Oracle resources listed here under *See Also*.

Package oracle.xml.srcviewer Summary

Table 15–1 *oracle.xml.srcviewer* Classes

Class	Description
XMLSourceView Class	Shows an XML document.
XMLSourceViewBeanInfo Class	Extends <code>java.beans.SimpleBeanInfo</code> .

XMLSourceView Class

Syntax

```
public class XMLSourceView extends javax.swing.JPanel implements
java.io.Serializable
```

```
java.lang.Object
|
+--java.awt.Component
|
+--java.awt.Container
|
+--javax.swing.JComponent
|
+--javax.swing.JPanel
|
+--oracle.xml.srcviewer.XMLSourceView
```

All Implemented Interfaces

```
javax.accessibility.Accessible, java.awt.image.ImageObserver,
java.awt.MenuContainer, java.io.Serializable
```

Description

Shows an XML document. Recognizes the following XML token types: Tag, Attribute Name, Attribute Value, Comment, CDATA, PCDATA, PI Data, PI Name and NOTATION Symbol. Each token type has a foreground color and font setting. The default color and font settings can be changed by the user. Takes as input an `org.w3c.dom.Document` object.

Fields

inputDOMDocument

```
protected org.w3c.dom.Document inputDOMDocument
```

jScrollPane

```
protected javax.swing.JScrollPane jScrollPane
```

jTextPane

```
protected javax.swing.JTextPane JTextPane
```

xmlStyledDocument

```
protected oracle.xml.srcviewer.XMLStyledDocument xmlStyledDocument
```

Constructors

XMLSourceView()

```
public XMLSourceView()
```

The class constructor. Creates an object of type XMLSourceView.

Methods

fontGet(AttributeSet)

```
public static java.awt.Font fontGet(javax.swing.text.AttributeSet attributeset)
```

Extracts and returns the font from a given attributeset.

Parameters

attributeset - The source AttributeSet.

Returns

The extracted Font.

fontSet(MutableAttributeSet, Font)

```
public static void fontSet(javax.swing.text.MutableAttributeSet
```

```
mutableattributeset, java.awt.Font font)
```

Sets the mutableattributeset font.

Parameters

mutableattributeset - The mutableattributeset to update.

font - The new Font for the mutableattributeset.

getAttributeNameFont()

```
public java.awt.Font getAttributeNameFont()
```

Returns the Attribute Value font.

Returns

The `Font` object.

getAttributeNameForeground()

```
public java.awt.Color getAttributeNameForeground()  
Returns the Attribute Name foreground color.
```

Returns

The `Color` object.

getAttributeValueFont()

```
public java.awt.Font getAttributeValueFont()  
Returns the Attribute Value font.
```

Returns

The `Font` object.

getAttributeValueForeground()

```
public java.awt.Color getAttributeValueForeground()  
Returns the Attribute Value foreground color.
```

Returns

The `Color` object.

getBackground()

```
public java.awt.Color getBackground()  
Returns the background color.
```

Overrides

`java.awt.Component.getBackground()` in class `java.awt.Component`

Returns

The `Color` object.

getCDATAFont()

```
public java.awt.Font getCDATAFont()  
Returns the CDATA font.
```

Returns

The `Font` object.

getCDATAForeground()

`public java.awt.Color getCDATAForeground()`
Returns the CDATA foreground color.

Returns

The `Color` object.

getCommentDataFont()

`public java.awt.Font getCommentDataFont()`
Returns the Comment Data font.

Returns

The `Font` object.

getCommentDataForeground()

`public java.awt.Color getCommentDataForeground()`
Returns the Comment Data foreground color.

Returns

The `Color` object.

getEditedText()

`public java.lang.String getEditedText()`
Returns the edited text.

Returns

The `String` object containing the edited text.

getJTextPane()

`public javax.swing.JTextPane getJTextPane()`
Returns the viewer `JTextPane` component.

Returns

The `JTextPane` object used by `XMLSourceViewer`

getMinimumSize()

```
public java.awt.Dimension getMinimumSize()
```

Returns the XMLSourceView minimal size.

Overrides

javax.swing.JComponent.getMinimumSize() in class javax.swing.JComponent

Returns

The Dimension object containing the XMLSourceView minimum size.

getNodeAtOffset(int)

```
public org.w3c.dom.Node getNodeAtOffset(int i)
```

Returns the XML node at a given offset.

Parameters

i - The node offset.

Returns

The Node object from offset *i*.

getPCDATAFont()

```
public java.awt.Font getPCDATAFont()
```

Returns the PCDATA font.

Returns

The Font object.

getPCDATAForeground()

```
public java.awt.Color getPCDATAForeground()
```

Returns the PCDATA foreground color.

Returns

The Color object.

getPIDataFont()

```
public java.awt.Font getPIDataFont()
```

Returns the PI Data font.

Returns

The `Font` object

getPIDataForeground()

```
public java.awt.Color getPIDataForeground()
```

Returns the PI Data foreground color.

Returns

The `Color` object.

getPINameFont()

```
public java.awt.Font getPINameFont()
```

Returns the PI Name font.

Returns

The `Font` object.

getPINameForeground()

```
public java.awt.Color getPINameForeground()
```

Returns the PI Data foreground color.

Returns

The `Color` object.

getSymbolFont()

```
public java.awt.Font getSymbolFont()
```

Returns the NOTATION Symbol font.

Returns

The `Font` object.

getSymbolForeground()

```
public java.awt.Color getSymbolForeground()
```

Returns the NOTATION Symbol foreground color.

Returns

The `Color` object.

getTagFont()

```
public java.awt.Font getTagFont()  
Returns the Tag font.
```

Returns

The `Font` object.

getTagForeground()

```
public java.awt.Color getTagForeground()  
Returns the Tag foreground color.
```

Returns

The `Color` object.

getText()

```
public java.lang.String getText()  
Returns the XML document as a String.
```

Returns

The `String` object containing the XML document.

isEditable()

```
public boolean isEditable()  
Returns boolean to indicate whether this object is editable.
```

selectNodeAt(int)

```
public void selectNodeAt(int i)  
Moves the cursor to XML Node at offset i.
```

Parameters

`i` - The node offset.

setAttributeNameFont(Font)

```
public void setAttributeNameFont(java.awt.Font font)  
Sets the Attribute Name font.
```

Parameters

font - The new Font for Attribute Name.

setAttributeNameForeground(Color)

```
public void setAttributeNameForeground(java.awt.Color color)
```

Sets the Attribute Name foreground color.

Parameters

color - The new Color for Attribute Name.

setAttributeValueFont(Font)

```
public void setAttributeValueFont(java.awt.Font font)
```

Sets the Attribute Value font.

Parameters

font - The new Font for Attribute Value.

setAttributeValueForeground(Color)

```
public void setAttributeValueForeground(java.awt.Color color)
```

Sets the Attribute Value foreground color.

Parameters

color - The new Color for Attribute Value.

setBackground(Color)

```
public void setBackground(java.awt.Color color)
```

Sets the background color.

Overrides

javax.swing.JComponent.setBackground(java.awt.Color) in class
javax.swing.JComponent

Parameters

color - The new background Color.

setCDATAFont(Font)

```
public void setCDATAFont(java.awt.Font font)
```

Sets the CDATA font.

Parameters

`font` - The new `Font` for CDATA.

setCDATAForeground(Color)

```
public void setCDATAForeground(java.awt.Color color)
```

Sets the CDATA foreground color.

Parameters

`color` - The new `Color` for CDATA.

setCommentDataFont(Font)

```
public void setCommentDataFont(java.awt.Font font)
```

Sets the Comment font.

Parameters

`font` - The new `Font` for the XML Comments.

setCommentDataForeground(Color)

```
public void setCommentDataForeground(java.awt.Color color)
```

Sets the Comment foreground color.

Parameters

`color` - The new `Color` for Comment.

setEditable(boolean)

```
public void setEditable(boolean edit)
```

Sets the specified boolean to indicate whether this object should be editable.

Parameters

`doc` - The new boolean value.

setPCDATAFont(Font)

```
public void setPCDATAFont(java.awt.Font font)
```

Sets the PCDATA font.

Parameters

font - The new Font for PCDATA.

setPCDATAForeground(Color)

```
public void setPCDATAForeground(java.awt.Color color)
```

Sets the PCDATA foreground color.

Parameters

color - The new Color for PCDATA.

setPIDataFont(Font)

```
public void setPIDataFont(java.awt.Font font)
```

Sets the PI Data font.

Parameters

font - The new Font for PI Data.

setPIDataForeground(Color)

```
public void setPIDataForeground(java.awt.Color color)
```

Sets the PI Data foreground color.

Parameters

color - The new Color for PI Data.

setPINameFont(Font)

```
public void setPINameFont(java.awt.Font font)
```

Sets the PI Name font.

Parameters

font - The new Font for the PI Names.

setPINameForeground(Color)

```
public void setPINameForeground(java.awt.Color color)
```

Sets the PI Name foreground color.

Parameters

color - The new Color for PI Name.

setSelectedNode(Node)

```
public void setSelectedNode(org.w3c.dom.Node node)
```

Sets the cursor position at the selected XML node.

Parameters

`node` - The selected node.

setSymbolFont(Font)

```
public void setSymbolFont(java.awt.Font font)
```

Sets the NOTATION Symbol font.

Parameters

`color` - The new `Font` for NOTATION Symbol.

setSymbolForeground(Color)

```
public void setSymbolForeground(java.awt.Color color)
```

Sets the NOTATION Symbol foreground color.

Parameters

`color` - The new `Color` for NOTATION Symbol.

setTagFont(Font)

```
public void setTagFont(java.awt.Font font)
```

Sets the Tag font.

Parameters

`font` - The new `Font` for the XML Tags.

setTagForeground(Color)

```
public void setTagForeground(java.awt.Color color)
```

Sets the Tag foreground color.

Parameters

`color` - The new `Color` for the XML Tags.

setXMLDocument(Document)

```
public void setXMLDocument(org.w3c.dom.Document document)
```

Associates the XMLviewer with a XML document.

Parameters

`doc` - The Document document to display.

See Also

[getText\(\)](#)

XMLSourceViewBeanInfo Class

Syntax

```
public class XMLSourceViewBeanInfo extends java.beans.SimpleBeanInfo
```

```
java.lang.Object
|
+--java.beans.SimpleBeanInfo
|
+--oracle.xml.srcviewer.XMLSourceViewBeanInfo
```

All Implemented Interfaces

```
java.beans.BeanInfo
```

Constructors

XMLSourceViewBeanInfo()

```
public XMLSourceViewBeanInfo()
```

Methods

getIcon(int)

```
public java.awt.Image getIcon(int iconKind)
```

Overrides

```
java.beans.SimpleBeanInfo.getIcon(int) in class java.beans.SimpleBeanInfo
```

getPropertyDescriptors()

```
public java.beans.PropertyDescriptor[] getPropertyDescriptors()
```

Overrides

```
java.beans.SimpleBeanInfo.getPropertyDescriptors() in class
java.beans.SimpleBeanInfo
```

Package oracle.xml.transviewer

This chapter describes package `oracle.xml.transviewer` for Oracle XML Transviewer Beans. The Oracle XML Transviewer Beans are provided as part of XDK for Java Beans. XML Transviewer Beans facilitate the addition of graphical or visual interfaces to your XML applications.

This chapter contains these sections:

- [Package `oracle.xml.transviewer` Description](#)
- [Package `oracle.xml.transviewer` Summary](#)

Package oracle.xml.transviewer Description

Package oracle.xml.transviewer provides the bean that creates and deletes CLOB tables, lists the content of a CLOB table, and also adds, replaces, or deletes text documents in the specified CLOB table. This bean also includes a file interface to load XML documents and XSL stylesheets from the file system or from CLOB tables in the database and to edit the retrieved XML and XSL documents or files. Additionally, Database connectivity is included with the XML Transviewer beans. The beans can now connect directly to a JDBC-enabled database to retrieve and store XML and XSL files. Information about developing applications using Oracle XML Transviewer Java Beans can be found in *Oracle9i XML Developer's Kits Guide - XDK*.

Package oracle.xml.transviewer Summary

Table 16–1 Class Summary

Class	Description
DBAccess	Maintains CLOB tables that can hold multiple XML and text documents.
DBAccessBeanInfo	Provides DB Access bean info.
XMLTransformPanel	Applies XSL transformations on XML documents.
XMLTransformPanelBeanInfo	Provides XMLTransformPanel bean info.
XMLTransViewer	Can be used from the command line to edit and parse XML files, edit and apply XSL transformations and retrieve and save XML, XSL and result files in the file system or in the Oracle 9i database.

DBAccess

Syntax

```
public class DBAccess extends java.lang.Object

java.lang.Object
|
+--oracle.xml.transviewer.DBAccess
```

Description

Maintains CLOB tables that can hold multiple XML and text documents. Each table is created using the statement: CREATE TABLE tablename FILENAME CHAR(16) (UNIQUE, FILEDATA CLOB) LOB(FILEDATA) STORE AS (DISABLE STORAGE IN ROW). Each XML (or text) document is stored as a row in the table and the FILENAME field holds a unique string that is used as a key to retrieve, update, or delete the row. The document text is stored in the FILEDATA field as a CLOB object. The CLOB tables are automatically maintained by the Transviewer bean. The CLOB tables maintained by this class can be later used by the Transviewer bean. The class creates and deletes CLOB tables, lists the content of a CLOB table and also adds, replaces, or deletes text documents in the specified CLOB table.

Constructors

DBAccess()

```
public DBAccess()
```

Methods

createBLOBTable(Connection, String)

```
public boolean createBLOBTable(java.sql.Connection con, java.lang.String
tablename)
Create BLOB table
```

Parameters

con - the Connection object
tablename - the table name

Returns

true if successful

createXMLTable(Connection, String)

```
public boolean createXMLTable(java.sql.Connection con, java.lang.String  
tablename)
```

Create XML table

Parameters

con - the Connection object

tablename - the table name

Returns

true if successful

deleteBLOBName(Connection, String, String)

```
public boolean deleteBLOBName(java.sql.Connection con, java.lang.String  
tablename, java.lang.String xmlname)
```

Delete binary file from BLOB table

Parameters

con - the Connection object

tablename - the table name

xmlname - the file name

Returns

true if successful

deleteXMLName(Connection, String, String)

```
public boolean deleteXMLName(java.sql.Connection con, java.lang.String  
tablename, java.lang.String xmlname)
```

Delete file from XML table

Parameters

con - the Connection object

tablename - the table name

xmlname - the file name

Returns

true if successful

dropBLOBTable(Connection, String)

```
public boolean dropBLOBTable(java.sql.Connection con, java.lang.String  
tablename)
```

Delete BLOB table

Parameters

con - the Connection object

tablename - the table name

Returns

true if successful

dropXMLTable(Connection, String)

```
public boolean dropXMLTable(java.sql.Connection con, java.lang.String tablename)
```

Delete XML table

Parameters

con - the Connection object

tablename - the table name

Returns

true if successful

getBLOBData(Connection, String, String)

```
public byte[] getBLOBData(java.sql.Connection con, java.lang.String tablename,  
java.lang.String xmlname)
```

Retrieve binary file from BLOB table

Parameters

con - the Connection object

tablename - the table name

xmlname - the file name

Returns

file as a byte array

getNameSize()

```
public int getNameSize()
```

Returns the size of the field where the filename is kept.

Returns

filename size

getXMLData(Connection, String, String)

```
public java.lang.String getXMLData(java.sql.Connection con, java.lang.String  
tablename, java.lang.String xmlname)
```

Retrieve text file from XML table

Parameters

con - the Connection object

tablename - the table name

xmlname - the file name

Returns

file as a string

getXMLNames(Connection, String)

```
public java.lang.String[] getXMLNames(java.sql.Connection con, java.lang.String  
tablename)
```

Returns all file names in XML table

Parameters

con - the Connection object

tablename - the table name

Returns

String array with all file names in this table

getXMLTableNames(Connection, String)

```
public java.lang.String[] getXMLTableNames(java.sql.Connection con,
```

```
java.lang.String tablePrefix)
```

Gets all XML tables with names starting with a given string

Parameters

con - the Connection object

tablePrefix - table prefix string

Returns

array of all XML tables that begin with tablePrefix

insertBLOBData(Connection, String, String, byte[])

```
public boolean insertBLOBData(java.sql.Connection con, java.lang.String  
tablename, java.lang.String xmlname, byte[] xmldata)
```

Inserts binary file as a row in BLOB table

Parameters

con - the Connection object

tablename - the table name

xmlname - the file name

xmldata - byte array with file data

Returns

true if successfull

insertXMLData(Connection, String, String, String)

```
public boolean insertXMLData(java.sql.Connection con, java.lang.String  
tablename, java.lang.String xmlname, java.lang.String xmldata)
```

Inserts text file as a row in XML table

Parameters

con - the Connection object

tablename - the table name

xmlname - the file name

xmldata - string with the file data

Returns

true if successfull

isXMLTable(Connection, String)

```
public boolean isXMLTable(java.sql.Connection con, java.lang.String tablename)
```

Check if the table is XML table.

Parameters

con - the Connection object
tableName - the table name to test

Returns

true if this is XML table

replaceXMLData(Connection, String, String, String)

```
public boolean replaceXMLData(java.sql.Connection con, java.lang.String  
tablename, java.lang.String xmlname, java.lang.String xmldata)
```

Replace text file as a row in XML table

Parameters

con - the Connection object
tablename - the table name
xmlname - the file name
xmldata - string with the file data

Returns

true if successfull

xmlTableExists(Connection, String)

```
public boolean xmlTableExists(java.sql.Connection con, java.lang.String  
tablename)
```

Checks if the XML table exists

Parameters

con - the Connection object
tablename - the table name

Returns

true if the table exists

DBAccessBeanInfo

Syntax

```
public class DBAccessBeanInfo extends java.beans.SimpleBeanInfo
```

```
java.lang.Object
|
+--java.beans.SimpleBeanInfo
   |
   +--oracle.xml.transviewer.DBAccessBeanInfo
```

All Implemented Interfaces

```
java.beans.BeanInfo
```

Constructors

DBAccessBeanInfo()

```
public DBAccessBeanInfo()
Constructor
```

Methods

getIcon(int)

```
public java.awt.Image getIcon(int iconKind)
```

Overrides

```
java.beans.SimpleBeanInfo.getIcon(int) in class java.beans.SimpleBeanInfo
```

getPropertyDescriptors()

```
public java.beans.PropertyDescriptor[] getPropertyDescriptors()
```

Overrides

```
java.beans.SimpleBeanInfo.getPropertyDescriptors() in class
java.beans.SimpleBeanInfo
```

XMLTransformPanel

Syntax

```
public class XMLTransformPanel extends javax.swing.JPanel
```

```
java.lang.Object
|
+--java.awt.Component
   |
   +--java.awt.Container
      |
      +--javax.swing.JComponent
         |
         +--javax.swing.JPanel
            |
            +--oracle.xml.transviewer.XMLTransformPanel
```

All Implemented Interfaces:

```
javax.accessibility.Accessible, java.awt.image.ImageObserver,
java.awt.MenuContainer, java.io.Serializable
```

Description

XMLTransformPanel visual bean. Applies XSL transformations on XML documents. Visualizes the result. Allows editing of input XML and XSL documents/files.

Constructors

XMLTransformPanel()

```
public XMLTransformPanel()
```

The class constructor. Creates an object of type XMLTransformPanel.

XMLTransformPanelBeanInfo

Syntax

```
public class XMLTransformPanelBeanInfo extends java.beans.SimpleBeanInfo
```

```
java.lang.Object
```

```
|
```

```
+--java.beans.SimpleBeanInfo
```

```
|
```

```
+--oracle.xml.transviewer.XMLTransformPanelBeanInfo
```

All Implemented Interfaces:

```
java.beans.BeanInfo
```

Constructors

XMLTransformPanelBeanInfo()

```
public XMLTransformPanelBeanInfo()
```

Methods

getIcon(int)

```
public java.awt.Image getIcon(int iconKind)
```

Overrides

java.beans.SimpleBeanInfo.getIcon(int) in class java.beans.SimpleBeanInfo

getPropertyDescriptors()

```
public java.beans.PropertyDescriptor[] getPropertyDescriptors()
```

Overrides

java.beans.SimpleBeanInfo.getPropertyDescriptors() in class
java.beans.SimpleBeanInfo

XMLTransViewer

Syntax

```
public class XMLTransViewer extends java.lang.Object

java.lang.Object
|
+--oracle.xml.transviewer.XMLTransViewer
```

Description

Simple application that uses XMLTransformPanel. Can be used from the command line to edit and parse XML files, edit and apply XSL transformations and retrieve and save XML, XSL and result files in the file system or in the Oracle 9i database.

Constructors

XMLTransViewer()

```
public XMLTransViewer()
```

Methods

getReleaseVersion()

```
public static java.lang.String getReleaseVersion()
Returns the release version of the Oracle XML Transviewer
```

Returns

The release version string

main(String[])

```
public static void main(java.lang.String[] args)
```

Package oracle.xml.treeviewer

This chapter describes package oracle.xml.treeviewer.

The Oracle XML Transviewer Beans are provided as part of Oracle XDK for Java Beans. XML Transviewer Beans facilitate the addition of graphical or visual interfaces to your XML applications.

This chapter contains these sections:

- [Package oracle.xml.treeviewer Description](#)
- [Package oracle.xml.treeviewer Summary](#)

Package oracle.xml.treeviewer Description

Package oracle.xml.treeviewer implements the Treeviewer bean, which displays XML formatted files graphically as a tree. The branches and leaves of this tree can be manipulated with a mouse. It takes as input an `org.w3c.dom.Document` object, and recognizes XML DOM nodes as described in the class reference section of this chapter.

Information about developing applications using Oracle XML Transviewer Java Beans can be found in *Oracle9i XML Developer's Kits Guide - XDK*.

See Also: Oracle XML Developer's Kit for Java Beans at
http://otn.oracle.com/tech/xml/xdk_jbeans/content.html

Package oracle.xml.treeviewer Summary

Table 17-1 Class Summary for oracle.xml.treeviewer

Class	Description
XMLTreeView	Shows an XML document as a tree.
XMLTreeViewBeanInfo	Provides info for XMLTreeView bean.

XMLTreeView

Syntax

```
public class XMLTreeView extends javax.swing.JPanel
```

```
java.lang.Object
|
+--java.awt.Component
   |
   +--java.awt.Container
      |
      +--javax.swing.JComponent
         |
         +--javax.swing.JPanel
            |
            +--oracle.xml.treeviewer.XMLTreeView
```

All Implemented Interfaces:

```
javax.accessibility.Accessible, java.awt.image.ImageObserver,
java.awt.MenuContainer, java.io.Serializable
```

Description

Shows an XML document as a tree. Recognizes XML DOM nodes as follows: Tag, Attribute Name, Attribute Value, Comment, CDATA, PCDATA, PI Data, PI Name and NOTATION Symbol.

Takes as input an `org.w3c.dom.Document` object.

Fields

model

```
protected oracle.xml.treeviewer.XMLTreeModel model
```

scrollPane

```
protected transient javax.swing.JScrollPane scrollPane
```

theTree

```
protected transient javax.swing.JTree theTree
```

Constructors

XMLTreeView()

```
public XMLTreeView()
```

The class constructor. Creates an object of type `XMLTreeView`.

Methods

getPreferredSize()

```
public java.awt.Dimension getPreferredSize()
```

Returns the `XMLTreeView` preferred size.

Overrides

`javax.swing.JComponent.getPreferredSize()` in class `javax.swing.JComponent`

Returns

The `Dimension` object containing the `XMLTreeView` preferred size.

getTree()

```
protected javax.swing.JTree getTree()
```

getXMLTreeModel()

```
protected oracle.xml.treeviewer.XMLTreeModel getXMLTreeModel()
```

setXMLDocument(Document)

```
public void setXMLDocument(org.w3c.dom.Document document)
```

Associates the `XMLTreeViewer` with a XML document.

Parameters

`doc` - The `Document` document to display.

updateUI()

```
public void updateUI()
```

Forces the `XMLTreeView` to update/refresh UI.

Overrides

`javax.swing.JPanel.updateUI()` in class `javax.swing.JPanel`

XMLTreeViewBeanInfo

Syntax

```
public class XMLTreeViewBeanInfo extends java.beans.SimpleBeanInfo
```

```
java.lang.Object
|
+--java.beans.SimpleBeanInfo
|
+--oracle.xml.treeviewer.XMLTreeViewBeanInfo
```

All Implemented Interfaces

```
java.beans.BeanInfo
```

Constructors

XMLTreeViewBeanInfo()

Syntax

```
public XMLTreeViewBeanInfo()
```

Methods

getIcon(int)

```
public java.awt.Image getIcon(int iconKind)
```

Overrides

java.beans.SimpleBeanInfo.getIcon(int) in class java.beans.SimpleBeanInfo

getPropertyDescriptors()

```
public java.beans.PropertyDescriptor[] getPropertyDescriptors()
```

Overrides

java.beans.SimpleBeanInfo.getPropertyDescriptors() in class java.beans.SimpleBeanInfo

Part IV

Java Packages for Oracle SOAP

The Java packages described in this part implement support for Oracle SOAP in the XDK for Java.

This part contains these chapters:

- [Chapter 18, "Package oracle.soap.server"](#)
- [Chapter 19, "Package oracle.soap.transport"](#)
- [Chapter 20, "Package oracle.soap.transport.http"](#)
- [Chapter 21, "Package oracle.soap.util.xml"](#)

See Also: For more information about how to develop applications using the features of Oracle SOAP in Oracle9i, refer to these Oracle resources.

- *Oracle9i XML Developer's Kits Guide - XDK*
- *Oracle9i XML API Reference - XDK and Oracle XML DB*

Package oracle.soap.server

This chapter documents package oracle.soap.server. This package contains the classes that provide support for Oracle SOAP in the XDK for Java.

Oracle SOAP is an implementation of the Simple Object Access Protocol. Oracle SOAP is based on the SOAP open source implementation developed by the Apache Software Foundation.

This chapter contains these sections:

- [Package oracle.soap.server Description](#)
- [Package oracle.soap.server Summary](#)

Package oracle.soap.server Description

The Simple Object Access Protocol (SOAP) is a transport protocol for sending and receiving requests and responses across the Internet. It is based on XML and HTTP. SOAP is transport protocol-independent and operating system-independent. It provides the standard XML message format for all applications. SOAP uses the XML Schema standard of the World Wide Web Consortium (W3C).

Package oracle.soap.server contains the interfaces and classes that implement the API for SOAP administrative clients and the provider implementation for Java classes. These include the Service Manager and the Provider Manager. These administrative clients are services that support dynamic deployment of new services and new providers.

See Also: For background information about SOAP, refer to these Web sites.

- <http://www.w3.org/TR/SOAP/>
- <http://xml.apache.org/soap>

For information about developing applications with Oracle SOAP, refer to *Oracle9i XML Developer's Kits Guide - XDK*

Package oracle.soap.server Summary

Table 18–1 *Interfaces in package oracle.soap.server*

Interface	Description
Interface Handler	Defines the interface for a pluggable handler in the SOAP server.
Interface Provider	Defines the capabilities that must be supported for each type of service provider.
Interface ProviderManager	Defines the Provider Manager used by the SOAP engine to deploy providers, undeploy providers, and access provider deployment information.
Interface ServiceManager	Defines the Service Manager used by the SOAP engine to deploy services, undeploy services, and to access service deployment information.

Table 18–2 *Classes in package oracle.soap.server*

Class	Description
Class ContainerContext	Defines the context of the container in which the SOAP server is running.
Class Logger	Defines the capabilities that must be supported by a logger implementation.
Class ProviderDeploymentDescriptor	Defines the deployment information for a specific provider.
Class RequestContext	Defines all of the context for a SOAP request, including information that is passed to the provider and information that the provider must set before returning.
Class SOAPServerContext	defines the context of the SOAP server that is independent of the type of container in which the server is running.
Class ServiceDeploymentDescriptor	Defines the deployment information for a SOAP service, independent of its provider type.

Table 18–2 (Cont.) Classes in package oracle.soap.server

Class	Description
Class UserContext	Defines the user context for a SOAP service request.

Interface Handler

Interface oracle.soap.server.Handler

```
public interface Handler
```

Description

Handler defines the interface for a pluggable handler in the SOAP server. This class does not imply any policies about when the handler is invoked.

A handler implementation must:

- provide a no-args constructor
- be thread-safe

Table 18–3 *Methods in Handler*

Method	Description
init	One-time handler initialization.
setOptions	Set the options for the handler for subsequent use by init .
getOptions	Get this handler's options.
getName	Get this handler's name.
setName	Set the name of the handler.
destroy	One-time handler cleanup.
invoke	Invoke the requested handler as part of the specified chain type.

Fields

REQUEST_TYPE

```
public static final int REQUEST_TYPE
```

Handler invocation is part of request chain.

RESPONSE_TYPE

```
public static final int RESPONSE_TYPE
```

Handler invocation is part of response chain.

ERROR_TYPE

```
public static final int ERROR_TYPE
```

Handler invocation is part of error chain.

Methods

init

```
public abstract void init(SOAPServerContext ssc)  
    throws SOAPException
```

One-time handler initialization. This method will be invoked by the SOAP server exactly once before the server makes any invocations on the handler, allowing the handler to set up any global state. It uses any options that were set previously via `setOptions`.

Parameters

`ssc` - The SOAP server context, which contains the logger for informational messages.

Throws

`SOAPException` if unable to initialize the handler.

setOptions

```
public abstract void setOptions(Properties options)
```

Set the options for the handler for subsequent use by `init`. This method must be called before `init`.

Parameters

`options` - Options that are specific to the handler implementation.

getOptions

```
public abstract Properties getOptions()
```

Get this handler's options.

Returns

Options that are specific to the handler implementation.

setName

```
public abstract void setName(String name)
```

Set the name of the handler. This method must be called before init.

Parameters

name - The name of the handler instance.

getName

```
public abstract String getName()
```

Get this handler's name.

Returns

The name of the handler instance.

destroy

```
public abstract void destroy()  
                    throws SOAPException
```

One-time handler cleanup. This method will be invoked by the SOAP server exactly once before the server shuts down. This gives the handler the opportunity to do cleanup of global state.

Throws

SOAPException if unable to destroy.

invoke

```
public abstract void invoke(int chainType,  
                            RequestContext requestContext)  
                            throws SOAPException
```

Invoke the requested handler as part of the specified chain type. Note that execution of a chain of request handlers or response handlers will terminate immediately if any handler throws a `SOAPException`. In contrast, all handlers in an error chain will be invoked, regardless of whether or not any handler throws an exception. In the case of an exception in an error handler, the exception is logged and discarded.

Parameters

`chainType` - as follows:

- `Handler.REQUEST_TYPE` if the handler is being invoked as part of a request chain (i.e., before the service is invoked);
- `Handler.RESPONSE_TYPE` if the handler is being invoked as part of a response chain (i.e., after the service has been invoked);
- `Handler.ERROR_TYPE` if the handler is being invoked as part of an error chain (i.e., in case of an error during any one of request chain, service invocation, or response chain).

`requestContext` - The relevant request context.

Throws

`SOAPException` if handler invocation failed.

Interface Provider

Interface oracle.soap.server.Provider

```
public interface Provider
```

Provider defines the capabilities that must be supported for each type of service provider, such as Java class or stored procedure. Providers are responsible for service authorization, and parameter unmarshalling/marshalling.

Providers, aka provider instances, must be deployed to the SOAP handler. Each provider deployment must define the provider name, Java classname that implements the provider (which must be an implementation of this interface), and any number of provider-specific key-value pairs. Given the provider deployment information, the SOAP handler will interact with the providers solely through this interface.

The SOAP handler will create one instance for each deployed provider instance. It is possible to have one or more instances of each provider implementation (which is not to say that is necessarily recommended). In any event, each instance of a provider must be able to handle requests concurrently.

A provider implementation must:

- provide a no-args constructor
- be thread-safe

Table 18–4 *Methods in Provider*

Method	Description
destroy	One-time provider instance cleanup.
getId	Get this provider's unique name.
init	One-time provider instance initialization.
invoke	Invoke the requested method in the specified service, where the SOAP request is completely described in the request context.

init

```
public abstract void init(ProviderDeploymentDescriptor pd,
                        SOAPServerContext ssc)
    throws SOAPException
```

One-time provider instance initialization. This method will be invoked by the SOAP handler exactly once before the handler makes any requests to services supported by the provider, allowing the provider to set up any provider-global context.

Parameters

`pd` - The provider descriptor which contains the provider deployment information.

`ssc` - The SOAP server context, which contains the logger for informational messages.

Throws

`SOAPException` if unable to initialize and therefore unable to provide services.

destroy

```
public abstract void destroy() throws SOAPException
```

One-time provider instance cleanup. This method will be invoked by the SOAP handler exactly once before the handler shuts down. This gives the provider the opportunity to do cleanup of provider-global state.

Throws

`SOAPException` if unable to destroy.

See Also:

`init`

getId

```
public abstract String getId()
```

Get this provider's unique name.

Returns

This providers name, which is unique within the SOAP handler.

invoke

```
public abstract void invoke(RequestContext requestContext)
                        throws SOAPException
```

Invoke the requested method in the specified service, where the SOAP request is completely described in the request context.

Parameters

`requestContext` - The `RequestContext` that contains everything the provider needs to process the request.

Throws

`SOAPException` if error during method invocation for any number of reasons, including user does not have permission, method does not exist.

Interface ProviderManager

```
Interface oracle.soap.server.ProviderManager  
public interface ProviderManager
```

Provider Manager defines the interface to manage providers. The provider manager is used by the SOAP engine to deploy providers, undeploy providers, and access provider deployment information. The provider manager may cache deployment information and is responsible to maintain the cache.

The HTTP server provides security for the provider manager. The provider manager can be configured with a URL that requests must be made to in order for the request to be accepted. If a SOAP request for the provider manager is made to any other URL, the request will be rejected. This URL should be an alias to the SOAP servlet, and HTTP security can be set to control which users can post to the URL.

Table 18–5 *Methods in ProviderManager*

Method	Description
deploy	Deploy the given provider.
destroy	Cleanup the provider manager.
getRequiredRequestURI	Get the URI that provider manager requests must be made to in order to be accepted.
init	Initialize the provider manager.
list	Get a list of provider ids for all providers that have been deployed.
query	Get the deployment descriptor for the given provider.
setServiceManager	Make the service manager that is being used to manage service deployment information available to the provider manager.
undeploy	Undeploy the given provider, and return its descriptor.

Methods

init

```
public abstract void init(Properties options) throws SOAPException
```

Initialize the provider manager.

Parameters

`options` - The options required to setup access to the deployment information.

Throws

SOAPException if unable to access the deployment information.

destroy

```
public abstract void destroy() throws SOAPException
```

Cleanup the provider manager.

Throws

SOAPException if unable to cleanup the provider manager.

setServiceManager

```
public abstract void setServiceManager(ServiceManager serviceManager)
```

Make the service manager that is being used to manage service deployment information available to the provider manager. The provider manager may use the service manager to ensure that a provider is not undeployed as long as any services are deployed under that provider.

Parameters

`providerManager` - The provider manager that is managing provider deployment information for the SOAP server.

getRequiredRequestURI

```
public abstract String getRequiredRequestURI()
```

Get the URI that provider manager requests must be made to in order to be accepted. Requests made to any other URI must be rejected.

Returns

The request URI for provider manager requests, or null if any URI can be used.

undeploy

```
public abstract ProviderDeploymentDescriptor undeploy(String providerId)
    throws SOAPException
```

Undeploy the given provider, and return its descriptor.

Parameters

`providerId` - The id of the provider to undeploy.

Returns

The descriptor containing the deployment information for the provider that has been undeployed.

Throws

`SOAPException` if the provider is not found or failed to undeploy.

deploy

```
public abstract void deploy(ProviderDeploymentDescriptor pd)
    throws SOAPException
```

Deploy the given provider.

Parameters

`pd` - The provider descriptor for the provider to deploy.

Throws

`SOAPException` if unable to deploy.

query

```
public abstract ProviderDeploymentDescriptor query(String providerId) throws
    SOAPException
```

Get the deployment descriptor for the given provider.

Parameters

`providerId` - The id of the provider.

Returns

The descriptor containing the deployment information for the given provider.

Throws

`SOAPException` if the provider is not found.

list

```
public abstract String[] list() throws SOAPException
```

Get a list of provider ids for all providers that have been deployed.

Returns

An array of deployed provider ids.

Throws

`SOAPException` if unable to list provider ids.

Interface ServiceManager

```
Interface oracle.soap.server.ServiceManager  
public interface ServiceManager
```

Service Manager defines the interface to manage services. The Service Manager is used by the SOAP engine to deploy services, undeploy services, and to access service deployment information. The Service Manager may cache deployment information and is responsible for maintaining the cache.

The HTTP server provides security for the service manager. The service manager can be configured with a URL that requests must be made to in order for the request to be accepted. If a SOAP request for the service manager is made to any other URL, the request will be rejected. This URL should be an alias to the SOAP servlet, and HTTP security can be set to control which users can post to the specified URL.

Methods

init

```
public abstract void init(Properties options,  
                          ProviderManager providerManager)  
    throws SOAPException
```

Initialize the service manager. The implementation should be able to handle a null value for the provider manager.

Parameters

`options` - The options required to setup access to the service deployment information.

`providerManager` - The provider manager that is managing provider deployment information for the SOAP server, or null if the provider manager is not supplied. The service manager may want to use the provider manager to confirm the existence of the provider when a new service is deployed.

Throws

SOAPException if unable to access the service deployment information.

destroy

```
public abstract void destroy() throws SOAPException
```

Cleanup the service manager.

Throws

SOAPException if unable to cleanup the service manager.

getRequiredRequestURI

```
public abstract String getRequiredRequestURI()
```

Get the URI that service manager requests must be made to in order to be accepted. Requests made to any other URI must be rejected.

Returns

The request URI for service manager requests, or null if any URI can be used.

undeploy

```
public abstract ServiceDeploymentDescriptor undeploy(String serviceId)
    throws SOAPException
```

Undeploy the given service, and return its descriptor.

Parameters

serviceId - The URI of the service to undeploy.

Returns

The descriptor containing the deployment information for the service that has been undeployed.

Throws

SOAPException if the service is not found or failed to undeploy.

deploy

```
public abstract void deploy(ServiceDeploymentDescriptor sd) throws SOAPException
```

Deploy the given service.

Parameters

`sd` - The service descriptor for the service to deploy.

Throws

`SOAPException` if unable to deploy.

query

```
public abstract ServiceDeploymentDescriptor query(String serviceId) throws SOAPException
```

Get the deployment descriptor for the given service.

Parameters

`serviceId` - The unique URI of the service.

Returns

The descriptor containing the deployment information for the given service.

Throws

`SOAPException` if the service is not found.

list

```
public abstract String[] list() throws SOAPException
```

Get a list of service ids for all services that have been deployed, regardless of the provider.

Returns

An array of deployed service ids.

Throws

`SOAPException` if unable to list service ids.

Class ContainerContext

Class oracle.soap.server.ContainerContext

```
java.lang.Object
|
+----oracle.soap.server.ContainerContext
```

```
public class ContainerContext
    extends Object
```

ContainerContext defines the context of the container in which the SOAP server is running. The actual content depends on the environment in which the server is running, such as in a servlet engine. This class should contain only container-specific content.

Fields

SERVLET_CONTAINER

```
public static final String SERVLET_CONTAINER
```

The value for a servlet container type.

Constructors

ContainerContext

```
public ContainerContext()
```

Methods

setContainerType

```
public void setContainerType(String containerType)
```

Set the container type.

Parameters

containerType - The type of container in which the SOAP server is running.

getContainerType

```
public String getContainerType()
```

Returns the container type.

Returns

The type of container in which the SOAP server is running.

getHttpServlet

```
public HttpServlet getHttpServlet()
```

Returns the HTTP servlet if the container type is `SERVLET_CONTAINER`.

Returns

The `HttpServlet` that is processing the SOAP request, or null if the servlet attribute is not set.

setHttpServlet

```
public void setHttpServlet(HttpServlet servlet)
```

Set the HTTP servlet for a SOAP server running in a `SERVLET_CONTAINER` type of container.

Parameters

`servlet` - The `HttpServlet` that is processing the SOAP request.

getAttribute

```
public Object getAttribute(String name)
```

Returns the attribute with the given name, or null if there is no attribute by that name.

Parameters

`name` - A `String` specifying the name of the attribute.

Returns

An `Object` containing the value of the attribute, or null if no attribute exists matching the given name.

See Also:`getAttributeNames`**getAttributeNames**

```
public Enumeration getAttributeNames()
```

Returns an Enumeration containing the attribute names available within this SOAP context.

Returns

An Enumeration of attribute names.

See Also:`getAttribute`**setAttribute**

```
public void setAttribute(String name,  
                        Object object)
```

Binds an object to a given attribute name in this SOAP context. If the name specified is already used for an attribute, this method will remove the old attribute and bind the name to the new attribute. Neither the name nor the object may be null.

Parameters

`name` - A non-null String specifying the name of the attribute.

`object` - An non-null Object representing the attribute to be bound.

removeAttribute

```
public void removeAttribute(String name)
```

Removes the attribute with the given name from the context. After removal, subsequent calls to `getAttribute(java.lang.String)` to retrieve the attribute's value will return null.

Parameters

`name` - A String specifying the name of the attribute to be removed.

`object` - An Object representing the attribute to be bound.

Class Logger

Class oracle.soap.server.Logger

```
java.lang.Object
|
+----oracle.soap.server.Logger
```

```
public abstract class Logger
extends Object
```

Logger defines the capabilities that must be supported by a logger implementation. The logger is used to persistently record error and informational messages.

Each log request specifies the severity, and the information should be logged iff the severity is at least as high as the specified severity.

The order of severity in increasing order is:

- SEVERITY_ERROR
- SEVERITY_STATUS
- SEVERITY_DEBUG

For example, if the severity is set to SEVERITY_STATUS, any log request with severity of either SEVERITY_STATUS or SEVERITY_ERROR will be logged.

Fields

SEVERITY_ERROR

```
public static final int SEVERITY_ERROR
```

SEVERITY_STATUS

```
public static final int SEVERITY_STATUS
```

SEVERITY_DEBUG

```
public static final int SEVERITY_DEBUG
```

SEVERITY_INVALID

```
protected static final int SEVERITY_INVALID
```

SEVERITY_NAMES

```
public static String SEVERITY_NAMES[]
```

DEFAULT_SEVERITY

```
public static final int DEFAULT_SEVERITY
```

OPTION_SEVERITY

```
public static final String OPTION_SEVERITY
```

Configuration option that specifies the severity for the logger.

m_severity

```
protected int m_severity
```

The logger's severity setting.

Constructors**Logger**

```
public Logger()
```

Methods**getSeverityValue**

```
protected final int getSeverityValue(String severityName)
```

Get the severity value associated with the given severity name.

Parameters

`severityName` - The name of the serverity level (e.g., error).

Returns

The severity (`SEVERITY_xxx`).

getSeverityName

```
protected final String getSeverityName(int severity)
```

Get the severity name associated with the given severity.

Parameters

`severity` - The severity level (`SEVERITY_xxx`).

Returns

The severity name.

init

```
public abstract void init(Properties options,  
                          ContainerContext context)  
    throws SOAPException
```

One-time initialization of the logger with its configuration parameters.

Parameters

`options` - The configuration options for the logger.

`context` - The context of the container in which the SOAP server is running, which includes information that may be used by the logger.

Throws

`SOAPException` if unable to initialize the logger.

getSeverity

```
public int getSeverity()
```

Get the current severity.

Returns

The current severity setting for the logger.

setSeverity

```
public void setSeverity(int severity)
```

Set the current severity.

Parameters

severity - The new severity setting for the logger.

isLoggable

```
public boolean isLoggable(int severity)
```

Determine if a message would be logged at the given severity level.

Parameters

severity - The severity level to check.

Returns

True if a message would be logged at the given severity level, else false.

log

```
public abstract void log(String msg,  
                        int severity)
```

Log the given message at the given severity.

Parameters

msg - The message to log.

severity - The severity at which to log the information.

log

```
public abstract void log(String msg,  
                        Throwable t,  
                        int severity)
```

Log the given message and exception at the given severity.

Parameters

msg - The message to log.

`t` - The throwable exception to log.

`severity` - The severity at which to log the information.

log

```
public abstract void log(Throwable t,  
                        int severity)
```

Log the given exception at the given severity.

Parameters

`t` - The throwable exception to log.

`severity` - The severity at which to log the information.

Class ProviderDeploymentDescriptor

Class oracle.soap.server.ProviderDeploymentDescriptor

```
java.lang.Object
|
+----oracle.soap.server.ProviderDeploymentDescriptor
```

```
public final class ProviderDeploymentDescriptor
extends Object
implements Serializable
```

`ProviderDeploymentDescriptor` defines the deployment information for a specific provider. Different providers may be deployed using the same implementation and be distinguished only by their provider descriptor.

Table 18–6 Class members in *ProviderDeploymentDescriptor*

Class Member	Description
<code>ProviderDeploymentDescriptor</code>	Class constructor. Construct a new instance of a provider descriptor.
<code>fromXML</code>	Build a provider descriptor from the given document.
<code>getClassName</code>	Returns the name of the class that implements this provider.
<code>getId</code>	Returns the unique id for this provider.
<code>getOptions</code>	Get the provider-specific options.
<code>getProviderType</code>	Returns the provider type.
<code>setClassName</code>	Set the name of the class that implements this provider.
<code>setId</code>	Set the provider id.
<code>setOptions</code>	Set the options.
<code>setProviderType</code>	Set the provider type.
<code>toString</code>	Overrides <code>toString</code> in class <code>Object</code> .
<code>toXML</code>	Write out the service deployment descriptor as XML.

Constructors

ProviderDeploymentDescriptor

```
public ProviderDeploymentDescriptor()
```

Construct a new instance of a provider descriptor.

Methods

setId

```
public void setId(String id)
```

Set the provider id.

Parameters

`id` - The unique provider id.

getId

```
public String getId()
```

Returns the unique id for this provider.

Returns

This provider's unique id.

setClassname

```
public void setClassname(String classname)
```

Set the name of the class that implements this provider.

Parameters

`classname` - The name of the implementing class.

getClassname

```
public String getClassname()
```

Returns the name of the class that implements this provider.

Returns

The classname.

setProviderType

```
public void setProviderType(String providerType)
```

Set the provider type.

Parameters

providerType - The provider type.

getProviderType

```
public String getProviderType()
```

Returns the provider type.

Returns

This provider's type.

setOptions

```
public void setOptions(Hashtable options)
```

Set the options.

Parameters

options - The name-value pairs that represent the provider implementation-specific options for this service.

getOptions

```
public Hashtable getOptions()
```

Get the provider-specific options.

Returns

The name -value pairs that represent the provider-specific options for this service.

fromXML

```
public static ProviderDeploymentDescriptor fromXML(Element root)
```

Build a provider descriptor from the given document.

Parameters

`root` - The root of the document that represents the XML provider descriptor.

Returns

The `ProviderDeploymentDescriptor` for the given XML.

toXML

```
public void toXML(Writer pr)
```

Write out the service deployment descriptor as XML.

Parameters

`pr` - The writer for the XML output.

toString

```
public String toString()
```

Overrides

`toString` in class `Object`

Class RequestContext

Class oracle.soap.server.RequestContext

```
java.lang.Object
|
+----oracle.soap.server.RequestContext
```

```
public class RequestContext
extends Object
```

RequestContext defines all of the context for a SOAP request, including information that is passed to the provider and information that the provider must set before returning. Note that the provider is given the request Envelope and is therefore responsible to unmarshall the request parameters. Similarly, the provider is required to marshall the response, although the response envelope must also be set by the provider, as it may be needed by a pluggable handler.

The following information is provided by the SOAP engine to the Provider, meaning that the provider can utilize this information in `Provider.invoke`:

- `getEnvelope` - the envelope containing the request
- `getServiceDeploymentDescriptor` - the service deployment descriptor for the service in which the method is being invoked
- `getServiceId` - the URI of the service
- `getUserContext` - the security context describing the user invoking the method in the service
- `getMethodName` - the name of the method being invoked in the service.

The following information must be given by the Provider to the SOAP engine:

- `setResponseBytes` - this is the marshalled response. Given a `Response`, it can be created by building the response envelope and then marshalling the envelope.
- `setResponseEnvelope` - this is the response envelope, which is logically equivalent to the response bytes.
- `getRequestEncodingStyle` - the encoding style to use for the response in case of an error (if not set, defaults to `Constants.NS_URI_SOAP_ENC`, which is SOAP encoding). If the provider cares about this, it should set this value as soon

as it can in case of an exception. The provider might use the same encoding as the request or as one of the parameters.

Table 18–7 Class members in RequestContext

Member	Description
RequestContext	Default constructor.
getMethodName	Get the method name for this SOAP request.
getRequestEncodingStyle	Get the encoding style that was used on the request.
getRequestEnvelope	Get the envelope that represents the actual SOAP request.
getResponseBytes	Get the response stream for this SOAP request.
getResponseEnvelope	Get the envelope that represents the SOAP response.
getResponseMap	Get the mapping registry that must be used to serialize the SOAP response.
getServiceDeploymentDescriptor	Get the service deployment descriptor for the requested service.
getServiceId	Get the service id (URI) for this SOAP request.
getUserContext	Get the user context for this SOAP request.
setMethodName	Set the method name for this SOAP request.
setRequestEncodingStyle	Set the encoding style that was used on the request.
setRequestEnvelope	Set the envelope that represents the actual SOAP request.
setResponseBytes	Set the response stream for this SOAP request.
setResponseEnvelope	Set the envelope that represents the SOAP response.
setResponseMap	Set the mapping registry that must be used to serialize the SOAP response envelope.

Table 18–7 (Cont.) Class members in RequestContext

Member	Description
setServiceDeploymentDescriptor	Set the service deployment descriptor for the requested service.
setServiceId	Set the service id (URI) for this SOAP request.
setUserContext	Set the user context for this SOAP request.

Constructors

RequestContext

```
public RequestContext()
```

Default constructor for this class.

Methods

setRequestEnvelope

```
public void setRequestEnvelope(Envelope envelope)
```

Set the envelope that represents the actual SOAP request.

Parameters

`envelope` - The SOAP envelope.

getRequestEnvelope

```
public Envelope getRequestEnvelope()
```

Get the envelope that represents the actual SOAP request.

Returns

The SOAP envelope.

setResponseEnvelope

```
public void setResponseEnvelope(Envelope envelope)
```

Set the envelope that represents the SOAP response.

Parameters

`envelope` - The SOAP response envelope.

getResponseEnvelope

```
public Envelope getResponseEnvelope()
```

Get the envelope that represents the SOAP response.

Returns

The SOAP response envelope.

setResponseMap

```
public void setResponseMap(SOAPMappingRegistry smr)
```

Set the mapping registry that must be used to serialize the SOAP response envelope.

Parameters

`smr` - The mapping registry for the SOAP response envelope.

getResponseMap

```
public SOAPMappingRegistry getResponseMap()
```

Get the mapping registry that must be used to serialize the SOAP response.

Returns

The mapping registry for the SOAP response envelope.

setResponseBytes

```
public void setResponseBytes(ByteArrayOutputStream bytes)
```

Set the response stream for this SOAP request.

Parameters

`bytes` - The `ByteArrayOutputStream` that contains the response.

getResponseBytes

```
public ByteArrayOutputStream getResponseBytes()
```

Get the response stream for this SOAP request.

Returns

The ByteArrayOutputStream that contains the response.

setRequestEncodingStyle

```
public void setRequestEncodingStyle(String requestEncodingStyle)
```

Set the encoding style that was used on the request.

Parameters

`requestEncodingStyle` - The request encoding style.

getRequestEncodingStyle

```
public String getRequestEncodingStyle()
```

Get the encoding style that was used on the request.

Returns

The request encoding style.

setServiceDeploymentDescriptor

```
public void setServiceDeploymentDescriptor(ServiceDeploymentDescriptor  
serviceDeploymentDescriptor)
```

Set the service deployment descriptor for the requested service.

Parameters

`serviceDeploymentDescriptor` - The service deployment descriptor for this request.

getServiceDeploymentDescriptor

```
public ServiceDeploymentDescriptor getServiceDeploymentDescriptor()
```

Get the service deployment descriptor for the requested service.

Returns

The service deployment descriptor for this request, or null if the provider is an `AutonomousProvider`.

setMethodName

```
public void setMethodName(String methodName)
```

Set the method name for this SOAP request. The method name is in the envelope, but it can be "cached" here by the server as a convenience.

Parameters

`methodName` - The method name that is being invoked in the service.

getMethodName

```
public String getMethodName()
```

Get the method name for this SOAP request.

Returns

The the method name being invoked.

setServiceId

```
public void setServiceId(String serviceId)
```

Set the service id (URI) for this SOAP request.

Parameters

`serviceId` - The URI for the service to which this request is directed.

getServiceId

```
public String getServiceId()
```

Get the service id (URI) for this SOAP request.

Returns

The URI for the service to which this request is directed.

setUserContext

```
public void setUserContext(UserContext userContext)
```

Set the user context for this SOAP request.

Parameters

`userContext` - The user context.

getUserContext

```
public UserContext getUserContext()
```

Get the user context for this SOAP request.

Returns

The user context

Class SOAPServerContext

Class oracle.soap.server.SOAPServerContext

```
java.lang.Object
|
+----oracle.soap.server.SOAPServerContext
```

```
public class SOAPServerContext
extends Object
```

SOAPServerContext defines the context of the SOAP server that is independent of the type of container in which the server is running.

Table 18–8 Class members in SOAPServerContext

Class Member	Description
SOAPServerContext	Default constructor.
getAttribute	Returns the attribute with the given name, or null if there is no attribute by that name.
getAttributeNames	Returns an Enumeration containing the attribute names available within this SOAP context.
getGlobalContext	Returns the global context.
getLogger	Returns the SOAP logger.
removeAttribute	Removes the attribute with the given name from the context.
setAttribute	Binds an object to a given attribute name in this SOAP context.
setGlobalContext	Set the global context, which contains SOAP server-wide objects.
setLogger	Set the logger, which is used for text-based logging of informational and debug messages.

Constructors

SOAPServerContext

```
public SOAPServerContext()
```

Default constructor.

Methods

getGlobalContext

```
public Hashtable getGlobalContext()
```

Returns the global context.

Returns

The global context that contains SOAP server-wide objects, or null if the attribute is not set.

setGlobalContext

```
public void setGlobalContext(Hashtable globalContext)
```

Set the global context, which contains SOAP server-wide objects.

Parameters

`globalContext` - The global context.

setLogger

```
public void setLogger(Logger logger)
```

Set the logger, which is used for text-based logging of informational and debug messages.

Parameters

`logger` - The SOAP logger.

getLogger

```
public Logger getLogger()
```

Returns the SOAP logger.

Returns

The SOAP logger, which is used to log informational and debug messages.

getAttribute

```
public Object getAttribute(String name)
```

Returns the attribute with the given name, or null if there is no attribute by that name.

Parameters

name - A String specifying the name of the attribute.

Returns

An Object containing the value of the attribute, or null if no attribute exists matching the given name.

getAttributeNames

```
public Enumeration getAttributeNames()
```

Returns an Enumeration containing the attribute names available within this SOAP context.

Returns

An Enumeration of attribute names.

getAttribute

```
setAttribute  
public void setAttribute(String name,  
                        Object object)
```

Binds an object to a given attribute name in this SOAP context. If the name specified is already used for an attribute, this method will remove the old attribute and bind the name to the new attribute. Neither the name nor the object may be null.

Parameters

name - A non-null String specifying the name of the attribute.

`object` - An non-null Object representing the attribute to be bound.

removeAttribute

```
public void removeAttribute(String name)
```

Removes the attribute with the given name from the context. After removal, subsequent calls to `getAttribute(java.lang.String)` to retrieve the attribute's value will return null.

Parameters

`name` - A String specifying the name of the attribute to be removed.

`object` - An Object representing the attribute to be bound.

setAttribute

```
public void setAttribute(String name,  
                        Object object)
```

Binds an object to a given attribute name in this SOAP context. If the name specified is already used for an attribute, this method will remove the old attribute and bind the name to the new attribute. Neither the name nor the object may be null.

Parameters

`name` - A non-null String specifying the name of the attribute.

`object` - An non-null Object representing the attribute to be bound.

Class ServiceDeploymentDescriptor

Class oracle.soap.server.**ServiceDeploymentDescriptor**

```
java.lang.Object
|
+----oracle.soap.server.ServiceDeploymentDescriptor
```

```
public final class ServiceDeploymentDescriptor
extends Object
implements Serializable
```

`ServiceDeploymentDescriptor` defines the deployment information for a SOAP service, independent of its provider type. The class supports any number of named provider options, which allows the descriptor to be easily extended (without code changes) for new types of providers.

Fields

SERVICE_TYPE_RPC

```
public static final int SERVICE_TYPE_RPC
```

SERVICE_TYPE_MESSAGE

```
public static final int SERVICE_TYPE_MESSAGE
```

SCOPE_REQUEST

```
public static final int SCOPE_REQUEST
```

SCOPE_SESSION

```
public static final int SCOPE_SESSION
```

SCOPE_APPLICATION

```
public static final int SCOPE_APPLICATION
```

Constructors

ServiceDeploymentDescriptor

```
public ServiceDeploymentDescriptor()
```

Construct a new service descriptor.

Methods

setId

```
public void setId(String id)
```

Set the service id, which must be a valid URI.

Parameters

`id` - The service URI.

getId

```
public String getId()
```

Get the service id.

Returns

The service id, which is a URI.

setProviderId

```
public void setProviderId(String providerId)
```

Set the id of the provider for this service.

Parameters

`providerId` - The provider's id for this service.

getProviderId

```
public String getProviderId()
```

Get the provider id for this service.

Returns

The provider id.

setMethods

```
public void setMethods(String methods[])
```

Set the list of methods that are provided by this service.

Parameters

methods - The list of provided methods.

getMethods

```
public String[] getMethods()
```

Get the list of methods that are provided by this service.

Returns

The list of provided methods.

setScope

```
public void setScope(int scope)
```

Set the execution scope.

Parameters

scope - The execution scope, which is one of the SCOPE_XXX constants.

getScope

```
public int getScope()
```

Get the scope.

Returns

The scope, which is one of the SCOPE_XXX constants.

setServiceType

```
public void setServiceType(int serviceType)
```

Set the service type. DLD: explain RPC vs one-way message.

Parameters

`serviceType` - The service type, which is one of the `SERVICE_TYPE_XXX` constants.

getServiceType

```
public int getServiceType()
```

Get the service type.

Returns

The service type, which is one of the `SERVICE_TYPE_XXX` constants.

setProviderType

```
public void setProviderType(String providerType)
```

Set the provider type.

Parameters

`providerType` - The provider type, which can be any string. The provider type is used to validate the XML service descriptor (for the provider-specific options).

getProviderType

```
public String getProviderType()
```

Get the provider type.

Returns

The provider type.

setProviderOptions

```
public void setProviderOptions(Hashtable providerOptions)
```

Set the provider-specific options.

Parameters

`providerOptions` - The name-value pairs that represent the provider-specific options for this service.

getProviderOptions

```
public Hashtable getProviderOptions()
```

Get the provider-specific options.

Returns

The name-value pairs that represent the provider-specific options for this service.

setFaultListener

```
public void setFaultListener(String faultListener[])
```

Set the fault listener list.

Parameters

`faultListener` - The list of class names that are fault listeners for this service.

getFaultListener

```
public String[] getFaultListener()
```

Get the fault listener list.

Returns

The list of class names that are fault listeners for this service.

buildFaultRouter

```
public SOAPFaultRouter buildFaultRouter()
```

Get the fault router.

Returns

The fault router that is built from the service's fault listeners.

setTypeMappings

```
public void setTypeMappings(TypeMapping typeMappings[])
```

Set the XML-Java type mappings, which define how to deserialize XML into Java and serialize Java into XML.

Parameters

`typeMappings` - The type mappings.

getTypeMappings

```
public TypeMapping[] getTypeMappings()
```

Get the XML-Java type mappings, which define how to deserialize XML into Java and serialize Java into XML.

Returns

The type mappings.

setSqlMap

```
public void setSqlMap(Hashtable sqlMap)
```

Set the map that maps from SQL type to Java type.

Parameters

`sqlMap` - The SQL type to Java class map.

getSqlMap

```
public Hashtable getSqlMap()
```

Get the SQL type to Java type map.

Returns

The SQL type to Java class map.

setDefaultSMRClass

```
public void setDefaultSMRClass(String defaultSMRClass)
```

Set the default SOAP mapping registry class.

Parameters

`defaultSMRClass` - The default SOAP mapping registry class.

getDefaultSMRClass

```
public String getDefaultSMRClass()
```

Get the default SOAP mapping registry class.

Returns

The default SOAP mapping registry class.

isMethodValid

```
public boolean isMethodValid(String methodName)
```

Determine if the given method is valid for this service.

Returns

true if the method is valid for this service, else false.

fromXML

```
public static ServiceDeploymentDescriptor fromXML(Element root)
```

Populate the `ServiceDeploymentDescriptor` with information from the given document, which is the XML representation of the descriptor.

Parameters

`root` - The root of the XML document that represents the service descriptor.

Returns

The `ServiceDeploymentDescriptor` that contains the information from the document.

Throws

`IllegalArgumentException` if invalid document.

toXML

```
public void toXML(Writer pr)
```


Write out the service deployment descriptor as XML.

Parameters

`pr` - The writer for the XML output.

toString

```
public String toString()
```

Get a printable representation of this descriptor.

Overrides

`toString` in class `Object`

buildSOAPMappingRegistry

```
public static SOAPMappingRegistry  
buildSOAPMappingRegistry(ServiceDeploymentDescriptor sdd)
```

Utility to generate an XML serialization registry from all the type mappings registered into a deployment descriptor.

Parameters

`dd` - the deployment descriptor

Returns

the xml serialization registry

buildSqlClassMap

```
public static Hashtable buildSqlClassMap(ServiceDeploymentDescriptor sdd)  
throws SOAPException
```

Utility to generate a map from SQL type to Java Class using the type mapping information from the deployment descriptor.

Parameters

`sdd` - The service deployment descriptor to use.

Returns

The type to Class map.

Throws

SOAPException if failed to generate map.

Class UserContext

Class oracle.soap.server.UserContext

```
java.lang.Object
|
+----oracle.soap.server.UserContext
```

```
public class UserContext
    extends Object
```

UserContext defines the user context for a SOAP service request. Several attributes are pre-defined, and set and get methods are provided for those. In addition, the provider may define additional attributes using `getAttribute` and `setAttribute`.

Note that the `HttpServletRequest` and `HttpSession` do not really belong here, but they are required by the `JavaProvider`.

Constructors

UserContext

```
public UserContext()
```

Default constructor.

Methods

getRequestURI

```
public String getRequestURI()
```

Returns the URI of the request.

Returns

The URI of the request.

setRequestURI

```
public void setRequestURI(String uri)
```

Set the URI of the request.

Parameters

`uri` - Request URI

getCertificate

```
public Object getCertificate()
```

Returns the user certificate.

Returns

The user certificate for the user making SOAP request, or null if this attribute is not set.

setCertificate

```
public void setCertificate(Object certificate)
```

Set the user certificate.

Parameters

`certificate` - The user certificate for the user making the SOAP request.

getHttpServlet

```
public HttpServlet getHttpServlet()
```

Returns the HTTP servlet.

Returns

The HttpServlet that is processing the SOAP request, or null if the servlet attribute is not set.

setHttpServlet

```
public void setHttpServlet(HttpServlet servlet)
```

Set the HTTP servlet.

Parameters

`servlet` - The HttpServlet that is processing the SOAP request.

getHttpSession public HttpSession getHttpSession()

Returns the HTTP session.

Returns

The `HttpSession` for the SOAP request, or null if the session attribute is not set.

setHttpSession

```
public void setHttpSession(HttpSession session)
```

Set the HTTP session.

Parameters

`servlet` - The `HttpSession` for the SOAP request.

getRemoteAddress

```
public String getRemoteAddress()
```

Returns the Internet Protocol (IP) address of the client that sent the request.

Returns

The remote client's IP address.

setRemoteAddress

```
public void setRemoteAddress(String remoteAddress)
```

Set the remote IP address of the client.

Parameters

`remoteAddress` - The IP address of the client making the SOAP request.

getRemoteHost

```
public String getRemoteHost()
```

Returns the host name of the client that sent the request.

Returns

The remote client's host name.

setRemoteHost

```
public void setRemoteHost(String remoteHost)
```

Set the host name of the client making the SOAP request

Parameters

`remoteHost` - The host name of the client making the SOAP request.

getSecureChannel

```
public boolean getSecureChannel()
```

Returns an indication whether the channel is secure.

Returns

`true` if the channel is secure, else `false`.

setSecureChannel

```
public void setSecureChannel(boolean secureChannel)
```

Set the indicator of whether the channel is secure.

Parameters

`secureChannel` - `true` if the channel is secure, else `false`.

getUsername

```
public String getUsername()
```

Returns the protocol-specific username.

Returns

The protocol-specific username for the SOAP request, or `null` if this attribute is not set.

setUsername

```
public void setUsername(String username)
```

Set the protocol-specific username.

Parameters

username - The protocol-specific username for the SOAP request.

getAttribute

```
public Object getAttribute(String name)
```

Returns the attribute with the given name, or null if there is no attribute by that name.

Parameters

name - A String specifying the name of the attribute.

Returns

An Object containing the value of the attribute, or null if no attribute exists matching the given name.

See Also:

getAttributeNames

getAttributeNames

```
public Enumeration getAttributeNames()
```

Returns an Enumeration containing the attribute names available within this SOAP context.

Returns

An Enumeration of attribute names.

See Also:

getAttribute

setAttribute

```
public void setAttribute(String name,  
                        Object object)
```

Binds an object to a given attribute name in this SOAP context. If the name specified is already used for an attribute, this method will remove the old attribute and bind the name to the new attribute. Neither the name nor the object may be null.

Parameters

`name` - A non-null `String` specifying the name of the attribute.

`object` - An non-null `Object` representing the attribute to be bound.

removeAttribute

```
public void removeAttribute(String name)
```

Removes the attribute with the given name from the context. After removal, subsequent calls to `getAttribute(java.lang.String)` to retrieve the attribute's value will return null.

Parameters

`name` - A `String` specifying the name of the attribute to be removed.

`object` - An `Object` representing the attribute to be bound.

Package oracle.soap.transport

This chapter documents package oracle.soap.transport. This package contains the classes that provide support for Oracle SOAP in the XDK for Java.

Oracle SOAP is an implementation of the Simple Object Access Protocol. Oracle SOAP is based on the SOAP open source implementation developed by the Apache Software Foundation.

This chapter contains these sections:

- [Package oracle.soap.transport Description](#)
- [Package oracle.soap.transport Summary](#)
- [Interface OracleSOAPTransport](#)

Package oracle.soap.transport Description

The Simple Object Access Protocol (SOAP) is a transport protocol for sending and receiving requests and responses across the Internet. It is based on XML and HTTP, and it is not blocked by firewalls.

SOAP is transport protocol-independent and operating system-independent. It provides the standard XML message format for all applications. SOAP uses the XML Schema standard of the World Wide Web Consortium (W3C).

See Also:

- <http://www.w3.org/TR/SOAP/>
- <http://xml.apache.org/soap>
- *Oracle9i XML Developer's Kits Guide - XDK*

Package oracle.soap.transport Summary

Package oracle.soap.transport contains interface oracle.soap.transport.**OracleSOAPTransport**, which defines Oracle specific transport extensions.

Table 19–1 *Methods in interface OracleSOAPTransport*

Method	Description
getProperties	Get the connection properties.
setProperties	Set the connection properties.
close	Close the transport and perform any clean up.

Interface OracleSOAPTransport

Interface oracle.soap.transport.**OracleSOAPTransport**

```
public interface OracleSOAPTransport
    extends SOAPTransport
```

This interface defines Oracle specific transport extensions.

getProperties

```
public abstract Properties getProperties()
```

Get the connection properties.

Returns

Connection properties

setProperties

```
public abstract void setProperties(Properties prop)
```

Set the connection properties.

Parameters

prop - connection properties

close

```
public abstract void close()
```

Close the transport and perform any clean up.

Package oracle.soap.transport.http

This chapter documents package `oracle.soap.transport.http`. This package contains the classes that provide support for Oracle SOAP in the XDK for Java.

Oracle SOAP is an implementation of the Simple Object Access Protocol. Oracle SOAP is based on the SOAP open source implementation developed by the Apache Software Foundation.

This chapter contains these sections:

- [Package `oracle.soap.transport.http` Description](#)
- [Package `oracle.soap.transport.http` Summary](#)
- [Class `OracleSOAPHTTPConnection`](#)

Package oracle.soap.transport.http Description

The Simple Object Access Protocol (SOAP) is a transport protocol for sending and receiving requests and responses across the Internet. It is based on XML and HTTP. SOAP is transport protocol-independent and operating system-independent. It provides the standard XML message format for all applications. SOAP uses the XML Schema standard of the World Wide Web Consortium (W3C).

Package `oracle.soap.transport.http` contains class `OracleSOAPHTTPConnection`, which implements `OracleSOAPTransport`.

The Oracle SOAP client API supports a pluggable transport, allowing the client to easily change the transport. Available transports include HTTP and HTTPS (secure HTTP).

See Also:

- <http://www.w3.org/TR/SOAP/>
- <http://xml.apache.org/soap>
- *Oracle9i XML Developer's Kits Guide - XDK*

Package oracle.soap.transport.http Summary

Package oracle.soap.transport.http contains class oracle.soap.transport.http.OracleSOAPHTTPConnection, which implements Oracle-specific transport extensions in OracleSOAPTransport.

Table 20–1 *Fields in class OracleSOAPHTTPConnection*

Field	Description
ALLOW_USER_INTERACTION	Sets user interaction.
AUTH_TYPE	Defines http auth type (basic/digest).
CIPHERS	Defines cipher suites used for HTTPS (colon separated list of cipher suites).
PASSWORD	Defines HTTP password.
PROXY_AUTH_TYPE	Defines proxy auth type (basic/digest).
PROXY_HOST	Defines proxy host.
PROXY_PASSWORD	Defines proxy password.
PROXY_PORT	Defines proxy ports.
PROXY_USERNAME	Defines proxy username.
STATUS_LINE	Used to get HTTP status line from HTTP headers (getHeaders).
USERNAME	Defines HTTP username.
WALLET_LOCATION	Defines wallet location used for HTTPS.
WALLET_PASSWORD	Defines password used for HTTPS.

Table 20–2 *Constructor and Methods in class OracleSOAPHTTPConnection*

Member	Description
Constructor	
OracleSOAPHTTPConnection	Constructor that takes Properties as an arg.
Methods	
getProperties	Get the connection properties.

Table 20–2 (Cont.) Constructor and Methods in class OracleSOAPHTTPConnection

Member	Description
close	Close the transport and perform any clean up.
finalize	Overrides finalize in class Object.
getHeaders	Return access to headers generated by the protocol.
getProperties	Get the connection properties.
receive	Return a buffered reader to receive back the response to whatever was sent to x.
send	Request that an envelope be posted to the given URL.
setProperties	Set the connection properties.

Class OracleSOAPHTTPConnection

```
oracle.soap.transport.http.OracleSOAPHTTPConnection
java.lang.Object
|
+----oracle.soap.transport.http.OracleSOAPHTTPConnection

public class OracleSOAPHTTPConnection
extends Object

Implements OracleSOAPTransport.
```

Fields

ALLOW_USER_INTERACTION

```
public static final String ALLOW_USER_INTERACTION

property to set user interaction
```

STATUS_LINE

```
public static final String STATUS_LINE

property used to get HTTP status line from HTTP headers (getHeaders)
```

PROXY_HOST

```
public static final String PROXY_HOST

property used for defining proxy host
```

PROXY_PORT

```
public static final String PROXY_PORT

property used for defining proxy port
```

PROXY_AUTH_TYPE

```
public static final String PROXY_AUTH_TYPE
```

property used for defining proxy auth type (basic/digest)

PROXY_USERNAME

```
public static final String PROXY_USERNAME
```

property used for defining proxy username

PROXY_PASSWORD

```
public static final String PROXY_PASSWORD
```

property used for defining proxy password

AUTH_TYPE

```
public static final String AUTH_TYPE
```

property used for defining http auth type (basic/digest)

USERNAME

```
public static final String USERNAME
```

property used for defining http username

PASSWORD

```
public static final String PASSWORD
```

property used for defining http password

WALLET_LOCATION

```
public static final String WALLET_LOCATION
```

property used for defining wallet location used for HTTPS

WALLET_PASSWORD

```
public static final String WALLET_PASSWORD
```

property used for defining wallet password used for HTTPS

CIPHERS

```
public static final String CIPHERS
```

property used for defining cipher suites used for HTTPS (colon separated list of cipher suites).

Constructors

OracleSOAPHTTPConnection

```
public OracleSOAPHTTPConnection(Properties prop)
```

Constructor that takes Properties as an arg.

Parameters

prop - connection properties.

Methods

setPropertyies

```
public void setPropertyies(Properties prop)
```

Set the connection properties.

Parameters

`prop` - connection properties

getPropertyies

```
public Properties getPropertyies()
```

Get the connection properties.

Returns

connection properties

send

```
public void send(URL sendTo,  
                String action,  
                Hashtable headers,  
                Envelope env,  
                SOAPMappingRegistry smr,  
                int timeout) throws SOAPException
```

This method is used to request that an envelope be posted to the given URL. The response (if any) must be gotten by calling the `receive()` function. Soap clients should not use this method directly, but should instead use `org.apache.soap.rpc.Call`.

Parameters

`sendTo` - the URL to send the envelope to

`action` - the SOAPAction header field value

`headers` - any other header fields to go to as protocol headers

`env` - the envelope to send

`smr` - the XML<->Java type mapping registry (passed on)

`ctx` - the request SOAPContext

Throws

SOAPException with appropriate reason code if problem.

receive

```
public BufferedReader receive()
```

Return a buffered reader to receive back the response to whatever was sent to whatever. Soap clients should not use this method directly but use `org.apache.soap.rpc.Call` instead.

Returns

A reader to read the results from or null if that's not possible.

getHeaders

```
public Hashtable getHeaders()
```

Return access to headers generated by the protocol. Soap clients should not use this method directly but use `org.apache.soap.rpc.Call` instead.

Returns

A hashtable containing all the headers.

close

```
public void close()
```

Closes the connection. Once this method has been called, the `BufferedReader` returned by `receive` method may be closed and should not be used. Calling this method will free resources without having the garbage collector run.

finalize

```
public void finalize()  
    throws Throwable
```

Overrides

`finalize` in class `Object`

Package oracle.soap.util.xml

This chapter documents package `oracle.soap.util.xml`. This package contains the classes that provide support for Oracle SOAP in the XDK for Java.

Oracle SOAP is an implementation of the Simple Object Access Protocol. Oracle SOAP is based on the SOAP open source implementation developed by the Apache Software Foundation.

This chapter contains these sections:

- [Package `oracle.soap.util.xml` Description](#)
- [Package `oracle.soap.util.xml` Summary](#)
- [Class `XmlUtils`](#)

Package oracle.soap.util.xml Description

The Simple Object Access Protocol (SOAP) is a transport protocol for sending and receiving requests and responses across the Internet. It is based on XML and HTTP. Oracle SOAP is based on the SOAP open source implementation developed by the Apache Software Foundation.

SOAP is transport protocol-independent and operating system-independent. It provides the standard XML message format for all applications. SOAP uses the XML Schema standard of the World Wide Web Consortium (W3C).

Package oracle.soap.util.xml contains class `XmlUtils`, which implements the API for SOAP clients to generate the XML documents that compose a request for a SOAP service and handle the SOAP response. Oracle SOAP processes requests from any client that sends a valid SOAP request.

See Also:

- <http://www.w3.org/TR/SOAP/>
- <http://xml.apache.org/soap>
- *Oracle9i XML Developer's Kits Guide - XDK*

Package oracle.soap.util.xml Summary

Package oracle.soap.util.xml contains class oracle.soap.util.xml.OracleSOAPHTTPConnection, which implements Oracle-specific transport extensions in OracleSOAPTransport.

Table 21-1 *Members in class Class XmlUtils*

Member	Description
Constructor	
XmlUtils	Default constructor.
Methods	
extractServiceId	Get the service id from the envelope.
extractMethodName	Get the method name from the envelope.
parseXml(String)	Parse the given XML file and return the XML document.
parseXml(Reader)	Parse the given XML source and return the XML document
parseXml(InputStream)	Parse the contents of the XML InputStream and return the XML document
createDocument	Create a Document.

Class XmlUtils

Class oracle.soap.util.xml.**XmlUtils**

```
java.lang.Object
|
+----oracle.soap.util.xml.XmlUtils
```

```
public class XmlUtils
```

Extends Object.

Constructors

XmlUtils

```
public XmlUtils()
```

Default constructor.

Methods

extractServiceId

```
public static String extractServiceId(Envelope envelope) throws SOAPException
```

Get the service id from the envelope. It is the namespace URI of the first body entry.

Throws

`SOAPException` if unable to get service URI from envelope.

extractMethodName

```
public static String extractMethodName(Envelope envelope) throws SOAPException
```

Get the method name from the envelope. It is the name of the first body entry.

Throws

`SOAPException` if unable to get method name from envelope.

parseXml

```
public static Document parseXml(String filename)  
    throws SOAPException
```

Parse the given XML file and return the XML document.

Parameters

`filename` - The full path to the XML file.

Throws

`SOAPException` if file not found or parse error.

parseXml

```
public static Document parseXml(Reader reader)  
    throws SOAPException
```

Parse the given XML source and return the XML document.

Parameters

`reader` - Reader for XML.

Throws

SOAPException if file not found or parse error.

parseXml

```
public static Document parseXml(InputStream is)
                               throws SOAPException
```

Parse the contents of the XML InputStream and return the XML document.

Parameters

is - input stream source

Throws

SOAPException if there are parse errors or IO errors.

createDocument

```
public static Document createDocument()
                               throws SOAPException
```

Create a Document.

Throws

SOAPException if cannot create Document.

Part V

Java Packages for Oracle XML DB

The packages in this part document Java packages that implement the Document Object Model (DOM) and the service provider interface (SPI) in Oracle XML DB.

This part contains these chapters:

- [Chapter 22, "Package oracle.xdb.dom"](#)
- [Chapter 23, "Package oracle.xdb.spi"](#)

See Also: For more information about how to develop applications using the features of Oracle XML DB, refer to:

- *Oracle9i XML Database Developer's Guide - Oracle XML DB*
- *Oracle9i XML API Reference - XDK and Oracle XML DB*

Package oracle.xdb.dom

The classes described in this chapter are contained in package oracle.xdb.dom and implement the Java DOM API for XMLType.

This chapter contains these sections:

- [Package oracle.xdb.dom Description](#)
- [Package oracle.xdb.dom Class Summary](#)

Package oracle.xml.dom Description

The classes described in this chapter implement the Java DOM API for XMLType and are contained in package oracle.xml.dom. Oracle XML DB supports the Document Object Model (DOM) as set forth by the W3C DOM Recommendation. For more information about the DOM, refer to <http://www.w3.org/DOM/>.

See Also: For more information about developing applications using the features of Oracle XML DB, see *Oracle9i XML Database Developer's Guide - Oracle XML DB*

In addition to implementing the W3C DOM Recommendation, the DOM API for Oracle XML DB provides Oracle-specific extensions.

Package oracle.xdb.dom Class Summary

Table 22–1 *Classes in oracle.xdb.dom*

Class	Description
XDBAttribute Class	Implements the W3C DOM Node interface for interacting with XOBs.
XDBCData Class	Implements the W3C text interface.
XDBCharData Class	Implements the W3C CharacterData interface
XDBComment Class	Implements the W3C Comment interface.
XDBDocument Class	Implements the W3C Document interface.
XDBDomImplementation Class	Implements the W3C DomImplementation interface, which opens a JDBC connection to the server.
XDBElement Class	Implements the W3C Element interface.
XDBNamedNodeMap Class	Implements the W3C Named Node Map interface.
XDBNode Class	Implements the W3C Node interface.
XDBNodeList Class	Implements the W3C Node List interface.
XDBProcInst Class	Implements the W3C the W3C DOM ProcessingInstruction interface.
XDBText Class	Implements the W3C Text interface.
XMLType Class	Implements the native datatype XMLType in Oracle XML DB.

Table 22–2 *Classes in oracle.xdb.dom mapped to org.w3c.dom Implementation*

Class	Implements
XDBAttribute Class	org.w3c.dom.Attribute
XDBCData Class	org.w3c.dom.CData
XDBCharData Class	org.w3c.dom.CharData
XDBComment Class	org.w3c.dom.Comment
XDBDocument Class	org.w3c.dom.Document
XDBDomImplementation Class	org.w3c.dom.DomImplementation

Table 22–2 (Cont.) Classes in oracle.xdb.dom mapped to org.w3c.dom

Class	Implements
XDBElement Class	org.w3c.dom.Element
XDBNamedNodeMap Class	org.w3c.dom.NamedNodeMap
XDBNode Class	org.w3c.dom.Node
XDBNodeList Class	org.w3c.dom.NodeList
XDBProcInst Class	org.w3c.dom.ProcInst
XDBText Class	org.w3c.dom.Text
XMLType Class	N/a. This class is specific to Oracle.

XDBAttribute Class

Syntax

```
public class XDBAttribute
```

```
oracle.xdb.dom.XDBAttribute
```

Description

This class implements `org.w3c.dom.Attribute`, the W3C DOM Node interface for interacting with XDBs.

XDBCData Class

Syntax

```
public class XDBCData
```

```
oracle.xdb.dom.XDBCData
```

Description

This class implements `org.w3c.dom.CData`, the W3C text interface.

XDBCharData Class

Syntax

```
public class XDBCharData
```

```
oracle.xdb.dom.XDBCharData
```

Description

This class implements `org.w3c.dom.CharacterData`, the W3C CharacterData interface.

XDBComment Class

Syntax

```
public class XDBComment
```

```
oracle.xdb.dom.XDBComment
```

Description

This class implements the `org.w3c.dom.Comment` interface.

XDBDocument Class

Syntax

```
public class XDBDocument
```

```
oracle.xdb.dom.XDBDocument
```

Description

This class implements the `org.w3c.dom.Document` interface.

Methods

XDBDocument()

Syntax

```
public XDBDocument()
```

Description

Creates new Document. Can be used in server only.

XDBDocument(byte[])

Syntax

```
public XDBDocument(byte[] source)
```

Description

Populates Document from source. Can be used in server only.

Parameters

`source` - Bytes containing XML text.

XDBDocument(Connection)

Syntax

```
public XDBDocument(java.sql.Connection conn)
```

Description

Opens connection for caching Document source.

Parameters

`conn` - Connection to be used for Document.

XDBObject(Connection, byte[])

Syntax

```
public XDBObject(java.sql.Connection conn, byte[] source)
```

Description

Connection for caching bytes for Document source.

Parameters

`conn` - Connection to be used for Document.

`source` - Bytes containing XML text.

XDBObject(Connection, String)

Syntax

```
public XDBObject(java.sql.Connection conn, java.lang.String source)
```

Description

Opens connection for caching string containing XML text.

Parameters

`source` - String containing XML text.

`conn` - Connection to be used for String.

XDBObject(String)

Syntax

```
public XDBObject(java.lang.String source)
```

Description

String containing XML text. Can be used in server only.

XDBDomImplementation Class

Syntax

```
public class XDBDomImplementation
```

```
oracle.xdb.dom.XDBDomImplementation
```

Description

This class implements `org.w3c.dom.DomImplementation`.

Methods

XDBDomImplementation()

Syntax

```
public XDBDomImplementation()
```

Description

Opens a JDBC connection to the server.

XDBElement Class

Syntax

```
public class XDBElement
```

```
oracle.xdb.dom.XDBElement
```

Description

This class implements `org.w3c.dom.Element`.

XDBNamedNodeMap Class

Syntax

```
public class XDBNamedNodeMap
```

```
oracle.xdb.dom.XDBNamedNodeMap
```

Description

This class implements `org.w3c.dom.NamedNodeMap`.

XDBNode Class

Syntax

```
public abstract class XDBNode
```

```
oracle.xml.dom.XDBNode
```

Description

This class implements `org.w3c.dom.Node`, the W3C DOM Node interface for interacting with XOBs.

Methods

`write(OutputStream, String, short)`

Description

Writes the XML for this Node (and all subnodes) to an `OutputStream`. If the `OutputStream` is a `ServletOutputStream`, the servlet output is committed and the data is written using a native stream mechanism.

Syntax

```
public void write(java.io.OutputStream s, java.lang.String charEncoding, short indent)
```

Parameters

`s` - The stream to write the output to

`charEncoding` - The IANA char code (e.g. "ISO-8859")

`indent` - Number of chars to indent nested elements

XDBNodeList Class

Syntax

```
public class XDBNodeList
```

```
oracle.xdb.dom.XDBNodeList
```

Description

This class implements `org.w3c.dom.NodeList`.

XDBProcInst Class

Syntax

```
public class XDBProcInst
```

```
oracle.xdb.dom.XDBProcInst
```

Description

This class implements `org.w3c.dom.ProcessingInstruction`, the W3C DOM ProcessingInstruction interface.

XDBText Class

Syntax

```
public class XDBText
```

```
oracle.xdb.dom.XDBText
```

Description

This class implements `org.w3c.dom.Text`.

XMLType Class

```
oracle.xdb.XMLType

public class XMLType
```

Description

XMLType class implements the native datatype `XMLType` in Oracle XML DB, which supports storage and manipulation of XML in the server. Multiple storage options including structured XML and Character Large Object (CLOB) are available with XMLType.

The native structured XML storage is a decomposition of XML into underlying object-relational structures (automatically created and managed by Oracle) for better SQL queriability. CLOB storage is an un-decomposed storage that retains an image of the original XML, including white space.

Fields

Table 22–3 Field Summary for XMLType Class

Field	Description
SQL_TYPECODE	static int
SQL_TYPENAME	static java.lang.String

`_SQL_TYPECODE`

```
public static final int _SQL_TYPECODE
```

`_SQL_TYPENAME`

```
public static final java.lang.String _SQL_TYPENAME
```

Constructors

XMLType

```
public XMLType()
    throws java.sql.SQLException
```


Methods

Table 22–4 Method Summary for XMLType

Method	Description
getORADataFactory	Get the oradata factory for this XMLType.
toDatum	Construct an image from the XMLType data.
createXML	Create an XMLType given a connection (<code>conn</code>) and the specified datatype and object.
getStringVal	Get the string value containing the XML data from the XMLType.
getClobVal	Get the CLOB value containing the XML data from the XMLType.
extract	Function to extract the given set of nodes from the XMLType.
existsNode	Function to check for the existence of the given set of nodes in the XMLType.
transform	Function to transform the XMLType using the given XSL document.
isFragment	Function to check if the XMLType is a regular document or a document fragment.
isSchemaValid	Function to check if the XMLType is schema based.
getDOM	Gets the DOM document associated with the XMLType.
getBytesValue	Gets the bytes value containing the XML data from the XMLType.

getORADataFactory

```
public static oracle.sql.ORADataFactory getORADataFactory()
```

Get the oradata factory for this XMLType. This is required as part of the oradata interface

Returns

the ORADataFactory assoicated with this XMLType.

toDatum

```
public oracle.sql.Datum toDatum(java.sql.Connection conn)
    throws java.sql.SQLException
```

Construct an image from the XMLType data. This function pickles the XMLType bytes and returns the bytes as a Datum.

Parameters

conn - the connection to be used for Datum creation

Returns

the pickled image

Throws

java.sql.SQLException -

createXML

```
public static XMLType createXML(oracle.sql.OPAQUE opq)
    throws java.sql.SQLException
```

Create an XMLType given the opaque type containing the XMLType bytes

Parameters

opq - the opaque data object from which the XMLType is to be generated

Returns

the created XMLType

Throws

java.sql.SQLException -

createXML

```
public static XMLType createXML(java.sql.Connection conn,
    java.lang.String xmlval)
    throws java.sql.SQLException
```

Create an XMLType given the string containing the XML data

Parameters

conn - the connection object to be used.

xmlval - the string containing the XML data

Returns

the created XMLType

Throws

java.sql.SQLException -

createXML

```
public static XMLType createXML(java.sql.Connection conn,  
                                oracle.sql.CLOB xmlval)  
                                throws java.sql.SQLException
```

Create an XMLType given a CLOB containing the XML data

Parameters

conn - the connection object to be used.

xmlval - the CLOB containing the XML data

Returns

the created XMLType

Throws

java.sql.SQLException

getStringVal

```
public java.lang.String getStringVal()  
                        throws java.sql.SQLException
```

Get the string value containing the XML data from the XMLType.

Returns

the string containing the XML data bytes.

Throws

java.sql.SQLException -

getClobVal

```
public oracle.sql.CLOB getClobVal()  
                        throws java.sql.SQLException
```

Get the CLOB value containing the XML data from the XMLType.

Returns

the CLOB containing the XML data bytes.

Throws

`java.sql.SQLException` -

createXML

```
public static XMLType createXML(java.sql.Connection conn,  
                                org.w3c.dom.Document domdoc)  
    throws java.sql.SQLException
```

Create an XMLType given an instance of the DOM document

Parameters

`domdoc` - the DOM Document which represents the DOM tree

Returns

the constructed XMLType.

Throws

`java.sql.SQLException` -

extract

```
public XMLType extract(java.lang.String xpath,  
                       java.lang.String nsmap)  
    throws java.sql.SQLException
```

Function to extract the given set of nodes from the XMLType. This set of nodes is specified by the XPath expression. The original XMLType remains unchanged. Works only in the thick case.

Parameters

`xpath` - the xpath expression which specifies the nodes to search for.

`nsmap` - the map of namespaces which resolves the prefixes in the xpath expression. format is "xmlns=a.com xmlns:b=b.com"

Returns

XMLType which contains the extracted nodes. null if no nodes match the specified expression.

existsNode

```
public boolean existsNode(java.lang.String xpath,  
                          java.lang.String nsmap)
```

throws `java.sql.SQLException`

Function to check for the existence of the given set of nodes in the XMLType. This set of nodes is specified by the xpath expression.

Parameters

`xpath` - the xpath expression which specifies the nodes to search for.

`nsmmap` - the map of namespaces which resolves the prefixes in the xpath expression. format is "xmlns=a.com xmlns:b=b.com"

Returns

TRUE if specified nodes exist in the XMLType else FALSE

transform

```
public XMLType transform(XMLType xsldoc,  
                          java.lang.String parammap)  
    throws java.sql.SQLException
```

Function to transform the XMLType using the given XSL document. The new (transformed) XML document is returned.

Parameters

`xsldoc` - the XSL document to be applied to the XMLType

`parammap` - the top level parameters to be passed to the XSL transformation. This should be of the format "a=b c=d e=f". This can be null.

Returns

the transformed XMLType

isFragment

```
public boolean isFragment()  
    throws java.sql.SQLException
```

Function to check if the XMLType is a regular document or a document fragment.

Returns

TRUE if doc is a fragment else FALSE

isSchemaValid

```
public boolean isSchemaValid(java.lang.String schurl,  
                              java.lang.String elname)
```

throws `java.sql.SQLException`

Function to check if the `XMLType` is schema based.

Parameters

`schurl` - the URL of the schema to be validated against; if this is null then the documents own schema URL is used (if one exists).

Returns

TRUE if doc is schema based else FALSE

getDOM

```
public org.w3c.dom.Document getDOM()
    throws java.sql.SQLException
```

Gets the DOM document associated with the `XMLType`. This document is the `org.w3c.dom.Document`. The caller can perform all the DOM operations on the Document. If the document is a binary document the `getDOM` function will return null.

Returns

the DOM Document object associated with the `XMLType`

getBytesValue

```
public byte[] getBytesValue()
    throws java.sql.SQLException
```

Gets the bytes value containing the XML data from the `XMLType`.

Package oracle.xdb.spi

This chapter documents package oracle.xdb.spi. The classes contained in oracle.xdb.spi implement the service provider drivers that provide the application with common access to JNDI and JDBC interfaces.

This chapter contains these sections:

- [Package oracle.xdb.spi Description](#)
- [Package oracle.xdb.spi Class Summary](#)

Package oracle.xdb.spi Description

The classes contained in `oracle.xdb.spi` implement the JNDI API and the JNDI service provider drivers that provide the application with access to file systems and standard Web protocols.

- JNDI (Java Naming and Directory Interface) is the API for connecting Java programs to naming and directory services such as DNS, LDAP and NDS.
- JNDI SPI (Service Provider Interface) is the API for writing directory drivers.
- JDBC (Java DataBase Connectivity) is a programming interface that enables Java applications to access a database using the SQL language. JDBC is the Java counterpart of Microsoft's ODBC.

The application is written to the JNDI API, and the directory drivers are written to the JNDI SPI (Service Provider Interface). The package `oracle.xdb.spi` contains Oracle-specific extensions to the public standard.

In this release, the Oracle XML DB implementation of JNDI SPI supports the `javax.naming` interface. Directory, attribute, and event operations are not yet supported.

See Also: *Oracle9i XML Database Developer's Guide - Oracle XML DB* for information about developing applications that use these interfaces.

Package oracle.xdb.spi Class Summary

The classes in oracle.xdb.spi implement core JNDI and JNDI SPI interfaces for Oracle XML DB.

Table 23–1 *Classes in package oracle.xdb.spi*

Class	Description
XDBContext Class	Implements the <code>javax.naming.context</code> interface
XDBContextFactory Class	Implements <code>javax.naming.spi.InitialContextFactory</code>
XDBNameParser Class	Implements <code>javax.naming.NameParser</code>
XDBNamingEnumeration Class	Implements <code>javax.naming.NamingEnumeration</code>
XDBResource Class	Implements the core features for the Oracle XML DB JNDI service provider interface (SPI)
XDBResourceContext Class	Implements <code>javax.naming.context</code>

XDBContext Class

Description of XDBContext

This class implements the Java naming and context interface for Oracle XML DB, which extends `javax.naming.context`. The current implementation has no federation support, which makes it completely unaware of the existence of other namespaces.

Syntax of XDBContext

```
public class XDBContext
```

```
oracle.xml.spi.XDBContext
```

Methods

XDBContext()

Description

Constructor for class XDBContext. The options are listed in the following table.

Syntax	Parameters
<code>public XDBContext(java.util.Hashtable env)</code>	Takes the environment to the resource.
<code>public XDBContext(java.util.Hashtable env, java.lang.String path)</code>	Takes the environment and path to the resource.

Parameters

`env` - environment to describe the properties of context.

`path` - initial path for the context.

XDBContextFactory Class

Description of XDBContextFactory

This class implements `javax.naming.context.`

Syntax of XDBContextFactory

```
public class XDBContextFactory
```

```
oracle.xdb.spi.XDBContextFactory
```

Constructors

XDBContextFactory()

Description

Constructor for class XDBContextFactory

Syntax

```
public XDBContextFactory()
```

XDBNameParser Class

Description

Implements `javax.naming.NameParser`

Syntax

```
public class XDBNameParser
```

```
oracle.xdb.spi.XDBNameParser
```

XDBNamingEnumeration Class

Description

Implements `javax.naming.NamingEnumeration`

Syntax

```
public class XDBNamingEnumeration
```

```
oracle.xdb.spi.XDBNamingEnumeration
```

XDBResource Class

Description

This class implements the core features for the Oracle XML DB JNDI service provider interface (SPI). The current implementation has no federation support, being completely unaware of the existence of other namespaces.

Syntax

```
public class XDBResource extends java.lang.Object
```

```
java.lang.Object
|
+--oracle.xml.db.spi.XDBResource
```

Methods

Table 23–2 *Methods of XDBResource*

Method	Description
XDBResource(Hashtable)	Takes the environment and path to the resource
XDBResource(Hashtable, String)	Takes the environment and path to the resource as a string.
getAuthor()	Returns author of the resource.
getComment()	Returns the DAV comment of the resource.
getContent()	Returns the content of the resource.
getContentType()	Returns the content type of the resource.
getCreateDate()	Returns the create date of the resource.
getDisplayName()	Returns the display name of the resource.
getLanguage()	Returns the language of the resource.
getLastModDate()	Returns the last modification date of the resource.
getOwnerId()	Returns the owner ID of the resource.
setACL(String)	Sets the ACL on the resource.

Table 23–2 (Cont.) Methods of XDBResource

Method	Description
setAuthor(String)	Sets the author of the resource.
setComment(String)	Sets the DAV comment of the resource.
setContent(Object)	Sets the content of the resource.
setContentTypes(String)	Sets the content type of the resource.
setCreateDate(Date)	Sets the creation date of the resource.
setDisplayname(String)	Sets the display name of the resource.
setInheritedACL(String)	Sets the ACL on the resource.
setLanguage(String)	Sets the language of the resource.
setLastModDate(Date)	Sets the last modification date of the resource.
setOwnerId(long)	Sets the owner ID of the resource.

XDBResource(Hashtable)

Description

Takes the environment and path to the resource.

Syntax

```
public XDBResource(java.util.Hashtable env)
```

Parameters

`env` - Environment passed in

`path` - Path to the resource

XDBResource(Hashtable, String)

Description

Takes the environment and path to the resource as a string.

Syntax

```
public XDBResource(java.util.Hashtable env, java.lang.String path)
```

getAuthor()

Description

Returns the author of the resource

Syntax

```
public java.lang.String getAuthor()
```

getComment()

Description

Returns the DAV comment of the resource (DAV is for Web distributed authoring and versioning)

Syntax

```
public java.lang.String getComment()
```

getContent()

Description

Returns the content of the resource

Syntax

```
public java.lang.Object getContent()
```

getContentType()

Description

Returns the content type of the resource

Syntax

```
public java.lang.String getContentType()
```

getCreateDate()

Description

Returns the creation date of the resource

Syntax

```
public java.util.Date getCreateDate()
```

getDisplayName()**Description**

Returns the display name of the resource

Syntax

```
public java.lang.String getDisplayName()
```

getLanguage()**Description**

Returns the Language of the resource

Syntax

```
public java.lang.String getLanguage()
```

getLastModDate()**Description**

Returns the last modification date of the resource

Syntax

```
public java.util.Date getLastModDate()
```

getOwnerId()**Description**

Returns the owner id of the resource

Syntax

```
public long getOwnerId()
```

setACL(String)**Description**

Sets the ACL on the resource

Syntax

```
public void setACL(java.lang.String aclpath)
```

Parameters

`aclpath` - the path to the ACL resource

setAuthor(String)

Description

Sets the author of the resource

Syntax

```
public void setAuthor(java.lang.String authname)
```

Parameters

`authname` - author of the resource

setComment(String)

Description

Sets the DAV comment of the resource (for Web distributed authoring and versioning)

Syntax

```
public void setComment(java.lang.String davcom)
```

Parameters

`davcom` - DAV comment of the resource

setContent(Object)

Description

Sets the content of the resource

Syntax

```
public void setContent(java.lang.Object xmlobj)
```

Parameters

`xmlobj` - content of the resource

setContentTypes(String)**Description**

Sets the content type of the resource

Syntax

```
public void setContentTypes(java.lang.String conttypes)
```

Parameters

`conttypes` - content type of the resource

setCreateDate(Date)**Description**

Sets the creation date of the resource

Syntax

```
public void setCreateDate(java.util.Date create)
```

Parameters

`create` - creation date of the resource

setDisplayNames(String)**Description**

Sets the display name of the resource

Syntax

```
public void setDisplayNames(java.lang.String dnames)
```

Parameters

`dnames` - display name of the resource

setInheritedACL(String)

Description

Sets the ACL on the resource, the ACL is copied from the specified resource.

Syntax

```
public void setInheritedACL(java.lang.String aclpath)
```

Parameters

`aclpath` - path to the ACL to be set

setLanguage(String)

Description

Sets the language of the resource

Syntax

```
public void setLanguage(java.lang.String lang)
```

Parameters

`lang` - language of the resource

setLastModDate(Date)

Description

Sets the last modification date of the resource

Syntax

```
public void setLastModDate(java.util.Date d)
```

Parameters

`d` - last modification date of the resource

setOwnerId(long)

Description

Sets the owner id of the resource

Syntax

```
public void setOwnerId(long ownerId)
```

Parameters

`ownerid` - owner id of the resource

XDBResourceContext Class

Description of XDBResourceContext

This class provides core Oracle XML DB features for JNDI SPI. The current implementation has no federation support, being completely unaware of the existence of other namespaces. This class implements `javax.naming.context`.

Syntax of XDBResourceContext

```
public class XDBResourceContext extends oracle.xdb.spi.XDBBaseContext
```

```
oracle.xdb.spi.XDBBaseContext
|
+--oracle.xdb.spi.XDBResourceContext
```

Methods of XDBResourceContext

Table 23–3 *Methods of XDBResourceContext*

Method	Description
<code>XDBResourceContext(Hashtable)</code>	Implements the environment.
<code>getAuthor()</code>	Returns author of the resource
<code>getContentType()</code>	Returns content type of the resource
<code>getCreateDate()</code>	Returns creation date of the resource
<code>getDavComment()</code>	Returns the DAV comment of the resource
<code>getDisplayName()</code>	Returns the display name of the resource
<code>getEnvironment()</code>	Returns a copy of the environment of this context
<code>getLanguage()</code>	Returns the language of the resource
<code>getLastModDate()</code>	Returns last modification date of the resource
<code>getOwnerId()</code>	Returns the owner ID of the resource
<code>setAuthor(String)</code>	Sets the author of the resource
<code>setContentType(String)</code>	Sets the content type of the resource
<code>setCreateDate(Date)</code>	Sets the creation date of the resource

Table 23–3 (Cont.) Methods of XDBResourceContext

Method	Description
setDavComment(String)	Sets the DAV comment of the resource
setDisplayDisplayName(String)	Sets the display name of the resource
setLanguage(String)	Sets the language of the resource
setLastModDate(Date)	Sets the last modification date of the resource
setOwnerId(long)	Sets the owner ID of the resource.

XDBResourceContext(Hashtable)

Description

Implements the environment.

Syntax

```
public XDBResourceContext(java.util.Hashtable env)
```

Parameters

env - Environment for the context

getAuthor()

Description

Returns the author of the resource

Syntax

```
public java.lang.String getAuthor()
```

getContentType()

Description

Returns the content type of the resource

Syntax

```
public java.lang.String getContentType()
```

getCreateDate()

Description

Returns the creation date of the resource

Syntax

```
public java.util.Date getCreateDate()
```

getDavComment()

Description

Returns the DAV comment of the resource

Syntax

```
public java.lang.String getDavComment()
```

getDisplayName()

Description

Returns the display name of the resource

Syntax

```
public java.lang.String getDisplayName()
```

getEnvironment()

Description

Retrieves the environment properties for this context. The result is a new copy of the environment; changes made to the returned object have no effect on the context.

Syntax

```
public java.util.Hashtable getEnvironment()
```

Overrides

`XDBBaseContext.getEnvironment()` in class `XDBBaseContext`

Returns

a copy of the environment of this context

getLanguage()

Description

Returns the Language of the resource

Syntax

```
public java.lang.String getLanguage()
```

getLastModDate()

Description

Returns the last modification date of the resource

Syntax

```
public java.util.Date getLastModDate()
```

getOwnerId()

Description

Returns the owner id of the resource

Syntax

```
public long getOwnerId()
```

setAuthor(String)

Description

Sets the author of the resource

Syntax

```
public void setAuthor(java.lang.String authname)
```

Parameters

authname - author of the resource

setContentTypes(String)

Description

Sets the content type of the resource

Syntax

```
public void setContentType(java.lang.String conttype)
```

Parameters

conttype - content type of the resource

setCreateDate(Date)

Description

Sets the creation date of the resource

Syntax

```
public void setCreateDate(java.util.Date ccreate)
```

Parameters

ccreate - creation date of the resource

setDavComment(String)

Description

Sets the DAV comment of the resource

Syntax

```
public void setDavComment(java.lang.String davcom)
```

Parameters

davcom - DAV comment of the resource

setDisplayDisplayName(String)

Description

Sets the display name of the resource

Syntax

```
public void setDisplayName(java.lang.String dname)
```

Parameters

dname - display name of the resource

setLanguage(String)

Description

Sets the language of the resource

Syntax

```
public void setLanguage(java.lang.String lang)
```

Parameters

lang - language of the resource

setLastModDate(Date)

Description

Sets the last modification date of the resource

Syntax

```
public void setLastModDate(java.util.Date d)
```

Parameters

d - last modification date of the resource

setOwnerId(long)

Description

Sets the owner id of the resource

Syntax

```
public void setOwnerId(long ownerid)
```

Parameters

ownerid - owner ID of the resource

Index

Symbols

- _atomic -
 - oracle.xml.parser.schema.XSDTypeConstants._atomic, 7-54
- _base64 -
 - oracle.xml.parser.schema.XSDTypeConstants._base64, 7-54
- _collapse -
 - oracle.xml.parser.schema.XSDTypeConstants._collapse, 7-54
- _hex -
 - oracle.xml.parser.schema.XSDTypeConstants._hex, 7-54
- _preserve -
 - oracle.xml.parser.schema.XSDTypeConstants._preserve, 7-54
- _replace -
 - oracle.xml.parser.schema.XSDTypeConstants._replace, 7-54

A

- activeFound(), 12-24
- addAttr(String, String, String, String, boolean, int) -
 - oracle.xml.parser.v2.SAXAttrList.addAttr(java.lang.String, java.lang.String, java.lang.String, java.lang.String, boolean, int), 11-86
- addAttr(String, String, String, String, boolean, int, String) -
 - oracle.xml.parser.v2.SAXAttrList.addAttr(java.lang.String, java.lang.String, java.lang.String, java.lang.String, boolean, int, java.lang.String), 11-86
- addAttribute(String, Object) -
 - oracle.xml.classgen.CGXSElement.addAttribute(java.lang.String, java.lang.Object), 6-17
- addCDATASection(String) -
 - oracle.xml.classgen.CGNode.addCDATASection(java.lang.String), 6-9
- addData(String) -
 - oracle.xml.classgen.CGNode.addData(java.lang.String), 6-9
- addDOMBuilderErrorListener(DOMBuilderErrorListener), 12-5
- addDOMBuilderListener(DOMBuilderListener), 12-6
- addElement(Object) -
 - oracle.xml.classgen.CGXSElement.addElement(java.lang.Object), 6-17
- addEventListener(String, EventListener, boolean) -
 - oracle.xml.parser.v2.XMLNode.addEventListener(java.lang.String, org.w3c.dom.events.EventListener, boolean), 11-191
- addID(String, XMLElement) -
 - oracle.xml.parser.v2.XMLDocument.addID(java.lang.String, oracle.xml.parser.v2.XMLNode), 11-126
- addIndent(int) -
 - oracle.xml.parser.v2.XMLOutputStream.addIndent(int), 11-229
- addNode(CGNode) -
 - oracle.xml.classgen.CGNode.addNode(oracle.xml.classgen.CGNode), 6-9
- addSubscriber, 2-29
- addText(char[], int, int) -
 - oracle.xml.parser.v2.XMLNSNode.addText(char

- r[], int, int), 11-218
- addText(char[], int, int) -
 - oracle.xml.parser.v2.XMLText.addText(char[], int, int), 11-262
- addText(String) -
 - oracle.xml.parser.v2.XMLAttr.addText(java.lang.String), 11-105
- addText(String) -
 - oracle.xml.parser.v2.XMLComment.addText(java.lang.String), 11-116
- addText(String) -
 - oracle.xml.parser.v2.XMLNSNode.addText(java.lang.String), 11-219
- addText(String) -
 - oracle.xml.parser.v2.XMLPI.addText(java.lang.String), 11-250
- addXSLTransformerErrorListener(XSLTransformerErrorListener), 12-27
- addXSLTransformerListener(XSLTransformerListener), 12-27
- adoptNode(Node) -
 - oracle.xml.parser.v2.XMLDocument.adoptNode(org.w3c.dom.Node), 11-127
- afterAQOperation(HttpServletRequest, HttpServletResponse, AQxmlCallbackContext), 3-6
- alter, 2-23, 4-72
- alterPropagationSchedule, 2-31, 4-73
- alterQueue, 2-28
- ANY_SIMPLE -
 - oracle.xml.parser.schema.XSDTypeConstants.ANY_SIMPLE, 7-54
- ANY_URI -
 - oracle.xml.parser.schema.XSDTypeConstants.ANY_URI, 7-54
- AppCtxManager, 1-1, 1-2, 1-3
- appendChild(Node) -
 - oracle.xml.parser.v2.XMLDocument.appendChild(org.w3c.dom.Node), 11-127
- appendChild(Node) -
 - oracle.xml.parser.v2.XMLNode.appendChild(org.w3c.dom.Node), 11-192
- appendChild(Node) -
 - oracle.xml.parser.v2.XMLNSNode.appendChild(org.w3c.dom.Node), 11-219
- AQ_ORA_TR1, 3-24
- AQAgent, 2-13
- AQConstants, 2-12
- AQDequeueOption, 2-41
- AQDriverManager, 2-6
- AQEnqueueOption, 2-39
- AQException, 2-54
- AQjmsAdtMessage, 4-11
- AQjmsAgent, 4-28, 4-29
- AQjmsBytesMessage, 4-32
- AQjmsConnection, 4-47
- AQjmsConnectionMetaData, 4-55, 4-56
- AQjmsConstants, 4-60, 4-61
- AQjmsConsumer, 4-63
- AQjmsDestination, 4-71
- AQjmsDestinationProperty, 4-81
- AQjmsException, 4-85
- AQjmsFactory, 4-87
- AQjmsInvalidDestinationException, 4-93
- AQjmsInvalidSelectorException, 4-94
- AQjmsMapMessage, 4-95
- AQjmsMessage, 4-111
- AQjmsMessageEOFException, 4-134
- AQjmsMessageFormatException, 4-135
- AQjmsMessageNotReadableException, 4-136
- AQjmsMessageNotWriteableException, 4-137
- AQjmsObjectMessage, 4-138
- AQjmsOracleDebug, 4-142
- AQjmsProducer, 4-144
- AQjmsQueueBrowser, 4-158
- AQjmsQueueConnectionFactory, 4-162
- AQjmsQueueReceiver, 4-165
- AQjmsQueueSender, 4-168
- AQjmsSession, 4-170
- AQjmsStreamMessage, 4-206
- AQjmsTextMessage, 4-220
- AQjmsTopicConnectionFactory, 4-229
- AQjmsTopicPublisher, 4-232
- AQjmsTopicReceiver, 4-236
- AQjmsTopicSubscriber, 4-239
- AQMessage, 2-45
- AQMessageProperty, 2-47
- AQObjectPayload, 2-53
- AQOracleSQLException, 2-55
- AQQueue, 2-35

AQQueueAdmin, 2-27
 AQQueueProperty, 2-20
 AQQueueTable, 2-23
 AQQueueTableProperty, 2-15
 AQRawPayload, 2-51
 AQSession, 2-8
 AQxmlCallback, 3-6
 AQxmlCallbackContext, 3-11
 AQxmlDataSource, 3-8, 3-9
 AQxmlDataSource(String, String, String, String,
 String), 3-9
 AQxmlDebug, 3-24
 AQxmlException, 3-26
 AQxmlServlet, 3-14
 AQxmlServlet20, 3-19
 ASTERISK -
 oracle.xml.parser.v2.ElementDecl.ASTERISK,
 11-70
 AttListDecl -
 oracle.xml.parser.v2.XMLToken.AttListDecl, 1
 1-267
 AttName -
 oracle.xml.parser.v2.XMLToken.AttName, 11-
 267
 ATTRDECL -
 oracle.xml.parser.v2.XMLNode.ATTRDECL, 1
 1-190
 AttrDecl() -
 oracle.xml.parser.v2.AttrDecl.AttrDecl(), 11-16
 Attribute -
 oracle.xml.parser.v2.XMLToken.Attribute, 11-
 267
 attributeDecl(String, String, String, String, String) -
 oracle.xml.parser.v2.DocumentBuilder.attribute
 Decl(java.lang.String, java.lang.String,
 java.lang.String, java.lang.String,
 java.lang.String), 11-34
 AttValue -
 oracle.xml.parser.v2.XMLToken.AttValue, 11-
 267
 Auto_Events -
 oracle.xml.parser.v2.XMLNode.Auto_
 Events, 11-189

B

BASE_URL -
 oracle.xml.jaxp.JXDocumentBuilderFactory.BA
 SE_URL, 11-278
 BASE_URL -
 oracle.xml.parser.v2.XMLParser.BASE_
 URL, 11-237
 BASE64_BINARY -
 oracle.xml.parser.schema.XSDTypeConstants.B
 ASE64_BINARY, 7-54
 beforeAQOperation(HttpServletRequest,
 HttpServletResponse,
 AQxmlCallbackContext), 3-7
 BINARY -
 oracle.xml.parser.schema.XSDTypeConstants.BI
 NARY, 7-54
 BOOLEAN -
 oracle.xml.parser.schema.XSDTypeConstants.B
 OOLEAN, 7-54
 BYTE -
 oracle.xml.parser.schema.XSDTypeConstants.B
 YTE, 7-54

C

capturing -
 oracle.xml.parser.v2.XMLNode.capturing, 11-
 190
 CDATA -
 oracle.xml.parser.schema.XSDTypeConstants.C
 DATA, 7-54
 CDATA -
 oracle.xml.parser.v2.AttrDecl.CDATA, 11-15
 cDATASection(char[], int, int) -
 oracle.xml.parser.v2.DefaultXMLDocumentHan
 dler.cDATASection(char[], int, int), 11-22
 cDATASection(char[], int, int) -
 oracle.xml.parser.v2.DocumentBuilder.cDATAS
 ection(char[], int, int), 11-34
 CDSect -
 oracle.xml.parser.v2.XMLToken.CDSect, 11-26
 8
 CENTURY -
 oracle.xml.parser.schema.XSDTypeConstants.C

ENTURY, 7-54
 CGDocument, 6-2
 CGDocument(String, DTD) -
 oracle.xml.classgen.CGDocument.CGDocument
 (java.lang.String,
 oracle.xml.parser.v2.DTD), 6-4
 CGNode(String) -
 oracle.xml.classgen.CGNode.CGNode(java.lang
 .String), 6-8
 CGXSDElement() -
 oracle.xml.classgen.CGXSDElement.CGXSDEle
 ment(), 6-16
 characters(char[], int, int) -
 oracle.xml.parser.schema.XSDValidator.charact
 ers(char[], int, int), 7-60
 characters(char[], int, int) -
 oracle.xml.parser.v2.DocumentBuilder.characte
 rs(char[], int, int), 11-35
 CharData -
 oracle.xml.parser.v2.XMLToken.CharData, 11-
 268
 clearBody, 4-13, 4-35, 4-98, 4-114, 4-140, 4-209,
 4-222
 clearContext, 1-3
 clearParameters() -
 oracle.xml.jaxp.JXTransformer.clearParameters(
), 11-298
 clearProperties, 4-35, 4-98, 4-114, 4-140, 4-209
 cloneNode(boolean) -
 oracle.xml.parser.v2.ElementDecl.cloneNode(b
 oolean), 11-72
 cloneNode(boolean) -
 oracle.xml.parser.v2.XMLAttr.cloneNode(boole
 an), 11-105
 cloneNode(boolean) -
 oracle.xml.parser.v2.XMLDeclPI.cloneNode(bo
 olean), 11-121
 cloneNode(boolean) -
 oracle.xml.parser.v2.XMLDocument.cloneNode
 (boolean), 11-128
 cloneNode(boolean) -
 oracle.xml.parser.v2.XMLElement.cloneNode(b
 oolean), 11-159
 cloneNode(boolean) -
 oracle.xml.parser.v2.XMLEntity.cloneNode(boo
 lean), 11-177
 cloneNode(boolean) -
 oracle.xml.parser.v2.XMLNode.cloneNode(bool
 ean), 11-192
 cloneNode(boolean) -
 oracle.xml.parser.v2.XMLNotation.cloneNode(b
 oolean), 11-213
 close, 4-48, 4-64, 4-146, 4-159
 close() -
 oracle.xdb.spi.XDBNamingEnumeration.close(),
 23-8
 close() -
 oracle.xml.parser.v2.PrintDriver.close(), 11-8
 close() -
 oracle.xml.parser.v2.XMLOutputStream.close(),
 11-229
 close() -
 oracle.xml.parser.v2.XMLPrintDriver.close(),
 11-254
 close() -
 oracle.xml.sql.dml.OracleXMLSave.close(), 8-5
 close() -
 oracle.xml.sql.query.OracleXMLQuery.close(),
 9-6
 col - oracle.xml.util.XMLError.col, 10-6
 collectTimingInfo(boolean) -
 oracle.xml.sql.dml.OracleXMLSave.collectTimin
 gInfo(boolean), 8-5
 COMMA -
 oracle.xml.parser.v2.ElementDecl.COMMA, 1
 1-70
 Comment -
 oracle.xml.parser.v2.XMLToken.Comment, 11
 -268
 comment(char[], int, int) -
 oracle.xml.parser.v2.DocumentBuilder.commen
 t(char[], int, int), 11-36
 comment(String) -
 oracle.xml.parser.v2.DefaultXMLDocumentHan
 dler.comment(java.lang.String), 11-23
 comment(String) -
 oracle.xml.parser.v2.DocumentBuilder.commen
 t(java.lang.String), 11-36
 COMPACT -
 oracle.xml.parser.v2.XMLOutputStream.COMP

ACT, 11-228
 compareTo(XSDDataValue) -
 oracle.xml.parser.schema.XSDDataValue.com
 pareTo(oracle.xml.parser.schema.XSDDataValue)
 , 7-28
 createAppCtxPermit, 1-3
 createAttribute(String) -
 oracle.xml.parser.v2.XMLDocument.createAttri
 bute(java.lang.String), 11-129
 createAttribute(String, String) -
 oracle.xml.parser.v2.NodeFactory.createAttribu
 te(java.lang.String, java.lang.String), 11-78
 createAttribute(String, String, String, String) -
 oracle.xml.parser.v2.NodeFactory.createAttribu
 te(java.lang.String, java.lang.String,
 java.lang.String, java.lang.String), 11-79
 createAttributeNS(String, String) -
 oracle.xml.parser.v2.XMLDocument.createAttri
 buteNS(java.lang.String,
 java.lang.String), 11-129
 createBLOBTable(Connection, String), 16-4
 createBrowser, 4-178
 createCDATASection(String) -
 oracle.xml.parser.v2.NodeFactory.createCDAT
 ASection(java.lang.String), 11-79
 createCDATASection(String) -
 oracle.xml.parser.v2.XMLDocument.createCDA
 TASession(java.lang.String), 11-130
 createComment(String) -
 oracle.xml.parser.v2.NodeFactory.createComm
 ent(java.lang.String), 11-79
 createComment(String) -
 oracle.xml.parser.v2.XMLDocument.createCom
 ment(java.lang.String), 11-130
 createDocument() -
 oracle.xml.parser.v2.NodeFactory.createDocum
 ent(), 11-80
 createDocument(String, String, DocumentType) -
 oracle.xml.parser.v2.XMLDOMImplementation.
 createDocument(java.lang.String,
 java.lang.String,
 org.w3c.dom.DocumentType), 11-156
 createDocumentFragment() -
 oracle.xml.parser.v2.NodeFactory.createDocum
 entFragment(), 11-80
 createDocumentFragment() -
 oracle.xml.parser.v2.XMLDocument.createDocu
 mentFragment(), 11-131
 createDocumentType(String, String, String) -
 oracle.xml.parser.v2.XMLDOMImplementation.
 createDocumentType(java.lang.String,
 java.lang.String, java.lang.String), 11-157
 createDurableSubscriber, 4-180, 4-181, 4-182, 4-184
 createElement(String) -
 oracle.xml.parser.v2.NodeFactory.createElemen
 t(java.lang.String), 11-80
 createElement(String) -
 oracle.xml.parser.v2.XMLDocument.createElem
 ent(java.lang.String), 11-131
 createElementNS(String, String) -
 oracle.xml.parser.v2.XMLDocument.createElem
 entNS(java.lang.String,
 java.lang.String), 11-132
 createElementNS(String, String, String) -
 oracle.xml.parser.v2.NodeFactory.createElemen
 tNS(java.lang.String, java.lang.String,
 java.lang.String), 11-81
 createEntityReference(String) -
 oracle.xml.parser.v2.NodeFactory.createEntityR
 eference(java.lang.String), 11-81
 createEntityReference(String) -
 oracle.xml.parser.v2.XMLDocument.createEntit
 yReference(java.lang.String), 11-132
 createEvent(String) -
 oracle.xml.parser.v2.XMLDocument.createEven
 t(java.lang.String), 11-133
 createMapMessage, 4-185
 createMessage, 2-35
 createMutationEvent(String) -
 oracle.xml.parser.v2.XMLDocument.createMuta
 tionEvent(java.lang.String), 11-133
 createNodeIterator(Node, int, NodeFilter, boolean) -
 oracle.xml.parser.v2.XMLDocument.createNod
 eIterator(org.w3c.dom.Node, int,
 org.w3c.dom.traversal.NodeFilter,
 boolean), 11-133
 createObjectMessage, 4-185
 createProcessingInstruction(String, String) -
 oracle.xml.parser.v2.NodeFactory.createProcess
 ingInstruction(java.lang.String,

- java.lang.String), 11-78, 11-82
- createProcessingInstruction(String, String) -
 - oracle.xml.parser.v2.XMLDocument.createProcessingInstruction(java.lang.String, java.lang.String), 11-134
- createPublisher, 4-186
- createQueue, 2-9, 2-24, 4-187
- createQueueConnection, 4-163
- createQueueSession, 4-49
- createQueueTable, 2-8, 4-187
- createRange() -
 - oracle.xml.parser.v2.XMLDocument.createRange(), 11-135
- createRangeEvent(String) -
 - oracle.xml.parser.v2.XMLDocument.createRangeEvent(java.lang.String), 11-135
- createReceiver, 4-188, 4-190
- createRemoteSubscriber, 4-191, 4-192, 4-193
- createSender, 4-194
- createStreamMessage, 4-194
- createSubscriber, 4-195
- createTextMessage, 4-196
- createTextNode(String) -
 - oracle.xml.parser.v2.NodeFactory.createTextNode(java.lang.String), 11-82
- createTextNode(String) -
 - oracle.xml.parser.v2.XMLDocument.createTextNode(java.lang.String), 11-135
- createTopic, 4-197
- createTopicConnection, 4-230
- createTopicReceiver, 4-198, 4-199
- createTopicSession, 4-49
- createTraversalEvent(String) -
 - oracle.xml.parser.v2.XMLDocument.createTraversalEvent(java.lang.String), 11-136
- createTreeWalker(Node, int, NodeFilter, boolean) -
 - oracle.xml.parser.v2.XMLDocument.createTreeWalker(org.w3c.dom.Node, int, org.w3c.dom.traversal.NodeFilter, boolean), 11-136
- createURL(String) -
 - oracle.xml.sql.dml.OracleXMLSave.createURL(java.lang.String), 8-5
- createXMLTable(Connection, String), 16-5

D

- data cartridges, 5-2
- DATE -
 - oracle.xml.parser.schema.XSDTypeConstants.DATE, 7-54
- DATE_TIME -
 - oracle.xml.parser.schema.XSDTypeConstants.DATE_TIME, 7-54
- DBAccess, 16-4
- DBAccess(), 16-4
- DBAccessBeanInfo, 16-10
- DBAccessBeanInfo(), 16-10
- DBMS_APPCTX, 1-2
- DBViewer(), 13-4
- DBViewerBeanInfo, 13-19
- DBViewerBeanInfo(), 13-19
- DEBUG_MODE -
 - oracle.xml.jaxp.JXDocumentBuilderFactory.DEBUG_MODE, 11-278
- DEBUG_MODE -
 - oracle.xml.parser.v2.DOMParser.DEBUG_MODE, 11-49
- DECIMAL -
 - oracle.xml.parser.schema.XSDTypeConstants.DECIMAL, 7-55
- DEFAULT -
 - oracle.xml.parser.v2.AttrDecl.DEFAULT, 11-15
- DEFAULT -
 - oracle.xml.parser.v2.XMLOutputStream.DEFAULT, 11-228
- DefaultXMLDocumentHandler -
 - oracle.xml.parser.v2.DefaultXMLDocumentHandler, 11-21
- DefaultXMLDocumentHandler() -
 - oracle.xml.parser.v2.DefaultXMLDocumentHandler.DefaultXMLDocumentHandler(), 11-21
- deleteBLOBName(Connection, String, String), 16-5
- deleteData(String) -
 - oracle.xml.classgen.CGNode.deleteData(java.lang.String), 6-10
- deleteXML(Document) -
 - oracle.xml.sql.dml.OracleXMLSave.deleteXML(org.w3c.dom.Document), 8-5

deleteXMLName(Connection, String, String), 16-5
 dequeue, 2-36, 2-37
 derivedFrom(XSDSimpleType, String, String) -
 oracle.xml.parser.schema.XSDSimpleType.derivedFrom(oracle.xml.parser.schema.XSDSimpleType, java.lang.String, java.lang.String), 7-48
 disablePropagationSchedule, 2-32, 4-73
 dispatchEvent(Event) -
 oracle.xml.parser.v2.XMLNode.dispatchEvent(org.w3c.dom.events.Event), 11-193
 DocumentBuilder -
 oracle.xml.parser.v2.DocumentBuilder, 11-32
 DocumentBuilder() -
 oracle.xml.parser.v2.DocumentBuilder.DocumentBuilder(), 11-32
 doGet(HttpServletRequest, HttpServletResponse), 3-15, 3-20
 DOM, 11-1
 DOMAttrModified -
 oracle.xml.parser.v2.XMLNode.DOMAttrModified(), 11-190
 DOMBuilder, 12-4
 DOMBuilder(), 12-5
 DOMBuilder(int), 12-5
 DOMBuilderBeanInfo, 12-15
 DOMBuilderBeanInfo(), 12-15
 domBuilderError(DOMBuilderEvent), 12-22
 domBuilderErrorCalled(DOMBuilderErrorEvent), 12-19
 DOMBuilderErrorEvent, 12-17
 DOMBuilderErrorEvent(Object, Exception), 12-17
 DOMBuilderErrorListener, 12-19
 DOMBuilderEvent, 12-20
 DOMBuilderEvent(Object, int), 12-20
 DOMBuilderListener, 12-22
 domBuilderOver(DOMBuilderEvent), 12-22
 domBuilderStarted(DOMBuilderEvent), 12-22
 DOMCharacterDataModified -
 oracle.xml.parser.v2.XMLNode.DOMCharacterDataModified(), 11-190
 DOMNodeInserted -
 oracle.xml.parser.v2.XMLNode.DOMNodeInserted(), 11-190
 DOMNodeRemoved -
 oracle.xml.parser.v2.XMLNode.DOMNodeRemoved(), 11-190
 DOMNodeRemovedFromDocument -
 oracle.xml.parser.v2.XMLNode.DOMNodeRemovedFromDocument(), 11-190
 DOMParser -
 oracle.xml.parser.v2.DOMParser, 11-49
 DOMParser() -
 oracle.xml.parser.v2.DOMParser.DOMParser(), 11-49
 doPost(HttpServletRequest, HttpServletResponse), 3-15, 3-20
 DOUBLE -
 oracle.xml.parser.schema.XSDTypeConstants.DOUBLE, 7-55
 drop, 2-23, 2-28, 4-74
 dropBLOBTable(Connection, String), 16-6
 dropQueue, 2-24
 dropXMLTable(Connection, String), 16-6
 DTD - oracle.xml.parser.v2.DTD, 11-59
 DTD() - oracle.xml.parser.v2.DTD.DTD(), 11-59
 DTD_OBJECT -
 oracle.xml.jaxp.JXDocumentBuilderFactory.DTD_OBJECT, 11-278
 DTD_OBJECT -
 oracle.xml.parser.v2.XMLParser.DTD_OBJECT, 11-237
 DTDClassGenerator() -
 oracle.xml.classgen.DTDClassGenerator.DTDClassGenerator(), 6-20
 DTDName -
 oracle.xml.parser.v2.XMLToken.DTDName, 11-268
 DURATION -
 oracle.xml.parser.schema.XSDTypeConstants.DURATION, 7-55
E
 ElemDeclName -
 oracle.xml.parser.v2.XMLToken.ElemDeclName, 11-268
 ELEMENT -
 oracle.xml.parser.v2.ElementDecl.ELEMENT, 11-70
 ELEMENT_DECLARED -

- oracle.xml.parser.v2.ElementDecl.ELEMENT_DECLARED, 11-70
- ElementDecl -
 - oracle.xml.parser.v2.ElementDecl, 11-70
- ELEMENTDECL -
 - oracle.xml.parser.v2.XMLNode.ELEMENTDECL, 11-190
- elementdecl -
 - oracle.xml.parser.v2.XMLToken.elementdecl, 11-268
- ElementDecl() -
 - oracle.xml.parser.v2.ElementDecl.ElementDecl(), 11-71
- elementDecl(String, String) -
 - oracle.xml.parser.v2.DocumentBuilder.elementDecl(java.lang.String, java.lang.String), 11-36
- ELEMENTS -
 - oracle.xml.parser.v2.ElementDecl.ELEMENTS, 11-70
- EmptyElemTag -
 - oracle.xml.parser.v2.XMLToken.EmptyElemTag, 11-268
- enablePropagationSchedule, 2-32, 4-74
- ENCODING -
 - oracle.xml.parser.schema.XSDTypeConstants.ENCODING, 7-55
- endCDATA() -
 - oracle.xml.parser.v2.DocumentBuilder.endCDATA(), 11-37
- endDoctype() -
 - oracle.xml.parser.v2.DefaultXMLDocumentHandler.endDoctype(), 11-23
- endDoctype() -
 - oracle.xml.parser.v2.DocumentBuilder.endDoctype(), 11-37
- endDocument() -
 - oracle.xml.parser.v2.DocumentBuilder.endDocument(), 11-37
- endDTD() -
 - oracle.xml.parser.v2.DocumentBuilder.endDTD(), 11-38
- endElement(NSName) -
 - oracle.xml.parser.v2.DefaultXMLDocumentHandler.endElement(oracle.xml.parser.v2.NSName), 11-24
- endElement(NSName) -
 - oracle.xml.parser.v2.DocumentBuilder.endElement(oracle.xml.parser.v2.NSName), 11-38
- endElement(String, String, String) -
 - oracle.xml.parser.schema.XSDValidator.endElement(java.lang.String, java.lang.String, java.lang.String), 7-60, 7-61
- endElement(String, String, String) -
 - oracle.xml.parser.v2.DefaultXMLDocumentHandler.endElement(java.lang.String, java.lang.String, java.lang.String), 11-24
- endElement(String, String) -
 - oracle.xml.parser.v2.DocumentBuilder.endElement(java.lang.String, java.lang.String, java.lang.String), 11-39
- endEntity(String) -
 - oracle.xml.parser.v2.DocumentBuilder.endEntity(java.lang.String), 11-39
- endPrefixMapping(String) -
 - oracle.xml.parser.v2.DefaultXMLDocumentHandler.endPrefixMapping(java.lang.String), 11-25
- enqueue, 2-36
- ENTITIES -
 - oracle.xml.parser.schema.XSDTypeConstants.ENTITIES, 7-55
- ENTITIES -
 - oracle.xml.parser.v2.AttrDecl.ENTITIES, 11-15
- ENTITY -
 - oracle.xml.parser.schema.XSDTypeConstants.ENTITY, 7-55
- ENTITY -
 - oracle.xml.parser.v2.AttrDecl.ENTITY, 11-15
- EntityDecl -
 - oracle.xml.parser.v2.XMLToken.EntityDecl, 11-268
- EntityDeclName -
 - oracle.xml.parser.v2.XMLToken.EntityDeclName, 11-268
- EntityValue -
 - oracle.xml.parser.v2.XMLToken.EntityValue, 11-268
- ENUMERATION -
 - oracle.xml.parser.schema.XSDTypeConstants.ENUMERATION, 7-55

errid - oracle.xml.util.XMLError.errid, 10-6
 ERROR -
 oracle.xml.parser.v2.XMLParseException.ERROR, 11-233
 error(int, int, String) -
 oracle.xml.util.XMLError.error(int, int, java.lang.String), 10-8
 error(int, int, String, String, String, int, int, boolean) -
 oracle.xml.parser.v2.XMLError.error(int, int, java.lang.String, java.lang.String, java.lang.String, int, int, boolean), 11-185
 error(int, int, String[]) -
 oracle.xml.util.XMLError.error(int, int, java.lang.String[]), 10-9
 ERROR_ENCODING -
 oracle.xml.jaxp.JXDocumentBuilderFactory.ERROR_ENCODING, 11-278
 ERROR_ENCODING -
 oracle.xml.parser.v2.DOMParser.ERROR_ENCODING, 11-49
 ERROR_STREAM -
 oracle.xml.jaxp.JXDocumentBuilderFactory.ERROR_STREAM, 11-278
 ERROR_STREAM -
 oracle.xml.parser.v2.DOMParser.ERROR_STREAM, 11-49
 error0(int, int) - oracle.xml.util.XMLError.error0(int, int), 10-9
 error1(int, int, String) -
 oracle.xml.util.XMLError.error1(int, int, java.lang.String), 10-9
 error2(int, int, String, String) -
 oracle.xml.util.XMLError.error2(int, int, java.lang.String, java.lang.String), 10-10
 error3(int, int, String, String, String) -
 oracle.xml.util.XMLError.error3(int, int, java.lang.String, java.lang.String, java.lang.String), 10-10
 ETag -
 oracle.xml.parser.v2.XMLToken.ETag, 11-268
 ETagName -
 oracle.xml.parser.v2.XMLToken.ETagName, 11-268
 exp - oracle.xml.util.XMLError.exp, 10-6
 expectedElements(Element) -

oracle.xml.parser.v2.ElementDecl.expectedElements(org.w3c.dom.Element), 11-73
 expectedElements(Element) -
 oracle.xml.parser.v2.XMLDocument.expectedElements(org.w3c.dom.Element), 11-137
 externalEntityDecl(String, String, String) -
 oracle.xml.parser.v2.DocumentBuilder.externalEntityDecl(java.lang.String, java.lang.String, java.lang.String), 11-40
 ExternalID -
 oracle.xml.parser.v2.XMLToken.ExternalID, 11-268

F

FATAL_ERROR -
 oracle.xml.parser.v2.XMLParseException.FATAL_ERROR, 11-233
 FATAL_ERROR -
 oracle.xml.util.XMLException.FATAL_ERROR, 10-18
 findAttrDecl(String) -
 oracle.xml.parser.v2.ElementDecl.findAttrDecl(java.lang.String), 11-73
 findEntity(String, boolean) -
 oracle.xml.parser.v2.DTD.findEntity(java.lang.String, boolean), 11-61
 findEquivClass(String, String) -
 oracle.xml.parser.schema.XSDElement.findEquivClass(java.lang.String, java.lang.String), 7-32
 findNotation(String) -
 oracle.xml.parser.v2.DTD.findNotation(java.lang.String), 11-62
 FLOAT -
 oracle.xml.parser.schema.XSDTypeConstants.FLOAT, 7-55
 flush() -
 oracle.xml.parser.v2.PrintDriver.flush(), 11-8
 flush() -
 oracle.xml.parser.v2.XMLOutputStream.flush(), 11-230
 flush() -
 oracle.xml.parser.v2.XMLPrintDriver.flush(), 11-254
 flushErrorListener(DOMLocator) -

oracle.xml.parser.v2.XMLError.flushErrorListener(oracle.xml.parser.v2.DOMLocator), 11-186
 flushErrorListenerStream(DOMLocator) -
 oracle.xml.parser.v2.XMLError.flushErrorListenerStream(oracle.xml.parser.v2.DOMLocator), 11-186
 flushErrors() -
 oracle.xml.parser.v2.XMLError.flushErrors(), 11-186
 flushErrorStream() -
 oracle.xml.util.XMLError.flushErrorStream(), 10-10
 fontGet(AttributeSet), 15-4
 fontSet(MutableAttributeSet, Font), 15-4
 formatErrorMesg(int) -
 oracle.xml.util.XMLError.formatErrorMesg(int), 10-11
 formatErrorMessage(int) -
 oracle.xml.parser.v2.XMLParseException.formatErrorMessage(int), 11-233
 formatErrorMessage(int) -
 oracle.xml.util.XMLException.formatErrorMessage(int), 10-20
 FRACTION_DIGITS -
 oracle.xml.parser.schema.XSDTypeConstants.FRACTION_DIGITS, 7-55

G

GDAY -
 oracle.xml.parser.schema.XSDTypeConstants.GDAY, 7-55
 generate(DTD, String) -
 oracle.xml.classgen.DTDClassGenerator.generate(oracle.xml.parser.v2.DTD, java.lang.String), 6-21
 generate(XMLSchema) -
 oracle.xml.classgen.SchemaClassGenerator.generate(oracle.xml.parser.schema.XMLSchema), 6-27
 getAddress, 2-14, 4-29
 getAdtPayload, 4-9, 4-14
 getAllTargetNS() -
 oracle.xml.parser.schema.XMLSchema.getAllTargetNS(), 7-5
 getAQDataSource(), 3-16, 3-21
 getAQSession, 2-6
 getAssociatedStylesheet(Source, String, String, String) -
 oracle.xml.jaxp.JXSAXTransformerFactory.getAssociatedStylesheet(javax.xml.transform.Source, java.lang.String, java.lang.String, java.lang.String), 11-289
 getAttempts, 2-49
 getAttrDecls() -
 oracle.xml.parser.v2.ElementDecl.getAttrDecls(), 11-73
 getAttribute(String) -
 oracle.xml.classgen.CGNode.getAttribute(java.lang.String), 6-10
 getAttribute(String) -
 oracle.xml.jaxp.JXDocumentBuilderFactory.getAttribute(java.lang.String), 11-279
 getAttribute(String) -
 oracle.xml.jaxp.JXSAXTransformerFactory.getAttribute(java.lang.String), 11-289
 getAttribute(String) -
 oracle.xml.parser.v2.DOMParser.getAttribute(java.lang.String), 11-51
 getAttribute(String) -
 oracle.xml.parser.v2.XMLElement.getAttribute(java.lang.String), 11-160
 getAttribute(String) -
 oracle.xml.parser.v2.XMLParser.getAttribute(java.lang.String), 11-238
 getAttributeDeclarations() -
 oracle.xml.parser.schema.XMLSchemaNode.getAttributeDeclarations(), 7-9
 getAttributeDeclarations() -
 oracle.xml.parser.schema.XSDComplexType.getAttributeDeclarations(), 7-21
 getAttributeNameFont(), 15-4
 getAttributeNameForeground(), 15-5
 getAttributeNode(String) -
 oracle.xml.parser.v2.XMLElement.getAttributeNode(java.lang.String), 11-160
 getAttributeNodeNS(String, String) -
 oracle.xml.parser.v2.XMLElement.getAttributeNodeNS(java.lang.String, java.lang.String), 11-161

getAttributeNS(String, String) -
 oracle.xml.parser.v2.XMLElement.getAttributeNS(java.lang.String, java.lang.String), 11-161
 getAttributes() -
 oracle.xml.classgen.CGXSElement.getAttributes(), 6-18
 getAttributes() -
 oracle.xml.parser.v2.XMLDocumentFragment.getAttributes(), 11-154
 getAttributes() -
 oracle.xml.parser.v2.XMLElement.getAttributes(), 11-162
 getAttributes() -
 oracle.xml.parser.v2.XMLNode.getAttributes(), 11-193
 getAttributeSet() -
 oracle.xml.parser.schema.XMLSchemaNode.getAttributeSet(), 7-9
 getAttributeSet() -
 oracle.xml.parser.schema.XSDComplexType.getAttributeSet(), 7-21
 getAttributeTemplateValue(XSLTContext, String, String) -
 oracle.xml.parser.v2.XSLExtensionElement.getAttributeTemplateValue(oracle.xml.parser.v2.XSLTContext, java.lang.String, java.lang.String), 11-311
 getAttributeValue(String, String) -
 oracle.xml.parser.v2.XSLExtensionElement.getAttributeValue(java.lang.String, java.lang.String), 11-312
 getAttributeValueFont(), 15-5
 getAttributeValueForeground(), 15-5
 getAttributeWildcard() -
 oracle.xml.parser.schema.XSDComplexType.getAttributeWildcard(), 7-21
 getAttrPresence() -
 oracle.xml.parser.v2.AttrDecl.getAttrPresence(), 11-16
 getAttrType() -
 oracle.xml.parser.v2.AttrDecl.getAttrType(), 11-17
 getAuthor() -
 oracle.xdb.spi.XDBResourceContext.getAuthor(), 23-17
 getAuthor() -
 oracle.xdb.spi.XDBResource.getAuthor(), 23-10
 getBackground(), 15-5
 getBase() -
 oracle.xml.parser.schema.XSDSimpleType.getBase(), 7-47, 7-49
 getBaseElementSet() -
 oracle.xml.parser.schema.XSDComplexType.getBaseElementSet(), 7-21
 getBaseType() -
 oracle.xml.parser.schema.XSDComplexType.getBaseType(), 7-21
 getBaseURL() -
 oracle.xml.parser.v2.XMLParser.getBaseURL(), 11-238
 getBasicType() -
 oracle.xml.parser.schema.XSDSimpleType.getBasicType(), 7-47, 7-49
 getBLOBData(Connection, String, String), 16-6
 getBoolean, 4-98
 getBooleanProperty, 4-14, 4-115
 getBuiltInDatatypes() -
 oracle.xml.parser.schema.XSDSimpleType.getBuiltInDatatypes(), 7-47, 7-49
 getByte, 4-99
 getByteProperty, 4-15, 4-115
 getBytes, 2-51, 4-99
 getCacheSize(), 3-10
 getCDATAFont(), 15-5
 getCDATAForeground(), 15-6
 getCGDocument() -
 oracle.xml.classgen.CGNode.getCGDocument(), 6-10
 getChar, 4-100
 getChildElements() -
 oracle.xml.classgen.CGXSElement.getChildElements(), 6-18
 getChildNodes() -
 oracle.xml.parser.v2.DTD.getChildNodes(), 11-62
 getChildNodes() -
 oracle.xml.parser.v2.XMLNode.getChildNodes(), 11-194
 getChildNodes() -

- oracle.xml.parser.v2.XMLNSNode.getChildNodes(), 11-220
- getChildNodes() -
 - oracle.xml.parser.v2.XSLExtensionElement.getChildNodes(), 11-313
- getChildrenByTagName(String) -
 - oracle.xml.parser.v2.XMLElement.getChildrenByTagName(java.lang.String), 11-162
- getChildrenByTagName(String, String) -
 - oracle.xml.parser.v2.XMLElement.getChildrenByTagName(java.lang.String, java.lang.String), 11-162
- getClientID, 4-50
- getColumnNumber() -
 - oracle.xml.parser.v2.XMLDocument.getColumnNumber(), 11-137
- getColumnNumber() -
 - oracle.xml.parser.v2.XMLNode.getColumnNumber(), 11-194
- getColumnNumber(int) -
 - oracle.xml.parser.v2.XMLParseException.getColumnNumber(int), 11-234
- getColumnNumber(int) -
 - oracle.xml.util.XMLError.getColumnNumber(int), 10-11
- getColumnNumber(int) -
 - oracle.xml.util.XMLException.getColumnNumber(int), 10-20
- getComment, 2-17, 2-22, 4-83
- getComment() -
 - oracle.xdb.spi.XDBResource.getComment(), 23-10
- getCommentDataFont(), 15-6
- getCommentDataForeground(), 15-6
- getCompatible, 2-18
- getCompleteName, 4-74
- getCompleteTableName, 4-75
- getComplexTypeSet() -
 - oracle.xml.parser.schema.XMLSchemaNode.getComplexTypeSet(), 7-9
- getComplexTypeTable() -
 - oracle.xml.parser.schema.XMLSchemaNode.getComplexTypeTable(), 7-9
- getConsumerName, 2-41
- getContent() -
 - oracle.xdb.spi.XDBResource.getContent(), 23-10
- getContent() -
 - oracle.xml.parser.schema.XSDComplexType.getContent(), 7-22
- getContentElements() -
 - oracle.xml.parser.v2.ElementDecl.getContentElements(), 11-74
- getContentHandler() -
 - oracle.xml.parser.v2.SAXParser.getContentHandler(), 11-96
- getContentType() -
 - oracle.xdb.spi.XDBResourceContext.getContentType(), 23-17
- getContentType() -
 - oracle.xdb.spi.XDBResource.getContentType(), 23-10
- getContentType() -
 - oracle.xml.parser.v2.ElementDecl.getContentType(), 11-74
- getContextNode() -
 - oracle.xml.parser.v2.XSLTContext.getContextNode(), 11-326
- getContextPosition() -
 - oracle.xml.parser.v2.XSLTContext.getContextPosition(), 11-327
- getContextSize() -
 - oracle.xml.parser.v2.XSLTContext.getContextSize(), 11-327
- getCorrelation, 2-44, 2-48
- getCreateDate() -
 - oracle.xdb.spi.XDBResourceContext.getCreateDate(), 23-18
- getCreateDate() -
 - oracle.xdb.spi.XDBResource.getCreateDate(), 23-10
- getCurrentJmsSession, 4-50
- getCurrentNode() -
 - oracle.xml.parser.v2.DocumentBuilder.getCurrentNode(), 11-33, 11-40
- getData() -
 - oracle.xml.classgen.CGNode.getData(), 6-11
- getData() -
 - oracle.xml.parser.v2.XMLDeclPI.getData(), 11-121

- getData() -
 - oracle.xml.parser.v2.XMLText.getData(), 11-262
- getDavComment() -
 - oracle.xdb.spi.XDBResourceContext.getDavComment(), 23-18
- getDB, 2-9
- getDBConnection, 4-200
- getDBConnection(), 3-12
- getDBDrv(), 3-10
- getDebugMode() -
 - oracle.xml.parser.v2.XMLDocument.getDebugMode(), 11-137
- getDebugMode() -
 - oracle.xml.parser.v2.XMLNode.getDebugMode(), 11-194
- getDecimalFormat(NSName) -
 - oracle.xml.parser.v2.XSLStylesheet.getDecimalFormat(oracle.xml.parser.v2.NSName), 11-324
- getDefaultVal() -
 - oracle.xml.parser.schema.XSDAttribute.getDefaultVal(), 7-12
- getDefaultVal() -
 - oracle.xml.parser.schema.XSDElement.getDefaultVal(), 7-32
- getDefaultValue() -
 - oracle.xml.parser.v2.AttrDecl.getDefaultValue(), 11-17
- getDelay, 2-47
- getDeliveryMode, 4-146
- getDequeueMode, 2-42
- getDerivationMethod() -
 - oracle.xml.parser.schema.XSDComplexType.getDerivationMethod(), 7-22
- getDisableMessageID, 4-146
- getDisableMessageTimestamp, 4-147
- getDisplayName() -
 - oracle.xdb.spi.XDBResourceContext.getDisplayName(), 23-18
- getDisplayName() -
 - oracle.xdb.spi.XDBResource.getDisplayName(), 23-11
- getDoctype(), 12-6
- getDoctype() -
 - oracle.xml.parser.v2.DOMParser.getDoctype(), 11-51
- getDoctype() -
 - oracle.xml.parser.v2.XMLDocument.getDoctype(), 11-138
- getDocument(), 12-6
- getDocument() -
 - oracle.xml.parser.v2.DocumentBuilder.getDocument(), 11-41
- getDocument() -
 - oracle.xml.parser.v2.DOMParser.getDocument(), 11-52
- getDocumentElement() -
 - oracle.xml.parser.v2.XMLDocument.getDocumentElement(), 11-138
- getDOMImplementation() -
 - oracle.xml.jaxp.JXDDocumentBuilder.getDOMImplementation(), 11-275
- getDouble, 4-100
- getDoubleProperty, 4-15, 4-116
- getDrivers, 2-6
- getDTDHandler() -
 - oracle.xml.parser.v2.SAXParser.getDTDHandler(), 11-96
- getDTDNode() -
 - oracle.xml.classgen.CGNode.getDTDNode(), 6-11
- getEditedText(), 15-6
- getElementById(String) -
 - oracle.xml.parser.v2.XMLDocument.getElementById(java.lang.String), 11-139
- getElementDecls() -
 - oracle.xml.parser.v2.DTD.getElementDecls(), 11-63
- getElementNode() -
 - oracle.xml.classgen.CGNode.getElementNode(), 6-11
- getElementsByTagName(String) -
 - oracle.xml.parser.v2.XMLDocument.getElementsByTagName(java.lang.String), 11-139, 11-163
- getElementsByTagName(String) -
 - oracle.xml.parser.v2.XMLElement.getElementsByTagName(java.lang.String), 11-163
- getElementsByTagName(String, String) -
 - oracle.xml.parser.v2.XMLElement.getElementsByTagName(java.lang.String, 11-163

- java.lang.String), 11-163
- getElementsByTagNameNS(String, String) -
 - oracle.xml.parser.v2.XMLDocument.getElement
 - sByTagNameNS(java.lang.String,
 - java.lang.String), 11-139
- getElementsByTagNameNS(String, String) -
 - oracle.xml.parser.v2.XMLElement.getElements
 - ByTagNameNS(java.lang.String,
 - java.lang.String), 11-163
- getElementSet() -
 - oracle.xml.parser.schema.XMLSchemaNode.get
 - ElementSet(), 7-10
- getElementSet() -
 - oracle.xml.parser.schema.XSDComplexType.get
 - ElementSet(), 7-22
- getEmailServerAddr(), 3-16, 3-21
- getEmailServerHost(), 3-16, 3-21
- getEncoding() -
 - oracle.xml.parser.schema.XSDDataValue.getEn
 - coding(), 7-28, 7-29
- getEncoding() -
 - oracle.xml.parser.v2.XMLDeclPI.getEncoding(),
 - 11-121
- getEncoding() -
 - oracle.xml.parser.v2.XMLDocument.getEncodi
 - ng(), 11-140
- getEnqueueTime, 2-50
- getEntities() -
 - oracle.xml.parser.v2.DTD.getEntities(), 11-63
- getEntityResolver() -
 - oracle.xml.parser.v2.XMLParser.getEntityResol
 - ver(), 11-238
- getEnumeration, 4-159
- getEnumerationValues() -
 - oracle.xml.parser.v2.AttrDecl.getEnumerationV
 - alues(), 11-18
- getEnvironment() -
 - oracle.xdb.spi.XDBResourceContext.getEnviron
 - ment(), 23-18
- getEquivClassRef() -
 - oracle.xml.parser.schema.XSDElement.getEquiv
 - ClassRef(), 7-33
- getError() -
 - oracle.xml.parser.v2.XSLTContext.getError(),
 - 11-327
- getErrorCode, 2-54
- getErrorCode(), 3-27
- getErrorCode() -
 - oracle.xml.sql.OracleXMLSQLException.getErr
 - orCode(), 9-20
- getErrorHandler() -
 - oracle.xml.parser.v2.XMLError.getErrorHandler
 - r(), 11-187
- getErrorHandler() -
 - oracle.xml.parser.v2.XMLParser.getErrorHandler
 - er(), 11-239
- getErrorID() -
 - oracle.xml.parser.v2.XPathException.getErrorI
 - D(), 11-308
- getErrorListener() -
 - oracle.xml.jaxp.JXSAXTransformerFactory.getE
 - rrorListener(), 11-290
- getErrorListener() -
 - oracle.xml.jaxp.JXTransformer.getErrorListener
 - (), 11-298
- getErrorListener() -
 - oracle.xml.parser.v2.XMLError.getErrorListener
 - (), 11-187
- getErrorNumber, 4-86
- getException(), 12-17, 12-33
- getException(int) -
 - oracle.xml.parser.v2.XMLParseException.getEx
 - ception(int), 11-234
- getException(int) -
 - oracle.xml.util.XMLError.getException(int), 10
 - 11
- getException(int) -
 - oracle.xml.util.XMLException.getException(int)
 - , 10-20
- getExceptionListener, 4-53
- getExceptionQueue, 2-50
- getExpandedName() -
 - oracle.xml.parser.v2.XMLAttr.getExpandedNa
 - me(), 11-106
- getExpandedName() -
 - oracle.xml.parser.v2.XMLElement.getExpanded
 - Name(), 11-164
- getExpandedName() -
 - oracle.xml.util.NSName.getExpandedName(),
 - 10-4, 11-13

getExpandedName(int) -
 oracle.xml.parser.v2.SAXAttrList.getExpandedName(int), 11-87
 getExpiration, 2-48
 getFacetId() -
 oracle.xml.parser.schema.XSDConstrainingFacet.getFacetId(), 7-26
 getFacets() -
 oracle.xml.parser.schema.XSDSimpleType.getFacets(), 7-47, 7-50
 getFeature(String) -
 oracle.xml.jaxp.JXSAXParserFactory.getFeature(java.lang.String), 11-285
 getFeature(String) -
 oracle.xml.jaxp.JXSAXTransformerFactory.getFeature(java.lang.String), 11-290
 getFeature(String) -
 oracle.xml.parser.v2.SAXParser.getFeature(java.lang.String), 11-97
 getFields() -
 oracle.xml.parser.schema.XSDIdentity.getFields(), 7-42
 getFirstAttribute() -
 oracle.xml.parser.v2.XMLElement.getFirstAttribute(), 11-164
 getFirstChild() -
 oracle.xml.parser.v2.XMLNode.getFirstChild(), 11-195
 getFirstChild() -
 oracle.xml.parser.v2.XMLNSNode.getFirstChild(), 11-220
 getFirstError() -
 oracle.xml.util.XMLError.getFirstError(), 10-11
 getFixedVal() -
 oracle.xml.parser.schema.XSDAttribute.getFixedVal(), 7-13
 getFixedVal() -
 oracle.xml.parser.schema.XSDElement.getFixedVal(), 7-33
 getFloat, 4-101
 getFloatProperty, 4-16, 4-116
 getGroup() -
 oracle.xml.parser.schema.XSDComplexType.getGroup(), 7-22
 getHandler() -
 oracle.xml.parser.v2.DefaultXMLDocumentHandler.getHandler(), 11-22, 11-25
 getHost(), 3-10
 getHostname(), 13-5
 getIcon(int), 12-15, 12-31, 13-19, 15-15, 16-10, 16-12, 17-6
 getID(), 12-21, 12-36
 getId(), 12-6, 12-27
 getIdentities() -
 oracle.xml.parser.schema.XSDElement.getIdentities(), 7-33
 getIDHashtable() -
 oracle.xml.parser.v2.XMLDocument.getIDHashtable(), 11-140
 getImplementation() -
 oracle.xml.parser.v2.XMLDocument.getImplementation(), 11-141
 getIndex(String) -
 oracle.xml.parser.v2.SAXAttrList.getIndex(java.lang.String), 11-87
 getIndex(String, String) -
 oracle.xml.parser.v2.SAXAttrList.getIndex(java.lang.String, java.lang.String), 11-87
 getInstanceName(), 13-5
 getInt, 4-101
 getInternalSubset() -
 oracle.xml.parser.v2.DTD.getInternalSubset(), 11-63
 getIntProperty, 4-16, 4-117
 getJmsConnection, 4-200
 getJMSCorrelationID, 4-117
 getJMSCorrelationIDAsBytes, 4-118
 getJMSDeliveryMode, 4-118
 getJMSDestination, 4-118
 getJMSExpiration, 4-119
 getJMSMajorVersion, 4-56
 getJMSMessageID, 4-119
 getJMSMessageIDAsBytes, 4-120
 getJMSMinorVersion, 4-56
 getJMSPriority, 4-120
 getJMSProviderName, 4-57
 getJMSRedelivered, 4-120
 getJMSReplyTo, 4-17, 4-121
 getJMSTimestamp, 4-121

getJMSType, 4-17, 4-122
 getJMSVersion, 4-57
 getJTextPane(), 15-6
 getLanguage() -
 oracle.xdb.spi.XDBResourceContext.getLanguage(), 23-19
 getLanguage() -
 oracle.xdb.spi.XDBResource.getLanguage(), 23-11
 getLastChild() -
 oracle.xml.parser.v2.XMLNode.getLastChild(), 11-195
 getLastChild() -
 oracle.xml.parser.v2.XMLNSNode.getLastChild(), 11-221
 getLastModDate() -
 oracle.xdb.spi.XDBResourceContext.getLastModDate(), 23-19
 getLastModDate() -
 oracle.xdb.spi.XDBResource.getLastModDate(), 23-11
 getLength() -
 oracle.xml.parser.schema.XSDDataValue.getLength(), 7-29
 getLength() -
 oracle.xml.parser.v2.SAXAttrList.getLength(), 11-88
 getLexicalEnumeration() -
 oracle.xml.parser.schema.XSDConstrainingFacet.getLexicalEnumeration(), 7-26
 getLexicalValue() -
 oracle.xml.parser.schema.XSDConstrainingFacet.getLexicalValue(), 7-25, 7-26
 getLexicalValue() -
 oracle.xml.parser.schema.XSDDataValue.getLexicalValue(), 7-29
 getLineNumber() -
 oracle.xml.parser.v2.XMLDocument.getLineNumber(), 11-141
 getLineNumber() -
 oracle.xml.parser.v2.XMLNode.getLineNumber(), 11-195
 getLineNumber(int) -
 oracle.xml.parser.v2.XMLParseException.getLineNumber(int), 11-234
 getLineNumber(int) -
 oracle.xml.util.XMLError.getLineNumber(int), 10-11
 getLineNumber(int) -
 oracle.xml.util.XMLException.getLineNumber(int), 10-21
 getLocalName() -
 oracle.xml.parser.v2.XMLAttr.getLocalName(), 11-106
 getLocalName() -
 oracle.xml.parser.v2.XMLElement.getLocalName(), 11-164
 getLocalName() -
 oracle.xml.parser.v2.XMLNode.getLocalName(), 11-196
 getLocalName() -
 oracle.xml.parser.v2.XMLNSNode.getLocalName(), 11-221
 getLocalName() -
 oracle.xml.util.NSName.getLocalName(), 10-4, 11-13
 getLocalName(int) -
 oracle.xml.parser.v2.SAXAttrList.getLocalName(int), 11-88
 getLocator() -
 oracle.xml.util.XMLError.getLocator(), 10-12
 getLogStream, 4-142
 getLogStream(), 3-24
 getLong, 4-102
 getLongProperty, 4-18, 4-122
 getMapNames, 4-102
 getMaxOccurs() -
 oracle.xml.parser.schema.XSDElement.getMaxOccurs(), 7-33
 getMaxOccurs() -
 oracle.xml.parser.schema.XSDGroup.getMaxOccurs(), 7-39
 getMaxOccurs() -
 oracle.xml.parser.schema.XSDSimpleType.getMaxOccurs(), 7-47, 7-50
 getMaxRetries, 2-20, 4-82
 getMessage, 2-54
 getMessage(), 12-18, 12-34
 getMessage() -
 oracle.xml.parser.v2.XPathException.getMessage()

e(), 11-308
 getMessage(int) -
 oracle.xml.parser.v2.XMLParseException.getMessage(int), 11-235
 getMessage(int) -
 oracle.xml.util.XMLError.getMessage(int), 10-12
 getMessage(int) -
 oracle.xml.util.XMLException.getMessage(int), 10-21
 getMessage(int, String[]) -
 oracle.xml.util.XMLError.getMessage(int, java.lang.String[]), 10-12
 getMessage(XMLError) -
 oracle.xml.parser.v2.XPathException.getMessage(oracle.xml.parser.v2.XMLError), 11-309
 getMessage0(int) -
 oracle.xml.util.XMLError.getMessage0(int), 10-12
 getMessage1(int, String) -
 oracle.xml.util.XMLError.getMessage1(int, java.lang.String), 10-13
 getMessage2(int, String, String) -
 oracle.xml.util.XMLError.getMessage2(int, java.lang.String, java.lang.String), 10-13
 getMessage3(int, String, String, String) -
 oracle.xml.util.XMLError.getMessage3(int, java.lang.String, java.lang.String, java.lang.String), 10-13
 getMessage4(int, String, String, String, String) -
 oracle.xml.util.XMLError.getMessage4(int, java.lang.String, java.lang.String, java.lang.String, java.lang.String), 10-13
 getMessage5(int, String, String, String, String, String) -
 oracle.xml.util.XMLError.getMessage5(int, java.lang.String, java.lang.String, java.lang.String, java.lang.String, java.lang.String), 10-14
 getMessageGrouping, 2-17
 getMessageId, 2-43, 2-45
 getMessageListener, 4-64
 getMessageProperty, 2-46
 getMessageSelector, 4-65, 4-159
 getMessageType(int) -
 oracle.xml.parser.v2.XMLParseException.getMessageType(int), 11-235
 getMessageType(int) -
 oracle.xml.util.XMLError.getMessageType(int), 10-14
 getMessageType(int) -
 oracle.xml.util.XMLException.getMessageType(int), 10-21
 getMetaData, 4-50
 getMinimumSize(), 15-7
 getMinOccurs() -
 oracle.xml.parser.schema.XSDElement.getMinOccurs(), 7-34
 getMinOccurs() -
 oracle.xml.parser.schema.XSDGroup.getMinOccurs(), 7-40
 getMinOccurs() -
 oracle.xml.parser.schema.XSDSimpleType.getMinOccurs(), 7-47, 7-50
 getName, 2-13, 2-23, 2-35, 4-30
 getName() -
 oracle.xml.parser.schema.XSDAttribute.getName(), 7-13
 getName() -
 oracle.xml.parser.schema.XSDConstrainingFacet.getName(), 7-26
 getName() -
 oracle.xml.parser.schema.XSDElement.getName(), 7-34
 getName() -
 oracle.xml.parser.schema.XSDNode.getName(), 7-44
 getName() -
 oracle.xml.parser.v2.DTD.getName(), 11-64
 getName() -
 oracle.xml.parser.v2.XMLAttr.getName(), 11-107
 getNameSize(), 16-7
 getNamespace() -
 oracle.xml.parser.v2.XMLElement.getNamespace(), 11-165
 getNamespace() -
 oracle.xml.util.NSName.getNamespace(), 10-4, 11-14
 getNamespaceURI() -

oracle.xml.parser.schema.XSDNode.getNamespaceURI(), 7-45
 getNamespaceURI() -
 oracle.xml.parser.v2.XMLAttr.getNamespaceURI(), 11-107
 getNamespaceURI() -
 oracle.xml.parser.v2.XMLElement.getNamespaceURI(), 11-165
 getNamespaceURI() -
 oracle.xml.parser.v2.XMLNode.getNamespaceURI(), 11-196
 getNamespaceURI() -
 oracle.xml.parser.v2.XMLNSNode.getNamespaceURI(), 11-222
 getNavigationMode, 2-42, 4-65, 4-237, 4-240
 getNextAttribute() -
 oracle.xml.parser.v2.XMLAttr.getNextAttribute(), 11-107
 getNextException, 2-54
 getNextException(), 3-27
 getNextSibling() -
 oracle.xml.parser.v2.XMLAttr.getNextSibling(), 11-108
 getNextSibling() -
 oracle.xml.parser.v2.XMLNode.getNextSibling(), 11-196
 getNode(String) -
 oracle.xml.classgen.CGNode.getNode(java.lang.String), 6-11
 getNodeAtOffset(int), 15-7
 getNodeName() -
 oracle.xml.parser.v2.AttrDecl.getNodeName(), 11-18
 getNodeName() -
 oracle.xml.parser.v2.DTD.getNodeName(), 11-64
 getNodeName() -
 oracle.xml.parser.v2.ElementDecl.getNodeName(), 11-74
 getNodeName() -
 oracle.xml.parser.v2.XMLCDATA.getNodeName(), 11-114
 getNodeName() -
 oracle.xml.parser.v2.XMLComment.getNodeName(), 11-117
 getNodeName() -
 oracle.xml.parser.v2.XMLNode.getNodeName(), 11-197
 getNodeName() -
 oracle.xml.parser.v2.XMLNotation.getNodeName(), 11-214
 getNodeName() -
 oracle.xml.parser.v2.XMLNSNode.getNodeName(), 11-222
 getNodeName() -
 oracle.xml.parser.v2.XMLPI.getNodeName(), 11-250
 getNodeName() -
 oracle.xml.parser.v2.XMLText.getNodeName(), 11-263
 getNodeTypeInfo() -
 oracle.xml.parser.schema.XSDIdentity.getNodeTypeInfo(), 7-42
 getNodeTypeInfo() -
 oracle.xml.parser.schema.XSDNode.getNodeTypeInfo(), 7-45
 getNodeTypeInfo() -
 oracle.xml.parser.v2.AttrDecl.getNodeTypeInfo(), 11-18
 getNodeTypeInfo() -
 oracle.xml.parser.v2.DTD.getNodeTypeInfo(), 11-64
 getNodeTypeInfo() -
 oracle.xml.parser.v2.ElementDecl.getNodeTypeInfo(), 11-75
 getNodeTypeInfo() -
 oracle.xml.parser.v2.XMLAttr.getNodeTypeInfo(), 11-108
 getNodeTypeInfo() -
 oracle.xml.parser.v2.XMLCDATA.getNodeTypeInfo(), 11-114
 getNodeTypeInfo() -
 oracle.xml.parser.v2.XMLComment.getNodeTypeInfo(), 11-117
 getNodeTypeInfo() -
 oracle.xml.parser.v2.XMLDocumentFragment.getNodeTypeInfo(), 11-154
 getNodeTypeInfo() -
 oracle.xml.parser.v2.XMLDocument.getNodeTypeInfo(), 11-141

getNodeTypeInfo() - oracle.xml.parser.v2.XMLElement.getNodeTypeInfo(), 11-165
 getNodeTypeInfo() - oracle.xml.parser.v2.XMLEntity.getNodeTypeInfo(), 11-178
 getNodeTypeInfo() - oracle.xml.parser.v2.XMLEntityReference.getNodeTypeInfo(), 11-182
 getNodeTypeInfo() - oracle.xml.parser.v2.XMLNode.getNodeTypeInfo(), 11-197
 getNodeTypeInfo() - oracle.xml.parser.v2.XMLNotation.getNodeTypeInfo(), 11-214
 getNodeTypeInfo() - oracle.xml.parser.v2.XMLPI.getNodeTypeInfo(), 11-250
 getNodeTypeInfo() - oracle.xml.parser.v2.XMLText.getNodeTypeInfo(), 11-263
 getNodeValue() - oracle.xml.classgen.CGXSDElement.getNodeValue(), 6-18
 getNodeValue() - oracle.xml.parser.v2.XMLAttr.getNodeValue(), 11-108
 getNodeValue() - oracle.xml.parser.v2.XMLDeclPI.getNodeValue(), 11-122
 getNodeValue() - oracle.xml.parser.v2.XMLEntity.getNodeValue(), 11-178
 getNodeValue() - oracle.xml.parser.v2.XMLNode.getNodeValue(), 11-197
 getNodeValue() - oracle.xml.parser.v2.XMLText.getNodeValue(), 11-263
 getNodeVector() - oracle.xml.parser.schema.XSDGroup.getNodeVector(), 7-40
 getNoLocal, 4-66
 getNotationName() - oracle.xml.parser.v2.XMLEntity.getNotationName(), 11-179
 getNotations() - oracle.xml.parser.v2.DTD.getNotations(), 11-65
 getNumMessages() - oracle.xml.parser.v2.XMLParseException.getNumMessages(), 11-235
 getNumMessages() - oracle.xml.util.XMLError.getNumMessages(), 10-14
 getNumMessages() - oracle.xml.util.XMLException.getNumMessages(), 10-21
 getNumRowsProcessed() - oracle.xml.sql.query.OracleXMLQuery.getNumRowsProcessed(), 9-6
 getObject, 4-103, 4-140
 getObjectPayload, 2-45
 getObjectProperty, 4-18, 4-122
 getOrder() - oracle.xml.parser.schema.XSDGroup.getOrder(), 7-40
 getOrigMessageId, 2-49
 getOutputEncoding() - oracle.xml.parser.v2.XSLStylesheet.getOutputEncoding(), 11-324
 getOutputMediaType() - oracle.xml.parser.v2.XSLStylesheet.getOutputMediaType(), 11-324
 getOutputProperties() - oracle.xml.jaxp.JXTransformer.getOutputProperties(), 11-298
 getOutputProperties() - oracle.xml.parser.v2.XSLStylesheet.getOutputProperties(), 11-325
 getOutputProperty(String) - oracle.xml.jaxp.JXTransformer.getOutputProperty(java.lang.String), 11-299
 getOutputStyle() - oracle.xml.parser.v2.XMLOutputStream.getOutputStyle(), 11-230
 getOverrideAQResponseFlag(), 3-12
 getOwner, 2-23, 2-35
 getOwnerDocument() - oracle.xml.parser.v2.XMLDocument.getOwner

Document(), 11-141
 getOwnerDocument() -
 oracle.xml.parser.v2.XMLNode.getOwnerDocument(), 11-198
 getOwnerElement() -
 oracle.xml.parser.v2.XMLAttr.getOwnerElement(), 11-109
 getOwnerId() -
 oracle.xdb.spi.XDBResourceContext.getOwnerId(), 23-19
 getOwnerId() -
 oracle.xdb.spi.XDBResource.getOwnerId(), 23-11
 getOwnerImplementation() -
 oracle.xml.parser.v2.DTD.getOwnerImplementation(), 11-65
 getParam(String) -
 oracle.xml.parser.v2.XSLProcessor.getParam(java.lang.String), 11-315
 getParameter(String) -
 oracle.xml.jaxp.JXTransformer.getParameter(java.lang.String), 11-300
 getParentException() -
 oracle.xml.sql.OracleXMLSQLException.getParentException(), 9-21
 getParentNode() -
 oracle.xml.parser.v2.XMLAttr.getParentNode(), 11-109
 getParentNode() -
 oracle.xml.parser.v2.XMLDocumentFragment.getParentNode(), 11-154
 getParentNode() -
 oracle.xml.parser.v2.XMLNode.getParentNode(), 11-198
 getParser() -
 oracle.xml.jaxp.JXSAXParser.getParser(), 11-282
 getParseTree() -
 oracle.xml.parser.v2.ElementDecl.getParseTree(), 11-75
 getPassword(), 13-5
 getPayloadData, 2-53
 getPayloadType, 2-15
 getPCDATAFont(), 15-7
 getPCDATAForeground(), 15-7
 getPIDataFont(), 15-7
 getPIDataForeground(), 15-8
 getPINameFont(), 15-8
 getPINameForeground(), 15-8
 getPingPeriod, 4-54
 getPort(), 3-10, 13-5
 getPrecision() -
 oracle.xml.parser.schema.XSDDataValue.getPrecision(), 7-29
 getPreferredSize(), 17-4
 getPrefix() -
 oracle.xml.parser.v2.XMLAttr.getPrefix(), 11-109
 getPrefix() -
 oracle.xml.parser.v2.XMLElement.getPrefix(), 11-165
 getPrefix() -
 oracle.xml.parser.v2.XMLNode.getPrefix(), 11-198
 getPrefix() -
 oracle.xml.parser.v2.XMLNSNode.getPrefix(), 11-222
 getPrefix() -
 oracle.xml.util.NSName.getPrefix(), 10-5, 11-14
 getPrefix(int) -
 oracle.xml.parser.v2.SAXAttrList.getPrefix(int), 11-89
 getPreviousSibling() -
 oracle.xml.parser.v2.XMLAttr.getPreviousSibling(), 11-110
 getPreviousSibling() -
 oracle.xml.parser.v2.XMLNode.getPreviousSibling(), 11-199
 getPrimaryInstance, 2-18
 getPrintWriter(), 3-25
 getPriority, 2-47, 4-147
 getProperty, 2-23, 2-35
 getProperty(String) -
 oracle.xml.jaxp.JXSAXParser.getProperty(java.lang.String), 11-283
 getProperty(String) -
 oracle.xml.parser.v2.SAXParser.getProperty(java.lang.String), 11-98
 getProperty(String) -

- oracle.xml.parser.v2.XMLNode.getProperty(java.lang.String), 11-199
- getPropertyDescriptors(), 12-16, 12-32, 13-19, 15-15, 16-10, 16-12, 17-6
- getPropertyNames, 4-19, 4-123
- getProtocol, 2-14, 4-30
- getProviderMajorVersion, 4-57
- getProviderMinorVersion, 4-58
- getPublicId() -
 - oracle.xml.parser.v2.DTD.getPublicId(), 11-65
- getPublicId() -
 - oracle.xml.parser.v2.XMLEntity.getPublicId(), 11-179
- getPublicId() -
 - oracle.xml.parser.v2.XMLNotation.getPublicId(), 11-215
- getPublicId(int) -
 - oracle.xml.parser.v2.XMLParseException.getPublicId(int), 11-235
- getPublicId(int) -
 - oracle.xml.util.XMLError.getPublicId(int), 10-14
- getPublicId(int) -
 - oracle.xml.util.XMLException.getPublicId(int), 10-22
- getQName(int) -
 - oracle.xml.parser.v2.SAXAttrList.getQName(int), 11-89
- getQualifiedName() -
 - oracle.xml.parser.v2.XMLElement.getQualifiedName(), 11-166
- getQualifiedName() -
 - oracle.xml.util.NSName.getQualifiedName(), 10-5, 11-14
- getQualifiedName(int) -
 - oracle.xml.parser.v2.SAXAttrList.getQualifiedName(int), 11-90
- getQueue, 2-9, 4-66, 4-148, 4-160, 4-200
- getQueueConnectionFactory, 4-88
- getQueueName, 4-75
- getQueueOwner, 4-75
- getQueueTable, 2-8, 4-200
- getQueueTableName, 2-35
- getQueueType, 2-20, 4-82
- getRawPayload, 2-45
- getRecipientList, 2-49
- getRefer() -
 - oracle.xml.parser.schema.XSDIdentity.getRefer(), 7-43
- getRefLocalname() -
 - oracle.xml.parser.schema.XSDAttribute.getRefLocalname(), 7-13
- getRefLocalname() -
 - oracle.xml.parser.schema.XSDComplexType.getRefLocalname(), 7-23
- getRefLocalname() -
 - oracle.xml.parser.schema.XSDElement.getRefLocalname(), 7-34
- getRefNamespace() -
 - oracle.xml.parser.schema.XSDAttribute.getRefNamespace(), 7-14
- getRefNamespace() -
 - oracle.xml.parser.schema.XSDElement.getRefNamespace(), 7-35
- getRefState() -
 - oracle.xml.parser.schema.XSDAttribute.getRefState(), 7-14
- getRefState() -
 - oracle.xml.parser.schema.XSDElement.getRefState(), 7-35
- getReleaseVersion(), 12-6, 16-13
- getReleaseVersion() -
 - oracle.xml.parser.v2.XMLParser.getReleaseVersion(), 11-239
- getRelMessageId, 2-40
- getResBuffer(), 13-5
- getResCLOBFileName(), 13-6
- getResCLOBTableName(), 13-6
- getResFileName(), 13-6
- getResource(), 12-24
- getResult(), 12-7, 12-27
- getRetentionTime, 2-21, 4-83
- getRetryInterval, 4-83
- getRootTag() -
 - oracle.xml.parser.v2.DTD.getRootTag(), 11-66
- getScale() -
 - oracle.xml.parser.schema.XSDDataValue.getScale(), 7-29
- getSchemaByTargetNS(String) -
 - oracle.xml.parser.schema.XMLSchema.getSchemaByTargetNS(String), 11-199

- maByTargetNS, 7-5
- getSecondaryInstance, 2-18
- getSelector() -
 - oracle.xml.parser.schema.XSDIdentity.getSelector(), 7-43
- getSender, 2-49
- getSenderID, 4-123
- getSequenceDeviation, 2-40
- getServerResponseDoc(), 3-12
- getShort, 4-103
- getShortProperty, 4-19, 4-124
- getSid(), 3-10
- getSimpleTypeSet() -
 - oracle.xml.parser.schema.XMLSchemaNode.getSimpleTypeSet(), 7-10
- getSimpleTypeTable() -
 - oracle.xml.parser.schema.XMLSchemaNode.getSimpleTypeTable(), 7-10
- getSortOrder, 2-16
- getSpecified() -
 - oracle.xml.parser.v2.XMLAttr.getSpecified(), 11-110
- getStandalone() -
 - oracle.xml.parser.v2.XMLDeclPI.getStandalone(), 11-122
- getStandalone() -
 - oracle.xml.parser.v2.XMLDocument.getStandalone(), 11-142
- getState, 2-50
- getStream, 2-51
- getString, 4-104
- getStringProperty, 4-20, 4-124
- getStyleSheetProcessingInstr(), 3-12
- getSubscribers, 2-38
- getSubstitutionGroup() -
 - oracle.xml.parser.schema.XSDElement.getSubstitutionGroup(), 7-35
- getSymbolFont(), 15-8
- getSymbolForeground(), 15-8
- getSystemId() -
 - oracle.xml.parser.v2.DTD.getSystemId(), 11-66
- getSystemId() -
 - oracle.xml.parser.v2.XMLDocument.getSystemId(), 11-142
- getSystemId() -
 - oracle.xml.parser.v2.XMLEntity.getSystemId(), 11-179
- getSystemId() -
 - oracle.xml.parser.v2.XMLNode.getSystemId(), 11-199
- getSystemId() -
 - oracle.xml.parser.v2.XMLNotation.getSystemId(), 11-215
- getSystemId(int) -
 - oracle.xml.parser.v2.XMLParseException.getSystemId(int), 11-236
- getSystemId(int) -
 - oracle.xml.util.XMLError.getSystemId(int), 10-15
- getSystemId(int) -
 - oracle.xml.util.XMLException.getSystemId(int), 10-22
- getTagFont(), 15-9
- getTagForeground(), 15-9
- getTagName() -
 - oracle.xml.parser.v2.XMLElement.getTagName(), 11-166
- getTarget() -
 - oracle.xml.parser.v2.XMLPI.getTarget(), 11-251
- getTargetNS() -
 - oracle.xml.parser.schema.XMLSchemaNode.getTargetNS(), 7-10
- getTargetNS() -
 - oracle.xml.parser.schema.XSDAttribute.getTargetNS(), 7-14
- getTargetNS() -
 - oracle.xml.parser.schema.XSDElement.getTargetNS(), 7-35
- getTargetNS() -
 - oracle.xml.parser.schema.XSDNode.getTargetNS(), 7-45
- getText, 4-222
- getText(), 15-9
- getText() -
 - oracle.xml.parser.v2.XMLDocument.getText(), 11-142
- getText() -
 - oracle.xml.parser.v2.XMLNode.getText(), 11-2

00

getText() -
 oracle.xml.parser.v2.XMLNSNode.getText(), 1-223

getTimeToLive, 4-148

getTopic, 4-67, 4-148, 4-201, 4-244

getTopicConnectionFactory, 4-89

getTopicName, 4-75

getTopicOwner, 4-76

getTraceLevel(), 3-25

getTransacted, 4-201

getTransformation, 4-160, 4-168, 4-233, 4-237

getTree(), 17-4

getType() -
 oracle.xml.parser.schema.XSDAttribute.getType(), 7-15

getType() -
 oracle.xml.parser.schema.XSDElement.getType(), 7-36

getType(int) -
 oracle.xml.parser.v2.SAXAttrList.getType(int), 11-90

getType(String) -
 oracle.xml.parser.v2.SAXAttrList.getType(java.lang.String), 11-91

getType(String, String) -
 oracle.xml.parser.v2.SAXAttrList.getType(java.lang.String, java.lang.String), 11-91

getTypeDefinitionTable() -
 oracle.xml.parser.schema.XMLSchemaNode.getTypeDefinitionTable(), 7-11

getTypeGroup() -
 oracle.xml.parser.schema.XSDComplexType.getTypeGroup(), 7-23

getURI(int) -
 oracle.xml.parser.v2.SAXAttrList.getURI(int), 11-92

getURIResolver() -
 oracle.xml.jaxp.JXSAXTransformerFactory.getURIResolver(), 11-290

getURIResolver() -
 oracle.xml.jaxp.JXTransformer.getURIResolver(), 11-300

getURL(String) -
 oracle.xml.sql.dml.OracleXMLSave.getURL(java.lang.String), 8-6

getUserCallback(), 3-16, 3-21

getUsername(), 13-6

getValidationMode(), 12-7

getValidationMode() -
 oracle.xml.parser.v2.XMLParser.getValidationMode(), 11-240

getValidationModeValue() -
 oracle.xml.parser.v2.XMLParser.getValidationModeValue(), 11-240

getValue() -
 oracle.xml.parser.v2.XMLAttr.getValue(), 11-110

getValue(int) -
 oracle.xml.parser.v2.SAXAttrList.getValue(int), 11-92

getValue(String) -
 oracle.xml.parser.v2.SAXAttrList.getValue(java.lang.String), 11-93

getValue(String, String) -
 oracle.xml.parser.v2.SAXAttrList.getValue(java.lang.String, java.lang.String), 11-93

getVariable(NSName, int) -
 oracle.xml.parser.v2.XSLTContext.getVariable(oracle.xml.parser.v2.NSName, int), 11-328

getVariety() -
 oracle.xml.parser.schema.XSDSimpleType.getVariety(), 7-47, 7-50

getVersion() -
 oracle.xml.parser.v2.XMLDeclPI.getVersion(), 11-122

getVersion() -
 oracle.xml.parser.v2.XMLDocument.getVersion(), 11-143

getVisibility, 2-39, 2-42

getWaitTime, 2-43

getXML(OracleXMLDocGen, boolean) -
 oracle.xml.sql.query.OracleXMLQuery.getXML(oracle.xml.sql.docgen.OracleXMLDocGen, boolean), 9-6

getXmlBuffer(), 13-6

getXmlCLOBFileName(), 13-6

getXmlCLOBTableName(), 13-7

getXMLData(Connection, String, String), 16-7

getXMLDOM() -

- oracle.xml.sql.query.OracleXMLQuery.getXMLDOM(), 9-6
- getXMLError() -
 - oracle.xml.util.XMLException.getXMLError(), 10-22
- getXMLErrorMessage() -
 - oracle.xml.sql.OracleXMLSQLException.getXMLErrorMessage(), 9-21
- getXmlFileName(), 13-7
- getXMLMetaData(int, boolean) -
 - oracle.xml.sql.query.OracleXMLQuery.getXMLMetaData(int, boolean), 9-7
- getXMLMetaData(int, boolean, OracleXMLDocGen) -
 - oracle.xml.sql.query.OracleXMLQuery.getXMLMetaData(int, boolean, oracle.xml.sql.docgen.OracleXMLDocGen), 9-8
- getXMLNames(Connection, String), 16-7
- getXMLProperty(String) -
 - oracle.xml.parser.v2.XMLParser.getXMLProperty(java.lang.String), 11-240
- getXMLReader() -
 - oracle.xml.jaxp.JXSAXParser.getXMLReader(), 11-283
- getXMLSAX(ContentHandler) -
 - oracle.xml.sql.query.OracleXMLQuery.getXMLSAX(org.xml.sax.ContentHandler), 9-8
- getXMLSchema() -
 - oracle.xml.sql.query.OracleXMLQuery.getXMLSchema(), 9-8
- getXMLSchemaNodeTable(), 7-6
- getXMLSchemaURLS() -
 - oracle.xml.parser.schema.XMLSchema.getXMLSchemaURLS(), 7-6
- getXMLSQLException() -
 - oracle.xml.sql.OracleXMLSQLException.getXMLSQLExceptionErrorString(), 9-21
- getXMLString() -
 - oracle.xml.sql.query.OracleXMLQuery.getXMLString(), 9-8
- getXMLString(Node, int) -
 - oracle.xml.sql.query.OracleXMLQuery.getXMLString(org.w3c.dom.Node, int), 9-9
- getXMLStringFromSQL(String), 13-7

- getXMLTableNames(Connection, String), 16-7
- getXMLTreeModel(), 17-4
- getXslBuffer(), 13-7
- getXslCLOBFileName(), 13-7
- getXslCLOBTableName(), 13-8
- getXslFileName(), 13-8
- GMONTH -
 - oracle.xml.parser.schema.XSDTypeConstants.GMONTH, 7-55
- GMONTH_DAY -
 - oracle.xml.parser.schema.XSDTypeConstants.GMONTH_DAY, 7-55
- grantQueuePrivilege, 2-30, 4-76
- grantSystemPrivilege, 4-202
- grantTopicPrivilege, 4-76
- GYEAR -
 - oracle.xml.parser.schema.XSDTypeConstants.GYEAR, 7-55
- GYEAR_MONTH -
 - oracle.xml.parser.schema.XSDTypeConstants.GYEAR_MONTH, 7-55

H

- hasAttribute(String) -
 - oracle.xml.parser.v2.XMLElement.hasAttribute(java.lang.String), 11-167
- hasAttributeNS(String, String) -
 - oracle.xml.parser.v2.XMLElement.hasAttributeNS(java.lang.String, java.lang.String), 11-167
- hasAttributes() -
 - oracle.xml.parser.v2.XMLElement.hasAttributes(), 11-168
- hasAttributes() -
 - oracle.xml.parser.v2.XMLNode.hasAttributes(), 11-200
- hasChildNodes() -
 - oracle.xml.parser.v2.DTD.hasChildNodes(), 11-66
- hasChildNodes() -
 - oracle.xml.parser.v2.XMLNode.hasChildNodes(), 11-200
- hasChildNodes() -
 - oracle.xml.parser.v2.XMLNSNode.hasChildNodes(), 11-223

hasFeature(String, String) -
oracle.xml.parser.v2.XMLDOMImplementation.
hasFeature(java.lang.String,
java.lang.String), 11-158
hasMoreElements, 4-161
hasMoreElements() -
oracle.xdb.spi.XDBNamingEnumeration.hasMore
Elements(), 23-8
HEX_BINARY -
oracle.xml.parser.schema.XSDTypeConstants.H
EX_BINARY, 7-55

I

iANY_SIMPLE -
oracle.xml.parser.schema.XSDTypeConstants.i
ANY_SIMPLE, 7-55
iANY_URI -
oracle.xml.parser.schema.XSDTypeConstants.i
ANY_URI, 7-55
iBASE64_BINARY -
oracle.xml.parser.schema.XSDTypeConstants.iB
ASE64_BINARY, 7-55
iBOOLEAN -
oracle.xml.parser.schema.XSDTypeConstants.iB
OOLEAN, 7-55
id, 12-36
DOMBuilderEvent, 12-20
ID -
oracle.xml.parser.schema.XSDTypeConstants.I
D, 7-55
ID_ATTR_DECL -
oracle.xml.parser.v2.ElementDecl.ID_ATTR_
DECL, 11-70, 11-323
iDATE -
oracle.xml.parser.schema.XSDTypeConstants.i
DATE, 7-55
iDATE_TIME -
oracle.xml.parser.schema.XSDTypeConstants.i
DATE_TIME, 7-55
iDECIMAL -
oracle.xml.parser.schema.XSDTypeConstants.i
DECIMAL, 7-55
iDOUBLE -
oracle.xml.parser.schema.XSDTypeConstants.i
DOUBLE, 7-55
IDREF -
oracle.xml.parser.schema.XSDTypeConstants.I
DREF, 7-55
IDREF -
oracle.xml.parser.v2.AttrDecl.IDREF, 11-15
IDREFS -
oracle.xml.parser.schema.XSDTypeConstants.I
DREFS, 7-55
IDREFS -
oracle.xml.parser.v2.AttrDecl.IDREFS, 11-15
iDUMMY -
oracle.xml.parser.schema.XSDTypeConstants.i
DUMMY, 7-55
iDURATION -
oracle.xml.parser.schema.XSDTypeConstants.i
DURATION, 7-56
iENUMERATION -
oracle.xml.parser.schema.XSDTypeConstants.iE
NUMERATION, 7-56
iFLOAT -
oracle.xml.parser.schema.XSDTypeConstants.iF
LOAT, 7-56
iFRACTION_DIGITS -
oracle.xml.parser.schema.XSDTypeConstants.iF
RACTION_DIGITS, 7-56
iGDAY -
oracle.xml.parser.schema.XSDTypeConstants.i
GDAY, 7-56
iGMONTH -
oracle.xml.parser.schema.XSDTypeConstants.i
GMONTH, 7-56
iGMONTH_DAY -
oracle.xml.parser.schema.XSDTypeConstants.i
GMONTH_DAY, 7-56
ignorableWhitespace(char[], int, int) -
oracle.xml.parser.v2.DocumentBuilder.ignorabl
eWhitespace(char[], int, int), 11-41
iGYEAR -
oracle.xml.parser.schema.XSDTypeConstants.i
GYEAR, 7-56
iGYEAR_MONTH -
oracle.xml.parser.schema.XSDTypeConstants.i
GYEAR_MONTH, 7-56
iHEX_BINARY -

- oracle.xml.parser.schema.XSDTypeConstants.i
HEX_BINARY, 7-56
- iLENGTH -
 - oracle.xml.parser.schema.XSDTypeConstants.i
LENGTH, 7-56
- iMAXEXCLUSIVE -
 - oracle.xml.parser.schema.XSDTypeConstants.i
MAXEXCLUSIVE, 7-56
- iMAXINCLUSIVE -
 - oracle.xml.parser.schema.XSDTypeConstants.i
MAXINCLUSIVE, 7-56
- iMAXLENGTH -
 - oracle.xml.parser.schema.XSDTypeConstants.i
MAXLENGTH, 7-56
- iMINEXCLUSIVE -
 - oracle.xml.parser.schema.XSDTypeConstants.i
MINEXCLUSIVE, 7-56
- iMININCLUSIVE -
 - oracle.xml.parser.schema.XSDTypeConstants.i
MININCLUSIVE, 7-56
- iMINLENGTH -
 - oracle.xml.parser.schema.XSDTypeConstants.i
MINLENGTH, 7-56
- IMPLIED -
 - oracle.xml.parser.v2.AttrDecl.IMPLIED, 11-15
- importNode(Node, boolean) -
 - oracle.xml.parser.v2.XMLDocument.importNo
de(org.w3c.dom.Node, boolean), 11-143
- init() -
 - oracle.xml.parser.schema.XSDComplexType.ini
t(), 7-23
- iNOTATION -
 - oracle.xml.parser.schema.XSDTypeConstants.i
NOTATION, 7-56
- inputDOMDocument, 15-3
- insertBefore(Node, Node) -
 - oracle.xml.parser.v2.XMLDocument.insertBefor
e(org.w3c.dom.Node,
org.w3c.dom.Node), 11-144
- insertBefore(Node, Node) -
 - oracle.xml.parser.v2.XMLNode.insertBefore(org
.w3c.dom.Node, org.w3c.dom.Node), 11-201
- insertBefore(Node, Node) -
 - oracle.xml.parser.v2.XMLNSNode.insertBefore(
org.w3c.dom.Node,
org.w3c.dom.Node), 11-224
- insertBLOBData(Connection, String, String,
byte[]), 16-8
- insertXML(Document) -
 - oracle.xml.sql.dml.OracleXMLSave.insertXML(
org.w3c.dom.Document), 8-6
- insertXMLData(Connection, String, String,
String), 16-8
- inSource, 12-4
- Installing CartridgeServices.jar, 5-2
- Installing ODCI.jar, 5-2
- installing ODCI.jar, if Java option is installed, 5-2
- inStream, 12-4
- inString, 12-4
- INT -
 - oracle.xml.parser.schema.XSDTypeConstants.I
NT, 7-56
- INTEGER -
 - oracle.xml.parser.schema.XSDTypeConstants.I
NTEGER, 7-56
- internalEntityDecl(String, String) -
 - oracle.xml.parser.v2.DocumentBuilder.internal
EntityDecl(java.lang.String,
java.lang.String), 11-33, 11-41
- InvalidContentException -
 - oracle.xml.classgen.InvalidContentException,
6-24
- InvalidContentException() -
 - oracle.xml.classgen.InvalidContentException.In
validContentException(), 6-24
- iPATTERN -
 - oracle.xml.parser.schema.XSDTypeConstants.iP
ATTERN, 7-56
- iQNAME -
 - oracle.xml.parser.schema.XSDTypeConstants.i
QNAME, 7-56
- isAbstract() -
 - oracle.xml.parser.schema.XSDComplexType.is
Abstract(), 7-23
- isAbstract() -
 - oracle.xml.parser.schema.XSDElement.isAbstra
ct(), 7-36
- isAbstract() -
 - oracle.xml.parser.schema.XSDSimpleType.isAb
stract(), 7-47, 7-51

- isEditable(), 15-9
- isExpandEntityReferences() -
 - oracle.xml.jaxp.JXDocumentBuilderFactory.isExpandEntityReferences(), 11-280
- isFixed(boolean) -
 - oracle.xml.parser.schema.XSDConstrainingFace.isFixed(boolean), 7-25, 7-26
- isIgnoringComments() -
 - oracle.xml.jaxp.JXDocumentBuilderFactory.isIgnoringComments(), 11-280
- isMulticonsumerEnabled, 2-16
- isNamespaceAware() -
 - oracle.xml.jaxp.JXDocumentBuilderFactory.isNamespaceAware(), 11-280
- isNamespaceAware() -
 - oracle.xml.jaxp.JXDocumentBuilder.isNamespaceAware(), 11-276
- isNamespaceAware() -
 - oracle.xml.jaxp.JXSAXParserFactory.isNamespaceAware(), 11-286
- isNamespaceAware() -
 - oracle.xml.jaxp.JXSAXParser.isNamespaceAware(), 11-284
- isNodeFlag(int) -
 - oracle.xml.parser.v2.XMLNode.isNodeFlag(int), 11-202
- isNodeType(int) -
 - oracle.xml.parser.schema.XSDNode.isNodeType(int), 7-45
- isNullable() -
 - oracle.xml.parser.schema.XSDElement.isNullable(), 7-36
- isRequired() -
 - oracle.xml.parser.schema.XSDAttribute.isRequired(), 7-15
- isSupported(String, String) -
 - oracle.xml.parser.v2.XMLNode.isSupported(java.lang.String, java.lang.String), 11-202
- ISTRING -
 - oracle.xml.parser.schema.XSDTypeConstants.ISTRING, 7-56
- isValidating() -
 - oracle.xml.jaxp.JXDocumentBuilder.isValidating(), 11-276
- isValidating() -
 - oracle.xml.jaxp.JXSAXParser.isValidating(), 11-284
- isWhiteSpaceNode() -
 - oracle.xml.parser.v2.XMLText.isWhiteSpaceNode(), 11-264
- isXMLPropertyReadOnly(String) -
 - oracle.xml.parser.v2.XMLParser.isXMLPropertyReadOnly(java.lang.String), 11-240
- isXMLPropertySupported(String) -
 - oracle.xml.parser.v2.XMLParser.isXMLPropertySupported(java.lang.String), 11-241
- isXMLTable(Connection, String), 16-9
- itemExists, 4-104
- iTIME -
 - oracle.xml.parser.schema.XSDTypeConstants.iTIME, 7-56
- iTOTAL_DIGITS -
 - oracle.xml.parser.schema.XSDTypeConstants.iTOTAL_DIGITS, 7-56
- iWHITESPACE -
 - oracle.xml.parser.schema.XSDTypeConstants.iWHITESPACE, 7-56

J

- javax, 4-58
- jScrollPane, 15-3
- jTextPane, 15-4
- JXDocumentBuilderFactory -
 - oracle.xml.jaxp.JXDocumentBuilderFactory, 11-278
- JXDocumentBuilderFactory() -
 - oracle.xml.jaxp.JXDocumentBuilderFactory.JXDocumentBuilderFactory(), 11-279
- JXSAXParser -
 - oracle.xml.jaxp.JXSAXParser, 11-282
- JXSAXParserFactory -
 - oracle.xml.jaxp.JXSAXParserFactory, 11-285
- JXSAXParserFactory() -
 - oracle.xml.jaxp.JXSAXParserFactory.JXSAXParserFactory(), 11-285
- JXSAXTransformerFactory -
 - oracle.xml.jaxp.JXSAXTransformerFactory, 11-288
- JXSAXTransformerFactory() -

oracle.xml.jaxp.JXSAXTransformerFactory.JSAXTransformerFactory(), 11-288
JXTransformer -
 oracle.xml.jaxp.JXTransformer, 11-297
JXTransformer() -
 oracle.xml.jaxp.JXTransformer.JXTransformer(), 11-297
JXTransformer(XSLStylesheet) -
 oracle.xml.jaxp.JXTransformer.JXTransformer(oracle.xml.parser.v2.XSLStylesheet), 11-297

K

keepCursorState(boolean) -
 oracle.xml.sql.query.OracleXMLQuery.keepCursorState(boolean), 9-9
keepObjectOpen(boolean) -
 oracle.xml.sql.query.OracleXMLQuery.keepObjectOpen(boolean), 9-9

L

LANGUAGE -
 oracle.xml.parser.schema.XSDTypeConstants.LANGUAGE, 7-56
LENGTH -
 oracle.xml.parser.schema.XSDTypeConstants.LENGTH, 7-57
line - oracle.xml.util.XMLError.line, 10-6
listen, 2-10
loadResBuffer(String), 13-8
loadResBuffer(String, String), 13-8
loadResBuffer(XMLDocument), 13-8
loadResBufferFromClob(), 13-9
loadResBufferFromFile(), 13-9
loadXmlBuffer(String), 13-9
loadXmlBuffer(String, String), 13-9
loadXmlBuffer(XMLDocument), 13-9
loadXmlBufferFromClob(), 13-9
loadXmlBufferFromFile(), 13-10
loadXMLBufferFromSQL(String), 13-10
loadXslBuffer(String), 13-10
loadXslBuffer(String, String), 13-10
loadXslBuffer(XMLDocument), 13-10
loadXslBufferFromClob(), 13-10

loadXslBufferFromFile(), 13-11

M

main(String[]), 16-13
main(String[]) -
 oracle.xml.parser.v2.oraxml.main(java.lang.String[]), 11-84
MAXEXCLUSIVE -
 oracle.xml.parser.schema.XSDTypeConstants.MAXEXCLUSIVE, 7-57
MAXINCLUSIVE -
 oracle.xml.parser.schema.XSDTypeConstants.MAXINCLUSIVE, 7-57
MAXLENGTH -
 oracle.xml.parser.schema.XSDTypeConstants.MAXLENGTH, 7-57
MAXROWS_ALL -
 oracle.xml.sql.query.OracleXMLQuery.MAXROWS_ALL, 9-4
msg - oracle.xml.util.XMLError.msg, 10-6
methodToCall, 12-4, 12-26
MINEXCLUSIVE -
 oracle.xml.parser.schema.XSDTypeConstants.MINEXCLUSIVE, 7-57
MININCLUSIVE -
 oracle.xml.parser.schema.XSDTypeConstants.MININCLUSIVE, 7-57
MINLENGTH -
 oracle.xml.parser.schema.XSDTypeConstants.MINLENGTH, 7-57
MIXED -
 oracle.xml.parser.v2.ElementDecl.MIXED, 11-70, 11-323
model, 17-3
MONTH -
 oracle.xml.parser.schema.XSDTypeConstants.MONTH, 7-57

N

N_STRING -
 oracle.xml.parser.schema.XSDTypeConstants.N_STRING, 7-57
NAME -

- oracle.xml.parser.schema.XSDTypeConstants.N
AME, 7-57
- NCNAME -
 - oracle.xml.parser.schema.XSDTypeConstants.N
CNAME, 7-57
- NEGATIVE_INTEGER -
 - oracle.xml.parser.schema.XSDTypeConstants.N
EGATIVE_INTEGER, 7-57
- newDocument() -
 - oracle.xml.jaxp.JXDocumentBuilder.newDocum
ent(), 11-276
- newDocumentBuilder() -
 - oracle.xml.jaxp.JXDocumentBuilderFactory.new
DocumentBuilder(), 11-281
- newSAXParser() -
 - oracle.xml.jaxp.JXSAXParserFactory.newSAXPa
rser(), 11-286
- newTemplates(Source) -
 - oracle.xml.jaxp.JXSAXTransformerFactory.new
Templates(javax.xml.transform.Source), 11-29
1
- newTemplatesHandler() -
 - oracle.xml.jaxp.JXSAXTransformerFactory.new
TemplatesHandler(), 11-291
- newTransformer() -
 - oracle.xml.jaxp.JXSAXTransformerFactory.new
Transformer(), 11-292
- newTransformer() -
 - oracle.xml.parser.v2.XSLStylesheet.newTransfo
rmer(), 11-325
- newTransformer(Source) -
 - oracle.xml.jaxp.JXSAXTransformerFactory.new
Transformer(javax.xml.transform.Source), 11-
292
- newTransformerHandler() -
 - oracle.xml.jaxp.JXSAXTransformerFactory.new
TransformerHandler(), 11-293
- newTransformerHandler(Source) -
 - oracle.xml.jaxp.JXSAXTransformerFactory.new
TransformerHandler(javax.xml.transform.Sourc
e), 11-293
- newTransformerHandler(Templates) -
 - oracle.xml.jaxp.JXSAXTransformerFactory.new
TransformerHandler(javax.xml.transform.Temp
lates), 11-294
- newXMLFilter(Source) -
 - oracle.xml.jaxp.JXSAXTransformerFactory.new
XMLFilter(javax.xml.transform.Source), 11-29
4
- newXMLFilter(Templates) -
 - oracle.xml.jaxp.JXSAXTransformerFactory.new
XMLFilter(javax.xml.transform.Templates), 11
-295
- newXSLStylesheet(InputStream) -
 - oracle.xml.parser.v2.XSLProcessor.newXSLStyl
esheet(java.io.InputStream), 11-315
- newXSLStylesheet(Reader) -
 - oracle.xml.parser.v2.XSLProcessor.newXSLStyl
esheet(java.io.Reader), 11-316
- newXSLStylesheet(URL) -
 - oracle.xml.parser.v2.XSLProcessor.newXSLStyl
esheet(java.net.URL), 11-316
- nextElement, 4-161
- nextElement() -
 - oracle.xdb.spi.XDBNamingEnumeration.nextEl
ement(), 23-8
- nFacets -
 - oracle.xml.parser.schema.XSDTypeConstants.n
Facets, 7-57
- NMTOKEN -
 - oracle.xml.parser.schema.XSDTypeConstants.N
MTOKEN, 7-57
- NMTOKEN -
 - oracle.xml.parser.v2.AttrDecl.NMTOKEN, 11-
15
- NMTOKENS -
 - oracle.xml.parser.schema.XSDTypeConstants.N
MTOKENS, 7-57
- NMTOKENS -
 - oracle.xml.parser.v2.AttrDecl.NMTOKENS, 1
1-15
- NODE_FACTORY -
 - oracle.xml.jaxp.JXDocumentBuilderFactory.NO
DE_FACTORY, 11-278
- NODE_FACTORY -
 - oracle.xml.parser.v2.DOMParser.NODE_
FACTORY, 11-49
- NodeFactory -
 - oracle.xml.parser.v2.NodeFactory, 11-77
- NodeFactory() -

- oracle.xml.parser.v2.NodeFactory.NodeFactory
(), 11-77
- NON_NEGATIVE_INTEGER -
 - oracle.xml.parser.schema.XSDTypeConstants.N
ON_NEGATIVE_INTEGER, 7-57
- NON_POSITIVE_INTEGER -
 - oracle.xml.parser.schema.XSDTypeConstants.N
ON_POSITIVE_INTEGER, 7-57
- noncapturing -
 - oracle.xml.parser.v2.XMLNode.noncapturing,
11-190
- NONE -
 - oracle.xml.sql.query.OracleXMLQuery.NONE,
9-4
- normalize() -
 - oracle.xml.parser.v2.DTD.normalize(), 11-67
- normalize() -
 - oracle.xml.parser.v2.XMLElement.normalize(),
11-168
- normalize() -
 - oracle.xml.parser.v2.XMLNode.normalize(), 1
1-202
- normalize() -
 - oracle.xml.parser.v2.XMLNSNode.normalize(),
11-225
- NOTATION -
 - oracle.xml.parser.v2.AttrDecl.NOTATION, 11
-15
- NotationDecl -
 - oracle.xml.parser.v2.XMLToken.NotationDecl,
11-268
- NSName - oracle.xml.parser.v2.NSName, 11-83
- NSName - oracle.xml.util.NSName, 10-4, 11-13
- NSResolver -
 - oracle.xml.parser.v2.NSResolver, 11-6

O

- ODCI, 5-1
- ODCI.jar and CartridgeServices.jar files, 5-2
- OR - oracle.xml.parser.v2.ElementDecl.OR, 11-71
- Oracle Data Cartridge Interface, 5-1
- Oracle JMS (Java Messaging Service), 4-1
- Oracle XDK Home, URL, 67, 331
- Oracle XML Developer's Kit for Java, URL, 67, 331

- Oracle XML Developer's Kit for JavaBeans,
URL, 67, 331, 17-2
- Oracle XML Transviewer Beans, 12-1, 13-1, 15-1,
16-1, 17-1
- oracle.AQ, 2-1
- oracle.AQ package, 2-1
- oracle.AQ.xml, 3-1
- oracle.AQ.xml package, 3-1
- oracle.jms package, 4-1
- oracle.ODCI, 5-1
- oracle.ODCI Description, 5-2
- oracle.ODCI package, 5-1
- oracle.security.rdbms.appctx, 1-1
- oracle.xdb.spi - oracle.xdb.spi, 23-2
- oracle.xdb.spi description, 23-2
- oracle.xml.async, 12-1, 13-1, 15-1, 16-1, 17-1
- oracle.xml.async - oracle.xml.async, 12-1, 13-1,
15-1, 16-1, 17-1
- oracle.xml.classgen - oracle.xml.classgen, 1-1, 6-1
- oracle.xml.parser.v2, 11-1
- OracleXMLQuery(Connection, ResultSet) -
 - oracle.xml.sql.query.OracleXMLQuery.OracleX
MLQuery(java.sql.Connection,
java.sql.ResultSet), 9-5
- OracleXMLSave(Connection, String) -
 - oracle.xml.sql.dml.OracleXMLSave.OracleXML
Save(java.sql.Connection,
java.lang.String), 8-4
- oracle.xml.sql.dml, 8-1
- OracleXMLSQLException(Exception) -
 - oracle.xml.sql.OracleXMLSQLException.Oracle
XMLSQLException(java.lang.Exception), 9-20
- OracleXMLSQLExceptionNoRowsException, 9-22
- OracleXMLSQLExceptionNoRowsException() -
 - oracle.xml.sql.OracleXMLSQLExceptionNoRowsExceptio
n.OracleXMLSQLExceptionNoRowsException(), 9-23
- oraxml() -
 - oracle.xml.parser.v2.oraxml.oraxml(), 11-83
- oraxsl, 11-306
- oraxsl - oracle.xml.parser.v2.oraxsl, 11-85
- oraxsl() -
 - oracle.xml.parser.v2.oraxsl.oraxsl(), 11-307
- out -
 - oracle.xml.parser.v2.XMLPrintDriver.out, 11-2
53

output -
oracle.xml.parser.v2.XSLStylesheet.output, 11-323

P

Package oracle.xdb.dom, 22-3
Package oracle.xml.sql.dml, 8-1
parse(InputSource), 12-7
parse(InputSource) -
oracle.xml.jaxp.JXDocumentBuilder.parse(org.xml.sax.InputSource), 11-276
parse(InputSource) -
oracle.xml.parser.v2.XMLParser.parse(org.xml.sax.InputSource), 11-241
parse(InputStream), 12-7
parse(InputStream) -
oracle.xml.parser.v2.XMLParser.parse(java.io.InputStream), 11-241
parse(Reader), 12-8
parse(Reader) -
oracle.xml.parser.v2.XMLParser.parse(java.io.Reader), 11-242
parse(String), 12-8
parse(String) -
oracle.xml.parser.v2.XMLParser.parse(java.lang.String), 11-243
parse(URL), 12-8
parse(URL) -
oracle.xml.parser.v2.XMLParser.parse(java.net.URL), 11-243
parseDocument() -
oracle.xml.parser.v2.XMLTokenizer.parseDocument(), 11-270
parseDTD(InputSource, String), 12-9
parseDTD(InputSource, String) -
oracle.xml.parser.v2.DOMParser.parseDTD(org.xml.sax.InputSource, java.lang.String), 11-52
parseDTD(InputStream, String), 12-9
parseDTD(InputStream, String) -
oracle.xml.parser.v2.DOMParser.parseDTD(java.io.InputStream, java.lang.String), 11-52
parseDTD(Reader, String), 12-10
parseDTD(Reader, String) -
oracle.xml.parser.v2.DOMParser.parseDTD(java.io.Reader, java.lang.String), 11-53
parseDTD(String, String), 12-10
parseDTD(String, String) -
oracle.xml.parser.v2.DOMParser.parseDTD(java.lang.String, java.lang.String), 11-54
parseDTD(URL, String), 12-11
parseDTD(URL, String) -
oracle.xml.parser.v2.DOMParser.parseDTD(java.net.URL, java.lang.String), 11-54
parseRequestStream(), 3-12
parseResBuffer(), 13-11
parseXmlBuffer(), 13-11
parseXslBuffer(), 13-11
PATTERN -
oracle.xml.parser.schema.XSDTypeConstants.PATTERN, 7-57
PERIOD -
oracle.xml.parser.schema.XSDTypeConstants.PERIOD, 7-57
PI - oracle.xml.parser.v2.XMLToken.PI, 11-268
PITarget -
oracle.xml.parser.v2.XMLToken.PITarget, 11-269
PLUS -
oracle.xml.parser.v2.ElementDecl.PLUS, 11-71
POSITIVE_INTEGER -
oracle.xml.parser.schema.XSDTypeConstants.POSITIVE_INTEGER, 7-57
PRECISION -
oracle.xml.parser.schema.XSDTypeConstants.PRECISION, 7-57
PRETTY -
oracle.xml.parser.v2.XMLOutputStream.PRETTY, 11-228
print(OutputStream) -
oracle.xml.classgen.CGDocument.print(java.io.OutputStream), 6-5
print(OutputStream) -
oracle.xml.parser.v2.XMLDocument.print(java.io.OutputStream), 11-145
print(OutputStream) -
oracle.xml.parser.v2.XMLNode.print(java.io.OutputStream), 11-202
print(OutputStream, String) -
oracle.xml.parser.v2.XMLDocument.print(java.i

- o.OutputStream, java.lang.String), 11-145
- print(OutputStream, String) -
 - oracle.xml.parser.v2.XMLNode.print(java.io.OutputStream, java.lang.String), 11-203
- print(PrintDriver) -
 - oracle.xml.parser.v2.XMLDocument.print(oracle.xml.parser.v2.PrintDriver), 11-146
- print(PrintWriter) -
 - oracle.xml.parser.v2.XMLDocument.print(java.io.PrintWriter), 11-146
- print(PrintWriter) -
 - oracle.xml.parser.v2.XMLNode.print(java.io.PrintWriter), 11-203
- print(XMLOutputStream) -
 - oracle.xml.classgen.CGXSElement.print(oracle.xml.parser.v2.XMLOutputStream), 6-18
- printAttribute(XMLAttr) -
 - oracle.xml.parser.v2.XMLPrintDriver.printAttribute(oracle.xml.parser.v2.XMLAttr), 11-254
- printAttributeNodes(XMLElement) -
 - oracle.xml.parser.v2.PrintDriver.printAttributeNodes(oracle.xml.parser.v2.XMLElement), 11-8
- printAttributeNodes(XMLElement) -
 - oracle.xml.parser.v2.XMLPrintDriver.printAttributeNodes(oracle.xml.parser.v2.XMLElement), 11-255
- printAttributes(XMLOutputStream, String, String) -
 - oracle.xml.classgen.CGXSElement.printAttributes(oracle.xml.parser.v2.XMLOutputStream, java.lang.String, java.lang.String), 6-19
- printCDATASection(XMLCDATA) -
 - oracle.xml.parser.v2.PrintDriver.printCDATASection(oracle.xml.parser.v2.XMLCDATA), 11-9
- printCDATASection(XMLCDATA) -
 - oracle.xml.parser.v2.XMLPrintDriver.printCDATASection(oracle.xml.parser.v2.XMLCDATA), 11-255
- printChildNodes(XMLNode) -
 - oracle.xml.parser.v2.PrintDriver.printChildNodes(oracle.xml.parser.v2.XMLNode), 11-9
- printChildNodes(XMLNode) -
 - oracle.xml.parser.v2.XMLPrintDriver.printChildNodes(oracle.xml.parser.v2.XMLNode), 11-2

- 56
- printComment(XMLComment) -
 - oracle.xml.parser.v2.PrintDriver.printComment(oracle.xml.parser.v2.XMLComment), 11-9
- printComment(XMLComment) -
 - oracle.xml.parser.v2.XMLPrintDriver.printComment(oracle.xml.parser.v2.XMLComment), 11-256
- printDoctype(DTD) -
 - oracle.xml.parser.v2.PrintDriver.printDoctype(oracle.xml.parser.v2.DTD), 11-10
- printDoctype(DTD) -
 - oracle.xml.parser.v2.XMLPrintDriver.printDoctype(oracle.xml.parser.v2.DTD), 11-256
- printDocument(XMLDocument) -
 - oracle.xml.parser.v2.PrintDriver.printDocument(oracle.xml.parser.v2.XMLDocument), 11-10
- printDocument(XMLDocument) -
 - oracle.xml.parser.v2.XMLPrintDriver.printDocument(oracle.xml.parser.v2.XMLDocument), 11-257
- printDocumentFragment(XMLDocumentFragment) -
 - oracle.xml.parser.v2.PrintDriver.printDocumentFragment(oracle.xml.parser.v2.XMLDocumentFragment), 11-10
- printDocumentFragment(XMLDocumentFragment) -
 - oracle.xml.parser.v2.XMLPrintDriver.printDocumentFragment(oracle.xml.parser.v2.XMLDocumentFragment), 11-257
- PrintDriver -
 - oracle.xml.parser.v2.PrintDriver, 11-85
- printElement(XMLElement) -
 - oracle.xml.parser.v2.PrintDriver.printElement(oracle.xml.parser.v2.XMLElement), 11-10
- printElement(XMLElement) -
 - oracle.xml.parser.v2.XMLPrintDriver.printElement(oracle.xml.parser.v2.XMLElement), 11-257
- printEntityReference(XMLEntityReference) -
 - oracle.xml.parser.v2.PrintDriver.printEntityReference(oracle.xml.parser.v2.XMLEntityReference), 11-11
- printEntityReference(XMLEntityReference) -
 - oracle.xml.parser.v2.XMLPrintDriver.printEntityReference(oracle.xml.parser.v2.XMLEntityReference), 11-257

- yReference(oracle.xml.parser.v2.XMLEntityReference), 11-258
- printErrorListener() -
 - oracle.xml.util.XMLError.printErrorListener(), 10-15
- printExternalDTD(OutputStream) -
 - oracle.xml.parser.v2.DTD.printExternalDTD(java.io.OutputStream), 11-67
- printExternalDTD(OutputStream) -
 - oracle.xml.parser.v2.XMLDocument.printExternalDTD(java.io.OutputStream), 11-146
- printExternalDTD(OutputStream, String) -
 - oracle.xml.parser.v2.DTD.printExternalDTD(java.io.OutputStream, java.lang.String), 11-61, 11-67
- printExternalDTD(OutputStream, String) -
 - oracle.xml.parser.v2.XMLDocument.printExternalDTD(java.io.OutputStream, java.lang.String), 11-147
- printExternalDTD(PrintWriter) -
 - oracle.xml.parser.v2.DTD.printExternalDTD(java.io.PrintWriter), 11-68
- printExternalDTD(PrintWriter) -
 - oracle.xml.parser.v2.XMLDocument.printExternalDTD(java.io.PrintWriter), 11-147
- printProcessingInstruction(XMLPI) -
 - oracle.xml.parser.v2.PrintDriver.printProcessingInstruction(oracle.xml.parser.v2.XMLPI), 11-11
- printProcessingInstruction(XMLPI) -
 - oracle.xml.parser.v2.XMLPrintDriver.printProcessingInstruction(oracle.xml.parser.v2.XMLPI), 11-258
- printSchema() -
 - oracle.xml.parser.schema.XMLSchema.printSchema(), 7-6
- printStackTrace() -
 - oracle.xml.util.XMLException.printStackTrace(), 10-23
- printStackTrace(PrintStream) -
 - oracle.xml.util.XMLException.printStackTrace(java.io.PrintStream), 10-23
- printStackTrace(PrintWriter) -
 - oracle.xml.util.XMLException.printStackTrace(java.io.PrintWriter), 10-23
- printTextNode(XMLText) -
 - oracle.xml.parser.v2.PrintDriver.printTextNode(oracle.xml.parser.v2.XMLText), 11-11
- printTextNode(XMLText) -
 - oracle.xml.parser.v2.XMLPrintDriver.printTextNode(oracle.xml.parser.v2.XMLText), 11-259
- processAction(XSLTContext) -
 - oracle.xml.parser.v2.XSLExtensionElement.processAction(oracle.xml.parser.v2.XSLTContext), 11-313
- processContent(XSLTContext) -
 - oracle.xml.parser.v2.XSLExtensionElement.processContent(oracle.xml.parser.v2.XSLTContext), 11-313
- processingInstruction(String, String) -
 - oracle.xml.parser.v2.DocumentBuilder.processingInstruction(java.lang.String, java.lang.String), 11-42
- processXSL(XSLStylesheet, InputStream, URL), 12-28
- processXSL(XSLStylesheet, InputStream, URL) -
 - oracle.xml.parser.v2.XSLProcessor.processXSL(oracle.xml.parser.v2.XSLStylesheet, java.io.InputStream, java.net.URL), 11-316
- processXSL(XSLStylesheet, Reader, URL), 12-28
- processXSL(XSLStylesheet, URL, URL), 12-28
- processXSL(XSLStylesheet, XMLDocument), 12-29
- processXSL(XSLStylesheet, XMLDocument, OutputStream), 12-29
- processXSL(XSLStylesheet, XMLElement, PrintWriter) -
 - oracle.xml.parser.v2.XSLProcessor.processXSL(oracle.xml.parser.v2.XSLStylesheet, oracle.xml.parser.v2.XMLElement, java.io.PrintWriter), 11-319
- processXSL(XSLStylesheet, XMLElement, XMLDocumentHandler) -
 - oracle.xml.parser.v2.XSLProcessor.processXSL(oracle.xml.parser.v2.XSLStylesheet, oracle.xml.parser.v2.XMLElement, oracle.xml.parser.v2.XMLDocumentHandler), 11-319
- propertyExists, 4-20, 4-125
- pubId - oracle.xml.util.XMLError.pubId, 10-6
- public, 4-235

publish, 4-149, 4-150, 4-151, 4-152, 4-233, 4-234

Q

QMARK -

oracle.xml.parser.v2.ElementDecl.QMARK, 11-71

QNAME -

oracle.xml.parser.schema.XSDTypeConstants.QNAME, 7-57

R

RANGE_DELETETEXT_EVENT -

oracle.xml.parser.v2.XMLNode.RANGE_DELETETEXT_EVENT, 11-190

RANGE_REPLACE_EVENT -

oracle.xml.parser.v2.XMLNode.RANGE_REPLACE_EVENT, 11-190

RANGE_SETTEXT_EVENT -

oracle.xml.parser.v2.XMLNode.RANGE_SETTEXT_EVENT, 11-190

readBoolean, 4-35, 4-209

readByte, 4-36, 4-210

readBytes, 4-36, 4-37, 4-210

readChar, 4-37, 4-210

readChildNodes(XMLObjectInput, CXMLContext) -

oracle.xml.parser.v2.XMLNode.readChildNodes(oracle.xml.io.XMLObjectInput, oracle.xml.comp.CXMLContext), 11-203

readDouble, 4-38, 4-211

reader, 12-5

readExternal(ObjectInput) -

oracle.xml.parser.v2.AttrDecl.readExternal(java.io.ObjectInput), 11-18

readExternal(ObjectInput) -

oracle.xml.parser.v2.DTD.readExternal(java.io.ObjectInput), 11-61, 11-68

readExternal(ObjectInput) -

oracle.xml.parser.v2.ElementDecl.readExternal(java.io.ObjectInput), 11-75

readExternal(ObjectInput) -

oracle.xml.parser.v2.XMLAttr.readExternal(java.io.ObjectInput), 11-111

readExternal(ObjectInput) -

oracle.xml.parser.v2.XMLCDATA.readExternal(java.io.ObjectInput), 11-114

readExternal(ObjectInput) -

oracle.xml.parser.v2.XMLComment.readExternal(java.io.ObjectInput), 11-117

readExternal(ObjectInput) -

oracle.xml.parser.v2.XMLDeclPI.readExternal(java.io.ObjectInput), 11-123

readExternal(ObjectInput) -

oracle.xml.parser.v2.XMLDocument.readExternal(java.io.ObjectInput), 11-147

readExternal(ObjectInput) -

oracle.xml.parser.v2.XMLElement.readExternal(java.io.ObjectInput), 11-169

readExternal(ObjectInput) -

oracle.xml.parser.v2.XMLEntity.readExternal(java.io.ObjectInput), 11-180

readExternal(ObjectInput) -

oracle.xml.parser.v2.XMLEntityReference.readExternal(java.io.ObjectInput), 11-183

readExternal(ObjectInput) -

oracle.xml.parser.v2.XMLNode.readExternal(java.io.ObjectInput), 11-203

readExternal(ObjectInput) -

oracle.xml.parser.v2.XMLNotation.readExternal(java.io.ObjectInput), 11-215

readExternal(ObjectInput) -

oracle.xml.parser.v2.XMLPI.readExternal(java.io.ObjectInput), 11-251

readExternal(ObjectInput) -

oracle.xml.parser.v2.XMLText.readExternal(java.io.ObjectInput), 11-264

readExternal(ObjectInput, CXMLContext) -

oracle.xml.classgen.CGDocument.readExternal(java.io.ObjectInput,

oracle.xml.comp.CXMLContext), 6-5

readExternal(XMLObjectInput, CXMLContext) -

oracle.xml.classgen.CGNode.readExternal(oracle.xml.io.XMLObjectInput,

oracle.xml.comp.CXMLContext), 6-12

readExternal(XMLObjectInput, CXMLContext) -

oracle.xml.parser.v2.AttrDecl.readExternal(oracle.xml.io.XMLObjectInput,

oracle.xml.comp.CXMLContext), 11-19

readExternal(XMLObjectInput, CXMLContext) -

- oracle.xml.parser.v2.DTD.readExternal(oracle.xml.io.XMLObjectInput, oracle.xml.comp.CXMLContext), 11-69
- readExternal(XMLObjectInput, CXMLContext) - oracle.xml.parser.v2.ElementDecl.readExternal(oracle.xml.io.XMLObjectInput, oracle.xml.comp.CXMLContext), 11-72
- readExternal(XMLObjectInput, CXMLContext) - oracle.xml.parser.v2.XMLComment.readExternal(oracle.xml.io.XMLObjectInput, oracle.xml.comp.CXMLContext), 11-118
- readExternal(XMLObjectInput, CXMLContext) - oracle.xml.parser.v2.XMLDeclPI.readExternal(oracle.xml.io.XMLObjectInput, oracle.xml.comp.CXMLContext), 11-123
- readExternal(XMLObjectInput, CXMLContext) - oracle.xml.parser.v2.XMLElement.readExternal(oracle.xml.io.XMLObjectInput, oracle.xml.comp.CXMLContext), 11-169
- readExternal(XMLObjectInput, CXMLContext) - oracle.xml.parser.v2.XMLEntity.readExternal(oracle.xml.io.XMLObjectInput, oracle.xml.comp.CXMLContext), 11-180
- readExternal(XMLObjectInput, CXMLContext) - oracle.xml.parser.v2.XMLEntityReference.readExternal(oracle.xml.io.XMLObjectInput, oracle.xml.comp.CXMLContext), 11-183
- readExternal(XMLObjectInput, CXMLContext) - oracle.xml.parser.v2.XMLNode.readExternal(oracle.xml.io.XMLObjectInput, oracle.xml.comp.CXMLContext), 11-204
- readExternal(XMLObjectInput, CXMLContext) - oracle.xml.parser.v2.XMLPI.readExternal(oracle.xml.io.XMLObjectInput, oracle.xml.comp.CXMLContext), 11-251
- readFloat, 4-38
- readInt, 4-39
- readLong, 4-39, 4-212
- readObject, 4-213
- readShort, 4-39
- readString, 4-214
- readUnsignedByte, 4-40
- readUnsignedShort, 4-40
- readUTF, 4-41
- receive, 4-67
- receiveNoData, 4-68, 4-166, 4-237, 4-240
- receiveNoWait, 4-69
- RECURRING_DATE - oracle.xml.parser.schema.XSDTypeConstants.RECURRING_DATE, 7-57
- RECURRING_DAY - oracle.xml.parser.schema.XSDTypeConstants.RECURRING_DAY, 7-57
- RECURRING_DURATION - oracle.xml.parser.schema.XSDTypeConstants.RECURRING_DURATION, 7-57
- Reference - oracle.xml.parser.v2.XMLToken.Reference, 11-269
- registerConnectionFactory, 4-90, 4-91, 4-92
- releaseResource(), 12-24
- removeAttribute(String) - oracle.xml.parser.v2.XMLElement.removeAttribute(java.lang.String), 11-169
- removeAttributeNode(Attr) - oracle.xml.parser.v2.XMLElement.removeAttributeNode(org.w3c.dom.Attr), 11-170
- removeAttributeNS(String, String) - oracle.xml.parser.v2.XMLElement.removeAttributeNS(java.lang.String, java.lang.String), 11-170
- removeChild(Node) - oracle.xml.parser.v2.XMLDocument.removeChild(org.w3c.dom.Node), 11-148
- removeChild(Node) - oracle.xml.parser.v2.XMLNode.removeChild(org.w3c.dom.Node), 11-204
- removeChild(Node) - oracle.xml.parser.v2.XMLNSNode.removeChild(org.w3c.dom.Node), 11-225
- removeDOMBuilderErrorListener(DOMBuilderErrorListener), 12-11
- removeDOMBuilderListener(DOMBuilderListener), 12-11
- removeDOMTransformerErrorListener(XSLTransformerErrorListener), 12-29
- removeEventListener(String, EventListener, boolean) - oracle.xml.parser.v2.XMLNode.removeEventListener(java.lang.String,

- org.w3c.dom.events.EventListener, boolean), 11-204
- removeParam(String) - oracle.xml.parser.v2.XSLStylesheet.removeParam(java.lang.String), 11-326
- removeParam(String, String) - oracle.xml.parser.v2.XSLProcessor.removeParam(java.lang.String, java.lang.String), 11-319
- removeSubscriber, 2-29
- removeXSLTParam(String) - oracle.xml.sql.dml.OracleXMLSave.removeXSLTParam(java.lang.String), 8-7
- removeXSLTParam(String) - oracle.xml.sql.query.OracleXMLQuery.removeXSLTParam(java.lang.String), 9-9
- removeXSLTransformerListener(XSLTransformerListener), 12-30
- replaceChild(Node, Node) - oracle.xml.parser.v2.XMLDocument.replaceChild(org.w3c.dom.Node, org.w3c.dom.Node), 11-148
- replaceChild(Node, Node) - oracle.xml.parser.v2.XMLNode.replaceChild(org.w3c.dom.Node, org.w3c.dom.Node), 11-205
- replaceChild(Node, Node) - oracle.xml.parser.v2.XMLNSNode.replaceChild(org.w3c.dom.Node, org.w3c.dom.Node), 11-226
- replaceXMLData(Connection, String, String, String), 16-9
- reportCharacters(String, boolean) - oracle.xml.parser.v2.XSLTContext.reportCharacters(java.lang.String, boolean), 11-328
- reportNode(XMLNode) - oracle.xml.parser.v2.XSLTContext.reportNode(oracle.xml.parser.v2.XMLNode), 11-328
- reportSAXEvents(ContentHandler) - oracle.xml.parser.v2.XMLComment.reportSAXEvents(org.xml.sax.ContentHandler), 11-118
- reportSAXEvents(ContentHandler) - oracle.xml.parser.v2.XMLDocument.reportSAXEvents(org.xml.sax.ContentHandler), 11-149
- reportSAXEvents(ContentHandler) - oracle.xml.parser.v2.XMLElement.reportSAXEvents(org.xml.sax.ContentHandler), 11-171
- reportSAXEvents(ContentHandler) - oracle.xml.parser.v2.XMLNode.reportSAXEvents(org.xml.sax.ContentHandler), 11-205
- reportSAXEvents(ContentHandler) - oracle.xml.parser.v2.XMLPI.reportSAXEvents(org.xml.sax.ContentHandler), 11-251
- reportSAXEvents(ContentHandler) - oracle.xml.parser.v2.XMLText.reportSAXEvents(org.xml.sax.ContentHandler), 11-264
- REQUIRED - oracle.xml.parser.v2.AttrDecl.REQUIRED, 11-16, 11-21
- reset, 4-41, 4-214
- reset() - oracle.xml.parser.v2.DOMParser.reset(), 11-55
- reset() - oracle.xml.parser.v2.SAXAttrList.reset(), 11-94
- reset() - oracle.xml.parser.v2.XMLParser.reset(), 11-243
- reset() - oracle.xml.util.XMLERror.reset(), 10-15
- resetNodeFlag(int) - oracle.xml.parser.v2.XMLNode.resetNodeFlag(int), 11-206
- resetParams() - oracle.xml.parser.v2.XSLProcessor.resetParams(), 11-319
- resetParams() - oracle.xml.parser.v2.XSLStylesheet.resetParams(), 11-326
- resolveNamespacePrefix(String), 11-6
- resolveNamespacePrefix(String) - oracle.xml.parser.v2.XMLElement.resolveNamespacePrefix(java.lang.String), 11-171
- ResourceManager, 12-24
- ResourceManager(int), 12-24
- result, 12-5, 12-26
- retainCDATASection(boolean) - oracle.xml.parser.v2.DocumentBuilder.retainCDATASection(boolean), 11-42
- retainCDATASection(boolean) - oracle.xml.parser.v2.DOMParser.retainCDATASection(boolean), 11-55
- revokeQueuePrivilege, 2-30, 4-77
- revokeSystemPrivilege, 4-203
- revokeTopicPrivilege, 4-77

rollback, 4-203
rootName, 12-5
run, 4-204
run(), 12-11, 12-30

S

saveResBuffer(String), 13-11
saveResBuffer(String, String), 13-11
saveResBufferToClob(), 13-12
saveResBufferToFile(), 13-12
saveXmlBuffer(String), 13-12
saveXmlBuffer(String, String), 13-12
saveXmlBufferToClob(), 13-12
saveXmlBufferToFile(), 13-12
saveXslBuffer(String), 13-13
saveXslBuffer(String, String), 13-13
saveXslBufferToClob(), 13-13
saveXslBufferToFile(), 13-13
SAX, 11-1
SAXAttrList, 11-308
SAXAttrList -
 oracle.xml.parser.v2.SAXAttrList, 11-85
SAXAttrList(int) -
 oracle.xml.parser.v2.SAXAttrList.SAXAttrList(int), 11-86
SAXParser - oracle.xml.parser.v2.SAXParser, 11-95
SAXParser() -
 oracle.xml.parser.v2.SAXParser.SAXParser(), 11-95
SCALE -
 oracle.xml.parser.schema.XSDTypeConstants.SCALE, 7-58
schedulePropagation, 2-30, 4-78
SCHEMA -
 oracle.xml.sql.query.OracleXMLQuery.SCHEMA, 9-4
SCHEMA_OBJECT -
 oracle.xml.jaxp.JXDocumentBuilderFactory.SCHEMA_OBJECT, 11-278
SCHEMA_OBJECT -
 oracle.xml.parser.v2.XMLParser.SCHEMA_OBJECT, 11-237
SchemaClassGenerator() -
 oracle.xml.classgen.SchemaClassGenerator.SchemaClassGenerator(), 6-26
scrollPane, 17-3
selectNodeAt(int), 15-9
selectNodes(String) -
 oracle.xml.parser.v2.XMLNode.selectNodes(java.lang.String), 11-206
selectNodes(String, NSResolver) -
 oracle.xml.parser.v2.XMLNode.selectNodes(java.lang.String, oracle.xml.parser.v2.NSResolver), 11-207
selectNodes(XSLNodeSetInt) -
 oracle.xml.parser.v2.XMLNode.selectNodes(oracle.xml.parser.v2.XSLNodeSetInt), 11-207
selectSingleNode(String) -
 oracle.xml.parser.v2.XMLNode.selectSingleNode(java.lang.String), 11-207
selectSingleNode(String, NSResolver) -
 oracle.xml.parser.v2.XMLNode.selectSingleNode(java.lang.String, oracle.xml.parser.v2.NSResolver), 11-208
send, 4-153, 4-154
setACL(String) -
 oracle.xdb.spi.XDBResource.setACL(java.lang.String), 23-11
setAddress, 2-14, 4-30
setAdtPayload, 4-9, 4-21
setAQDataSource(AQxmlDataSource), 3-16, 3-21
setAQSchemaLocation(String), 3-16, 3-21
setAttribute(String, Object) -
 oracle.xml.jaxp.JXDocumentBuilderFactory.setAttribute(java.lang.String, java.lang.Object), 11-281
setAttribute(String, Object) -
 oracle.xml.jaxp.JXSAXTransformerFactory.setAttribute(java.lang.String, java.lang.Object), 11-295
setAttribute(String, Object) -
 oracle.xml.parser.v2.DOMParser.setAttribute(java.lang.String, java.lang.Object), 11-55
setAttribute(String, Object) -
 oracle.xml.parser.v2.XMLParser.setAttribute(java.lang.String, java.lang.Object), 11-244
setAttribute(String, String) -
 oracle.xml.classgen.CGNode.setAttribute(java.lang.String, java.lang.String), 6-12

setAttribute(String, String) -
 oracle.xml.parser.v2.XMLElement.setAttribute(
 java.lang.String, java.lang.String), 11-171
 setAttributeNameFont(Font), 15-9
 setAttributeNameForeground(Color), 15-10
 setAttributeNode(Attr) -
 oracle.xml.parser.v2.XMLElement.setAttribute
 Node(org.w3c.dom.Attr), 11-172
 setAttributeNodeNS(Attr) -
 oracle.xml.parser.v2.XMLElement.setAttribute
 NodeNS(org.w3c.dom.Attr), 11-173
 setAttributeNS(String, String, String) -
 oracle.xml.parser.v2.XMLElement.setAttribute
 NS(java.lang.String, java.lang.String,
 java.lang.String), 11-173
 setAttributeValueFont(Font), 15-10
 setAttributeValueForeground(Color), 15-10
 setAuthor(String) -
 oracle.xdb.spi.XDBResourceContext.setAuthor(
 java.lang.String), 23-19
 setAuthor(String) -
 oracle.xdb.spi.XDBResource.setAuthor(java.lan
 g.String), 23-12
 setBackground(Color), 15-10
 setBaseURL(URL), 12-12
 setBaseURL(URL) -
 oracle.xml.parser.v2.XMLParser.setBaseURL(ja
 va.net.URL), 11-244
 setBaseURL(URL) -
 oracle.xml.parser.v2.XSLProcessor.setBaseURL(
 java.net.URL), 11-319
 setBatchSize(int) -
 oracle.xml.sql.dml.OracleXMLSave.setBatchSize
 (int), 8-8
 setBoolean, 4-105
 setBooleanProperty, 4-21, 4-125
 setByte, 4-105
 setByteProperty, 4-22, 4-125
 setBytes, 4-106
 setCacheSize(int), 3-10
 setCDATAFont(Font), 15-10
 setCDATAForeground(Color), 15-11
 setChar, 4-107
 setClientID, 4-51
 setCollIdAttrName(String) -
 oracle.xml.sql.query.OracleXMLQuery.setColl
 idAttrName(java.lang.String), 9-10
 setComment, 2-18, 2-22, 4-84
 setComment(String) -
 oracle.xdb.spi.XDBResource.setComment(java.l
 ang.String), 23-12
 setCommentDataFont(Font), 15-11
 setCommentDataForeground(Color), 15-11
 setCommitBatch(int) -
 oracle.xml.sql.dml.OracleXMLSave.setCommitB
 atch(int), 8-8
 setCompatible, 2-18
 setConsumerName, 2-41
 setContent(Object) -
 oracle.xdb.spi.XDBResource.setContent(java.lan
 g.Object), 23-12
 setContentHandler(ContentHandler) -
 oracle.xml.parser.v2.SAXParser.setContentHan
 dler(org.xml.sax.ContentHandler), 11-99
 setContentType(String) -
 oracle.xdb.spi.XDBResourceContext.setConten
 tType(java.lang.String), 23-19
 setContentType(String) -
 oracle.xdb.spi.XDBResource.setContentType(ja
 va.lang.String), 23-13
 setContext, 1-4
 setCorrelation, 2-44, 2-48
 setCreateDate(Date) -
 oracle.xdb.spi.XDBResourceContext.setCreateD
 ate(java.util.Date), 23-20
 setCreateDate(Date) -
 oracle.xdb.spi.XDBResource.setCreateDate(java.
 util.Date), 23-13
 setDataHeader(Reader, String) -
 oracle.xml.sql.query.OracleXMLQuery.setData
 Header(java.io.Reader, java.lang.String), 9-10
 setDateFormat(String) -
 oracle.xml.sql.dml.OracleXMLSave.setDateFor
 mat(java.lang.String), 8-8
 setDateFormat(String) -
 oracle.xml.sql.query.OracleXMLQuery.setDateF
 ormat(java.lang.String), 9-11
 setDavComment(String) -
 oracle.xdb.spi.XDBResourceContext.setDavCo
 mment(java.lang.String), 23-20

setDebug(boolean), 3-25
 setDebugInfo(int, int, String) -
 oracle.xml.parser.v2.XMLNode.setDebugInfo(int, int, java.lang.String), 11-208
 setDebugMode(boolean), 12-12
 setDebugMode(boolean) -
 oracle.xml.parser.v2.DocumentBuilder.setDebugMode(boolean), 11-33, 11-43
 setDebugMode(boolean) -
 oracle.xml.parser.v2.DOMParser.setDebugMode(boolean), 11-56
 setDelay, 2-48
 setDeliveryMode, 4-155
 setDequeueMode, 2-42
 setDisableMessageID, 4-155
 setDisableMessageTimestamp, 4-156
 setDisplayName(String) -
 oracle.xdb.spi.XDBResourceContext.setDisplayName(java.lang.String), 23-20
 setDisplayName(String) -
 oracle.xdb.spi.XDBResource.setDisplayName(java.lang.String), 23-13
 setDoctype(DTD), 12-12
 setDoctype(DTD) -
 oracle.xml.parser.v2.DefaultXMLDocumentHandler.setDoctype(oracle.xml.parser.v2.DTD), 11-22, 11-26
 setDoctype(DTD) -
 oracle.xml.parser.v2.DocumentBuilder.setDoctype(oracle.xml.parser.v2.DTD), 11-33, 11-43
 setDoctype(DTD) -
 oracle.xml.parser.v2.XMLParser.setDoctype(oracle.xml.parser.v2.DTD), 11-244
 setDoctype(String, String, String) -
 oracle.xml.parser.v2.XMLDocument.setDoctype(java.lang.String, java.lang.String, java.lang.String), 11-150
 setDocument(CGDocument) -
 oracle.xml.classgen.CGNode.setDocument(oracle.xml.classgen.CGDocument), 6-12
 setDocumentLocator(Locator) -
 oracle.xml.parser.schema.XSDValidator.setDocumentLocator(org.xml.sax.Locator), 7-60, 7-62
 setDocumentLocator(Locator) -
 oracle.xml.parser.v2.DocumentBuilder.setDocumentLocator(org.xml.sax.Locator), 11-33, 11-43
 setDouble, 4-107
 setDoubleProperty, 4-22, 4-126
 setDTDHandler(DTDHandler) -
 oracle.xml.parser.v2.SAXParser.setDTDHandler(org.xml.sax.DTDHandler), 11-100
 setEditable(boolean), 15-11
 setElementNode(XMLElement) -
 oracle.xml.classgen.CGNode.setElementNode(oracle.xml.parser.v2.XMLElement), 6-13
 setEmailServerAddr(String), 3-16, 3-21
 setEncoding(String) -
 oracle.xml.parser.v2.PrintDriver.setEncoding(java.lang.String), 11-12
 setEncoding(String) -
 oracle.xml.parser.v2.XMLDeclPI.setEncoding(java.lang.String), 11-123
 setEncoding(String) -
 oracle.xml.parser.v2.XMLDocument.setEncoding(java.lang.String), 11-150
 setEncoding(String) -
 oracle.xml.parser.v2.XMLPrintDriver.setEncoding(java.lang.String), 11-259
 setEncoding(String) -
 oracle.xml.sql.query.OracleXMLQuery.setEncoding(java.lang.String), 9-11
 setEncoding(String, boolean, boolean) -
 oracle.xml.parser.v2.XMLOutputStream.setEncoding(java.lang.String, boolean, boolean), 11-230
 setEntityResolver(EntityResolver) -
 oracle.xml.jaxp.JXDocumentBuilder.setEntityResolver(org.xml.sax.EntityResolver), 11-277
 setEntityResolver(EntityResolver) -
 oracle.xml.parser.v2.XMLParser.setEntityResolver(org.xml.sax.EntityResolver), 11-245
 setEntityResolver(EntityResolver) -
 oracle.xml.parser.v2.XSLProcessor.setEntityResolver(org.xml.sax.EntityResolver), 11-320
 setError(XMLError) -
 oracle.xml.parser.schema.XSDValidator.setError(oracle.xml.parser.v2.XMLError), 7-60, 7-62
 setError(XMLError) -
 oracle.xml.parser.v2.DefaultXMLDocumentHan

dler.setError(oracle.xml.parser.v2.XMLError), 11-26
 setError(XMLError) -
 oracle.xml.parser.v2.XSLTContext.setError(oracle.xml.parser.v2.XMLError), 11-329
 setErrorHandler(ErrorHandler) -
 oracle.xml.jaxp.JXDocumentBuilder.setErrorHandler(org.xml.sax.ErrorHandler), 11-277
 setErrorHandler(ErrorHandler) -
 oracle.xml.parser.v2.XMLError.setErrorHandler(org.xml.sax.ErrorHandler), 11-187
 setErrorHandler(ErrorHandler) -
 oracle.xml.parser.v2.XMLParser.setErrorHandler(org.xml.sax.ErrorHandler), 11-245
 setErrorHandler(ErrorHandler) -
 oracle.xml.parser.v2.XMLTokenizer.setErrorHandler(org.xml.sax.ErrorHandler), 11-271
 setErrorListener(ErrorListener) -
 oracle.xml.jaxp.JXSAXTransformerFactory.setErrorListener(javax.xml.transform.ErrorListener), 11-296
 setErrorListener(ErrorListener) -
 oracle.xml.jaxp.JXTransformer.setErrorListener(javax.xml.transform.ErrorListener), 11-300
 setErrorListener(ErrorListener) -
 oracle.xml.parser.v2.XMLError.setErrorListener(javax.xml.transform.ErrorListener), 11-188
 setErrorStream(OutputStream), 12-12, 12-30
 setErrorStream(OutputStream) -
 oracle.xml.parser.v2.DOMParser.setErrorStream(java.io.OutputStream), 11-56
 setErrorStream(OutputStream) -
 oracle.xml.parser.v2.XMLTokenizer.setErrorStream(java.io.OutputStream), 11-271
 setErrorStream(OutputStream) -
 oracle.xml.parser.v2.XSLProcessor.setErrorStream(java.io.OutputStream), 11-320
 setErrorStream(OutputStream) -
 oracle.xml.util.XMLError.setErrorStream(java.io.OutputStream), 10-15
 setErrorStream(OutputStream, String), 12-12
 setErrorStream(OutputStream, String) -
 oracle.xml.parser.v2.DOMParser.setErrorStream(java.io.OutputStream, java.lang.String), 11-50, 11-56
 setErrorStream(OutputStream, String) -
 oracle.xml.util.XMLError.setErrorStream(java.io.OutputStream, java.lang.String), 10-16
 setErrorStream(PrintWriter), 12-13
 setErrorStream(PrintWriter) -
 oracle.xml.parser.v2.DOMParser.setErrorStream(java.io.PrintWriter), 11-57
 setErrorStream(PrintWriter) -
 oracle.xml.util.XMLError.setErrorStream(java.io.PrintWriter), 10-16
 setErrorTag(String) -
 oracle.xml.sql.OracleXMLSQLException.setErrorTag(java.lang.String), 9-21
 setErrorTag(String) -
 oracle.xml.sql.query.OracleXMLQuery.setErrorTag(java.lang.String), 9-11
 setException(Exception) -
 oracle.xml.sql.query.OracleXMLQuery.setException(java.lang.Exception), 9-12
 setException(Exception) -
 oracle.xml.util.XMLError.setException(java.lang.Exception), 10-16
 setException(Exception) -
 oracle.xml.util.XMLException.setException(java.lang.Exception), 10-24
 setExceptionListener, 4-52
 setExceptionQueue, 2-50
 setExpiration, 2-48
 setFacet(String, String) -
 oracle.xml.parser.schema.XSDSimpleType.setFacet(java.lang.String, java.lang.String), 7-47, 7-51
 setFeature(String) -
 oracle.xml.parser.v2.XMLDOMImplementation.setFeature(java.lang.String), 11-158
 setFeature(String, boolean) -
 oracle.xml.jaxp.JXSAXParserFactory.setFeature(java.lang.String, boolean), 11-287
 setFeature(String, boolean) -
 oracle.xml.parser.v2.SAXParser.setFeature(java.lang.String, boolean), 11-100
 setFloat, 4-108
 setFloatProperty, 4-23, 4-126
 setGenerateComments(boolean) -
 oracle.xml.classgen.DTDClassGenerator.setGen

erateComments(boolean), 6-21
 setGenerateComments(boolean) -
 oracle.xml.classgen.SchemaClassGenerator.setGenerateComments(boolean), 6-27
 setHandler(XMLDocumentHandler) -
 oracle.xml.parser.v2.DefaultXMLDocumentHandler.setHandler(oracle.xml.parser.v2.XMLDocumentHandler), 11-27
 setHostname(String), 13-13
 setIgnoreCase(boolean) -
 oracle.xml.sql.dml.OracleXMLSave.setIgnoreCase(boolean), 8-9
 setInheritedACL(String) -
 oracle.xdb.spi.XDBResource.setInheritedACL(java.lang.String), 23-14
 setInstancename(String), 13-13
 setInt, 4-108
 setIntProperty, 4-24, 4-127
 setJavaPackage(XMLSchema, Vector) -
 oracle.xml.classgen.SchemaClassGenerator.setJavaPackage(oracle.xml.parser.schema.XMLSchema, java.util.Vector), 6-28
 setJMSCorrelationID, 4-127
 setJMSDestination, 4-128
 setJMSExpiration, 4-129
 setJMSMessageID, 4-129
 setJMSPriority, 4-129
 setJMSRedelivered, 4-130
 setJMSReplyTo, 4-24, 4-130
 setJMSTimestamp, 4-131
 setJMSType, 4-24, 4-131
 setKeyColumnList(String[]) -
 oracle.xml.sql.dml.OracleXMLSave.setKeyColumnList(java.lang.String[]), 8-9
 setLanguage(String) -
 oracle.xdb.spi.XDBResourceContext.setLanguage(java.lang.String), 23-21
 setLanguage(String) -
 oracle.xdb.spi.XDBResource.setLanguage(java.lang.String), 23-14
 setLastModDate(Date) -
 oracle.xdb.spi.XDBResourceContext.setLastModDate(java.util.Date), 23-21
 setLastModDate(Date) -
 oracle.xdb.spi.XDBResource.setLastModDate(java.util.Date), 23-14
 setLdapContext(DirContext), 3-17, 3-22
 setLocale(Locale) -
 oracle.xml.parser.v2.XMLDocument.setLocale(java.util.Locale), 11-150
 setLocale(Locale) -
 oracle.xml.parser.v2.XMLParser.setLocale(java.util.Locale), 11-246
 setLocale(Locale) -
 oracle.xml.parser.v2.XSLProcessor.setLocale(java.util.Locale), 11-321
 setLocale(Locale) -
 oracle.xml.util.XMLError.setLocale(java.util.Locale), 10-17
 setLocator(Locator) -
 oracle.xml.util.XMLError.setLocator(org.xml.sax.Locator), 10-17
 setLogStream, 4-143
 setLogStream(OutputStream), 3-25
 setLong, 4-109
 setLongProperty, 4-25, 4-131
 setManualInvalidation(boolean), 3-22
 setMaxOccurs(int) -
 oracle.xml.parser.schema.XSDElement.setMaxOccurs(int), 7-36
 setMaxOccurs(int) -
 oracle.xml.parser.schema.XSDGroup.setMaxOccurs(int), 7-39, 7-40
 setMaxOccurs(int) -
 oracle.xml.parser.schema.XSDSimpleType.setMaxOccurs(int), 7-47, 7-51
 setMaxRetries, 2-21, 4-82
 setMaxRows(int) -
 oracle.xml.sql.query.OracleXMLQuery.setMaxRows(int), 9-12
 setMessageGrouping, 2-17
 setMessageId, 2-43
 setMessageListener, 4-69, 4-204
 setMessageProperty, 2-46
 setMetaHeader(Reader) -
 oracle.xml.sql.query.OracleXMLQuery.setMetaHeader(java.io.Reader), 9-12
 setMinOccurs(int) -
 oracle.xml.parser.schema.XSDElement.setMinOccurs(int), 7-37

setMinOccurs(int) -
 oracle.xml.parser.schema.XSDGroup.setMinOccurs(int), 7-41
 setMinOccurs(int) -
 oracle.xml.parser.schema.XSDSimpleType.setMinOccurs(int), 7-47, 7-52
 setMultiConsumer, 2-17
 setName, 2-13, 4-30
 setNavigationMode, 2-42, 4-70, 4-167, 4-238, 4-241
 setNextException(Exception), 3-27
 setNodeContext(NodeContext) -
 oracle.xml.parser.v2.XMLDocument.setNodeContext(oracle.xml.util.NodeContext), 11-151
 setNodeFactory(NodeFactory), 12-13
 setNodeFactory(NodeFactory) -
 oracle.xml.parser.v2.DocumentBuilder.setNodeFactory(oracle.xml.parser.v2.NodeFactory), 11-33, 11-44
 setNodeFactory(NodeFactory) -
 oracle.xml.parser.v2.DOMParser.setNodeFactory(oracle.xml.parser.v2.NodeFactory), 11-57
 setNodeFlag(int) -
 oracle.xml.parser.v2.XMLNode.setNodeFlag(int), 11-208
 setNodeValue(String) -
 oracle.xml.classgen.CGXSDElement.setNodeValue(java.lang.String), 6-19
 setNodeValue(String) -
 oracle.xml.parser.v2.XMLAttr.setNodeValue(java.lang.String), 11-111
 setNodeValue(String) -
 oracle.xml.parser.v2.XMLEntity.setNodeValue(java.lang.String), 11-180
 setNodeValue(String) -
 oracle.xml.parser.v2.XMLNode.setNodeValue(java.lang.String), 11-209
 setObject, 4-109, 4-141
 setObjectPayload, 2-46
 setObjectProperty, 4-25, 4-132
 setOutputDirectory(String) -
 oracle.xml.classgen.DTDClassGenerator.setOutputDirectory(java.lang.String), 6-22
 setOutputDirectory(String) -
 oracle.xml.classgen.SchemaClassGenerator.setOutputDirectory(java.lang.String), 6-28
 setOutputProperties(Properties) -
 oracle.xml.jaxp.JXTransformer.setOutputProperties(java.util.Properties), 11-301
 setOutputProperty(String, String) -
 oracle.xml.jaxp.JXTransformer.setOutputProperty(java.lang.String, java.lang.String), 11-302
 setOutputStyle(int) -
 oracle.xml.parser.v2.XMLOutputStream.setOutputStyle(int), 11-231
 setOverrideAQResponseFlag(boolean), 3-13
 setOwnerId(long) -
 oracle.xdb.spi.XDBResourceContext.setOwnerId(long), 23-21
 setOwnerId(long) -
 oracle.xdb.spi.XDBResource.setOwnerId(long), 23-14
 setParam(String, String) -
 oracle.xml.parser.v2.XSLStylesheet.setParam(java.lang.String, java.lang.String), 11-326
 setParam(String, String, Object) -
 oracle.xml.parser.v2.XSLProcessor.setParam(java.lang.String, java.lang.String, java.lang.Object), 11-321
 setParameter(String, Object) -
 oracle.xml.jaxp.JXTransformer.setParameter(java.lang.String, java.lang.Object), 11-303
 setParsedDoctype(String, String, String) -
 oracle.xml.parser.v2.XMLDocument.setParsedDoctype(java.lang.String, java.lang.String, java.lang.String), 11-151
 setPassword(String), 13-14
 setPayloadData, 2-53
 setPayloadType, 2-15
 setPCDATAFont(Font), 15-11
 setPCDATAForeground(Color), 15-12
 setPIDataFont(Font), 15-12
 setPIDataForeground(Color), 15-12
 setPINameFont(Font), 15-12
 setPINameForeground(Color), 15-12
 setPingPeriod, 4-53
 setPort(String), 13-14
 setPrefix(String) -
 oracle.xml.parser.v2.XMLNode.setPrefix(java.lang.String), 11-209
 setPrefix(String) -

oracle.xml.parser.v2.XMLNSNode.setPrefix(java.lang.String), 11-227
 setPreserveWhitespace(boolean), 12-13
 setPreserveWhitespace(boolean) - oracle.xml.parser.v2.XMLParser.setPreserveWhitespace(boolean), 11-246
 setPreserveWhitespace(boolean) - oracle.xml.sql.dml.OracleXMLSave.setPreserveWhitespace(boolean), 8-10
 setPrimaryInstance, 2-18
 setPriority, 2-47, 4-156
 setProperty(String, Object) - oracle.xml.jaxp.JXSAXParser.setProperty(java.lang.String, java.lang.Object), 11-284
 setProperty(String, Object) - oracle.xml.parser.v2.SAXParser.setProperty(java.lang.String, java.lang.Object), 11-101
 setProperty(String, Object) - oracle.xml.parser.v2.XMLNode.setProperty(java.lang.String, java.lang.Object), 11-210
 setProtocol, 2-14, 4-31
 setPublicId(String) - oracle.xml.parser.v2.XMLNotation.setPublicId(java.lang.String), 11-216
 setQueueType, 2-20, 4-82
 setRaiseException(boolean) - oracle.xml.sql.query.OracleXMLQuery.setRaiseException(boolean), 9-13
 setRaiseNoRowsException(boolean) - oracle.xml.sql.query.OracleXMLQuery.setRaiseNoRowsException(boolean), 9-13
 setRawPayload, 2-45
 setRecipientList, 2-49
 setResCLOBFileName(String), 13-14
 setResCLOBTableName(String), 13-14
 setResFileName(String), 13-14
 setResHtmlView(boolean), 13-15
 setResSourceEditView(boolean), 13-15
 setResSourceView(boolean), 13-15
 setResTreeView(boolean), 13-15
 setRetentionTime, 2-21, 4-83
 setRetryInterval, 2-21, 4-83
 setRootTag(String) - oracle.xml.parser.v2.DTD.setRootTag(java.lang.String), 11-69
 setRowIdAttrName(String) - oracle.xml.sql.query.OracleXMLQuery.setRowIdAttrName(java.lang.String), 9-13
 setRowIdAttrValue(String) - oracle.xml.sql.query.OracleXMLQuery.setRowIdAttrValue(java.lang.String), 9-14
 setRowIdColumn(String) - oracle.xml.sql.query.OracleXMLQuery.setRowIdColumn(java.lang.String), 9-14
 setRowsetTag(String) - oracle.xml.sql.query.OracleXMLQuery.setRowsetTag(java.lang.String), 9-14
 setRowTag(String) - oracle.xml.sql.dml.OracleXMLSave.setRowTag(java.lang.String), 8-10
 setRowTag(String) - oracle.xml.sql.query.OracleXMLQuery.setRowTag(java.lang.String), 9-14
 setSecondaryInstance, 2-19
 setSelectedNode(Node), 15-13
 setSender, 2-50
 setSenderId, 4-132
 setSequenceDeviation, 2-40
 setSerializationMode(boolean) - oracle.xml.classgen.DTDCClassGenerator.setSerializationMode(boolean), 6-22
 setSessionMaxInactiveTime(int), 3-17, 3-22
 setShort, 4-110
 setShortProperty, 4-26, 4-133
 setSkipRows(int) - oracle.xml.sql.query.OracleXMLQuery.setSkipRows(int), 9-15
 setSortOrder, 2-16
 setSource(XSDNode) - oracle.xml.parser.schema.XSDSimpleType.setSource(oracle.xml.parser.schema.XSDNode), 7-47, 7-52
 setSQLToXMLNameEscaping(boolean) - oracle.xml.sql.dml.OracleXMLSave.setSQLToXMLNameEscaping(boolean), 8-10
 setSQLToXMLNameEscaping(boolean) - oracle.xml.sql.query.OracleXMLQuery.setSQLToXMLNameEscaping(boolean), 9-15
 setStandalone(String) -

- oracle.xml.parser.v2.XMLDeclPI.setStandalone(java.lang.String), 11-124
- setStandalone(String) -
 - oracle.xml.parser.v2.XMLDocument.setStandalone(java.lang.String), 11-151
- setStorageClause, 2-16
- setStream, 2-51
- setString, 4-110
- setStringProperty, 4-27, 4-133
- setStyleSheet(String) -
 - oracle.xml.sql.query.OracleXMLQuery.setStyleSheet(java.lang.String), 9-15
- setStyleSheet(String, String), 3-13, 3-17, 3-22
- setStyleSheet(String, String) -
 - oracle.xml.sql.query.OracleXMLQuery.setStyleSheet(java.lang.String, java.lang.String), 9-15
- setStylesheetHeader(String) -
 - oracle.xml.sql.query.OracleXMLQuery.setStyleSheetHeader(java.lang.String), 9-15
- setStyleSheetProcessingInstr(String), 3-13, 3-18, 3-23
- setSymbolFont(Font), 15-13
- setSymbolForeground(Color), 15-13
- setSystemId(String) -
 - oracle.xml.parser.v2.XMLNotation.setSystemId(java.lang.String), 11-216
- setTagFont(Font), 15-13
- setTagForeground(Color), 15-13
- setText, 4-223
- setTextDecl(String, String) -
 - oracle.xml.parser.v2.DefaultXMLDocumentHandler.setTextDecl(java.lang.String, java.lang.String), 11-27
- setTextDecl(String, String) -
 - oracle.xml.parser.v2.DocumentBuilder.setTextDecl(java.lang.String, java.lang.String), 11-33, 11-44
- setTimeToLive, 4-157
- setToken(int, boolean) -
 - oracle.xml.parser.v2.XMLTokenizer.setToken(int, boolean), 11-271
- setTokenHandler(XMLToken) -
 - oracle.xml.parser.v2.XMLTokenizer.setTokenHandler(oracle.xml.parser.v2.XMLToken), 11-271
- setTraceLevel, 4-143
- setTraceLevel(int), 3-25
- setTransformation, 4-161, 4-167, 4-169, 4-238
- setUpdateColumnList(String[]) -
 - oracle.xml.sql.dml.OracleXMLSave.setUpdateColumnList(java.lang.String[]), 8-11
- setURIResolver(URIResolver) -
 - oracle.xml.jaxp.JXSAXTransformerFactory.setURIResolver(javax.xml.transform.URIResolver), 11-296
- setURIResolver(URIResolver) -
 - oracle.xml.jaxp.JXTransformer.setURIResolver(javax.xml.transform.URIResolver), 11-303
- setUserCallback(AQxmlCallback), 3-18, 3-23
- setUsername(String), 13-15
- setValidationMode(boolean), 12-14
- setValidationMode(boolean) -
 - oracle.xml.classgen.DTDClassGenerator.setValidationMode(boolean), 6-23
- setValidationMode(boolean) -
 - oracle.xml.parser.v2.XMLParser.setValidationMode(boolean), 11-247
- setValidationMode(int) -
 - oracle.xml.parser.v2.XMLParser.setValidationMode(int), 11-247
- setValue(String) -
 - oracle.xml.parser.v2.XMLAttr.setValue(java.lang.String), 11-112
- setVersion(String) -
 - oracle.xml.parser.v2.XMLDeclPI.setVersion(java.lang.String), 11-124
- setVersion(String) -
 - oracle.xml.parser.v2.XMLDocument.setVersion(java.lang.String), 11-151
- setVisibility, 2-40, 2-43
- setWaitTime, 2-43
- setXmlBuffer(String), 13-15
- setXmlCLOBFileName(String), 13-15
- setXmlCLOBTableName(String), 13-16
- setXMLDecl(String, String, String) -
 - oracle.xml.parser.v2.DefaultXMLDocumentHandler.setXMLDecl(java.lang.String, java.lang.String, java.lang.String), 11-22, 11-28
- setXMLDecl(String, String, String) -
 - oracle.xml.parser.v2.DocumentBuilder.setXML

- Decl(java.lang.String, java.lang.String, java.lang.String), 11-33, 11-45
- setXMLDocument(Document), 15-13, 17-4
- setXmlFileName(String), 13-16
- setXMLProperties(XMLProperties) -
 - oracle.xml.parser.schema.XSDValidator.setXMLProperties(oracle.xml.util.XMLProperties), 7-62
- setXMLProperty(String, Object) -
 - oracle.xml.parser.schema.XSDValidator.setXMLProperty(java.lang.String, java.lang.Object), 7-60, 7-63
- setXMLProperty(String, Object) -
 - oracle.xml.parser.v2.XMLParser.setXMLProperty(java.lang.String, java.lang.Object), 11-247
- setXMLSchema(Object) -
 - oracle.xml.parser.v2.DefaultXMLDocumentHandler.setXMLSchema(java.lang.Object), 11-28
- setXMLSchema(Object) -
 - oracle.xml.parser.v2.XMLParser.setXMLSchema(java.lang.Object), 11-248
- setXmlSourceEditView(boolean), 13-16
- setXmlSourceView(boolean), 13-16
- setXmlTreeView(boolean), 13-16
- setXslBuffer(String), 13-16
- setXslCLOBFileName(String), 13-17
- setXslCLOBTableName(String), 13-17
- setXslFileName(String), 13-17
- setXSLOutput(XSLOutput) -
 - oracle.xml.parser.v2.XSLProcessor.setXSLOutput(oracle.xml.parser.v2.XSLOutput), 11-322
- setXslSourceEditView(boolean), 13-17
- setXslSourceView(boolean), 13-17
- setXSLT(Reader, String) -
 - oracle.xml.sql.dml.OracleXMLSave.setXSLT(java.io.Reader, java.lang.String), 8-11
- setXSLT(Reader, String) -
 - oracle.xml.sql.query.OracleXMLQuery.setXSLT(java.io.Reader, java.lang.String), 9-16
- setXSLTParam(String, String) -
 - oracle.xml.sql.dml.OracleXMLSave.setXSLTParam(java.lang.String, java.lang.String), 8-12
- setXSLTParam(String, String) -
 - oracle.xml.sql.query.OracleXMLQuery.setXSLTParam(java.lang.String, java.lang.String), 9-16
- setXslTreeView(boolean), 13-17
- sFacets -
 - oracle.xml.parser.schema.XSDTypeConstants.sFacets, 7-58
- SHORT -
 - oracle.xml.parser.schema.XSDTypeConstants.SHORT, 7-58
- SHOW_WARNINGS -
 - oracle.xml.jaxp.JXDocumentBuilderFactory.SHOW_WARNINGS, 11-279
- SHOW_WARNINGS -
 - oracle.xml.parser.v2.DOMParser.SHOW_WARNINGS, 11-49
- showWarnings(boolean), 12-14, 12-30
- showWarnings(boolean) -
 - oracle.xml.parser.v2.DOMParser.showWarnings(boolean), 11-58
- showWarnings(boolean) -
 - oracle.xml.parser.v2.XSLProcessor.showWarnings(boolean), 11-322
- showWarnings(boolean) -
 - oracle.xml.util.XMLError.showWarnings(boolean), 10-17
- skippedEntity(String) -
 - oracle.xml.parser.v2.DefaultXMLDocumentHandler.skippedEntity(java.lang.String), 11-22, 11-29
- sleep(int), 12-25
- SNOTATION -
 - oracle.xml.parser.schema.XSDTypeConstants.SNOTATION, 7-58
- splitText(int) -
 - oracle.xml.parser.v2.XMLText.splitText(int), 11-265
- STag -
 - oracle.xml.parser.v2.XMLToken.STag, 11-269
- STagName -
 - oracle.xml.parser.v2.XMLToken.STagName, 11-269
- STANDALONE -
 - oracle.xml.parser.v2.XMLParser.STANDALONE, 11-237
- start, 2-27, 4-51, 4-78
- startCDATA() -
 - oracle.xml.parser.v2.DocumentBuilder.startCD

ATA(), 11-45
 startDequeue, 2-27
 startDocument() -
 oracle.xml.parser.v2.DocumentBuilder.startDoc
 ument(), 11-34, 11-45
 startDTD(String, String, String) -
 oracle.xml.parser.v2.DocumentBuilder.startDT
 D(java.lang.String, java.lang.String,
 java.lang.String), 11-46
 startElement(NSName, SAXAttrList) -
 oracle.xml.parser.v2.DefaultXMLDocumentHan
 dler.startElement(oracle.xml.parser.v2.NSName
 , oracle.xml.parser.v2.SAXAttrList), 11-22,
 11-29
 startElement(NSName, SAXAttrList) -
 oracle.xml.parser.v2.DocumentBuilder.startEle
 ment(oracle.xml.parser.v2.NSName,
 oracle.xml.parser.v2.SAXAttrList), 11-46
 startElement(String, String, String, Attributes) -
 oracle.xml.parser.schema.XSDValidator.startEle
 ment(java.lang.String, java.lang.String,
 java.lang.String, org.xml.sax.Attributes), 7-60,
 7-63
 startElement(String, String, String, Attributes) -
 oracle.xml.parser.v2.DefaultXMLDocumentHan
 dler.startElement(java.lang.String,
 java.lang.String, java.lang.String,
 org.xml.sax.Attributes), 11-22, 11-30
 startElement(String, String, String, Attributes) -
 oracle.xml.parser.v2.DocumentBuilder.startEle
 ment(java.lang.String, java.lang.String,
 java.lang.String, org.xml.sax.Attributes), 11-47
 startEnqueue, 2-27
 startEntity(String) -
 oracle.xml.parser.v2.DocumentBuilder.startEnti
 ty(java.lang.String), 11-48
 startPrefixMapping(String, String) -
 oracle.xml.parser.v2.DefaultXMLDocumentHan
 dler.startPrefixMapping(java.lang.String,
 java.lang.String), 11-22, 11-30
 stop, 2-27, 4-52, 4-79
 stopDequeue, 2-28
 stopEnqueue, 2-28
 storeID(String, String) -
 oracle.xml.classgen.CGNode.storeID(java.lang.

String, java.lang.String), 6-13
 storeIDREF(String, String) -
 oracle.xml.classgen.CGNode.storeIDREF(java.la
 ng.String, java.lang.String), 6-13
 STRING -
 oracle.xml.parser.schema.XSDTypeConstants.S
 TRING, 7-58
 sTypes -
 oracle.xml.parser.schema.XSDTypeConstants.s
 Types, 7-58
 supports(String, String) -
 oracle.xml.parser.v2.XMLNode.supports(java.la
 ng.String, java.lang.String), 11-210
 sysId - oracle.xml.util.XMLError.sysId, 10-6

T

TextDecl -
 oracle.xml.parser.v2.XMLToken.TextDecl, 11-
 269
 theTree, 17-3
 TIME -
 oracle.xml.parser.schema.XSDTypeConstants.TI
 ME, 7-58
 TIME_DURATION -
 oracle.xml.parser.schema.XSDTypeConstants.TI
 ME_DURATION, 7-58
 TIME_INSTANT -
 oracle.xml.parser.schema.XSDTypeConstants.TI
 ME_INSTANT, 7-58
 TIME_PERIOD -
 oracle.xml.parser.schema.XSDTypeConstants.TI
 ME_PERIOD, 7-58
 TOKEN -
 oracle.xml.parser.schema.XSDTypeConstants.T
 OKEN, 7-58
 token(int, String) -
 oracle.xml.parser.v2.XMLToken.token(int,
 java.lang.String), 11-269
 tokenize(InputSource) -
 oracle.xml.parser.v2.XMLTokenizer.tokenize(or
 g.xml.sax.InputSource), 11-272
 tokenize(InputStream) -
 oracle.xml.parser.v2.XMLTokenizer.tokenize(ja
 va.io.InputStream), 11-272

tokenize(Reader) -
 oracle.xml.parser.v2.XMLTokenizer.tokenize(java.io.Reader), 11-273

tokenize(String) -
 oracle.xml.parser.v2.XMLTokenizer.tokenize(java.lang.String), 11-273

tokenize(URL) -
 oracle.xml.parser.v2.XMLTokenizer.tokenize(java.net.URL), 11-274

TopicReceiver, 4-243

toString, 4-31, 4-79

toString() -
 oracle.xml.util.XMLException.toString(), 10-24

TOTAL_DIGITS -
 oracle.xml.parser.schema.XSDTypeConstants.TOTAL_DIGITS, 7-58

transform(Source, Result) -
 oracle.xml.jaxp.JXTransformer.transform(javax.xml.transform.Source, javax.xml.transform.Result), 11-304

transformNode(XSLStylesheet) -
 oracle.xml.parser.v2.XMLNode.transformNode(oracle.xml.parser.v2.XSLStylesheet), 11-210

transformToDoc(), 13-17

transformToRes(), 13-18

transformToString(), 13-18

TRAVERSAL_DELETE_EVENT -
 oracle.xml.parser.v2.XMLNode.TRAVERSAL_DELETE_EVENT, 11-190

TRAVERSAL_REPLACE_EVENT -
 oracle.xml.parser.v2.XMLNode.TRAVERSAL_REPLACE_EVENT, 11-191

type -
 oracle.xml.classgen.CGXSDElement.type, 6-16

types - oracle.xml.util.XMLError.types, 10-6

typeToString(int) -
 oracle.xml.parser.v2.AttrDecl.typeToString(int), 11-19

U

unschedulePropagation, 2-31, 4-79

UNSIGNED_BYTE -
 oracle.xml.parser.schema.XSDTypeConstants.UNSIGNED_BYTE, 7-58

UNSIGNED_INT -
 oracle.xml.parser.schema.XSDTypeConstants.UNSIGNED_INT, 7-58

UNSIGNED_LONG -
 oracle.xml.parser.schema.XSDTypeConstants.UNSIGNED_LONG, 7-58

UNSIGNED_SHORT -
 oracle.xml.parser.schema.XSDTypeConstants.UNSIGNED_SHORT, 7-58

unsubscribe, 4-205

updateUI(), 17-4

updateXML(Document) -
 oracle.xml.sql.dml.OracleXMLSave.updateXML(org.w3c.dom.Document), 8-12

URI_REFERENCE -
 oracle.xml.parser.schema.XSDTypeConstants.URI_REFERENCE, 7-58

url, 12-5

USE_DTD_ONLY_FOR_VALIDATION -
 oracle.xml.jaxp.JXDocumentBuilderFactory.USE_DTD_ONLY_FOR_VALIDATION, 11-279

USE_DTD_ONLY_FOR_VALIDATION -
 oracle.xml.parser.v2.XMLParser.USE_DTD_ONLY_FOR_VALIDATION, 11-237

useLowerCaseTagNames() -
 oracle.xml.sql.query.OracleXMLQuery.useLowerCaseTagNames(), 9-17

useNullAttributeIndicator(boolean) -
 oracle.xml.sql.query.OracleXMLQuery.useNullAttributeIndicator(boolean), 9-17

useTypeForCollElemTag(boolean) -
 oracle.xml.sql.query.OracleXMLQuery.useTypeForCollElemTag(boolean), 9-17

useUpperCaseTagNames() -
 oracle.xml.sql.query.OracleXMLQuery.useUpperCaseTagNames(), 9-18

V

validateContent() -
 oracle.xml.classgen.CGNode.validateContent(), 6-14

validateContent(DTD) -
 oracle.xml.parser.v2.XMLElement.validateContent(oracle.xml.parser.v2.DTD), 11-174

validateContent(Element) -
 oracle.xml.parser.v2.ElementDecl.validateContent(org.w3c.dom.Element), 11-72, 11-76
 validateContent(XMLSchema) -
 oracle.xml.parser.v2.XMLElement.validateContent(oracle.xml.parser.schema.XMLSchema), 11-175
 validateContent(XMLSchema, String) -
 oracle.xml.parser.v2.XMLElement.validateContent(oracle.xml.parser.schema.XMLSchema, java.lang.String), 11-175
 validateElementContent(Element) -
 oracle.xml.parser.v2.XMLDocument.validateElementContent(org.w3c.dom.Element), 11-152
 validateFacet(XSDDataValue) -
 oracle.xml.parser.schema.XSDConstrainingFacet.validateFacet(oracle.xml.parser.schema.XSDDataValue), 7-25, 7-27
 validateValue(String) -
 oracle.xml.parser.schema.XSDSimpleType.validateValue(java.lang.String), 7-48, 7-53
 validEntity(String) -
 oracle.xml.classgen.CGNode.validEntity(java.lang.String), 6-14
 validID(String) -
 oracle.xml.classgen.CGNode.validID(java.lang.String), 6-14
 validNMTOKEN(String) -
 oracle.xml.classgen.CGNode.validNMTOKEN(java.lang.String), 6-15
 valueOf(String) -
 oracle.xml.parser.v2.XMLNode.valueOf(java.lang.String), 11-211
 valueOf(String, NSResolver) -
 oracle.xml.parser.v2.XMLNode.valueOf(java.lang.String, NSResolver), 11-211

W

WARNING -
 oracle.xml.parser.v2.XMLParseException.WARNING, 11-233
 WARNING -
 oracle.xml.util.XMLException.WARNING, 10-

18
 WHITESPACE -
 oracle.xml.parser.schema.XSDTypeConstants.WHITESPACE, 7-58
 write(int) -
 oracle.xml.parser.v2.XMLOutputStream.writeInt(), 11-231
 write(OutputStream, String, short) -
 oracle.xdb.dom.XDBNode.write, 22-14
 writeBoolean, 4-41, 4-214
 writeByte, 4-42, 4-215
 writeBytes, 4-42, 4-43, 4-215, 4-216
 writeChar, 4-43, 4-216
 writeChars(String) -
 oracle.xml.parser.v2.XMLOutputStream.writeChars(java.lang.String), 11-231
 writeDouble, 4-43, 4-216
 writeExternal(ObjectOutput) -
 oracle.xml.parser.v2.AttrDecl.writeExternal(java.io.ObjectOutput), 11-19
 writeExternal(ObjectOutput) -
 oracle.xml.parser.v2.DTD.writeExternal(java.io.ObjectOutput), 11-61, 11-69
 writeExternal(ObjectOutput) -
 oracle.xml.parser.v2.ElementDecl.writeExternal(java.io.ObjectOutput), 11-76
 writeExternal(ObjectOutput) -
 oracle.xml.parser.v2.XMLAttr.writeExternal(java.io.ObjectOutput), 11-112
 writeExternal(ObjectOutput) -
 oracle.xml.parser.v2.XMLCDATA.writeExternal(java.io.ObjectOutput), 11-115
 writeExternal(ObjectOutput) -
 oracle.xml.parser.v2.XMLComment.writeExternal(java.io.ObjectOutput), 11-118
 writeExternal(ObjectOutput) -
 oracle.xml.parser.v2.XMLDeclPI.writeExternal(java.io.ObjectOutput), 11-124
 writeExternal(ObjectOutput) -
 oracle.xml.parser.v2.XMLDocument.writeExternal(java.io.ObjectOutput), 11-152
 writeExternal(ObjectOutput) -
 oracle.xml.parser.v2.XMLElement.writeExternal(java.io.ObjectOutput), 11-175
 writeExternal(ObjectOutput) -

oracle.xml.parser.v2.XMLEntityReference.writeExternal(java.io.ObjectOutput), 11-183
 writeExternal(ObjectOutput) -
 oracle.xml.parser.v2.XMLEntity.writeExternal(java.io.ObjectOutput), 11-181
 writeExternal(ObjectOutput) -
 oracle.xml.parser.v2.XMLNode.writeExternal(java.io.ObjectOutput), 11-212
 writeExternal(ObjectOutput) -
 oracle.xml.parser.v2.XMLNotation.writeExternal(java.io.ObjectOutput), 11-216
 writeExternal(ObjectOutput) -
 oracle.xml.parser.v2.XMLPI.writeExternal(java.io.ObjectOutput), 11-252
 writeExternal(ObjectOutput) -
 oracle.xml.parser.v2.XMLText.writeExternal(java.io.ObjectOutput), 11-265
 writeExternal(XMLObjectOutput, CXMLContext) -
 oracle.xml.classgen.CGNode.writeExternal(oracle.xml.io.XMLObjectOutput, CXMLContext), 6-15
 writeExternal(XMLObjectOutput, CXMLContext) -
 oracle.xml.parser.v2.DTD.writeExternal(oracle.xml.io.XMLObjectOutput, CXMLContext), 11-70
 writeExternal(XMLObjectOutput, CXMLContext) -
 oracle.xml.parser.v2.XMLElement.writeExternal(oracle.xml.io.XMLObjectOutput, CXMLContext), 11-176
 writeFloat, 4-44, 4-217
 writeIndent() -
 oracle.xml.parser.v2.XMLOutputStream.writeIndent(), 11-232
 writeInt, 4-44, 4-217
 writeLong, 4-45, 4-218
 writeNewLine() -
 oracle.xml.parser.v2.XMLOutputStream.writeNewLine(), 11-232
 writeObject, 4-45, 4-218
 writeQuotedString(String) -
 oracle.xml.parser.v2.XMLOutputStream.writeQuotedString(java.lang.String), 11-232
 writeShort, 4-46, 4-219
 writeString, 4-219
 writeUTF, 4-46

X

XDBAttribute - oracle.xdb.dom.XDBAttribute, 22-5
 XDBBaseContext -
 oracle.xdb.spi.XDBBaseContext, 23-4
 XDBCData - oracle.xdb.dom.XDBCData, 22-6
 XDBCharData -
 oracle.xdb.dom.XDBCharData, 22-7
 XDBComment -
 oracle.xdb.dom.XDBComment, 22-8
 XDBContext - oracle.xdb.spi.XDBContext, 23-4
 XDBContextFactory -
 oracle.xdb.spi.XDBContextFactory, 23-5
 XDBContextFactory() -
 oracle.xdb.spi.XDBContextFactory.XDBContextFactory(), 23-5
 XDBDocument -
 oracle.xdb.dom.XDBDocument, 22-9
 XDBDocument(Connection, String), 22-10
 XDBDomImplementation -
 oracle.xdb.dom.XDBDomImplementation, 22-11
 XDBDomImplementation() -
 oracle.xdb.dom.XDBDomImplementation.XDBDomImplementation(), 22-11
 XDBElement - oracle.xdb.dom.XDBElement, 22-12
 XDBNamedNodeMap -
 oracle.xdb.dom.XDBNamedNodeMap, 22-13
 XDBNameParser -
 oracle.xdb.spi.XDBNameParser, 23-6
 XDBNamingEnumeration -
 oracle.xdb.spi.XDBNamingEnumeration, 23-7
 XDBNode - oracle.xdb.dom.XDBNode, 22-14
 XDBNodeList -
 oracle.xdb.dom.XDBNodeList, 22-15
 XDBProcInst - oracle.xdb.dom.XDBProcInst, 22-16
 XDBResource - oracle.xdb.spi.XDBResource, 23-8
 XDBResource(Hashtable) -
 oracle.xdb.spi.XDBResource.XDBResource(java.util.Hashtable), 23-9
 XDBResourceContext -
 oracle.xdb.spi.XDBResourceContext, 23-16
 XDBResourceContext(Hashtable) -
 oracle.xdb.spi.XDBResourceContext.XDBResourceContext(java.util.Hashtable), 23-17

XDBText - oracle.xdb.dom.XDBText, 22-17
 XML Class Generator in the XDK, 6-1
 XML Parser, 11-1
 XML SQL Utility for Java, 8-1
 XMLAttr - oracle.xml.parser.v2.XMLAttr, 11-103
 XMLAttr() -
 oracle.xml.parser.v2.XMLAttr.XMLAttr(), 11-103
 XMLAttr(String, String) -
 oracle.xml.parser.v2.XMLAttr.XMLAttr(java.lang.String, java.lang.String), 11-104
 XMLAttr(String, String, String, String) -
 oracle.xml.parser.v2.XMLAttr.XMLAttr(java.lang.String, java.lang.String, java.lang.String, java.lang.String), 11-104
 XMLAttr(String, String, String, String, String) -
 oracle.xml.parser.v2.XMLAttr.XMLAttr(java.lang.String, java.lang.String, java.lang.String, java.lang.String, java.lang.String), 11-104
 XMLCDATA, 11-308
 XMLCDATA -
 oracle.xml.parser.v2.XMLCDATA, 11-113
 XMLCDATA() -
 oracle.xml.parser.v2.XMLCDATA.XMLCDATA(), 11-113
 XMLCDATA(String) -
 oracle.xml.parser.v2.XMLCDATA.XMLCDATA(java.lang.String), 11-114
 XMLComment -
 oracle.xml.parser.v2.XMLComment, 11-116
 XMLComment() -
 oracle.xml.parser.v2.XMLComment.XMLComment(), 11-116
 XMLDecl -
 oracle.xml.parser.v2.XMLToken.XMLDecl, 11-269
 XMLDECL_NODE -
 oracle.xml.parser.v2.XMLNode.XMLDECL_NODE, 11-191
 XMLDeclPI -
 oracle.xml.parser.v2.XMLDeclPI, 11-120
 XMLDeclPI() -
 oracle.xml.parser.v2.XMLDeclPI.XMLDeclPI(), 11-120
 XMLDeclPI(String, String, String, boolean) -
 oracle.xml.parser.v2.XMLDeclPI.XMLDeclPI(java.lang.String, java.lang.String, java.lang.String, boolean), 11-120
 XMLDocument, 11-308
 XMLDocument -
 oracle.xml.parser.v2.XMLDocument, 11-126
 XMLDocument() -
 oracle.xml.parser.v2.XMLDocument.XMLDocument(), 11-126
 XMLDocumentFragment -
 oracle.xml.parser.v2.XMLDocumentFragment, 11-153
 XMLDocumentFragment() -
 oracle.xml.parser.v2.XMLDocumentFragment.XMLDocumentFragment(), 11-153
 XMLDocumentHandler, 11-308
 XMLDOMException -
 oracle.xml.parser.v2.XMLDOMException, 11-155
 XMLDOMException(short) -
 oracle.xml.parser.v2.XMLDOMException.XMLDOMException(short), 11-155
 XMLDOMException(short, String) -
 oracle.xml.parser.v2.XMLDOMException.XMLDOMException(short, java.lang.String), 11-155
 XMLDOMImplementation -
 oracle.xml.parser.v2.XMLDOMImplementation, 11-156
 XMLDOMImplementation() -
 oracle.xml.parser.v2.XMLDOMImplementation.XMLDOMImplementation(), 11-156
 XMLElement -
 oracle.xml.parser.v2.XMLElement, 11-159
 XMLElement() -
 oracle.xml.parser.v2.XMLElement.XMLElement(), 11-159
 XMLElement(String) -
 oracle.xml.parser.v2.XMLElement.XMLElement(java.lang.String), 11-159
 XMLElement(String, String, String, String) -
 oracle.xml.parser.v2.XMLElement.XMLElement(java.lang.String, java.lang.String, java.lang.String, java.lang.String), 11-159
 XMLEntity -
 oracle.xml.parser.v2.XMLEntity, 11-177

XMLEntity() -
 oracle.xml.parser.v2.XMLEntity.XMLEntity(), 11-177

XMLEntityReference -
 oracle.xml.parser.v2.XMLEntityReference, 11-182

XMLEntityReference() -
 oracle.xml.parser.v2.XMLEntityReference.XMLEntityReference(), 11-182

XMLError - oracle.xml.parser.v2.XMLError, 11-185

XMLError - oracle.xml.util.XMLError, 10-6, 10-7

XMLError() -
 oracle.xml.parser.v2.XMLError.XMLError(), 11-185

XMLError() -
 oracle.xml.util.XMLError.XMLError(), 10-6

XMLException -
 oracle.xml.util.XMLException, 10-19

XMLException(String, String, String, int, int, int) -
 oracle.xml.util.XMLException.XMLException(java.lang.String, java.lang.String, java.lang.String, int, int, int), 10-18

XMLException(XMLError, Exception) -
 oracle.xml.util.XMLException.XMLException(oracle.xml.util.XMLError, java.lang.Exception), 10-19

XMLException(XMLError, int) -
 oracle.xml.util.XMLException.XMLException(oracle.xml.util.XMLError, int), 10-19

XMLException(XMLError, int, Exception) -
 oracle.xml.util.XMLException.XMLException(oracle.xml.util.XMLError, int, java.lang.Exception), 10-19

XMLNode -
 oracle.xml.parser.v2.XMLNode, 11-189

XMLNode() -
 oracle.xml.parser.v2.XMLNode.XMLNode(), 11-191

XMLNotation -
 oracle.xml.parser.v2.XMLNotation, 11-213

XMLNotation() -
 oracle.xml.parser.v2.XMLNotation.XMLNotation(), 11-213

XMLNotation(String) -
 oracle.xml.parser.v2.XMLNotation.XMLNotation(java.lang.String), 11-213

XMLNSNode -
 oracle.xml.parser.v2.XMLNSNode, 11-218

XMLNSNode(String) -
 oracle.xml.parser.v2.XMLNSNode.XMLNSNode(java.lang.String), 11-218

XMLOutputStream -
 oracle.xml.parser.v2.XMLOutputStream, 11-228

XMLOutputStream(OutputStream) -
 oracle.xml.parser.v2.XMLOutputStream.XMLOutputStream(java.io.OutputStream), 11-228

XMLOutputStream(PrintWriter) -
 oracle.xml.parser.v2.XMLOutputStream.XMLOutputStream(java.io.PrintWriter), 11-229

XMLParseException -
 oracle.xml.parser.v2.XMLParseException, 11-233

XMLParseException(String, String, String, int, int) -
 oracle.xml.parser.v2.XMLParseException.XMLParseException(java.lang.String, java.lang.String, java.lang.String, int, int, int), 11-233

XMLParser -
 oracle.xml.parser.v2.XMLParser, 11-237

XMLPI - oracle.xml.parser.v2.XMLPI, 11-249

XMLPI() -
 oracle.xml.parser.v2.XMLPI.XMLPI(), 11-249

XMLPI(String, String) -
 oracle.xml.parser.v2.XMLPI.XMLPI(java.lang.String, java.lang.String), 11-250

XMLPrintDriver -
 oracle.xml.parser.v2.XMLPrintDriver, 11-253

XMLPrintDriver(OutputStream) -
 oracle.xml.parser.v2.XMLPrintDriver.XMLPrintDriver(java.io.OutputStream), 11-253

XMLPrintDriver(PrintWriter) -
 oracle.xml.parser.v2.XMLPrintDriver.XMLPrintDriver(java.io.PrintWriter), 11-254

XMLRangeException -
 oracle.xml.parser.v2.XMLRangeException, 11-260

XMLRangeException(short) -
 oracle.xml.parser.v2.XMLRangeException.XMLRangeException(short), 11-260

- XMLSchemaNode() -
 - oracle.xml.parser.schema.XMLSchemaNode.XMLSchemaNode(), 7-8
- XMLSourceView, 15-3
- XMLSourceView(), 15-4
- XMLSourceViewBeanInfo, 15-15
- XMLSourceViewBeanInfo(), 15-15
- xmlStyledDocument, 15-4
- xmlTableExists(Connection, String), 16-9
- XMLText - oracle.xml.parser.v2.XMLText, 11-261
- XMLText() -
 - oracle.xml.parser.v2.XMLText.XMLText(), 11-261
- XMLText(String) -
 - oracle.xml.parser.v2.XMLText.XMLText(java.lang.String), 11-262
- XMLToken -
 - oracle.xml.parser.v2.XMLToken, 11-267
- XMLTokenizer -
 - oracle.xml.parser.v2.XMLTokenizer, 11-270
- XMLTokenizer() -
 - oracle.xml.parser.v2.XMLTokenizer.XMLTokenizer(), 11-270
- XMLTokenizer(XMLToken) -
 - oracle.xml.parser.v2.XMLTokenizer.XMLTokenizer(oracle.xml.parser.v2.XMLToken), 11-270
- XMLTransformPanel, 16-11
- XMLTransformPanel(), 16-11
- XMLTransformPanelBeanInfo, 16-12
- XMLTransformPanelBeanInfo(), 16-12
- XMLTransViewer, 16-13
- XMLTransViewer(), 16-13
- XMLTreeView, 17-3
- XMLTreeView(), 17-4
- XMLTreeViewBeanInfo, 17-6
- XMLTreeViewBeanInfo(), 17-6
- XMLType - oracle.xdb.dom.XMLType, 22-18
- XPathException -
 - oracle.xml.parser.v2.XPathException, 11-308
- XSDNode -
 - oracle.xml.parser.schema.XSDNode, 7-44
- XSDSimpleType -
 - oracle.xml.parser.schema.XSDSimpleType, 7-47
- XSDSimpleType() -
 - oracle.xml.parser.schema.XSDSimpleType.XSDSimpleType(), 7-47
- XSDValidator() -
 - oracle.xml.parser.schema.XSDValidator.XSDValidator(), 7-60
- XSLException, 11-308
- XSLException -
 - oracle.xml.parser.v2.XSLException, 11-308
- XSLException Class, 11-310
- XSLException(String) -
 - oracle.xml.parser.v2.XSLException.XSLException(java.lang.String), 11-310
- XSLExtensionElement -
 - oracle.xml.parser.v2.XSLExtensionElement, 11-311
- XSLExtensionElement() -
 - oracle.xml.parser.v2.XSLExtensionElement.XSLExtensionElement(), 11-311
- XSLProcessor -
 - oracle.xml.parser.v2.XSLProcessor, 11-314
- XSLProcessor() -
 - oracle.xml.parser.v2.XSLProcessor.XSLProcessor(), 11-315
- XSLStylesheet -
 - oracle.xml.parser.v2.XSLStylesheet, 11-323
- XSLTContext -
 - oracle.xml.parser.v2.XSLTContext, 11-326
- XSLTransformer, 12-26
- XSLTransformer(), 12-26
- XSLTransformer(int), 12-26
- XSLTransformerBeanInfo, 12-31
- XSLTransformerBeanInfo(), 12-31
- xsltTransformerError(XSLTransformerEvent), 12-38
- xsltTransformerErrorCalled(XSLTransformerEvent), 12-35
- XSLTransformerErrorEvent, 12-33
- XSLTransformerErrorEvent(Object, Exception), 12-33
- XSLTransformerErrorListener, 12-35
- XSLTransformerEvent, 12-36
- XSLTransformerEvent(Object, int), 12-36
- XSLTransformerListener, 12-38
- xsltTransformerOver(XSLTransformerEvent), 12-38
- xsltTransformerStarted(XSLTransformerEvent), 12-38

XSU, 8-1

Y

YEAR -

oracle.xml.parser.schema.XSDTypeConstants.Y
EAR, 7-58

