Oracle9*i*

Oracle9*i* Supplied Java Packages Reference

Release 1 (9.0.1)

June 2001

Part No. A88898-01

ORACLE

Oracle9*i* Oracle9*i* Supplied Java Packages Reference, Release 1 (9.0.1)

Part No. A88898-01

Primary Author: Jack Melnick

Contributing Authors: Shelley Higgins, Kev MacDowell, Denis Raphaely, Jim Rawles, John Russell, Chitra Sharma

Contributors: Anish Karmarkar, Anjana Manian, Ravi Murthy, Bhagat Nainani, Visar Nimani, Mark Scardina, Rama Vissapragada, Vikram Yavagal

Contents

2 Package oracle.jms

3 Package oracle.ODCI

4 Package oracle.xml.parser.v2

5 Package oracle.AQ.xml

6 Oracle XML SQL Utility (XSU) Java API

7 Package oracle.XML.parser.schema

8 Package oracle.xml.xsql

9 Package oracle.xml.classgen

10 Transviewer Beans

11 Class AppCtxManager

Send Us Your Comments

Oracle9*i* Oracle9*i* Supplied Java Packages Reference, Release 1 (9.0.1)

Part No. A88898-01

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 discusses the following topics:

What This Guide Has to Offer

This reference book presents the Java packages supplied with Oracle9*i*. Interfaces, classes, and exceptions are summarized. Then each is listed in alphabetical order, with its syntax, a member summary, and an inherited-member summary.

This is followed by details about the fields, constructors, and methods.

This book is generated from the Java source code of these packages, using Javadoc.

Organization

See the table of contents for a listing of the packages and classes.

Related Documentation

For more information, Java programmers should see:

- *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 Application Developer's Guide - XML*
- *Oracle9i XML Reference*

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://technet.oracle.com/membership/index.htm`

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://technet.oracle.com/docs/index.htm`

Documentation Accessibility

Oracle's goal is to make our products, services, and supporting documentation accessible to the disabled community with good usability. 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 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/`

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.

What's New in Java Supplied Packages?

The following sections describe the new features in Java Supplied Packages:

- Oracle9i Release 1 (9.0.1) New Features in Java Supplied Packages
- Oracle8i Release 2 (8.1.6) New Features in Java Supplied Packages

Oracle9*i* Release 1 (9.0.1) New Features in Java Supplied Packages

See Also:

- Chapter 5, "Package oracle.AQ.xml"
- Chapter 6, "Oracle XML SQL Utility (XSU) Java API"
- Chapter 7, "Package oracle.XML.parser.schema"
- Chapter 8, "Package oracle.xml.xsql"
- Chapter 9, "Package oracle.xml.classgen"
- Chapter 10, "Transviewer Beans"
- Chapter 11, "Class AppCtxManager"

Oracle8*i* Release 2 (8.1.6) New Features in Java Supplied Packages

See Also:

- Chapter 1, "Package oracle.AQ"
- Chapter 2, "Package oracle.jms"
- Chapter 3, "Package oracle.ODCI"
- Chapter 4, "Package oracle.xml.parser.v2"

1

Package oracle.AQ

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

The following sections describe the common interfaces and classes based on current PL/SQL interfaces.

- The common interfaces are prefixed with "AQ". These interfaces will have different implementations in Oracle8*i* and Oracle Lite.
- This document describes the common interfaces and their corresponding Oracle8*i* implementations, which are prefixed with "AQOracle".

Location of Java AQ Classes

The Java AQ classes are located in `$ORACLE_HOME/rdbms/jlib/aqapi.jar`. These classes can be used with any Oracle8*i* JDBC driver.

If your application uses the OCI8 or thin JDBC driver, for JDK 1.2 you must include `$ORACLE_HOME/rdbms/jlib/aqapi.jar` in the `CLASSPATH`; for JDK 1.1 you must include `$ORACLE_HOME/rdbms/jlib/aqapi11.jar` in the `CLASSPATH`.

If the application is using the Oracle Server driver and accessing the java AQ API from java stored procedures, the Java files are generally automatically pre-loaded in a Java-enabled database. If they are not loaded, you must first load the `aqapi.jar` and `jmscommon.jar` files into the database using the **loadjava** utility.

Oracle9i Application Developer's Guide - Advanced Queuing, Appendix A, contains the following examples:

- Enqueue and Dequeue of Object Type Messages (CustomDatum interface) Using Java

- Enqueue and Dequeue of Object Type Messages (using SQLData interface) Using Java
- Create a Queue Table and Queue Using Java
- Create a Queue and Start Enqueue / Dequeue Using Java
- Create a Multi-Consumer Queue and Add Subscribers Using Java
- Enqueue of RAW Messages using Java
- Dequeue of Messages Using Java
- Dequeue of Messages in Browse Mode Using Java
- Enqueue of Messages with Priority Using Java
- Enqueuing and Dequeuing Object Type Messages That Contain LOB Attributes Using Java

Set up for the test_aqjava class is described in "Setup for oracle.AQ Examples" on page 1-9.

The way to create a multi-consumer queue is described in the "AQSession" on page 1-7.

Package oracle.AQ Description

Class Summary	
Interfaces	
AQSession	Open a session to the queuing system
AQQueueTable	AQ Queue Table interface
AQQueueAdmin	AQ Queue administrative interfaces
AQQueue	AQ Queue operational interfaces
AQMessage	AQ message
AQRawPayload	AQ Raw Payload
AQObjectPayload	AQ Object Payload
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

Class Summary
Exceptions
AQException
AQOracleSQLException

AQDriverManager

The various implementations of the Java AQ API are managed via an AQDriverManager. Both OLite and Oracle8*i* will have an AQDriver which is registered with the AQDriverManager. The driver manager is used to create an AQSession which 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 Oracle8*i* 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_ADMINSTRATOR_ROLE. Users will also need the appropriate system and queue privileges for 8.1 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 Oracle8*i*, 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 1-9.

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 Oracle8i 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 (as described in the first two tables). This will ensure that your applications are portable between Oracle8*i* and Olite AQ implementations.

oracle.AQ classes should not be used unless there is a method in these classes that is not available in the common interfaces. Note that since the AQQueue interface extends AQQueueAdmin, all queue administrative and operation 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	Meaning
`owner`	schema (user) in which to create the queue table
`q_name`	name of the queue table
`property`	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	Meaning
`owner`	schema (user) in which the queue table resides
`name`	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	Meaning
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	Meaning
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	Meaning
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

Setup for oracle.AQ Examples

1. Create an oracle.AQ User

Here an 'aqjava' user is setup 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 we set up the main class from which we will call subsequent examples and handle exceptions.

```
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, an AQ Session is created for the 'aqjava' user as shown in the AQDriverManager section above:

```
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 Oracle8i 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;
  }
```

Example

1. Create a queue table and a queue

Now, with the 'runTest' class, called from the above main class, we will 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	Meaning
`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	Meaning
`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	Meaning
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	Meaning
protocol	Agent protocol

AQQueueTableProperty

This class represents queue table properties.

Constants for Message Grouping

```
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	Meaning
`p_type`	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	Meaning
`p_type`	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	Meaning
s_clauses	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	Meaning
s_order	specifies the columns to be used as the sort_key in ascending order; the string has the format <sort_column1, sort_column2>; the allowed columns name are priority and enq_time.

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	Meaning
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	Meaning
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	Meaning
qt_comment	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	Meaning
qt_compatible	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	Meaning
inst	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	Meaning
inst	secondary instance

Examples:

Set up the test_aqjava class as described in "Setup for oracle.AQ Examples" on page 1-9.

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

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

     /* Create AQQueueTable Property object: */
     qtable_prop = new AQQueueTableProperty("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
 {
     AQQueueTableProperty qtable_prop;

     /* Create AQQueueTable Property object: */
     qtable_prop = new AQQueueTableProperty("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
{
     AQQueueTableProperty qtable_prop;
     qtable_prop = new AQQueueTableProperty(“PERSON”);
     qtable_prop.setComment(“Qtable with Person ADT payload”);
     qtable_prop.setMessageGrouping(TRANSACTIONAL);
}
```

AQQueueProperty

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 AQQueueProperty()
```
This method creates a new `AQQueueProperty` object with default property values.

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	
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
qt_comment	queue comment

Example

Set up the test_aqjava class as described in the Setup for oracle.AQ Examples section on on page 1-9.

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	Meaning
`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	Meaning
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	Meaning
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	Meaning
queue_name	name of the queue to be dropped

Example

Set up the test_aqjava class as described in the Setup for oracle.AQ Examples section on on page 1-9, above.

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	Meaning
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 startEnqueue() 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	Meaning
enqueue	TRUE — disable enqueue on this queue FALSE — leave current setting unchanged

Parameter	Meaning
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	Meaning
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	Meaning
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	Meaning
property	AQQueueProperty 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	Meaning
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	Meaning
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	Meaning
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	Meaning
`privilege`	specifies the privilege to be granted: ENQUEUE, DEQUEUE or ALL
`grantee`	specifies the grantee(s); the grantee(s) can be a user, a role or the PUBLIC roles
`grant_option`	TRUE — the grantee is allowed to use this method to grant access to others FALSE — 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	Meaning
`privilege`	specifies the privilege to be revoked: ENQUEUE, DEQUEUE or ALL
`grantee`	specifies the grantee(s); the grantee(s) can be a user, a role or the PUBLIC 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	Meaning
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	Meaning
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	Meaning
destination	specifies the destination database link. NULL => destination is the local database.
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)

enablePropagationSchedule

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

This method is used to enable a propagation schedule.

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

Examples

Set up the test_aqjava class. For more information, see "Setup for oracle.AQ Examples" on page 1-9

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	Meaning
enq_option	AQEnqueOption 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	Meaning
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 Custom Datum 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.

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	Meaning
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	Meaning
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	Meaning
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.

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	Meaning
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	Meaning
dequeue_mode	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	Meaning
navigation	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	Meaning
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	Meaning
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	Meaning
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	Meaning
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	Meaning
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	Meaning
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	Meaning
`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	Meaning
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	Meaning
delay	the delay represents the number of seconds after which the message is available for dequeuing; with NO_DELAY 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	Meaning
expiration	the duration the message is available for dequeuing; this parameter is an offset from the delay; if NEVER, 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	Meaning
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	Meaning
r_list	vector of AQAgents; 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	Meaning
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	Meaning
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.

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	Meaning
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	Meaning
value	byte array containing the raw payload
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	Meaning
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 Oracle8i AQ driver accepts objects which implement the SQLData or CustomDatum interface inside the payload.

Please refer to the JDBC developer's guide for more information on SQLData 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 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 Oracle8*i* AQ driver, some errors may be raised from the client side and some from the RDBMS. The Oracle8*i* 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`.

2

Package oracle.jms

JMS is a set of interfaces and associated semantics that define how a JMS client accesses the facilities of an enterprise messaging product. Oracle Java Messaging Service (OJMS) provides a Java API for Oracle Advanced Queuing based on the JMS standard. Oracle JMS supports the standard JMS interfaces and has extensions to support the AQ administrative operations and other AQ features that are not a part of the standard.

Standard JMS features include:

- Point-to-point model of communication - using Queues
- Publish-subscribe model of communication - using Topics
- Five types of messages - ObjectMessage, StreamMessage, TextMessage, BytesMessage, MapMessage
- Synchronous and Asynchronous delivery of messages.
- Message selection based on message header fields/properties

Oracle JMS extensions include the following:

- Administrative API - to create Queue Tables, Queues and Topics
- Point-to-multipoint communication - using recipient lists for Topics
- Message propagation between Destinations. Allows the application to define remote subscribers.
- Supports transacted sessions that enable you to perform JMS as well as SQL operations in one atomic transaction.
- Message retention after messages have been dequeued
- Message delay - messages can be made visible after a certain delay

- Exception handling - messages are moved to exception queues if they cannot be processed successfully
- In addition to the standard JMS message types, Oracle supports AdtMessages. These are stored in the database as Oracle Objects and hence the payload of the message can be queried after it is enqueued. Subscriptions can be defined on the contents of these messages as opposed to just the message properties.
- Topic browsing - allows durable subscribers to browse through the messages in a publish-subscribe (topic) destination, and optionally allows these subscribers to purge the browsed messages (so that they are no longer retained by AQ for that subscriber).

Accessing Standard and Oracle JMS

Oracle JMS uses JDBC to connect to the database, hence it applications can run as follows:

- Outside the database using the "OCI8" or "thin" JDBC driver
- Inside Oracle8i JServer using the Oracle Server driver

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

Using OCI8 or Thin JDBC Driver:

To use JMS 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:

For JDK 1.1 include the following:

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

For JDK 1.2 include the following:

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

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_ADMINSTRATOR_ROLE.

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

Package oracle.jms Description

Class Summary	
Interfaces	
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-mulitpoint 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
TopicReceiver	This interface extends MessageConsumer to allow remote subscribers and explicitly specified recipients (in point-to-multipoint communication) to receive messages
Classes	
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
AQjmsBytesMessage	This class implements the BytesMessage interface. A BytesMessage is used to send a message containing a stream of uninterpreted bytes

Class Summary	
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
AQjmsSession	This class implements the javax.jms.Session interface. A Session is a single threaded context for producing a consuming messages

Class Summary	
AQjmsStreamMessage	This class implements the StreamMessage interface. A StreamMessage is used to send a stream of java primitives
AQjmsTextMessage	This class implements the TextMessage interface. A TextMessage is used to send a message containing a java.lang.StringBuffer
AQjmsTopicConnectionFactory	This class implements the TopicConnectionFactory interface. A TopicConnectionFactory is used to create TopicConnections
Exceptions	
AQjmsException	This exception extends JMSException - adds Oracle error codes. This is the root of all JMS exceptions
AQjmsInvalidDestinationException	This exception extends InvalidDestinationException. It is thrown when a Destination is not valid
AQjmsInvalidSelectorException	This exception extends InvalidSelectorException. It is thrown when the specified MessageSelector is not valid
AQjmsMessageEOFException	This exception extends MessageEOFException. It is thrown when an unexpected end of stream has been reached when a StreamMessage or BytesMessage is being read
AQjmsMessageFormatException	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
AQjmsMessageNotReadableException	This exception extends MessageNotReadableException. It is thrown when a client attempts to read a write-only message
AQjmsMessageNotWriteableException	This exception extends MessageNotWriteableException. 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 - this is an AQ extension to JMS.

Member Summary

Methods

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

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

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

`JMSException` - 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

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.
`getObjectProperty(String)`	Return the Java object property value with the given name.
`getPropertyNames()`	Return an Enumeration of all the property names.

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

Inherited Member Summary

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

`JMSException` - 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

`JMSException` - 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

javax.jms.Message.getBooleanProperty(java.lang.String) in interface javax.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

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

`MessageFormatException` - if this type conversion is invalid.

getByteProperty(String)

`public byte getByteProperty(java.lang.String name)`
Return the byte property value with the given name.

Specified By

javax.jms.Message.getByteProperty(java.lang.String) in interface javax.jms.Message

Overrides

getByteProperty(String) in class AQjmsMessage

Parameters

`name` - the name of the byte property

Returns

the byte 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.

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

`JMSException` - 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

`JMSException` - 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

`JMSException` - 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

`JMSException` - 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

`JMSException` - 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

`JMSException` - 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

`JMSException` - 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

`JMSException` - 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

`JMSException` - 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

`JMSException` - 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

`JMSException` - 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

`JMSException` - 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

`JMSException` - 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

`JMSException` - 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

`JMSException` - 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

`JMSException` - 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

`JMSException` - 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

`JMSException` - 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

Fields

Constructors

`AQjmsAgent(String, String)`	Constructor
`AQjmsAgent(String, String, int)`	Constructor

Methods

`getAddress()`	Get the address of the agent
`getName()`	Get the name of the agent
`getProtocol()`	Get the protocol of the agent
`setAddress(String)`	Set the address of the agent
`setName(String)`	Set the name of the agent
`setProtocol(int)`	Set the protocol of the agent
`toString()`	Convert the agent to its string representation which is of the form: "[AQjmsAgent] \n name: NAME \n address: ADDRESS \n protocol: PROTOCOL"

Fields

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

Methods

`clearBody()`	Clear out the message body.
`clearProperties()`	Clear a message's properties.
`readBoolean()`	Read a `boolean` from the stream message.
`readByte()`	Read a signed 8-bit value from the stream message.
`readBytes(byte[])`	Read a byte array from the stream message.
`readBytes(byte[], int)`	Read a portion of the bytes message.
`readChar()`	Read a Unicode character value from the stream message.
`readDouble()`	Read a `double` from the stream message.
`readFloat()`	Read a float from the stream message.
`readInt()`	Read a signed 32-bit integer from the stream message.
`readLong()`	Read a signed 64-bit integer from the stream message.
`readShort()`	Put the message in read-only mode, and reposition the stream of bytes to the beginning.

Member Summary	
readUnsignedByte()	Read an unsigned 8-bit number from the stream message.
readUnsignedShort()	Read an unsigned 16-bit number from the stream message.
readUTF()	Read in a string that has been encoded using a modified UTF-8 format from the stream message.
reset()	Put the message in read-only mode, and reposition the stream of bytes to the beginning.
writeBoolean(boolean)	Write a `boolean` to the stream message as a 1-byte value.
writeByte(byte)	Write out a `byte` to the stream message as a 1-byte value.
writeBytes(byte[])	Write a byte array to the stream message.
writeBytes(byte, int, int)	Write a portion of a byte array to the stream message.
writeChar(char)	Write a `char` to the stream message as a 2-byte value, high byte first.
writeDouble(double)	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.
writeFloat(float)	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.
writeInt(int)	Write an `int` to the stream message as four bytes, high byte first.
writeLong(long)	Write a `long` to the stream message as eight bytes, high byte first.
writeObject(Object)	Write a Java object to the stream message.
writeShort(short)	Write a `short` to the stream message as two bytes, high byte first.
writeUTF(String)	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

Inherited Member Summary

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

`JMSException` - 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

`JMSException` - 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.

`JMSException` - 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

`JMSException` - 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

`JMSException` - 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

`JMSException` - 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

`JMSException` - 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

`JMSException` - 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

`JMSException` - 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

`JMSException` - 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

`JMSException` - 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 MessageNotWriteableException - if message in write-only mode. JMSException - 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

`JMSException` - 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

`JMSException` - 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.

`JMSException` - 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.

`JMSException` - 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.

`JMSException` - 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.

`JMSException` - 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.

`JMSException` - 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.

`JMSException` - 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.

`JMSException` - 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.

`JMSException` - 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.

`JMSException` - 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.

`JMSException` - 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.

`JMSException` - 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.

`JMSException` - 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

Methods

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.
createQueueSession(boolean, int)	create a queue session
createTopicSession(boolean, int)	Create a TopicSession
getClientID()	Get the client identifier for this connection.
getCurrentJmsSession()	gets the current session
getMetaData()	Get the meta data for this connection.
setClientID(String)	Set the client identifier for this connection.
start()	Start (or restart) a Connection's delivery of incoming messages.
stop()	Used to temporarily stop a Connection's delivery of incoming messages.
setExceptionListener(ExceptionListener)	Set the exception listener for this connection.

Member Summary	
`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 java.lang.Object
`clone, equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

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

`JMSException` - 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 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

`JMSException` - 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

`JMSException` - 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

`JMSException` - 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

`JMSException` - 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

`JMSException` - 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

`JMSException` - if JMS implementation fails to stop the message delivery due to some internal error.

See Also

`javax.jms.Connection.start()`

setExceptionListener(ExceptionListener)

```
public void setExceptionListener(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 ExceptionListener if one has been registered. It does this by calling the listener’s onException() method passing it a JMSException 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

`JMSException` - general exception if JMS implementation fails to set the Exception listener 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

`JMSException` - general exception if JMS implementation fails to get the Exception listener 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

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

`JMSException` - 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

`JMSException` - 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

`JMSException` - 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

`JMSException` - 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

Member Summary

Fields

EXCEPTION

NONE

NORMAL

STATE_EXPIRED

STATE_PROCESSED

STATE_READY

STATE_WAITING

TRANSACTIONAL

WAIT_FOREVER

WAIT_NONE

Constructors

AQjmsConstants()

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()
```

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

Methods

`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.
`getMessageListener()`	Get the message consumer's MessageListener.
`getMessageSelector()`	Get the message consumer's message selector expression.
`getNavigationMode()`	Get the navigation mode for the consumer
`getNoLocal()`	Get the NoLocal attribute for this TopicSubscriber.
`getQueue()`	Get the queue associated with this queue receiver.
`getTopic()`	Get the topic associated with this subscriber.
`receive()`	Receive the next message produced for this message consumer.
`receive(long)`	Receive the next message that arrives within the specified timeout interval.
`receiveNoData()`	Consume the message without returning it to the user.
`receiveNoData()`	Consume the message without returning it to the user.

Member Summary	
receiveNoWait()	Receive the next message if one is immediately available.
setMessageListener(MessageListener)	Set the message consumer's MessageListener.
setNavigationMode(int)	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

`JMSException` - 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

`JMSException` - 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

`JMSException` - 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

`JMSException` - 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

`JMSException` - 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

`JMSException` - 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

`JMSException` - 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

`JMSException` - 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

Methods

`alter(Session, AQjmsDestinationProperty)`	alter the properties of the queue/topic
`alterPropagationSchedule(Session, String, Double, String, Double)`	alter propagation schedule between the topic and the destination database
`disablePropagationSchedule(Session, String)`	disable propagation schedule
`drop(Session)`	drop the queue/topic
`enablePropagationSchedule(Session, String)`	enable propagation schedule
`getCompleteName()`	Get the complete name of the queue/topic, of the form, [schema].name
`getCompleteTableName()`	Get the complete name of the queue table of the queue/topic of the form, [schema].name
`getQueueName()`	Get the name of the queue
`getQueueOwner()`	Get the owner of the queue
`getTopicName()`	Get the name of the Topic

Member Summary	
getTopicOwner()	Get the schema of the topic
grantQueuePrivilege(Session, String, String, boolean)	Grant enqueue or dequeue privilege on the queue to a database user
grantTopicPrivilege(Session, String, String, boolean)	Grant a topic privilege
revokeQueuePrivilege(Session, String, String)	Revoke a queue privilege
revokeTopicPrivilege(Session, String, String)	Revoke a topic privilege
schedulePropagation(Session, String, Date, Double, String, Double)	Schedule propagation from the topic for the given destination database
start(Session, boolean, boolean)	start the queue/topic for enqueue or dequeue or both
stop(Session, boolean, boolean, boolean)	stop the queue/topic for enqueue or dequeue or both
toString()	Get the queue/topic as a string, of the form [schema].name
unschedulePropagation(Session, String)	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

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

`JMSException` - 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

`JMSException` - 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

`JMSException` - 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

`JMSException` - 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

`JMSException` - 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

`JMSException` - 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

`JMSException` - 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

`JMSException` - 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

`JMSException` - 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

`JMSException` - 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

`JMSException` - 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

`JMSException` - 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

`JMSException` - if propagation could not be unscheduled

AQjmsDestinationProperty

```
public class AQjmsDestinationProperty
oracle.jms.AQjmsDestinationProperty
```

This class defines Destination properties

Member Summary

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
qt_comment	queue comment

AQjmsException

Syntax

```
public class AQjmsException extends javax.jms.JMSException

java.lang.Object
  |
  +--java.lang.Throwable
        |
        +--java.lang.Exception
              |
              +--javax.jms.JMSException
                    |
                    +--oracle.jms.AQjmsException
```

All Implemented Interfaces

`java.io.Serializable`

Description

This exception extends JMSException - adds Oracle error codes. This is the root of all JMS exceptions

Member Summary

Methods

`getErrorNumber()`	Get the Oracle Error code for the exception

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`

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

Methods

`getQueueConnectionFactory(String, Properties)`	get a Queue Connection Factory
`getQueueConnectionFactory(String, String, int, String)`	get a Queue Connection Factory
`getTopicConnectionFactory(String, Properties)`	get a Topic Connection Factory
`getTopicConnectionFactory(String, String, int, String)`	get a Topic Connection Factory
`registerConnectionFactory (java.sql.Connection,String, String, String, int, String, String)`	Register a Queue or Topic Connection Factory in LDAP through the database
`registerConnectionFactory (java.sql.Connection,String, String, java.util.Properties, String)`	Register a Queue or Topic Connection Factory in LDAP through the database
`registerConnectionFactory (java.util.Hashtable,String, String, String, int, String, String)`	Register a Queue or Topic Connection Factory to LDAP

Member Summary

`registerConnectionFactory (java.util.Hashtable,String, String, java.util.Properties, String)`	Register a Queue or Topic Connection Factory in LDAP

Inherited Member Summary

Methods inherited from class java.lang.Object

`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

`JMSException` - 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

`JMSException` - 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

`JMSException` - 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

`JMSException` - 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 JMSException
```

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

JMSException - 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 JMSException
```

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

`JMSException` - 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 JMSException
```

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

`JMSException` - 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 JMSException
```

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

`JMSException` - 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

Methods

`clearBody()`	Clear out the message body.
`clearProperties()`	Clear a message's properties.
`getBoolean(String)`	Return the boolean value with the given name.
`getByte(String)`	Return the byte value with the given name.
`getBytes(String)`	Return the byte array value with the given name.
`getChar(String)`	Return the Unicode character value with the given name.
`getDouble(String)`	Return the double value with the given name.
`getFloat(String)`	Return the float value with the given name.
`getInt(String)`	Return the integer value with the given name.
`getLong(String)`	Return the long value with the given name.
`getMapNames()`	Return an Enumeration of all the Map message's names.
`getObject(String)`	Return the Java object value with the given name.

Member Summary	
getShort(String)	Return the short value with the given name.
getString(String)	Set a String value with the given name, into the Map.
itemExists(String)	Check if an item exists in this MapMessage.
setBoolean(String, boolean)	Set a boolean value with the given name, into the Map.
setByte(String, byte)	Set a byte value with the given name, into the Map.
setBytes(String, byte[])	Set a byte array value with the given name, into the Map.
setBytes(String, byte[], int, int)	Set a portion of the byte array value with the given name, into the Map.
setChar(String, char)	Set a Unicode character value with the given name, into the Map.
setDouble(String, double)	Set a double value with the given name, into the Map.
setFloat(String, float)	Set a float value with the given name, into the Map.
setInt(String, int)	Set an integer value with the given name, into the Map.
setLong(String, long)	Set a long value with the given name, into the Map.
setObject(String, Object)	Set a Java object value with the given name, into the Map.
setShort(String, short)	Set a short value with the given name, into the Map.
setString(String, String)	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

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

`JMSException` - 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

`JMSException` - 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

JMSException - if JMS fails to read message due to some internal JMS error.

MessageFormatException - if this type conversion is invalid.

getByte(String)

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

Return the byte value with the given name.

Specified By

javax.jms.MapMessage.getByte(java.lang.String) in interface javax.jms.MapMessage

Parameters

name - the name of the byte

Returns

the byte value with the given name.

Throws

JMSException - 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

`JMSException` - 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

`JMSException` - 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
`JMSException` - 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
`JMSException` - 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

`JMSException` - 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

`JMSException` - 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

`JMSException` - 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

`JMSException` - 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

`JMSException` - 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

`JMSException` - 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

`JMSException` - 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

`JMSException` - 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

`JMSException` - 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

`JMSException` - 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

`JMSException` - 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

`JMSException` - 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

`JMSException` - 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

`JMSException` - 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

`JMSException` - 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

`JMSException` - 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

`JMSException` - 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

`JMSException` - 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

`JMSException` - 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

Methods

clearBody()	Clear out the message body.
clearProperties()	Clear a message's properties.
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.
getJMSCorrelationID()	Get the correlation ID for the message.
getJMSCorrelationIDAsBytes()	Get the correlation ID as an array of bytes for the message.
getJMSDeliveryMode()	Get the delivery mode for this message.

Member Summary	
getJMSDestination()	Get the destination for this message.
getJMSExpiration()	Get the message's expiration value.
getJMSMessageID()	Get the message ID.
getJMSMessageIDAsBytes()	Get the message ID.
getJMSPriority()	Get the message priority.
getJMSRedelivered()	Get an indication of whether this message is being redelivered.
getJMSReplyTo()	Get the replyTo field for this message
getJMSTimestamp()	Get the message timestamp.
getJMSType()	Get the message type.
getLongProperty(String)	Return the long property value with the given name.
getObjectProperty(String)	Return the Java object property value with the given name.
getPropertyNames()	Return an Enumeration of all the property names.
getSenderID()	Get the message's senderID.
getShortProperty(String)	Return the short property value with the given name.
getStringProperty(String)	Return the String property value with the given name.
propertyExists(String)	Check if a property value exists.
setBooleanProperty(String, boolean)	Set a boolean property value with the given name, into the Message.
setByteProperty(String, byte)	Set a byte property value with the given name, into the Message.
setDoubleProperty(String, double)	Set a double property value with the given name, into the Message.
setFloatProperty(String, float)	Set a float property value with the given name, into the Message.
setIntProperty(String, int)	Set an integer property value with the given name, into the Message.
setJMSCorrelationID(String)	Set the correlation ID for the message.
setJMSCorrelationIDAsBytes(byte[])	Set the correlation ID as an array of bytes for the message.

Member Summary	
setJMSDestination(Destination)	Set the destination for this message.
setJMSExpiration(long)	Set the message's expiration value Providers set this field when a message is sent.
setJMSMessageID(String)	Set the message ID.
setJMSPriority(int)	Set the priority for this message.
setJMSRedelivered(boolean)	Set to indicate whether this message is being redelivered.
setJMSReplyTo(Destination)	Set where a reply to this message should be sent.
setJMSTimestamp(long)	Set the message timestamp.
setJMSType(String)	Set the message type.
setLongProperty(String, long)	Set a long property value with the given name, into the Message.
setObjectProperty(String, Object)	Set a Java object property value with the given name, into the Message.
setSenderID(AQjmsAgent)	Set the message's senderID.
setShortProperty(String, short)	Set a short property value with the given name, into the Message.
setStringProperty(String, String)	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

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

`JMSException` - 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

`JMSException` - 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

JMSException - if JMS fails to get Property

MessageFormatException - if this type conversion is invalid.

getByteProperty(String)

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

Return the byte property value with the given name.

Specified By

javax.jms.Message.getByteProperty(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

JMSException - 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

`JMSException` - 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

`JMSException` - 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

`JMSException` - 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

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

getJMSCorrelationIDAsBytes()

```
public byte[] getJMSCorrelationIDAsBytes()
```

Get the correlation ID as an array of bytes for the message.

Specified By

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

Returns

the correlation ID of a message as an array of bytes.

Throws

`JMSException` - 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. In the current version this will always return DeliveryMode.PERSISTENT

Throws

`JMSException` - 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 returns it 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

`JMSException` - 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

`JMSException` - 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

`JMSException` - 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

`JMSException` - 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

`JMSException` - 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

`JMSException` - 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

`JMSException` - 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

`JMSException` - 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

`JMSException` - 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

`JMSException` - 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

`JMSException` - 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

`JMSException` - 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

`JMSException` - 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

`JMSException` - 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

`JMSException` - 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

`JMSException` - 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

`JMSException` - 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

`JMSException` - 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

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

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

`JMSException` - 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

`JMSException` - 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

`JMSException` - 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

`JMSException` - 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

`JMSException` - 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

`JMSException` - 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

`JMSException` - 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

`JMSException` - 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

`JMSException` - 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

`JMSException` - 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

`JMSException` - 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

`JMSException` - 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

`JMSException` - 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.JMSException
                    |
                    +--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.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`

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.JMSException
                    |
                    +--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.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`

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

Methods

`clearBody()`	Clear out the message body.
`clearProperties()`	Clear a message's properties.
`getObject()`	Get the serializable object containing this message's data.
`setObject(Serializable)`	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

`JMSException` - 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

`JMSException` - 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

`JMSException` - 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

`JMSException` - 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

Methods

`getLogStream()`	Get log stream
`setLogStream(OutputStream)`	Set log stream
`setTraceLevel(int)`	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

Methods

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.
getDeliveryMode()	Get the producer's default delivery mode.
getDisableMessageID()	Get an indication of whether message IDs are disabled.
getDisableMessageTimestamp()	Get an indication of whether message timestamps are disabled.
getPriority()	Get the producer's default priority.
getQueue()	Get the queue associated with this queue sender.
getTimeToLive()	Get the default length of time in milliseconds from its dispatch time that a produced message should be retained by the message system.
getTopic()	Get the topic associated with this publisher.
publish(Message)	Publish a Message to the topic
publish(Message, AQjmsAgent[])	Publish a Message to a specific list of recipients

Member Summary	
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(Message, int, int, long)	Publish a Message to the topic specifying delivery mode, priority and time to live to the topic.
publish(Topic, Message)	Publish a Message to a topic for an unidentified message producer.
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
publish(Topic, Message, int, int, long)	Publish a Message to a topic for an unidentified message producer, specifying delivery mode, priority and time to live.
send(Message)	Send a message
send(Message, int, int, long)	Send a message.
send(Queue, Message)	Send a message.
send(Queue, Message, int, int, long)	Send a message.
setDeliveryMode(int)	Set the producer's default delivery mode.
setDisableMessageID(boolean)	Set whether message IDs are disabled.
setDisableMessageTimestamp(boolean)	Set whether message timestamps are disabled.
setPriority(int)	Set the producer's default priority.
setTimeToLive(int)	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

```
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

`JMSException` - 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

`JMSException` - 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

`JMSException` - 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

`JMSException` - 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

`JMSException` - 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

`JMSException` - 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

`JMSException` - 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

`JMSException` - 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

`JMSException` - 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

`JMSException` - 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

`JMSException` - 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

`JMSException` - 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

`JMSException` - 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

`JMSException` - 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

`JMSException` - 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

`JMSException` - 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

`JMSException` - 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

`JMSException` - 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

`JMSException` - 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 deliveryMode, 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

`JMSException` - 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

`JMSException` - 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

`JMSException` - 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

`JMSException` - 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

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

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 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

`JMSException` - 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

`JMSException` - 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

`JMSException` - 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

`JMSException` - 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`

Description

This class implements the QueueConnectionFactory interface. A QueueConnectionFactory is used to create QueueConnections

Member Summary

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

`JMSException` - 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

`JMSException` - 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

Methods

getNavigationMode()	get the navigation mode used for receiving messages
getTransformation()	get the transformation for this receiver
receiveNoData()	Consume the message without returning it to the user.
receiveNoData(long)	Consume the message without returning it to the user.
setNavigationMode(int)	set the navigation mode used for receiving messages
setTransformation(String)	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

`JMSException` - if there was an error in getting the navigation mode

getTransformation()

```
public String getTransformation()
```
Get the transformation for this receiver

Returns

the transformation

Throws

`JMSException` - 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

`JMSException` - 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

`JMSException` - 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

`JMSException` - 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

`JMSException` - 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

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

`JMSException` - 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

`JMSException` - 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

Methods

`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.
`commit()`	Commit all messages done in this transaction and releases any locks currently held.
`createAdtMessage()`	Create an AdtMessage.
`createAdtMessage(CustomDatum)`	Create an initialized AdtMessage.
`createBrowser(Queue)`	Create a QueueBrowser to peek at the messages on the specified queue.
`createBrowser(Queue, CustomDatumFactory)`	Create a QueueBrowser to peek at the messages on the specified queue containing ADT messages.
`createBrowser(Queue, String)`	Create a QueueBrowser to peek at the messages on the specified queue.
`createBrowser(Queue, String, boolean)`	Create a QueueBrowser to peek at the messages on the specified queue.

Member Summary

`createBrowser(Queue, String, CustomDatumFactory)`	Create a QueueBrowser to peek at the messages on the specified queue containing ADT messages.
`createBrowser(Queue, String, CustomDatumFactory, boolean)`	Create a QueueBrowser to peek at the messages on the specified queue containing ADT messages.
`createBytesMessage()`	Create a BytesMessage.
`createDurableSubscriber(Topic, String)`	Create a durable Subscriber to the specified topic.
`createDurableSubscriber(Topic, String, CustomDatumFactory)`	Create a durable Subscriber to the specified topic.
`createDurableSubscriber(Topic, String, String, boolean)`	Create a durable Subscriber to the specified topic.
`createDurableSubscriber(Topic, String, String, boolean, String)`	Create a durable Subscriber to the specified topic. Specify transformation for the subscriber
`createDurableSubscriber(Topic, String, String, boolean, CustomDatumFactory, String)`	Create a durable Subscriber to the specified Oracle Object (ADT) topic.
`createDurableSubscriber(Topic, String, String, boolean, CustomDatumFactory, String)`	Create a durable Subscriber to the specified Oracle Object (ADT) topic. Specify transformation for the subscriber
`createMapMessage()`	Create a MapMessage.
`createObjectMessage()`	Create an ObjectMessage.
`createObjectMessage(Serializable)`	Create an initialized ObjectMessage.
`createPublisher(Topic)`	Create a Publisher for the specified topic.
`createQueue(AQQueueTable, String, AQjmsDestinationProperty)`	Create a queue.
`createQueueTable(String, String, AQQueueTableProperty)`	Create a Queue Table.
`createReceiver(Queue)`	Create a QueueReceiver to receive messages from the specified queue.
`createReceiver(Queue, CustomDatumFactory)`	Create a QueueReceiver to receive messages from the specified queue containing ADT messages.

Member Summary	
createReceiver(Queue, String)	Create a QueueReceiver to receive messages from the specified queue.
createReceiver(Queue, String, CustomDatumFactory)	Create a QueueReceiver to receive messages from the specified queue containing ADT messages.
createRemoteSubscriber(Topic, AQjmsAgent, String)	Create a remote subscriber for a topic.
createRemoteSubscriber(Topic, AQjmsAgent, String, String)	Create a remote subscriber for a topic. Specify transformation for the remote subscriber
createRemoteSubscriber(Topic, AQjmsAgent, String, CustomDatumFactory)	Create a remote subscriber for a topic.
createRemoteSubscriber(Topic, AQjmsAgent, String, CustomDatumFactory, String)	Create a remote subscriber for a Oracle Object (ADT) topic. Specify transformation for the remote subscriber
createRemoteSubscriber(Topic, AQjmsAgent, String, CustomDatumFactory, String)	Create a QueueSender to send messages to the specified queue.
createStreamMessage()	Create a StreamMessage.
createSubscriber(Topic)	Create a non-durable Subscriber to the specified topic.
createSubscriber(Topic, String, boolean)	Create a non-durable Subscriber to the specified topic.
createTextMessage()	Create a TextMessage.
createTextMessage(StringBuffer)	Create an initialized TextMessage.
createTopic(AQQueueTable, String, AQjmsDestinationProperty)	Create a topic
createTopicReceiver(Topic, String, String)	Create a TopicReceiver to receive messages from the specified topic.
createTopicReceiver(Topic, String, String, CustomDatumFactory)	
getDBConnection()	
getJmsConnection()	
getMessageListener()	Return the session's distinguished message listener.

Member Summary	
getQueue(String, String)	Get an existing queue.
getQueueTable(String, String)	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.
getTopic(String, String)	Get an existing topic.
getTransacted()	Checks if the session in transacted mode?
grantSystemPrivilege(String, String, boolean)	Grant AQ system privileges to users/roles.
revokeSystemPrivilege(String, String)	Revoke AQ system privilege from user/roles
rollback()	Rollback any messages done in this transaction and releases any locks currently held.
run()	
setMessageListener(MessageListener)	Set the session's distinguished message listener.
unsubscribe(Topic, AQjmsAgent)	Unsubscribe a remote durable subscription that has been created by a client on the specified topic
unsubscribe(Topic, String)	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 an Java object that maps to a Oracle SQL ADT. This object must support the OracleCustomDatum interface.

Throws

`JMSException` - 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

`JMSException` - 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

`JMSException` - 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

`JMSException` - 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

`JMSException` - 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

`JMSException` - 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 = '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

`locked` - if true then messages are locked as they are browsed (similar to a SELECT for UPDATE)

Throws

`JMSException` - 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

`JMSException` - 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 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) in interface javax.jms.TopicSession

Parameters

`topic` - the topic to subscribe to

`name` - the name used to identify this subscription.

Throws

`JMSException` - 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

`JMSException` - 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 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

`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, 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 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 subsccribers 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.

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

`JMSException` - 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

`JMSException` - 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

`JMSException` - 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

`JMSException` - if a session fails to create a publisher due to some JMS error.

`InvalidDestinationException` - if invalid Topic specified.

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

`JMSException` - 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 queuetable will be used to hold queues, then the queuetable must not be multiconsumer enabled (default). If the queue table will be used to hold topics the queuetable must be multiconsumer enabled

Throws

`JMSException` - 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

`JMSException` - 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

`JMSException` - 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

`JMSException` - 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.

`tranformation` - 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.

`tranformation` - 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

`JMSException` - 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. This method is not supported in the current release

Specified By

javax.jms.TopicSession.createSubscriber(javax.jms.Topic) in interface javax.jms.TopicSession

Throws

`JMSException` - NOT_SUPPORTED

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. This method is not supported in the current release

Specified By

javax.jms.TopicSession.createSubscriber(javax.jms.Topic, java.lang.String, boolean) in interface javax.jms.TopicSession

Throws

`JMSException` - NOT_SUPPORTED

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

`JMSException` - 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

`JMSException` - if some error occurs during message creation

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

`JMSException` - 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

`JMSException` - 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

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

`JMSException` - 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

`JMSException` - 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

`JMSException` - 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

`JMSException` - if the topic could not be returned due to some error

getTransacted()

```
public synchronized boolean getTransacted()
```

Checks if the session in transacted mode?

Specified By

javax.jms.Session.getTransacted() in interface javax.jms.Session

Returns

true if in transacted mode

Throws

`JMSException` - 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

`JMSException` - if the system privilege could not be granted.

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

`JMSException` - 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

`JMSException` - 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

`JMSException` - if JMS fails to set the message listener due to an internal error in JMS Provider.

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

Methods

`clearBody()`	Clear out the message body.
`clearProperties()`	
`readBoolean()`	Reads a boolean from the stream message
`readByte()`	Reads a signed 8-bit value from the stream message
`readBytes(byte[])`	Read a byte array from the stream message
`readChar()`	Read a Unicode character value from the stream message
`readDouble()`	Read a double from the stream message
`readFloat()`	Read a float from the stream message
`readInt()`	Read a signed 32 bit integer from the stream message
`readLong()`	Read a signed 64 bit integer from the stream message
`readObject()`	Read a Java object from the stream message.
`readShort()`	Reads a signed 16-bit value from the stream message

Member Summary	
readString()	Read a string from the stream message
reset()	Put the message in read-only mode, and reposition the stream of bytes to the beginning.
writeBoolean(boolean)	Write a boolean to the stream message as a 1-byte value.
writeByte(byte)	Write out a byte to the stream message as a 1-byte value.
writeBytes(byte[])	Write a byte array to the stream message
writeBytes(byte[], int, int)	Write a portion of byte array to the stream message
writeChar(char)	Write a char to the stream as a 2-byte, high byte first
writeDouble(double)	Write a double to the stream as a 8-byte, high byte first
writeFloat(float)	Write a float to the stream as a 4-byte, high byte first
writeInt(int)	Write a int to the stream as a 4-byte, high byte first
writeLong(long)	Write a int to the stream as a 4-byte, high byte first
writeObject(Object)	Write a java object to the stream message
writeShort(short)	Write a short to the stream as a 2-byte, high byte first
writeString(String)	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

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

`JMSException` - 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.

`JMSException` - 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.

`JMSException` - 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.

`JMSException` - 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.

`JMSException` - 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.

`JMSException` - 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.

`JMSException` - 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

`JMSException` - 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.

`JMSException` - 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.

`JMSException` - 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.

`JMSException` - 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.

`JMSException` - 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.

`JMSException` - 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.

`JMSException` - 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.

`JMSException` - 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.

`JMSException` - 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.

`JMSException` - 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.

`JMSException` - 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

`JMSException` - 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.

`JMSException` - 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.

`JMSException` - 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

Methods

`clearBody()`	
`clearProperties()`	Clear a message's properties.
`getText()`	Get the string containing this message's data.
`setText(String)`	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

`JMSException` - 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

`JMSException` - 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

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

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

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`

Description

This class implements the TopicConnectionFactory interface. A TopicConnectionFactory is used to create TopicConnections

Member Summary

Methods

`createTopicConnection()`	create a Topic Connection to the JMS Server hosting this Topic- connection factory.
`createTopicConnection(Connection)`	create a TopicConnection using the already open JDBC connection.
`createTopicConnection(String, String)`	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

`JMSException` - 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

`JMSException` - 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

`JMSException` - 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

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

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

`JMSException` - 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

`JMSException` - 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

`JMSException` - 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

`JMSException` - 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

`JMSException` - 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

`JMSException` - 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-mulitpoint communication). A TopicReceiver is used to receive messages from a Topic

Member Summary

Methods

`getNavigationMode()`	get the navigation mode used for receiving messages
`getTransformation()`	get the transformation for this receiver
`receiveNoData()`	Consume the message without returning it to the user.
`receiveNoData(long)`	Consume the message without returning it to the user.
`setNavigationMode(int)`	set the navigation mode used for receiving messages
`setTransformation(String)`	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

`JMSException` - if there was an error in getting the navigation mode

getTransformation()

```
public String getTransformation()
```
Get the transformation for this receiver

Returns

the transformation

Throws

`JMSException` - 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

`JMSException` - 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

`JMSException` - 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

`JMSException` - 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

`JMSException` - 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

Methods

`getNavigationMode()`	Consume the message without returning it to the user.
`receiveNoData()`	Consume the message without returning it to the user.
`receiveNoData(long)`	Consume the message without returning it to the user.
`setNavigationMode(int)`	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

`JMSException` - 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

`JMSException` - 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

`JMSException` - 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

`JMSException` - if there was an error in getting the navigation mode

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

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

`JMSException` - if JMS fails to get topic for this topic receiver due to some internal error.

3

Package oracle.ODCI

Package oracle.ODCI Description

Class Summary
Classes
ODCIArgDesc
ODCIArgDescList
ODCIArgDescRef
ODCIColInfo
ODCIColInfoList
ODCIColInfoRef
ODCICost
ODCICostRef
ODCIFuncInfo
ODCIFuncInfoRef
ODCIIndexCtx
ODCIIndexCtxRef
ODCIIndexInfo
ODCIIndexInfoRef
ODCIObject
ODCIObjectList
ODCIObjectRef
ODCIPredInfo
ODCIPredInfoRef
ODCIQueryInfo
ODCIQueryInfoRef
ODCIRidList
ODCIStatsOptions
ODCIStatsOptionsRef

ODCIArgDesc

Syntax

```
public class ODCIArgDesc
```

oracle.ODCI.ODCIArgDesc

Description

Member Summary

Fields

_SQL_NAME

_SQL_TYPECODE

Constructors

ODCIArgDesc()

ODCIArgDesc(Connection)

ODCIArgDesc(ConnectionContext)

Methods

create(Datum, int)

getArgtype()

getColname()

getFactory()

getTablename()

getTableschema()

setArgtype(BigDecimal)

setColname(String)

setTablename(String)

setTableschema(String)

toDatum(OracleConnection)

Fields

_SQL_NAME

```
public static final java.lang.String _SQL_NAME
```

_SQL_TYPECODE

```
public static final int _SQL_TYPECODE
```

Constructors

ODCIArgDesc()

```
public  ODCIArgDesc()
```

ODCIArgDesc(Connection)

```
public  ODCIArgDesc(java.sql.Connection c)
```

ODCIArgDesc(ConnectionContext)

```
public  ODCIArgDesc(sqlj.runtime.ConnectionContext c)
```

Methods

create(Datum, int)

```
public oracle.sql.CustomDatum create(oracle.sql.Datum d,int sqlType)
```

getArgtype()

```
public java.math.BigDecimal getArgtype()
```

getColname()

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

getFactory()

```
public static oracle.sql.CustomDatumFactory getFactory()
```

getTablename()

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

getTableschema()

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

setArgtype(BigDecimal)

```
public void setArgtype(java.math.BigDecimal argtype)
```

setColname(String)

```
public void setColname(java.lang.String colname)
```

setTablename(String)

```
public void setTablename(java.lang.String tablename)
```

setTableschema(String)

```
public void setTableschema(java.lang.String tableschema)
```

toDatum(OracleConnection)

```
public oracle.sql.Datum toDatum(oracle.jdbc.driver.OracleConnection c)
```

ODCIArgDescList

Syntax

```
public class ODCIArgDescList
```

oracle.ODCI.ODCIArgDescList

Description

Member Summary
Fields
_SQL_NAME
_SQL_TYPECODE
Constructors
ODCIArgDescList()
ODCIArgDescList(ODCIArgDesc[])
Methods
create(Datum, int)
getArray()
getArray(long, int)
getBaseType()
getBaseTypeName()
getDescriptor()
getElement(long)
getFactory()
length()
setArray(ODCIArgDesc[])
setArray(ODCIArgDesc[], long)
setElement(ODCIArgDesc, long)
toDatum(OracleConnection)

Fields

_SQL_NAME

```
public static final java.lang.String _SQL_NAME
```

_SQL_TYPECODE

```
public static final int _SQL_TYPECODE
```

Constructors

ODCIArgDescList()

```
public  ODCIArgDescList()
```

ODCIArgDescList(ODCIArgDesc[])

```
public  ODCIArgDescList(ODCIArgDesc a)
```

Methods

create(Datum, int)

```
public oracle.sql.CustomDatum create(oracle.sql.Datum d,int sqlType)
```

getArray()

```
public ODCIArgDesc getArray()
```

getArray(long, int)

```
public ODCIArgDesc getArray(long index, int count)
```

getBaseType()

```
public int getBaseType()
```

getBaseTypeName()

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

getDescriptor()

```
public oracle.sql.ArrayDescriptor getDescriptor()
```

getElement(long)

```
public ODCIArgDesc getElement(long index)
```

getFactory()

```
public static oracle.sql.CustomDatumFactory getFactory()
```

length()

```
public int length()
```

setArray(ODCIArgDesc[])

```
public void setArray(ODCIArgDesc a)
```

setArray(ODCIArgDesc[], long)

```
public void setArray(ODCIArgDesc a, long index)
```

setElement(ODCIArgDesc, long)

```
public void setElement(ODCIArgDesc a, long index)
```

toDatum(OracleConnection)

```
public oracle.sql.Datum toDatum(oracle.jdbc.driver.OracleConnection c)
```

ODCIArgDescRef

Syntax

```
public class ODCIArgDescRef
```

oracle.ODCI.ODCIArgDescRef

Description

Member Summary
Fields
_SQL_BASETYPE
_SQL_TYPECODE
Constructors
ODCIArgDescRef()
Methods
create(Datum, int)
getFactory()
getValue()
setValue(ODCIArgDesc)
toDatum(OracleConnection)

Fields

_SQL_BASETYPE

```
public static final java.lang.String _SQL_BASETYPE
```

_SQL_TYPECODE

```
public static final int _SQL_TYPECODE
```

Constructors

ODCIArgDescRef()

```
public  ODCIArgDescRef()
```

Methods

create(Datum, int)

```
public oracle.sql.CustomDatum create(oracle.sql.Datum d,int sqlType)
```

getFactory()

```
public static oracle.sql.CustomDatumFactory getFactory()
```

getValue()

```
public ODCIArgDesc getValue()
```

setValue(ODCIArgDesc)

```
public void setValue(ODCIArgDesc c)
```

toDatum(OracleConnection)

```
public oracle.sql.Datum toDatum(oracle.jdbc.driver.OracleConnection c)
```

ODCIColInfo

Syntax

```
public class ODCIColInfo
```

oracle.ODCI.ODCIColInfo

Description

Member Summary

Fields

_SQL_NAME

_SQL_TYPECODE

Constructors

ODCIColInfo()

ODCIColInfo(Connection)

ODCIColInfo(ConnectionContext)

Methods

create(Datum, int)

getColname()

getColtypename()

getColtypeschema()

getFactory()

getTablename()

getTableschema()

setColname(String)

setColtypename(String)

setColtypeschema(String)

setTablename(String)

setTableschema(String)

Member Summary
toDatum(OracleConnection)

Fields

_SQL_NAME

```
public static final java.lang.String _SQL_NAME
```

_SQL_TYPECODE

```
public static final int _SQL_TYPECODE
```

Constructors

ODCIColInfo()

```
public  ODCIColInfo()
```

ODCIColInfo(Connection)

```
public  ODCIColInfo(java.sql.Connection c)
```

ODCIColInfo(ConnectionContext)

```
public  ODCIColInfo(sqlj.runtime.ConnectionContext c)
```

Methods

create(Datum, int)

```
public oracle.sql.CustomDatum create(oracle.sql.Datum d,int sqlType)
```

getColname()

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

getColtypename()

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

getColtypeschema()

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

getFactory()

```
public static oracle.sql.CustomDatumFactory getFactory()
```

getTablename()

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

getTableschema()

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

setColname(String)

```
public void setColname(java.lang.String colname)
```

setColtypename(String)

```
public void setColtypename(java.lang.String coltypename)
```

setColtypeschema(String)

```
public void setColtypeschema(java.lang.String coltypeschema)
```

setTablename(String)

```
public void setTablename(java.lang.String tablename)
```

setTableschema(String)

```
public void setTableschema(java.lang.String tableschema)
```

toDatum(OracleConnection)

```
public oracle.sql.Datum toDatum(oracle.jdbc.driver.OracleConnection c)
```

ODCIColInfoList

Syntax

```
public class ODCIColInfoList
```

oracle.ODCI.ODCIColInfoList

Description

Member Summary

Fields

_SQL_NAME

_SQL_TYPECODE

Constructors

ODCIColInfoList()

ODCIColInfoList(ODCIColInfo[])

Methods

create(Datum, int)

getArray()

getArray(long, int)

getBaseType()

getBaseTypeName()

getDescriptor()

getElement(long)

getFactory()

length()

setArray(ODCIColInfo[])

setArray(ODCIColInfo[], long)

setElement(ODCIColInfo, long)

toDatum(OracleConnection)

Fields

_SQL_NAME

```
public static final java.lang.String _SQL_NAME
```

_SQL_TYPECODE

```
public static final int _SQL_TYPECODE
```

Constructors

ODCIColInfoList()

```
public  ODCIColInfoList()
```

ODCIColInfoList(ODCIColInfo[])

```
public  ODCIColInfoList(ODCIColInfo a)
```

Methods

create(Datum, int)

```
public oracle.sql.CustomDatum create(oracle.sql.Datum d,int sqlType)
```

getArray()

```
public ODCIColInfo getArray()
```

getArray(long, int)

```
public ODCIColInfo getArray(long index, int count)
```

getBaseType()

```
public int getBaseType()
```

getBaseTypeName()

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

getDescriptor()

```
public oracle.sql.ArrayDescriptor getDescriptor()
```

getElement(long)

```
public ODCIColInfo getElement(long index)
```

getFactory()

```
public static oracle.sql.CustomDatumFactory getFactory()
```

length()

```
public int length()
```

setArray(ODCIColInfo[])

```
public void setArray(ODCIColInfo a)
```

setArray(ODCIColInfo[], long)

```
public void setArray(ODCIColInfo a, long index)
```

setElement(ODCIColInfo, long)

```
public void setElement(ODCIColInfo a, long index)
```

toDatum(OracleConnection)

```
public oracle.sql.Datum toDatum(oracle.jdbc.driver.OracleConnection c)
```

ODCIColInfoRef

Syntax

```
public class ODCIColInfoRef
```

oracle.ODCI.ODCIColInfoRef

Description

Member Summary
Fields
_SQL_BASETYPE
_SQL_TYPECODE
Constructors
ODCIColInfoRef()
Methods
create(Datum, int)
getFactory()
getValue()
setValue(ODCIColInfo)
toDatum(OracleConnection)

Fields

_SQL_BASETYPE

```
public static final java.lang.String _SQL_BASETYPE
```

_SQL_TYPECODE

```
public static final int _SQL_TYPECODE
```

Constructors

ODCIColInfoRef()

```
public  ODCIColInfoRef()
```

Methods

create(Datum, int)

```
public oracle.sql.CustomDatum create(oracle.sql.Datum d,int sqlType)
```

getFactory()

```
public static oracle.sql.CustomDatumFactory getFactory()
```

getValue()

```
public ODCIColInfo getValue()
```

setValue(ODCIColInfo)

```
public void setValue(ODCIColInfo c)
```

toDatum(OracleConnection)

```
public oracle.sql.Datum toDatum(oracle.jdbc.driver.OracleConnection c)
```

ODCICost

Syntax

```
public class ODCICost
```

oracle.ODCI.ODCICost

Description

Member Summary
Fields
_SQL_NAME
_SQL_TYPECODE
Constructors
ODCICost()
ODCICost(Connection)
ODCICost(ConnectionContext)
Methods
create(Datum, int)
getCpucost()
getFactory()
getIocost()
getNetworkcost()
setCpucost(BigDecimal)
setIocost(BigDecimal)
setNetworkcost(BigDecimal)
toDatum(OracleConnection)

Fields

_SQL_NAME

```
public static final java.lang.String _SQL_NAME
```

_SQL_TYPECODE

```
public static final int _SQL_TYPECODE
```

Constructors

ODCICost()

```
public  ODCICost()
```

ODCICost(Connection)

```
public  ODCICost(java.sql.Connection c)
```

ODCICost(ConnectionContext)

```
public  ODCICost(sqlj.runtime.ConnectionContext c)
```

Methods

create(Datum, int)

```
public oracle.sql.CustomDatum create(oracle.sql.Datum d,int sqlType)
```

getCpucost()

```
public java.math.BigDecimal getCpucost()
```

getFactory()

```
public static oracle.sql.CustomDatumFactory getFactory()
```

getIocost()

```
public java.math.BigDecimal getIocost()
```

getNetworkcost()

```
public java.math.BigDecimal getNetworkcost()
```

setCpucost(BigDecimal)

```
public void setCpucost(java.math.BigDecimal cpucost)
```

setIocost(BigDecimal)

```
public void setIocost(java.math.BigDecimal iocost)
```

setNetworkcost(BigDecimal)

```
public void setNetworkcost(java.math.BigDecimal networkcost)
```

toDatum(OracleConnection)

```
public oracle.sql.Datum toDatum(oracle.jdbc.driver.OracleConnection c)
```

ODCICostRef

Syntax

```
public class ODCICostRef
```

oracle.ODCI.ODCICostRef

Description

Member Summary
Fields
_SQL_BASETYPE
_SQL_TYPECODE
Constructors
ODCICostRef()
Methods
create(Datum, int)
getFactory()
getValue()
setValue(ODCICost)
toDatum(OracleConnection)

Fields

_SQL_BASETYPE

```
public static final java.lang.String _SQL_BASETYPE
```

_SQL_TYPECODE

```
public static final int _SQL_TYPECODE
```

Constructors

ODCICostRef()

```
public  ODCICostRef()
```

Methods

create(Datum, int)

```
public oracle.sql.CustomDatum create(oracle.sql.Datum d,int sqlType)
```

getFactory()

```
public static oracle.sql.CustomDatumFactory getFactory()
```

getValue()

```
public ODCICost getValue()
```

setValue(ODCICost)

```
public void setValue(ODCICost c)
```

toDatum(OracleConnection)

```
public oracle.sql.Datum toDatum(oracle.jdbc.driver.OracleConnection c)
```

ODCIFuncInfo

Syntax

```
public class ODCIFuncInfo
```

oracle.ODCI.ODCIFuncInfo

Description

Member Summary

Fields

_SQL_NAME

_SQL_TYPECODE

Constructors

ODCIFuncInfo()

ODCIFuncInfo(Connection)

ODCIFuncInfo(ConnectionContext)

Methods

create(Datum, int)

getFactory()

getFlags()

getMethodname()

getObjectname()

getObjectschema()

setFlags(BigDecimal)

setMethodname(String)

setObjectname(String)

setObjectschema(String)

toDatum(OracleConnection)

Fields

_SQL_NAME

```
public static final java.lang.String _SQL_NAME
```

_SQL_TYPECODE

```
public static final int _SQL_TYPECODE
```

Constructors

ODCIFuncInfo()

```
public  ODCIFuncInfo()
```

ODCIFuncInfo(Connection)

```
public  ODCIFuncInfo(java.sql.Connection c)
```

ODCIFuncInfo(ConnectionContext)

```
public  ODCIFuncInfo(sqlj.runtime.ConnectionContext c)
```

Methods

create(Datum, int)

```
public oracle.sql.CustomDatum create(oracle.sql.Datum d,int sqlType)
```

getFactory()

```
public static oracle.sql.CustomDatumFactory getFactory()
```

getFlags()

```
public java.math.BigDecimal getFlags()
```

getMethodname()

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

getObjectname()

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

getObjectschema()

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

setFlags(BigDecimal)

```
public void setFlags(java.math.BigDecimal flags)
```

setMethodname(String)

```
public void setMethodname(java.lang.String methodname)
```

setObjectname(String)

```
public void setObjectname(java.lang.String objectname)
```

setObjectschema(String)

```
public void setObjectschema(java.lang.String objectschema)
```

toDatum(OracleConnection)

```
public oracle.sql.Datum toDatum(oracle.jdbc.driver.OracleConnection c)
```

ODCIFuncInfoRef

Syntax

```
public class ODCIFuncInfoRef
```

oracle.ODCI.ODCIFuncInfoRef

Description

Member Summary
Fields
_SQL_BASETYPE
_SQL_TYPECODE
Constructors
ODCIFuncInfoRef()
Methods
create(Datum, int)
getFactory()
getValue()
setValue(ODCIFuncInfo)
toDatum(OracleConnection)

Fields

_SQL_BASETYPE

```
public static final java.lang.String _SQL_BASETYPE
```

_SQL_TYPECODE

```
public static final int _SQL_TYPECODE
```

Constructors

ODCIFuncInfoRef()

```
public  ODCIFuncInfoRef()
```

Methods

create(Datum, int)

```
public oracle.sql.CustomDatum create(oracle.sql.Datum d,int sqlType)
```

getFactory()

```
public static oracle.sql.CustomDatumFactory getFactory()
```

getValue()

```
public ODCIFuncInfo getValue()
```

setValue(ODCIFuncInfo)

```
public void setValue(ODCIFuncInfo c)
```

toDatum(OracleConnection)

```
public oracle.sql.Datum toDatum(oracle.jdbc.driver.OracleConnection c)
```

ODCIIndexCtx

Syntax

```
public class ODCIIndexCtx
```

oracle.ODCI.ODCIIndexCtx

Description

Member Summary

Fields

_SQL_NAME

_SQL_TYPECODE

Constructors

ODCIIndexCtx()

ODCIIndexCtx(Connection)

ODCIIndexCtx(ConnectionContext)

Methods

create(Datum, int)

getFactory()

getIndexinfo()

getRid()

setIndexinfo(ODCIIndexInfo)

setRid(String)

toDatum(OracleConnection)

Fields

_SQL_NAME

```
public static final java.lang.String _SQL_NAME
```

_SQL_TYPECODE

```
public static final int _SQL_TYPECODE
```

Constructors

ODCIIndexCtx()

```
public  ODCIIndexCtx()
```

ODCIIndexCtx(Connection)

```
public  ODCIIndexCtx(java.sql.Connection c)
```

ODCIIndexCtx(ConnectionContext)

```
public  ODCIIndexCtx(sqlj.runtime.ConnectionContext c)
```

Methods

create(Datum, int)

```
public oracle.sql.CustomDatum create(oracle.sql.Datum d,int sqlType)
```

getFactory()

```
public static oracle.sql.CustomDatumFactory getFactory()
```

getIndexinfo()

```
public ODCIIndexInfo getIndexinfo()
```

getRid()

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

setIndexinfo(ODCIIndexInfo)

```
public void setIndexinfo(ODCIIndexInfo indexinfo)
```

setRid(String)

```
public void setRid(java.lang.String rid)
```

toDatum(OracleConnection)

```
public oracle.sql.Datum toDatum(oracle.jdbc.driver.OracleConnection c)
```

ODCIIndexCtxRef

Syntax

```
public class ODCIIndexCtxRef
```

`oracle.ODCI.ODCIIndexCtxRef`

Description

Member Summary

Fields

_SQL_BASETYPE

_SQL_TYPECODE

Constructors

ODCIIndexCtxRef()

Methods

create(Datum, int)

getFactory()

getValue()

setValue(ODCIIndexCtx)

toDatum(OracleConnection)

Fields

_SQL_BASETYPE

```
public static final java.lang.String _SQL_BASETYPE
```

_SQL_TYPECODE

```
public static final int _SQL_TYPECODE
```

Constructors

ODCIIndexCtxRef()

```
public  ODCIIndexCtxRef()
```

Methods

create(Datum, int)

```
public oracle.sql.CustomDatum create(oracle.sql.Datum d,int sqlType)
```

getFactory()

```
public static oracle.sql.CustomDatumFactory getFactory()
```

getValue()

```
public ODCIIndexCtx getValue()
```

setValue(ODCIIndexCtx)

```
public void setValue(ODCIIndexCtx c)
```

toDatum(OracleConnection)

```
public oracle.sql.Datum toDatum(oracle.jdbc.driver.OracleConnection c)
```

ODCIIndexInfo

Syntax

```
public class ODCIIndexInfo
```

oracle.ODCI.ODCIIndexInfo

Description

Member Summary

Fields

_SQL_NAME

_SQL_TYPECODE

Constructors

ODCIIndexInfo()

ODCIIndexInfo(Connection)

ODCIIndexInfo(ConnectionContext)

Methods

create(Datum, int)

getFactory()

getIndexcols()

getIndexname()

getIndexschema()

setIndexcols(ODCIColInfoList)

setIndexname(String)

setIndexschema(String)

toDatum(OracleConnection)

Fields

_SQL_NAME

```
public static final java.lang.String _SQL_NAME
```

_SQL_TYPECODE

```
public static final int _SQL_TYPECODE
```

Constructors

ODCIIndexInfo()

```
public  ODCIIndexInfo()
```

ODCIIndexInfo(Connection)

```
public  ODCIIndexInfo(java.sql.Connection c)
```

ODCIIndexInfo(ConnectionContext)

```
public  ODCIIndexInfo(sqlj.runtime.ConnectionContext c)
```

Methods

create(Datum, int)

```
public oracle.sql.CustomDatum create(oracle.sql.Datum d,int sqlType)
```

getFactory()

```
public static oracle.sql.CustomDatumFactory getFactory()
```

getIndexcols()

```
public ODCIColInfoList getIndexcols()
```

getIndexname()

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

getIndexschema()

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

setIndexcols(ODCIColInfoList)

```
public void setIndexcols(ODCIColInfoList indexcols)
```

setIndexname(String)

```
public void setIndexname(java.lang.String indexname)
```

setIndexschema(String)

```
public void setIndexschema(java.lang.String indexschema)
```

toDatum(OracleConnection)

```
public oracle.sql.Datum toDatum(oracle.jdbc.driver.OracleConnection c)
```

ODCIIndexInfoRef

Syntax

```
public class ODCIIndexInfoRef
```

oracle.ODCI.ODCIIndexInfoRef

Description

Member Summary
Fields
_SQL_BASETYPE
_SQL_TYPECODE
Constructors
ODCIIndexInfoRef()
Methods
create(Datum, int)
getFactory()
getValue()
setValue(ODCIIndexInfo)
toDatum(OracleConnection)

Fields

_SQL_BASETYPE

```
public static final java.lang.String _SQL_BASETYPE
```

_SQL_TYPECODE

```
public static final int _SQL_TYPECODE
```

Constructors

ODCIIndexInfoRef()

```
public  ODCIIndexInfoRef()
```

Methods

create(Datum, int)

```
public oracle.sql.CustomDatum create(oracle.sql.Datum d,int sqlType)
```

getFactory()

```
public static oracle.sql.CustomDatumFactory getFactory()
```

getValue()

```
public ODCIIndexInfo getValue()
```

setValue(ODCIIndexInfo)

```
public void setValue(ODCIIndexInfo c)
```

toDatum(OracleConnection)

```
public oracle.sql.Datum toDatum(oracle.jdbc.driver.OracleConnection c)
```

ODCIObject

Syntax

```
public class ODCIObject
```

oracle.ODCI.ODCIObject

Description

Member Summary

Fields

_SQL_NAME

_SQL_TYPECODE

Constructors

ODCIObject()

ODCIObject(Connection)

ODCIObject(ConnectionContext)

Methods

create(Datum, int)

getFactory()

getObjectname()

getObjectschema()

setObjectname(String)

setObjectschema(String)

toDatum(OracleConnection)

Fields

_SQL_NAME

```
public static final java.lang.String _SQL_NAME
```

_SQL_TYPECODE

```
public static final int _SQL_TYPECODE
```

Constructors

ODCIObject()

```
public  ODCIObject()
```

ODCIObject(Connection)

```
public  ODCIObject(java.sql.Connection c)
```

ODCIObject(ConnectionContext)

```
public  ODCIObject(sqlj.runtime.ConnectionContext c)
```

Methods

create(Datum, int)

```
public oracle.sql.CustomDatum create(oracle.sql.Datum d,int sqlType)
```

getFactory()

```
public static oracle.sql.CustomDatumFactory getFactory()
```

getObjectname()

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

getObjectschema()

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

setObjectname(String)

```
public void setObjectname(java.lang.String objectname)
```

setObjectschema(String)

```
public void setObjectschema(java.lang.String objectschema)
```

toDatum(OracleConnection)

```
public oracle.sql.Datum toDatum(oracle.jdbc.driver.OracleConnection c)
```

ODCIObjectList

Syntax

```
public class ODCIObjectList
```

`oracle.ODCI.ODCIObjectList`

Description

Member Summary
Fields
`_SQL_NAME`
`_SQL_TYPECODE`
Constructors
`ODCIObjectList()`
`ODCIObjectList(ODCIObject[])`
Methods
`create(Datum, int)`
`getArray()`
`getArray(long, int)`
`getBaseType()`
`getBaseTypeName()`
`getDescriptor()`
`getElement(long)`
`getFactory()`
`length()`
`setArray(ODCIObject[])`
`setArray(ODCIObject[], long)`
`setElement(ODCIObject, long)`
`toDatum(OracleConnection)`

Fields

_SQL_NAME

```
public static final java.lang.String _SQL_NAME
```

_SQL_TYPECODE

```
public static final int _SQL_TYPECODE
```

Constructors

ODCIObjectList()

```
public  ODCIObjectList()
```

ODCIObjectList(ODCIObject[])

```
public  ODCIObjectList(ODCIObject a)
```

Methods

create(Datum, int)

```
public oracle.sql.CustomDatum create(oracle.sql.Datum d,int sqlType)
```

getArray()

```
public ODCIObject getArray()
```

getArray(long, int)

```
public ODCIObject getArray(long index, int count)
```

getBaseType()

```
public int getBaseType()
```

getBaseTypeName()

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

getDescriptor()

```
public oracle.sql.ArrayDescriptor getDescriptor()
```

getElement(long)

```
public ODCIObject getElement(long index)
```

getFactory()

```
public static oracle.sql.CustomDatumFactory getFactory()
```

length()

```
public int length()
```

setArray(ODCIObject[])

```
public void setArray(ODCIObject a)
```

setArray(ODCIObject[], long)

```
public void setArray(ODCIObject a, long index)
```

setElement(ODCIObject, long)

```
public void setElement(ODCIObject a, long index)
```

toDatum(OracleConnection)

```
public oracle.sql.Datum toDatum(oracle.jdbc.driver.OracleConnection c)
```

ODCIObjectRef

Syntax

```
public class ODCIObjectRef
```

oracle.ODCI.ODCIObjectRef

Description

Member Summary
Fields
_SQL_BASETYPE
_SQL_TYPECODE
Constructors
ODCIObjectRef()
Methods
create(Datum, int)
getFactory()
getValue()
setValue(ODCIObject)
toDatum(OracleConnection)

Fields

_SQL_BASETYPE

```
public static final java.lang.String _SQL_BASETYPE
```

_SQL_TYPECODE

```
public static final int _SQL_TYPECODE
```

Constructors

ODCIObjectRef()

```
public  ODCIObjectRef()
```

Methods

create(Datum, int)

```
public oracle.sql.CustomDatum create(oracle.sql.Datum d,int sqlType)
```

getFactory()

```
public static oracle.sql.CustomDatumFactory getFactory()
```

getValue()

```
public ODCIObject getValue()
```

setValue(ODCIObject)

```
public void setValue(ODCIObject c)
```

toDatum(OracleConnection)

```
public oracle.sql.Datum toDatum(oracle.jdbc.driver.OracleConnection c)
```

ODCIPredInfo

Syntax

```
public class ODCIPredInfo
```

oracle.ODCI.ODCIPredInfo

Description

Member Summary

Fields

_SQL_NAME

_SQL_TYPECODE

Constructors

ODCIPredInfo()

ODCIPredInfo(Connection)

ODCIPredInfo(ConnectionContext)

Methods

create(Datum, int)

getFactory()

getFlags()

getMethodname()

getObjectname()

getObjectschema()

setFlags(BigDecimal)

setMethodname(String)

setObjectname(String)

setObjectschema(String)

toDatum(OracleConnection)

Fields

_SQL_NAME

```
public static final java.lang.String _SQL_NAME
```

_SQL_TYPECODE

```
public static final int _SQL_TYPECODE
```

Constructors

ODCIPredInfo()

```
public  ODCIPredInfo()
```

ODCIPredInfo(Connection)

```
public  ODCIPredInfo(java.sql.Connection c)
```

ODCIPredInfo(ConnectionContext)

```
public  ODCIPredInfo(sqlj.runtime.ConnectionContext c)
```

Methods

create(Datum, int)

```
public oracle.sql.CustomDatum create(oracle.sql.Datum d,int sqlType)
```

getFactory()

```
public static oracle.sql.CustomDatumFactory getFactory()
```

getFlags()

```
public java.math.BigDecimal getFlags()
```

getMethodname()

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

getObjectname()

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

getObjectschema()

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

setFlags(BigDecimal)

```
public void setFlags(java.math.BigDecimal flags)
```

setMethodname(String)

```
public void setMethodname(java.lang.String methodname)
```

setObjectname(String)

```
public void setObjectname(java.lang.String objectname)
```

setObjectschema(String)

```
public void setObjectschema(java.lang.String objectschema)
```

toDatum(OracleConnection)

```
public oracle.sql.Datum toDatum(oracle.jdbc.driver.OracleConnection c)
```

ODCIPredInfoRef

Syntax

```
public class ODCIPredInfoRef
```

`oracle.ODCI.ODCIPredInfoRef`

Description

Member Summary
Fields
`_SQL_BASETYPE`
`_SQL_TYPECODE`
Constructors
`ODCIPredInfoRef()`
Methods
`create(Datum, int)`
`getFactory()`
`getValue()`
`setValue(ODCIPredInfo)`
`toDatum(OracleConnection)`

Fields

_SQL_BASETYPE

```
public static final java.lang.String _SQL_BASETYPE
```

_SQL_TYPECODE

```
public static final int _SQL_TYPECODE
```

Constructors

ODCIPredInfoRef()

```
public  ODCIPredInfoRef()
```

Methods

create(Datum, int)

```
public oracle.sql.CustomDatum create(oracle.sql.Datum d,int sqlType)
```

getFactory()

```
public static oracle.sql.CustomDatumFactory getFactory()
```

getValue()

```
public ODCIPredInfo getValue()
```

setValue(ODCIPredInfo)

```
public void setValue(ODCIPredInfo c)
```

toDatum(OracleConnection)

```
public oracle.sql.Datum toDatum(oracle.jdbc.driver.OracleConnection c)
```

ODCIQueryInfo

Syntax

```
public class ODCIQueryInfo
```

oracle.ODCI.ODCIQueryInfo

Description

Member Summary

Fields

_SQL_NAME

_SQL_TYPECODE

Constructors

ODCIQueryInfo()

ODCIQueryInfo(Connection)

ODCIQueryInfo(ConnectionContext)

Methods

create(Datum, int)

getAncops()

getFactory()

getFlags()

setAncops(ODCIObjectList)

setFlags(BigDecimal)

toDatum(OracleConnection)

Fields

_SQL_NAME

```
public static final java.lang.String _SQL_NAME
```

_SQL_TYPECODE

```
public static final int _SQL_TYPECODE
```

Constructors

ODCIQueryInfo()

```
public  ODCIQueryInfo()
```

ODCIQueryInfo(Connection)

```
public  ODCIQueryInfo(java.sql.Connection c)
```

ODCIQueryInfo(ConnectionContext)

```
public  ODCIQueryInfo(sqlj.runtime.ConnectionContext c)
```

Methods

create(Datum, int)

```
public oracle.sql.CustomDatum create(oracle.sql.Datum d,int sqlType)
```

getAncops()

```
public ODCIObjectList getAncops()
```

getFactory()

```
public static oracle.sql.CustomDatumFactory getFactory()
```

getFlags()

```
public java.math.BigDecimal getFlags()
```

setAncops(ODCIObjectList)

```
public void setAncops(ODCIObjectList ancops)
```

setFlags(BigDecimal)

```
public void setFlags(java.math.BigDecimal flags)
```

toDatum(OracleConnection)

```
public oracle.sql.Datum toDatum(oracle.jdbc.driver.OracleConnection c)
```

ODCIQueryInfoRef

Syntax

```
public class ODCIQueryInfoRef
```

oracle.ODCI.ODCIQueryInfoRef

Description

Member Summary
Fields
_SQL_BASETYPE
_SQL_TYPECODE
Constructors
ODCIQueryInfoRef()
Methods
create(Datum, int)
getFactory()
getValue()
setValue(ODCIQueryInfo)
toDatum(OracleConnection)

Fields

_SQL_BASETYPE

```
public static final java.lang.String _SQL_BASETYPE
```

_SQL_TYPECODE

```
public static final int _SQL_TYPECODE
```

Constructors

ODCIQueryInfoRef()

```
public  ODCIQueryInfoRef()
```

Methods

create(Datum, int)

```
public oracle.sql.CustomDatum create(oracle.sql.Datum d,int sqlType)
```

getFactory()

```
public static oracle.sql.CustomDatumFactory getFactory()
```

getValue()

```
public ODCIQueryInfo getValue()
```

setValue(ODCIQueryInfo)

```
public void setValue(ODCIQueryInfo c)
```

toDatum(OracleConnection)

```
public oracle.sql.Datum toDatum(oracle.jdbc.driver.OracleConnection c)
```

ODCIRidList

Syntax

```
public class ODCIRidList
```

oracle.ODCI.ODCIRidList

Description

Member Summary

Fields

_SQL_NAME

_SQL_TYPECODE

Constructors

ODCIRidList()

ODCIRidList(String[])

Methods

create(Datum, int)

getArray()

getArray(long, int)

getBaseType()

getBaseTypeName()

getDescriptor()

getElement(long)

getFactory()

length()

setArray(String[])

setArray(String[], long)

setElement(String, long)

toDatum(OracleConnection)

Fields

_SQL_NAME

```
public static final java.lang.String _SQL_NAME
```

_SQL_TYPECODE

```
public static final int _SQL_TYPECODE
```

Constructors

ODCIRidList()

```
public  ODCIRidList()
```

ODCIRidList(String[])

```
public  ODCIRidList(java.lang.String[] a)
```

Methods

create(Datum, int)

```
public oracle.sql.CustomDatum create(oracle.sql.Datum d,int sqlType)
```

getArray()

```
public java.lang.String[] getArray()
```

getArray(long, int)

```
public java.lang.String[] getArray(long index, int count)
```

getBaseType()

```
public int getBaseType()
```

getBaseTypeName()

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

getDescriptor()

```
public oracle.sql.ArrayDescriptor getDescriptor()
```

getElement(long)

```
public java.lang.String getElement(long index)
```

getFactory()

```
public static oracle.sql.CustomDatumFactory getFactory()
```

length()

```
public int length()
```

setArray(String[])

```
public void setArray(java.lang.String[] a)
```

setArray(String[], long)

```
public void setArray(java.lang.String[] a, long index)
```

setElement(String, long)

```
public void setElement(java.lang.String a, long index)
```

toDatum(OracleConnection)

```
public oracle.sql.Datum toDatum(oracle.jdbc.driver.OracleConnection c)
```

ODCIStatsOptions

Syntax

```
public class ODCIStatsOptions
```

oracle.ODCI.ODCIStatsOptions

Description

Member Summary

Fields

_SQL_NAME

_SQL_TYPECODE

Constructors

ODCIStatsOptions()

ODCIStatsOptions(Connection)

ODCIStatsOptions(ConnectionContext)

Methods

create(Datum, int)

getFactory()

getFlags()

getOptions()

getSample()

setFlags(BigDecimal)

setOptions(BigDecimal)

setSample(BigDecimal)

toDatum(OracleConnection)

Fields

_SQL_NAME

```
public static final java.lang.String _SQL_NAME
```

_SQL_TYPECODE

```
public static final int _SQL_TYPECODE
```

Constructors

ODCIStatsOptions()

```
public  ODCIStatsOptions()
```

ODCIStatsOptions(Connection)

```
public  ODCIStatsOptions(java.sql.Connection c)
```

ODCIStatsOptions(ConnectionContext)

```
public  ODCIStatsOptions(sqlj.runtime.ConnectionContext c)
```

Methods

create(Datum, int)

```
public oracle.sql.CustomDatum create(oracle.sql.Datum d,int sqlType)
```

getFactory()

```
public static oracle.sql.CustomDatumFactory getFactory()
```

getFlags()

```
public java.math.BigDecimal getFlags()
```

getOptions()

```
public java.math.BigDecimal getOptions()
```

getSample()

```
public java.math.BigDecimal getSample()
```

setFlags(BigDecimal)

```
public void setFlags(java.math.BigDecimal flags)
```

setOptions(BigDecimal)

```
public void setOptions(java.math.BigDecimal options)
```

setSample(BigDecimal)

```
public void setSample(java.math.BigDecimal sample)
```

toDatum(OracleConnection)

```
public oracle.sql.Datum toDatum(oracle.jdbc.driver.OracleConnection c)
```

ODCIStatsOptionsRef

Syntax

```
public class ODCIStatsOptionsRef
```

oracle.ODCI.ODCIStatsOptionsRef

Description

Member Summary
Fields
_SQL_BASETYPE
_SQL_TYPECODE
Constructors
ODCIStatsOptionsRef()
Methods
create(Datum, int)
getFactory()
getValue()
setValue(ODCIStatsOptions)
toDatum(OracleConnection)

Fields

_SQL_BASETYPE

```
public static final java.lang.String _SQL_BASETYPE
```

_SQL_TYPECODE

```
public static final int _SQL_TYPECODE
```

Constructors

ODCIStatsOptionsRef()

```
public  ODCIStatsOptionsRef()
```

Methods

create(Datum, int)

```
public oracle.sql.CustomDatum create(oracle.sql.Datum d,int sqlType)
```

getFactory()

```
public static oracle.sql.CustomDatumFactory getFactory()
```

getValue()

```
public ODCIStatsOptions getValue()
```

setValue(ODCIStatsOptions)

```
public void setValue(ODCIStatsOptions c)
```

toDatum(OracleConnection)

```
public oracle.sql.Datum toDatum(oracle.jdbc.driver.OracleConnection c)
```

4

Package oracle.xml.parser.v2

Description

Class Summary	
Interfaces	
NSName	This interface provides Namespace support for Element and Attr names
NSResolver	This interface provides support for resolving Namespaces
XMLDocumentHandler	This interface extends the org.xml.sax.DocumentHandler interface.
XMLToken	Basic interface for XMLToken
Classes	
AttrDecl	This class hold information about each attribute declared in an attribute list in the Document Type Definition.
Package oracle.xml.parser.v2	This class implements the default behaviour for the XMLDocumentHandler interface.
DOMParser	This class implements an eXtensible Markup Language (XML) 1.0 parser according to the World Wide Web Consortium (W3C) recommendation.
DTD	Implements the DOM DocumentType interface and holds the Document Type Definition information for an XML document.
ElementDecl	This class represents an element declaration in a DTD.
NodeFactory	This class specifies methods to create various nodes of the DOM tree built during parsing.
oraxsl	The oraxsl class provides a command-line interface to applying stylesheets on multiple XML documents.
SAXAttrlist	This class implements the SAX AttributeList interface and also provides Namespace support.
SAXParser	This class implements an eXtensible Markup Language (XML) 1.0 SAX parser according to the World Wide Web Consortium (W3C) recommendation.
XMLAttr	This class implements the DOM Attr interface and holds information on each attribute of an element.
XMLCData	This class implements the DOM CDATASection interface.
XMLComment	This class implements the DOM Comment interface.
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.

Class Summary	
XMLElement	This class implements the DOM Element interface.
XMLEntityReference	
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.
XSLProcessor	This class provides methods to transform an input XML document using a previously constructed XSLStylesheet.
XSLStylesheet	The class holds XSL stylesheet information such as templates, keys, variables, and attribute sets.
Exceptions	
XMLParseException	Indicates that a parsing exception occurred while processing an XML document
XSLException	Indicates that an exception occurred during XSL tranformation

AttrDecl

Syntax

```
public class AttrDecl extends XMLNode implements
oracle.xml.parser.v2.XMLConstants, java.io.Serializable

java.lang.Object
  |
  +--XMLNode
        |
        +--oracle.xml.parser.v2.AttrDecl
```

All Implemented Interfaces

java.lang.Cloneable, org.w3c.dom.Node, java.io.Serializable, oracle.xml.parser.v2.XMLConstants

Description

This class holds information about each attribute declared in an attribute list in the Document Type Definition.

Member Summary

Fields

CDATA	AttType - StringType - CDATA
DEFAULT	Attribute presence - Default
ENTITIES	AttType - TokenizedType - Entities
ENTITY	AttType - TokenizedType - Entity
ENUMERATION	AttType - EnumeratedType - Enumeration
FIXED	Attribute presence - Fixed
ID	AttType - TokenizedType - ID
IDREF	AttType - TokenizedType - ID reference
IDREFS	AttType - TokenizedType - ID references
IMPLIED	Attribute presence - Implied
NMTOKEN	AttType - TokenizedType - Name token
NMTOKENS	AttType - TokenizedType - Name tokens

Member Summary

NOTATION	AttType - EnumeratedType - Notation
REQUIRED	Attribute presence - Required
Methods	
getAttrPresence()	Gets attribute presence
getAttrType()	Gets attribute type
getDefaultValue()	Gets attribute default value
getEnumerationValues()	Gets attribute values

Inherited Member Summary

Fields inherited from class XMLNode

AMP, ASTERISK, ATTRDECL, cANY, cATTLIST, cCDATA, cCDATAEND, cCDATASTART, cCOMMENTEND, cCOMMENTSTART, cDECCREF, cDECLSTART, cDOCTYPE, cELEMENT, cEMPTY, cEMPTYTAGEND, cENCODING, cENDTAGSTART, cENTITIES, cENTITY, cFIXED, cHEXCREF, cID, cIDREF, cIDREFS, cIGNORE, cIMPLIED, cINCLUDE, cNDATA, cNMTOKEN, cNMTOKENS, cNOTATION, COLON, COMMA, cPIEND, cPISTART, cPUBLIC, cREQUIRED, cSTANDALONE, cSYSTEM, cVERSION, cXML, DOUBLEQUOTE, ELEMENTDECL, EOF, EQ, ERROR, FATAL_ERROR, FDIGIT, FLETTER, FMISCNAME, FSTARTNAME, FWHITESPACE, HASH, ICOUNT, ISTART, LEFTSQB, LPAREN, nameCDATA, nameCOMMENT, nameDOCUMENT, nameDOCUMENTFRAGMENT, nameENCODING, nameNameSpace, nameSpaceSeparator, nameSTANDALONE, nameTEXT, nameVERSION, nameXML, nameXMLLang, nameXMLNamespace, nameXMLNSNamespace, nameXMLSpace, nameXSLPI, NONVALIDATING, OR, PERCENT, PLUS, QMARK, QUOTE, RIGHTSQB, RPAREN, SEMICOLON, SLASH, TAGEND, TAGSTART, VALIDATING, WARNING

Fields inherited from interface oracle.xml.parser.v2.XMLConstants

AMP, ASTERISK, cANY, cATTLIST, cCDATA, cCDATAEND, cCDATASTART, cCOMMENTEND, cCOMMENTSTART, cDECCREF, cDECLSTART, cDOCTYPE, cELEMENT, cEMPTY, cEMPTYTAGEND, cENCODING, cENDTAGSTART, cENTITIES, cENTITY, cFIXED, cHEXCREF, cID, cIDREF, cIDREFS, cIGNORE, cIMPLIED, cINCLUDE, cNDATA, cNMTOKEN, cNMTOKENS, cNOTATION, COLON, COMMA, cPIEND, cPISTART, cPUBLIC, cREQUIRED, cSTANDALONE, cSYSTEM, cVERSION, cXML, DOUBLEQUOTE, EOF, EQ, ERROR, FATAL_ERROR, FDIGIT, FLETTER, FMISCNAME, FSTARTNAME, FWHITESPACE, HASH, ICOUNT, ISTART, LEFTSQB, LPAREN, nameCDATA, nameCOMMENT, nameDOCUMENT, nameDOCUMENTFRAGMENT, nameENCODING, nameNameSpace, nameSpaceSeparator, nameSTANDALONE, nameTEXT, nameVERSION, nameXML, nameXMLLang, nameXMLNamespace, nameXMLNSNamespace, nameXMLSpace, nameXSLPI, NONVALIDATING, OR, PERCENT, PLUS, QMARK, QUOTE, RIGHTSQB, RPAREN, SEMICOLON, SLASH, TAGEND, TAGSTART, VALIDATING, WARNING

Fields inherited from interface Node

ATTRIBUTE_NODE, CDATA_SECTION_NODE, COMMENT_NODE, DOCUMENT_FRAGMENT_NODE, DOCUMENT_NODE, DOCUMENT_TYPE_NODE, ELEMENT_NODE, ENTITY_NODE, ENTITY_REFERENCE_NODE, NOTATION_NODE, PROCESSING_INSTRUCTION_NODE, TEXT_NODE

Methods inherited from class XMLNode

Inherited Member Summary

appendChild(Node), cloneNode(boolean), getAttributes(), getChildNodes(), getFirstChild(), getLastChild(), getNextSibling(), getNodeName(), getNodeType(), getNodeValue(), getOwnerDocument(), getParentNode(), getPreviousSibling(), hasChildNodes(), insertBefore(Node, Node), print(OutputStream), print(OutputStream, String), print(PrintWriter), removeChild(Node), replaceChild(Node, Node), selectNodes(String, NSResolver), selectSingleNode(String, NSResolver), setNodeValue(String), transformNode(XSLStylesheet), valueOf(String, NSResolver)

Methods inherited from class java.lang.Object

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

Methods inherited from interface Node

appendChild(Node), cloneNode(boolean), getAttributes(), getChildNodes(), getFirstChild(), getLastChild(), getNextSibling(), getNodeName(), getNodeType(), getNodeValue(), getOwnerDocument(), getParentNode(), getPreviousSibling(), hasChildNodes(), insertBefore(Node, Node), removeChild(Node), replaceChild(Node, Node), setNodeValue(String)

Fields

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

REQUIRED

```
public static final int REQUIRED
```
Attribute presence - Required

Methods

getAttrPresence()

```
public int getAttrPresence()
```
Gets attribute presence

Returns

The presence of the attribute

getAttrType()

```
public int getAttrType()
```
Gets attribute type

Returns

The type of the attribute

getDefaultValue()

```
public java.lang.String getDefaultValue()
```
Gets attribute default value

Returns

The default value of the attribute

getEnumerationValues()

```
public java.util.Vector getEnumerationValues()
```
Gets attribute values

Returns

The values of the attribute as an `Enumeration`

DefaultXMLDocumentHandler

Syntax

```
public class DefaultXMLDocumentHandler extends org.xml.sax.HandlerBase implements
XMLDocumentHandler

java.lang.Object
  |
  +--org.xml.sax.HandlerBase
        |
        +--oracle.xml.parser.v2.DefaultXMLDocumentHandler
```

Direct Known Subclasses:

XMLTokenizer

All Implemented Interfaces

org.xml.sax.DocumentHandler, org.xml.sax.DTDHandler, org.xml.sax.EntityResolver, org.xml.sax.ErrorHandler, XMLDocumentHandler

Description

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

Member Summary

Constructors	
DefaultXMLDocumentHandler()	Constructs a default document handler
Methods	
cDATASection(char[], int, int)	Receive notification of a CDATA Section.
comment(String)	Receive notification of a comment.
endDoctype()	Receive notification of end of the DTD.
endElement(NSName)	Receive notification of the end of an element.
setDoctype(DTD)	Receive notification of DTD.
setTextDecl(String, String)	Receive notification of a Text XML Declaration.

Member Summary	
setXMLDecl(String, String, String)	Receive notification of an XML Declaration.
startElement(NSName, SAXAttrList)	Receive notification of the beginning of an element.

Inherited Member Summary

Methods inherited from class HandlerBase

characters(char[], int, int), endDocument(), endElement(String), error(SAXParseException), fatalError(SAXParseException), ignorableWhitespace(char[], int, int), notationDecl(String, String, String), processingInstruction(String, String), resolveEntity(String, String), setDocumentLocator(Locator), startDocument(), startElement(String, AttributeList), unparsedEntityDecl(String, String, String, String), warning(SAXParseException)

Methods inherited from class java.lang.Object

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

Methods inherited from interface DocumentHandler

characters(char[], int, int), endDocument(), endElement(String), ignorableWhitespace(char[], int, int), processingInstruction(String, String), setDocumentLocator(Locator), startDocument(), startElement(String, AttributeList)

Methods inherited from interface EntityResolver

resolveEntity(String, String)

Methods inherited from interface DTDHandler

notationDecl(String, String, String), unparsedEntityDecl(String, String, String, String)

Methods inherited from interface ErrorHandler

error(SAXParseException), fatalError(SAXParseException), warning(SAXParseException)

Constructor

DefaultXMLDocumentHandler()

```
public DefaultXMLDocumentHandler()
```

Constructs a default document handler

Methods

cDATASection(char[], int, int)

`public void cDATASection(char[] ch, int start, int length)`
Receive notification of a CDATA Section.

The Parser will invoke this method once for each CDATA Section found.

Specified By

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)

`public void comment(java.lang.String data)`
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.

Specified By

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()

```
public void endDoctype()
```
Receive notification of end of the DTD.

Specified By

endDoctype() in interface XMLDocumentHandler

Throws

org.xml.sax.SAXException - Any SAX exception, possibly wrapping another exception.

endElement(NSName)

```
public void endElement(NSName elem)
```
Receive notification of the end of an element. The SAX parser will invoke this method at the end of every element in the XML document; there will be a corresponding startElement() event for every endElement() event (even when the element is empty).

By implementing this method instead of `org.xml.sax.DocumentHandler.endElement`, SAX Applications can get the Namespace support provided by `NSName`.

Specified By

endElement(NSName) in interface XMLDocumentHandler

Parameters

`elem` - NSName object

Throws

org.xml.sax.SAXException - A `SAXException` could be thrown.

See Also

org.xml.sax.DocumentHandler.endElement(String)

setDoctype(DTD)

```
public void setDoctype(DTD dtd)
```
Receive notification of DTD. Sets the DTD

Specified By

setDoctype(DTD) in interface XMLDocumentHandler

Throws

org.xml.sax.SAXException - Any SAX exception, possibly wrapping another exception.

setTextDecl(String, String)

`public void setTextDecl(java.lang.String version, java.lang.String encoding)`
Receive notification of a Text XML Declaration.

The Parser will invoke this method once for each text XML Decl

Specified By

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)

`public void setXMLDecl(java.lang.String version, java.lang.String standalone, java.lang.String encoding)`
Receive notification of an XML Declaration.

The Parser will invoke this method once for XML Decl

Specified By

setXMLDecl(String, String, String) in interface XMLDocumentHandler

Parameters

`version` - The version number

`standalone` - The standalone value (or null, if not specifed)

`encoding` - The encoding name (or null, if not specifed)

Throws

org.xml.sax.SAXException - Any SAX exception, possibly wrapping another exception.

startElement(NSName, SAXAttrList)

```
public void startElement(NSName elem, SAXAttrList attrlist)
```

Receive notification of the beginning of an element. The Parser will invoke this method at the beginning of every element in the XML document; there will be a corresponding endElement() event for every startElement() event (even when the element is empty). All of the element's content will be reported, inorder, before the corresponding endElement() event.

By implementing this method instead of `org.xml.sax.DocumentHandler.startElement`, SAX Applications can get the Namespace support provided by `NSName` and `SAXAttrList`.

Specified By

startElement(NSName, SAXAttrList) in interface XMLDocumentHandler

Parameters

`elem` - NSName object

`attrlist` - SAXAttrList for the element

Throws

org.xml.sax.SAXException - A `SAXException` could be thrown.

See Also

org.xml.sax.DocumentHandler.startElement(String, AttributeList)

DOMParser

Syntax

```
public class DOMParser extends XMLParser implements
oracle.xml.parser.v2.XMLConstants

java.lang.Object
  |
  +--XMLParser
        |
        +--oracle.xml.parser.v2.DOMParser
```

All Implemented Interfaces

`oracle.xml.parser.v2.XMLConstants`

Description

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.

Member Summary

Constructors	
DOMParser()	Creates a new parser object.
Methods	
getDoctype()	Get the DTD
getDocument()	Gets the document
parseDTD(InputSource, String)	Parses the XML External DTD from given input source
parseDTD(InputStream, String)	Parses the XML External DTD from given input stream.
parseDTD(Reader, String)	Parses the XML External DTD from given input stream.
parseDTD(String, String)	Parses the XML External DTD from the URL indicated
parseDTD(URL, String)	Parses the XML External DTD document pointed to by the given URL and creates the corresponding XML document hierarchy.
setErrorStream(OutputStream)	Creates an output stream for the output of errors and warnings.

Member Summary

setErrorStream(OutputStream, String)	Creates an output stream for the output of errors and warnings.
setErrorStream(PrintWriter)	Creates an output stream for the output of errors and warnings.
setNodeFactory(NodeFactory)	Set the node factory.
showWarnings(boolean)	Switch to determine whether to print warnings

Inherited Member Summary

Fields inherited from class XMLParser

AMP, ASTERISK, cANY, cATTLIST, cCDATA, cCDATAEND, cCDATASTART, cCOMMENTEND, cCOMMENTSTART, cDECCREF, cDECLSTART, cDOCTYPE, cELEMENT, cEMPTY, cEMPTYTAGEND, cENCODING, cENDTAGSTART, cENTITIES, cENTITY, cFIXED, cHEXCREF, cID, cIDREF, cIDREFS, cIGNORE, cIMPLIED, cINCLUDE, cNDATA, cNMTOKEN, cNMTOKENS, cNOTATION, COLON, COMMA, cPIEND, cPISTART, cPUBLIC, cREQUIRED, cSTANDALONE, cSYSTEM, cVERSION, cXML, DOUBLEQUOTE, EOF, EQ, ERROR, FATAL_ERROR, FDIGIT, FLETTER, FMISCNAME, FSTARTNAME, FWHITESPACE, HASH, ICOUNT, ISTART, LEFTSQB, LPAREN, nameCDATA, nameCOMMENT, nameDOCUMENT, nameDOCUMENTFRAGMENT, nameENCODING, nameNameSpace, nameSpaceSeparator, nameSTANDALONE, nameTEXT, nameVERSION, nameXML, nameXMLLang, nameXMLNamespace, nameXMLNSNamespace, nameXMLSpace, nameXSLPI, NONVALIDATING, OR, PERCENT, PLUS, QMARK, QUOTE, RIGHTSQB, RPAREN, SEMICOLON, SLASH, TAGEND, TAGSTART, VALIDATING, WARNING

Fields inherited from interface oracle.xml.parser.v2.XMLConstants

AMP, ASTERISK, cANY, cATTLIST, cCDATA, cCDATAEND, cCDATASTART, cCOMMENTEND, cCOMMENTSTART, cDECCREF, cDECLSTART, cDOCTYPE, cELEMENT, cEMPTY, cEMPTYTAGEND, cENCODING, cENDTAGSTART, cENTITIES, cENTITY, cFIXED, cHEXCREF, cID, cIDREF, cIDREFS, cIGNORE, cIMPLIED, cINCLUDE, cNDATA, cNMTOKEN, cNMTOKENS, cNOTATION, COLON, COMMA, cPIEND, cPISTART, cPUBLIC, cREQUIRED, cSTANDALONE, cSYSTEM, cVERSION, cXML, DOUBLEQUOTE, EOF, EQ, ERROR, FATAL_ERROR, FDIGIT, FLETTER, FMISCNAME, FSTARTNAME, FWHITESPACE, HASH, ICOUNT, ISTART, LEFTSQB, LPAREN, nameCDATA, nameCOMMENT, nameDOCUMENT, nameDOCUMENTFRAGMENT, nameENCODING, nameNameSpace, nameSpaceSeparator, nameSTANDALONE, nameTEXT, nameVERSION, nameXML, nameXMLLang, nameXMLNamespace, nameXMLNSNamespace, nameXMLSpace, nameXSLPI, NONVALIDATING, OR, PERCENT, PLUS, QMARK, QUOTE, RIGHTSQB, RPAREN, SEMICOLON, SLASH, TAGEND, TAGSTART, VALIDATING, WARNING

Methods inherited from class XMLParser

getReleaseVersion(), getValidationMode(), parse(InputSource), parse(InputStream), parse(Reader), parse(String), parse(URL), setBaseURL(URL), setDoctype(DTD), setLocale(Locale), setPreserveWhitespace(boolean), setValidationMode(boolean)

Methods inherited from class java.lang.Object

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

Constructor

DOMParser()

```
public  DOMParser()
```
Creates a new parser object.

Methods

getDoctype()

```
public DTD getDoctype()
```
Get the DTD

Returns

The `DTD`

getDocument()

```
public XMLDocument getDocument()
```
Gets the document

Returns

The document being parsed

parseDTD(InputSource, String)

```
public final 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.InputSouce` to parse

`rootName` - the element to be used as root Element

Throws

XMLParseException - if syntax or other error encountered.

org.xml.sax.SAXException - Any SAX exception, possibly wrapping another exception.

`IOException` - IO Error.

parseDTD(InputStream, String)

`public final 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.

org.xml.sax.SAXException - Any SAX exception, possibly wrapping another exception.

`IOException` - IO Error.

See Also

setBaseURL(URL)

parseDTD(Reader, String)

`public final 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.

org.xml.sax.SAXException - Any SAX exception, possibly wrapping another exception.

`IOException` - IO Error.

See Also

setBaseURL(URL)

parseDTD(String, String)

`public final 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.

org.xml.sax.SAXException - Any SAX exception, possibly wrapping another exception.

`IOException` - IO Error.

parseDTD(URL, String)

`public final 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.

org.xml.sax.SAXException - Any SAX exception, possibly wrapping another exception.

`IOException` - IO Error.

setErrorStream(OutputStream)

`public final 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 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 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 void setNodeFactory(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

showWarnings(boolean)

```
public void showWarnings(boolean yes)
```

Switch to determine whether to print warnings

Parameters

`yes` - determines whether warnings should be shown

DTD

Syntax

```
public class DTD extends XMLNode implements org.w3c.dom.DocumentType,
java.io.Serializable

java.lang.Object
  |
  +--XMLNode
        |
        +--oracle.xml.parser.v2.DTD
```

All Implemented Interfaces

`java.lang.Cloneable`, org.w3c.dom.DocumentType, org.w3c.dom.Node, `java.io.Serializable`, `oracle.xml.parser.v2.XMLConstants`

Description

Implements the DOM DocumentType interface and holds the Document Type Definition information for an XML document.

Member Summary

Methods

cloneNode(boolean)	Returns a duplicate of this node, i.e., serves as a generic copy constructor for nodes.
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.
getChildNodes()	A `NodeList` that contains all children of this node.
getElementDecls()	A `NamedNodeMap` containing the element declarations in the DTD.
getEntities()	A `NamedNodeMap` containing the general entities, both external and internal, declared in the DTD.
getName()	Gets the name of the DTD; i.e., the name immediately following the `DOCTYPE` keyword.
getNotations()	A `NamedNodeMap` containing the notations declared in the DTD.
getPublicId()	Gets The public identifier associated with the DTD, if specified.

Member Summary

getSystemId()	Gets the system identifier associated with the DTD, if specified.
hasChildNodes()	This is a convenience method to allow easy determination of whether a node has any children.
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.

Inherited Member Summary

Fields inherited from class XMLNode

AMP, ASTERISK, ATTRDECL, cANY, cATTLIST, cCDATA, cCDATAEND, cCDATASTART, cCOMMENTEND, cCOMMENTSTART, cDECCREF, cDECLSTART, cDOCTYPE, cELEMENT, cEMPTY, cEMPTYTAGEND, cENCODING, cENDTAGSTART, cENTITIES, cENTITY, cFIXED, cHEXCREF, cID, cIDREF, cIDREFS, cIGNORE, cIMPLIED, cINCLUDE, cNDATA, cNMTOKEN, cNMTOKENS, cNOTATION, COLON, COMMA, cPIEND, cPISTART, cPUBLIC, cREQUIRED, cSTANDALONE, cSYSTEM, cVERSION, cXML, DOUBLEQUOTE, ELEMENTDECL, EOF, EQ, ERROR, FATAL_ERROR, FDIGIT, FLETTER, FMISCNAME, FSTARTNAME, FWHITESPACE, HASH, ICOUNT, ISTART, LEFTSQB, LPAREN, nameCDATA, nameCOMMENT, nameDOCUMENT, nameDOCUMENTFRAGMENT, nameENCODING, nameNameSpace, nameSpaceSeparator, nameSTANDALONE, nameTEXT, nameVERSION, nameXML, nameXMLLang, nameXMLNamespace, nameXMLNSNamespace, nameXMLSpace, nameXSLPI, NONVALIDATING, OR, PERCENT, PLUS, QMARK, QUOTE, RIGHTSQB, RPAREN, SEMICOLON, SLASH, TAGEND, TAGSTART, VALIDATING, WARNING

Fields inherited from interface Node

ATTRIBUTE_NODE, CDATA_SECTION_NODE, COMMENT_NODE, DOCUMENT_FRAGMENT_NODE, DOCUMENT_NODE, DOCUMENT_TYPE_NODE, ELEMENT_NODE, ENTITY_NODE, ENTITY_REFERENCE_NODE, NOTATION_NODE, PROCESSING_INSTRUCTION_NODE, TEXT_NODE

Fields inherited from interface oracle.xml.parser.v2.XMLConstants

AMP, ASTERISK, cANY, cATTLIST, cCDATA, cCDATAEND, cCDATASTART, cCOMMENTEND, cCOMMENTSTART, cDECCREF, cDECLSTART, cDOCTYPE, cELEMENT, cEMPTY, cEMPTYTAGEND, cENCODING, cENDTAGSTART, cENTITIES, cENTITY, cFIXED, cHEXCREF, cID, cIDREF, cIDREFS, cIGNORE, cIMPLIED, cINCLUDE, cNDATA, cNMTOKEN, cNMTOKENS, cNOTATION, COLON, COMMA, cPIEND, cPISTART, cPUBLIC, cREQUIRED, cSTANDALONE, cSYSTEM, cVERSION, cXML, DOUBLEQUOTE, EOF, EQ, ERROR, FATAL_ERROR, FDIGIT, FLETTER, FMISCNAME, FSTARTNAME, FWHITESPACE, HASH, ICOUNT, ISTART, LEFTSQB, LPAREN, nameCDATA, nameCOMMENT, nameDOCUMENT, nameDOCUMENTFRAGMENT, nameENCODING, nameNameSpace, nameSpaceSeparator, nameSTANDALONE, nameTEXT, nameVERSION, nameXML, nameXMLLang, nameXMLNamespace, nameXMLNSNamespace, nameXMLSpace, nameXSLPI, NONVALIDATING, OR, PERCENT, PLUS, QMARK, QUOTE, RIGHTSQB, RPAREN, SEMICOLON, SLASH, TAGEND, TAGSTART, VALIDATING, WARNING

Methods inherited from class XMLNode

Inherited Member Summary

appendChild(Node), getAttributes(), getFirstChild(), getLastChild(), getNextSibling(), getNodeName(), getNodeType(), getNodeValue(), getOwnerDocument(), getParentNode(), getPreviousSibling(), insertBefore(Node, Node), print(OutputStream), print(OutputStream, String), print(PrintWriter), removeChild(Node), replaceChild(Node, Node), selectNodes(String, NSResolver), selectSingleNode(String, NSResolver), setNodeValue(String), transformNode(XSLStylesheet), valueOf(String, NSResolver)

Methods inherited from class java.lang.Object

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

Methods inherited from interface Node

appendChild(Node), getAttributes(), getFirstChild(), getLastChild(), getNextSibling(), getNodeName(), getNodeType(), getNodeValue(), getOwnerDocument(), getParentNode(), getPreviousSibling(), insertBefore(Node, Node), removeChild(Node), replaceChild(Node, Node), setNodeValue(String)

Methods

cloneNode(boolean)

`public` org.w3c.dom.Node `cloneNode(boolean deep)`
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.

Specified By

org.w3c.dom.Node.cloneNode(boolean) in interface org.w3c.dom.Node

Overrides

cloneNode(boolean) in class XMLNode

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.

findElementDecl(String)

`public final ElementDecl findElementDecl(java.lang.String name)`
Finds an element declaration for the given tag name.

Parameters

`name` - The tag name.

Returns

the element declaration object.

findEntity(String, boolean)

`public final` org.w3c.dom.Entity `findEntity(java.lang.String n, boolean par)`
Finds a named entity in the DTD.

Parameters

`n` - The name of the entity.

Returns

the specified `Entity` object; returns null if it is not found.

findNotation(String)

`public final` org.w3c.dom.Notation `findNotation(java.lang.String name)`
Retrieves the named notation from the DTD.

Parameters

`name` - The name of the notation.

Returns

the `Notation` object; returns null if it is not found.

getChildNodes()

`public` org.w3c.dom.NodeList `getChildNodes()`
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.

Specified By

org.w3c.dom.Node.getChildNodes() in interface org.w3c.dom.Node

Overrides

getChildNodes() in class XMLNode

Returns

The children of this node

getElementDecls()

public org.w3c.dom.NamedNodeMap getElementDecls()
A NamedNodeMap containing the element declarations in the DTD. Every node in this map is an ElementDecl object.

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()

public org.w3c.dom.NamedNodeMap getEntities()
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.

Specified By

org.w3c.dom.DocumentType.getEntities() in interface org.w3c.dom.DocumentType

Returns

The entities declared in the DTD

getName()

```
public java.lang.String getName()
```
Gets the name of the DTD; i.e., the name immediately following the `DOCTYPE` keyword.

Specified By

org.w3c.dom.DocumentType.getName() in interface org.w3c.dom.DocumentType

Returns

Name of the DTD

getNotations()

```
public org.w3c.dom.NamedNodeMap getNotations()
```
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.

Specified By

org.w3c.dom.DocumentType.getNotations() in interface org.w3c.dom.DocumentType

Returns

The notations declared in the DTD

getPublicId()

```
public java.lang.String getPublicId()
```
Gets The public identifier associated with the DTD, if specified. If the public identifier was not specified, this is `null`.

Returns

the public identifier associated with the DTD

getSystemId()

```
public java.lang.String getSystemId()
```
Gets the system identifier associated with the DTD, if specified. If the system identifier was not specified, this is `null`.

Returns

the system identifier associated with the DTD

hasChildNodes()

```
public boolean 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

Specified By

org.w3c.dom.Node.hasChildNodes() in interface org.w3c.dom.Node

Overrides

hasChildNodes() in class XMLNode

Returns

`false` as DTD node can not have any children,

printExternalDTD(OutputStream)

```
public void printExternalDTD(java.io.OutputStream out)
```

Writes the contents of this document to the given output stream.

Parameters

`out` - `OutputStream` to write to

Throws

`IOException` - if an error occurs

printExternalDTD(OutputStream, String)

```
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)

`public void printExternalDTD(java.io.PrintWriter out)`
Writes the contents of this document to the given output stream.

Parameters

`out` - `PrintWriter` to write to

Throws

`IOException` - if an error occurs

ElementDecl

Syntax

```
public class ElementDecl extends XMLNode implements java.io.Serializable

java.lang.Object
  |
  +--XMLNode
        |
        +--oracle.xml.parser.v2.ElementDecl
```

All Implemented Interfaces

`java.lang.Cloneable, org.w3c.dom.Node, java.io.Serializable, oracle.xml.parser.v2.XMLConstants`

Description

This class represents an element declaration in a DTD.

Member Summary

Fields	
ANY	Element content type - Children can be any element
ASTERISK	ContentModelParseTreeNode type - "*" node (has one children)
COMMA	ContentModelParseTreeNode type - "," node (has two children)
ELEMENT	ContentModelParseTreeNode type - 'leaf' node (has no children)
ELEMENTS	Element content type - Children can be elements as per Content Model
EMPTY	Element content type - No Children
MIXED	Element content type - Children can be PCDATA & elements as per Content Model
OR	ContentModelParseTreeNode type - "|" node (has two children)
PLUS	ContentModelParseTreeNode type - "+" node (has one children)
QMARK	ContentModelParseTreeNode type - "?" node (has one children)
Methods	
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

Member Summary	
getAttrDecls()	Gets an enumeration of attribute declarations
getContentElements()	Returns Vector of elements that can be appended to this element
getContentType()	Returns content model of element
getParseTree()	Returns the root node of Content Model Parse Tree.
validateContent(Element)	Validates the content of a element node.

Inherited Member Summary

Fields inherited from class XMLNode

AMP, ATTRDECL, cANY, cATTLIST, cCDATA, cCDATAEND, cCDATASTART, cCOMMENTEND, cCOMMENTSTART, cDECCREF, cDECLSTART, cDOCTYPE, cELEMENT, cEMPTY, cEMPTYTAGEND, cENCODING, cENDTAGSTART, cENTITIES, cENTITY, cFIXED, cHEXCREF, cID, cIDREF, cIDREFS, cIGNORE, cIMPLIED, cINCLUDE, cNDATA, cNMTOKEN, cNMTOKENS, cNOTATION, COLON, cPIEND, cPISTART, cPUBLIC, cREQUIRED, cSTANDALONE, cSYSTEM, cVERSION, cXML, DOUBLEQUOTE, ELEMENTDECL, EOF, EQ, ERROR, FATAL_ERROR, FDIGIT, FLETTER, FMISCNAME, FSTARTNAME, FWHITESPACE, HASH, ICOUNT, ISTART, LEFTSQB, LPAREN, nameCDATA, nameCOMMENT, nameDOCUMENT, nameDOCUMENTFRAGMENT, nameENCODING, nameNameSpace, nameSpaceSeparator, nameSTANDALONE, nameTEXT, nameVERSION, nameXML, nameXMLLang, nameXMLNamespace, nameXMLNSNamespace, nameXMLSpace, nameXSLPI, NONVALIDATING, PERCENT, QUOTE, RIGHTSQB, RPAREN, SEMICOLON, SLASH, TAGEND, TAGSTART, VALIDATING, WARNING

Fields inherited from interface Node

ATTRIBUTE_NODE, CDATA_SECTION_NODE, COMMENT_NODE, DOCUMENT_FRAGMENT_NODE, DOCUMENT_NODE, DOCUMENT_TYPE_NODE, ELEMENT_NODE, ENTITY_NODE, ENTITY_REFERENCE_NODE, NOTATION_NODE, PROCESSING_INSTRUCTION_NODE, TEXT_NODE

Fields inherited from interface oracle.xml.parser.v2.XMLConstants

AMP, cANY, cATTLIST, cCDATA, cCDATAEND, cCDATASTART, cCOMMENTEND, cCOMMENTSTART, cDECCREF, cDECLSTART, cDOCTYPE, cELEMENT, cEMPTY, cEMPTYTAGEND, cENCODING, cENDTAGSTART, cENTITIES, cENTITY, cFIXED, cHEXCREF, cID, cIDREF, cIDREFS, cIGNORE, cIMPLIED, cINCLUDE, cNDATA, cNMTOKEN, cNMTOKENS, cNOTATION, COLON, cPIEND, cPISTART, cPUBLIC, cREQUIRED, cSTANDALONE, cSYSTEM, cVERSION, cXML, DOUBLEQUOTE, EOF, EQ, ERROR, FATAL_ERROR, FDIGIT, FLETTER, FMISCNAME, FSTARTNAME, FWHITESPACE, HASH, ICOUNT, ISTART, LEFTSQB, LPAREN, nameCDATA, nameCOMMENT, nameDOCUMENT, nameDOCUMENTFRAGMENT, nameENCODING, nameNameSpace, nameSpaceSeparator, nameSTANDALONE, nameTEXT, nameVERSION, nameXML, nameXMLLang, nameXMLNamespace, nameXMLNSNamespace, nameXMLSpace, nameXSLPI, NONVALIDATING, PERCENT, QUOTE, RIGHTSQB, RPAREN, SEMICOLON, SLASH, TAGEND, TAGSTART, VALIDATING, WARNING

Methods inherited from class XMLNode

Inherited Member Summary

appendChild(Node), cloneNode(boolean), getAttributes(), getChildNodes(), getFirstChild(), getLastChild(), getNextSibling(), getNodeName(), getNodeType(), getNodeValue(), getOwnerDocument(), getParentNode(), getPreviousSibling(), hasChildNodes(), insertBefore(Node, Node), print(OutputStream), print(OutputStream, String), print(PrintWriter), removeChild(Node), replaceChild(Node, Node), selectNodes(String, NSResolver), selectSingleNode(String, NSResolver), setNodeValue(String), transformNode(XSLStylesheet), valueOf(String, NSResolver)

Methods inherited from class java.lang.Object

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

Methods inherited from interface Node

appendChild(Node), cloneNode(boolean), getAttributes(), getChildNodes(), getFirstChild(), getLastChild(), getNextSibling(), getNodeName(), getNodeType(), getNodeValue(), getOwnerDocument(), getParentNode(), getPreviousSibling(), hasChildNodes(), insertBefore(Node, Node), removeChild(Node), replaceChild(Node, Node), setNodeValue(String)

Fields

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)

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

MIXED

```
public static final byte MIXED
```
Element content type - Children can be PCDATA & elements as per Content Model

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)

Methods

expectedElements(Element)

```
public java.util.Vector expectedElements(org.w3c.dom.Element e)
```
Returns vector of element names that can be appended to the element.

Parameters

e - Element

Returns

Vector of names

findAttrDecl(String)

```
public final AttrDecl findAttrDecl(java.lang.String name)
```
Gets an attribute declaration object or null if not found

Parameters

`name` - Attribute declaration to find

Returns

The `AttrDecl` object, or null, if it was not found

getAttrDecls()

`public` org.w3c.dom.NamedNodeMap `getAttrDecls()`
Gets an enumeration of attribute declarations

Returns

An enumeration of attribute declarations

getContentElements()

`public final java.util.Vector getContentElements()`
Returns Vector of elements that can be appended to this element

Returns

The `Vector` containing the element names.

getContentType()

`public int getContentType()`
Returns content model of element

Returns

The `type` of the element declaration.

getParseTree()

`public final` org.w3c.dom.Node `getParseTree()`
Returns the root node of Content Model Parse Tree. `Node.getFirstChild()` and `Node.getLastChild()` return the the parse tree branches. `Node.getNodeType()` and `Node.getNodeName()` return the the parse tree node type and name.

Returns

The `Node` containing the Content Model parse tree root node.

validateContent(Element)

```
public boolean validateContent(org.w3c.dom.Element e)
```

Validates the content of a element node.

Returns

True if valid, else false

NodeFactory

Syntax

```
public class NodeFactory extends java.lang.Object implements
java.io.Serializable

java.lang.Object
  |
  +--oracle.xml.parser.v2.NodeFactory
```

All Implemented Interfaces

java.io.Serializable

Description

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.

See Also

setNodeFactory(NodeFactory)

Member Summary

Constructors	
NodeFactory()	
Methods	
createAttribute(String, String)	Creates an attribute node with the specified tag, and text.
createCDATASection(String)	Creates a CDATA node with the specified text.
createComment(String)	Creates a comment node with the specified text.
createDocument()	Creates a document node.
createElement(String)	Creates an Element node with the specified tag.
createProcessingInstruction(String, String)	Creates a PI node with the specified tag, and text.

Member Summary	
createTextNode(String)	Creates a text node with the specified text.

Inherited Member Summary
Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor

NodeFactory()

```
public  NodeFactory()
```

Methods

createAttribute(String, String)

```
public XMLAttr createAttribute(java.lang.String tag, java.lang.String text)
```
Creates an attribute node with the specified tag, and text.

Parameters

`tag` - The name of the node.

`text` - The text associated with the node.

Returns

The created attribute node.

createCDATASection(String)

```
public XMLCDATA createCDATASection(java.lang.String text)
```
Creates a CDATA node with the specified text.

Parameters

`text` - The text associated with the node.

Returns

The created CDATA node.

createComment(String)

```
public XMLComment createComment(java.lang.String text)
```

Creates a comment node with the specified text.

Parameters

`text` - The text associated with the node.

Returns

The created comment node.

createDocument()

```
public XMLDocument createDocument()
```

Creates a document node. This method cannot return a null pointer.

Returns

The created element.

createElement(String)

```
public XMLElement createElement(java.lang.String tag)
```

Creates an Element node with the specified tag.

Parameters

`tag` - The name of the element.

Returns

The created element.

createProcessingInstruction(String, String)

```
public XMLPI createProcessingInstruction(java.lang.String tag, java.lang.String
text)
```

Creates a PI node with the specified tag, and text.

Parameters

`tag` - The name of the node.

`text` - The text associated with the node.

Returns

The created PI node.

createTextNode(String)

```
public XMLText createTextNode(java.lang.String text)
```

Creates a text node with the specified text.

Parameters

`text` - The text associated with the node.

Returns

The created text node.

NSName

Syntax

```
public interface NSName
```

All Known Implementing Classes:

```
XMLAttr, XMLElement
```

Description

This interface provides Namespace support for Element and Attr names

Member Summary	
Methods	
getExpandedName()	Get the fully resolved name for this name
getLocalName()	Get the local name for this name
getNamespace()	Get the resolved Namespace for this name
getPrefix()	Get the prefix for this name
getQualifiedName()	Get the qualified name

Methods

getExpandedName()

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

Get the fully resolved name for this name

Returns

The fully resolved name

getLocalName()

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

Get the local name for this name

Returns

The local name

getNamespace()

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

Get the resolved Namespace for this name

Returns

The resolved Namespace

getPrefix()

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

Get the prefix for this name

Returns

The prefix

getQualifiedName()

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

Get the qualified name

Returns

The qualified name

NSResolver

Syntax

```
public interface NSResolver
```

All Known Implementing Classes

```
XMLElement
```

Description

This interface provides support for resolving Namespaces

Member Summary

Methods	
resolveNamespacePrefix(String)	Find the namespace definition in scope for a given namespace prefix

Methods

resolveNamespacePrefix(String)

```
public java.lang.String resolveNamespacePrefix(java.lang.String prefix)
```

Find the namespace definition in scope for a given namespace prefix

Parameters

`prefix` - Namespace prefix to be resolved

Returns

the resolved Namespace (null, if prefix could not be resolved)

oraxsl

Syntax

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

java.lang.Object
  |
  +--oracle.xml.parser.v2.oraxsl
```

Description

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. The following is its invocation syntax:

```
java oraxsl options* source? stylesheet? result?
     -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
```

Member Summary

Constructors

oraxsl()

Methods

main(String[]) Invokes the oraxsl driver

Inherited Member Summary

Methods inherited from class java.lang.Object

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

Constructor

oraxsl()

```
public  oraxsl()
```

Methods

main(String[])

```
public static void main(java.lang.String[] args)
```
Invokes the oraxsl driver

Parameters

`args` - command line arguments

SAXAttrList

Syntax

public class SAXAttrList extends java.lang.Object implements org.xml.sax.AttributeList

```
java.lang.Object
  |
  +--oracle.xml.parser.v2.SAXAttrList
```

All Implemented Interfaces

org.xml.sax.AttributeList

Description

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.

Member Summary

Methods	
getExpandedName(int)	Get the expanded name for an attribute in the list (by position)
getLength()	Return the number of attributes in this list.
getLocalName(int)	Get the local name for an attribute in the list (by position)
getName(int)	Return the name of an attribute in this list (by position).
getNamespace(int)	Get the resolved namespace for an attribute in the list (by position)
getPrefix(int)	Get the namespace prefix for an attribute in the list (by position)
getQualifiedName(int)	Get the qualified name for an attribute in the list (by position)
getType(int)	
getType(String)	Return the type of an attribute in the list (by name).
getValue(int)	Return the value of an attribute in the list (by position).
getValue(String)	Return the value of an attribute in the list (by name).

Inherited Member Summary

Methods inherited from class java.lang.Object

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

Methods

getExpandedName(int)

```
public java.lang.String getExpandedName(int i)
```
Get the expanded name for an attribute in the list (by position)

Parameters

`i` - The index of the attribute in the list.

Returns

The expanded name for the attribute

getLength()

```
public int getLength()
```
Return the number of attributes in this list.

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.

Specified By

org.xml.sax.AttributeList.getLength() in interface org.xml.sax.AttributeList

Returns

The number of attributes in the list.

getLocalName(int)

```
public java.lang.String getLocalName(int i)
```
Get the local name for an attribute in the list (by position)

Parameters

`i` - The index of the attribute in the list.

Returns

The local name for the attribute

getName(int)

`public java.lang.String getName(int i)`
Return the name of an attribute in this list (by position).

The names must be unique the SAX parser shall not include the same attribute twice. Attributes without values (those declared #IMPLIED without a value specified in the start tag) will be omitted from the list.

If the attribute name has a namespace prefix, the prefix will still be attached.

Specified By

org.xml.sax.AttributeList.getName(int) in interface org.xml.sax.AttributeList

Parameters

`i` - The index of the attribute in the list (starting at 0).

Returns

The name of the indexed attribute, or null if the index is out of range.

See Also

getLength()

getNamespace(int)

`public java.lang.String getNamespace(int i)`
Get the resolved namespace for an attribute in the list (by position)

Parameters

`i` - The index of the attribute in the list.

Returns

The resolved namespace for the attribute

getPrefix(int)

```
public java.lang.String getPrefix(int i)
```
Get the namespace prefix for an attribute in the list (by position)

Parameters

`i` - The index of the attribute in the list.

Returns

The namespace prefix for the attribute

getQualifiedName(int)

```
public java.lang.String getQualifiedName(int i)
```
Get the qualified name for an attribute in the list (by position)

Parameters

`i` - The index of the attribute in the list.

Returns

The qualified name for the attribute

getType(int)

```
public java.lang.String getType(int i)
```

Specified By

org.xml.sax.AttributeList.getType(int) in interface org.xml.sax.AttributeList

getType(String)

```
public java.lang.String getType(java.lang.String s)
```
Return the type of an attribute in the list (by name).

The return value is the same as the return value for getType(int).

If the attribute name has a namespace prefix in the document, the application must include the prefix here.

Specified By

org.xml.sax.AttributeList.getType(String) in interface org.xml.sax.AttributeList

Parameters

name - The name of the attribute.

Returns

The attribute type as a string, or null if no such attribute exists.

See Also

getType(int)

getValue(int)

```
public java.lang.String getValue(int i)
```

Return the value of an attribute in the list (by position).

If the attribute value is a list of tokens (IDREFS, ENTITIES, or NMTOKENS), the tokens will be concatenated into a single string separated by whitespace.

Specified By

org.xml.sax.AttributeList.getValue(int) in interface org.xml.sax.AttributeList

Parameters

i - The index of the attribute in the list (starting at 0).

Returns

The attribute value as a string, or null if the index is out of range.

See Also

getLength(), getValue(String)

getValue(String)

```
public java.lang.String getValue(java.lang.String s)
```

Return the value of an attribute in the list (by name).

The return value is the same as the return value for getValue(int).

If the attribute name has a namespace prefix in the document, the application must include the prefix here.

Specified By

org.xml.sax.AttributeList.getValue(String) in interface org.xml.sax.AttributeList

Parameters

`i` - The index of the attribute in the list.

Returns

The attribute value as a string, or null if no such attribute exists.

See Also

getValue(int)

SAXParser

Syntax

```
public class SAXParser extends XMLParser implements org.xml.sax.Parser,
oracle.xml.parser.v2.XMLConstants

java.lang.Object
  |
  +--XMLParser
        |
        +--oracle.xml.parser.v2.SAXParser
```

All Implemented Interfaces

org.xml.sax.Parser, oracle.xml.parser.v2.XMLConstants

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.

Member Summary

Constructors	
SAXParser()	Creates a new parser object.
Methods	
setDocumentHandler(DocumentHandler)	SAX applications can use this to register a new document event handler.
setDTDHandler(DTDHandler)	SAX applications can use this to register a new DTD event handler.
setEntityResolver(EntityResolver)	SAX applications can use this to register a new entity resolver
setErrorHandler(ErrorHandler)	SAX applications can use this to register a new error event handler.

Inherited Member Summary

Fields inherited from class XMLParser

Inherited Member Summary

AMP, ASTERISK, cANY, cATTLIST, cCDATA, cCDATAEND, cCDATASTART, cCOMMENTEND, cCOMMENTSTART, cDECCREF, cDECLSTART, cDOCTYPE, cELEMENT, cEMPTY, cEMPTYTAGEND, cENCODING, cENDTAGSTART, cENTITIES, cENTITY, cFIXED, cHEXCREF, cID, cIDREF, cIDREFS, cIGNORE, cIMPLIED, cINCLUDE, cNDATA, cNMTOKEN, cNMTOKENS, cNOTATION, COLON, COMMA, cPIEND, cPISTART, cPUBLIC, cREQUIRED, cSTANDALONE, cSYSTEM, cVERSION, cXML, DOUBLEQUOTE, EOF, EQ, ERROR, FATAL_ERROR, FDIGIT, FLETTER, FMISCNAME, FSTARTNAME, FWHITESPACE, HASH, ICOUNT, ISTART, LEFTSQB, LPAREN, nameCDATA, nameCOMMENT, nameDOCUMENT, nameDOCUMENTFRAGMENT, nameENCODING, nameNameSpace, nameSpaceSeparator, nameSTANDALONE, nameTEXT, nameVERSION, nameXML, nameXMLLang, nameXMLNamespace, nameXMLNSNamespace, nameXMLSpace, nameXSLPI, NONVALIDATING, OR, PERCENT, PLUS, QMARK, QUOTE, RIGHTSQB, RPAREN, SEMICOLON, SLASH, TAGEND, TAGSTART, VALIDATING, WARNING

Fields inherited from interface oracle.xml.parser.v2.XMLConstants

AMP, ASTERISK, cANY, cATTLIST, cCDATA, cCDATAEND, cCDATASTART, cCOMMENTEND, cCOMMENTSTART, cDECCREF, cDECLSTART, cDOCTYPE, cELEMENT, cEMPTY, cEMPTYTAGEND, cENCODING, cENDTAGSTART, cENTITIES, cENTITY, cFIXED, cHEXCREF, cID, cIDREF, cIDREFS, cIGNORE, cIMPLIED, cINCLUDE, cNDATA, cNMTOKEN, cNMTOKENS, cNOTATION, COLON, COMMA, cPIEND, cPISTART, cPUBLIC, cREQUIRED, cSTANDALONE, cSYSTEM, cVERSION, cXML, DOUBLEQUOTE, EOF, EQ, ERROR, FATAL_ERROR, FDIGIT, FLETTER, FMISCNAME, FSTARTNAME, FWHITESPACE, HASH, ICOUNT, ISTART, LEFTSQB, LPAREN, nameCDATA, nameCOMMENT, nameDOCUMENT, nameDOCUMENTFRAGMENT, nameENCODING, nameNameSpace, nameSpaceSeparator, nameSTANDALONE, nameTEXT, nameVERSION, nameXML, nameXMLLang, nameXMLNamespace, nameXMLNSNamespace, nameXMLSpace, nameXSLPI, NONVALIDATING, OR, PERCENT, PLUS, QMARK, QUOTE, RIGHTSQB, RPAREN, SEMICOLON, SLASH, TAGEND, TAGSTART, VALIDATING, WARNING

Methods inherited from class XMLParser

getReleaseVersion(), getValidationMode(), parse(InputSource), parse(InputStream), parse(Reader), parse(String), parse(URL), setBaseURL(URL), setDoctype(DTD), setLocale(Locale), setPreserveWhitespace(boolean), setValidationMode(boolean)

Methods inherited from class java.lang.Object

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

Methods inherited from interface Parser

parse(InputSource), parse(String), setLocale(Locale)

Constructor

SAXParser()

```
public SAXParser()
```

Creates a new parser object.

Methods

setDocumentHandler(DocumentHandler)

```
public void setDocumentHandler(org.xml.sax.DocumentHandler handler)
```

SAX applications can use this to register a new document event handler.

Specified By

org.xml.sax.Parser.setDocumentHandler(DocumentHandler) in interface org.xml.sax.Parser

Parameters

`handler` - `DocumentHandler` being registered

See Also

org.xml.sax.Parser.setDocumentHandler(DocumentHandler), org.xml.sax.DocumentHandler

setDTDHandler(DTDHandler)

`public void setDTDHandler(`org.xml.sax.DTDHandler `handler)`
SAX applications can use this to register a new DTD event handler.

Specified By

org.xml.sax.Parser.setDTDHandler(DTDHandler) in interface org.xml.sax.Parser

Parameters

`handler` - `DTDHandler` being registered

See Also

org.xml.sax.Parser.setDTDHandler(DTDHandler), org.xml.sax.DTDHandler

setEntityResolver(EntityResolver)

`public void setEntityResolver(`org.xml.sax.EntityResolver `resolver)`
SAX applications can use this to register a new entity resolver

Specified By

org.xml.sax.Parser.setEntityResolver(EntityResolver) in interface org.xml.sax.Parser

Parameters

`resolver` - `EntityResolver` being registered

See Also

org.xml.sax.Parser.setEntityResolver(EntityResolver), org.xml.sax.DTDHandler

setErrorHandler(ErrorHandler)

`public void setErrorHandler(`org.xml.sax.ErrorHandler `handler)`
SAX applications can use this to register a new error event handler. This replaces any previous setting for error handling.

Specified By

org.xml.sax.Parser.setErrorHandler(ErrorHandler) in interface org.xml.sax.Parser

Parameters

`handler` - `ErrorHandler` being registered

See Also

org.xml.sax.Parser.setErrorHandler(ErrorHandler), org.xml.sax.ErrorHandler

XMLAttr

Syntax

public class XMLAttr extends XMLNode implements org.w3c.dom.Attr, NSName, java.io.Serializable

```
java.lang.Object
  |
  +--XMLNode
        |
        +--oracle.xml.parser.v2.XMLAttr
```

All Implemented Interfaces

org.w3c.dom.Attr, java.lang.Cloneable, org.w3c.dom.Node, NSName, java.io.Serializable, oracle.xml.parser.v2.XMLConstants

Description

This class implements the DOM Attr interface and holds information on each attribute of an element.

See Also

org.w3c.dom.Attr, NodeFactory, setNodeFactory(NodeFactory)

Member Summary

Constructors	
XMLAttr(String, String)	Construct attribute with given name and value.
XMLAttr(String, String, String, String)	Namespace support
Methods	
cloneNode(boolean)	Returns a duplicate of this node, i.e., serves as a generic copy constructor for nodes.
getExpandedName()	Get the fully resolved Name for this attribute
getLocalName()	Get the local Name for this attribute
getName()	Gets the attribute name.
getNamespace()	Get the resolved Namespace for this attribute
getNodeValue()	Gets the value of this node, depending on its type

Member Summary	
getParentNode()	Gets the parent of this node.
getPrefix()	Get the namespace prefix for this attribute
getQualifiedName()	Gets the qualified name for this attribute
getSpecified()	Returns true if the attribute was specified explicity in the element
getValue()	Gets the attribute value.
setNodeValue(String)	Sets the value of this node, depending on its type
setValue(String)	Sets the value.

Inherited Member Summary

Fields inherited from class XMLNode

AMP, ASTERISK, ATTRDECL, cANY, cATTLIST, cCDATA, cCDATAEND, cCDATASTART, cCOMMENTEND, cCOMMENTSTART, cDECCREF, cDECLSTART, cDOCTYPE, cELEMENT, cEMPTY, cEMPTYTAGEND, cENCODING, cENDTAGSTART, cENTITIES, cENTITY, cFIXED, cHEXCREF, cID, cIDREF, cIDREFS, cIGNORE, cIMPLIED, cINCLUDE, cNDATA, cNMTOKEN, cNMTOKENS, cNOTATION, COLON, COMMA, cPIEND, cPISTART, cPUBLIC, cREQUIRED, cSTANDALONE, cSYSTEM, cVERSION, cXML, DOUBLEQUOTE, ELEMENTDECL, EOF, EQ, ERROR, FATAL_ERROR, FDIGIT, FLETTER, FMISCNAME, FSTARTNAME, FWHITESPACE, HASH, ICOUNT, ISTART, LEFTSQB, LPAREN, nameCDATA, nameCOMMENT, nameDOCUMENT, nameDOCUMENTFRAGMENT, nameENCODING, nameNameSpace, nameSpaceSeparator, nameSTANDALONE, nameTEXT, nameVERSION, nameXML, nameXMLLang, nameXMLNamespace, nameXMLNSNamespace, nameXMLSpace, nameXSLPI, NONVALIDATING, OR, PERCENT, PLUS, QMARK, QUOTE, RIGHTSQB, RPAREN, SEMICOLON, SLASH, TAGEND, TAGSTART, VALIDATING, WARNING

Fields inherited from interface Node

ATTRIBUTE_NODE, CDATA_SECTION_NODE, COMMENT_NODE, DOCUMENT_FRAGMENT_NODE, DOCUMENT_NODE, DOCUMENT_TYPE_NODE, ELEMENT_NODE, ENTITY_NODE, ENTITY_REFERENCE_NODE, NOTATION_NODE, PROCESSING_INSTRUCTION_NODE, TEXT_NODE

Fields inherited from interface oracle.xml.parser.v2.XMLConstants

AMP, ASTERISK, cANY, cATTLIST, cCDATA, cCDATAEND, cCDATASTART, cCOMMENTEND, cCOMMENTSTART, cDECCREF, cDECLSTART, cDOCTYPE, cELEMENT, cEMPTY, cEMPTYTAGEND, cENCODING, cENDTAGSTART, cENTITIES, cENTITY, cFIXED, cHEXCREF, cID, cIDREF, cIDREFS, cIGNORE, cIMPLIED, cINCLUDE, cNDATA, cNMTOKEN, cNMTOKENS, cNOTATION, COLON, COMMA, cPIEND, cPISTART, cPUBLIC, cREQUIRED, cSTANDALONE, cSYSTEM, cVERSION, cXML, DOUBLEQUOTE, EOF, EQ, ERROR, FATAL_ERROR, FDIGIT, FLETTER, FMISCNAME, FSTARTNAME, FWHITESPACE, HASH, ICOUNT, ISTART, LEFTSQB, LPAREN, nameCDATA, nameCOMMENT, nameDOCUMENT, nameDOCUMENTFRAGMENT, nameENCODING, nameNameSpace, nameSpaceSeparator, nameSTANDALONE, nameTEXT, nameVERSION, nameXML, nameXMLLang, nameXMLNamespace, nameXMLNSNamespace, nameXMLSpace, nameXSLPI, NONVALIDATING, OR, PERCENT, PLUS, QMARK, QUOTE, RIGHTSQB, RPAREN, SEMICOLON, SLASH, TAGEND, TAGSTART, VALIDATING, WARNING

Methods inherited from class XMLNode

Inherited Member Summary

appendChild(Node), getAttributes(), getChildNodes(), getFirstChild(), getLastChild(), getNextSibling(), getNodeName(), getNodeType(), getOwnerDocument(), getPreviousSibling(), hasChildNodes(), insertBefore(Node, Node), print(OutputStream), print(OutputStream, String), print(PrintWriter), removeChild(Node), replaceChild(Node, Node), selectNodes(String, NSResolver), selectSingleNode(String, NSResolver), transformNode(XSLStylesheet), valueOf(String, NSResolver)

Methods inherited from class java.lang.Object

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

Methods inherited from interface Node

appendChild(Node), getAttributes(), getChildNodes(), getFirstChild(), getLastChild(), getNextSibling(), getNodeName(), getNodeType(), getOwnerDocument(), getPreviousSibling(), hasChildNodes(), insertBefore(Node, Node), removeChild(Node), replaceChild(Node, Node)

Constructor

XMLAttr(String, String)

`public XMLAttr(java.lang.String n, java.lang.String v)`
Construct attribute with given name and value.

Parameters

`n` - Name of the attribute

`v` - Value of the attribute

XMLAttr(String, String, String, String)

`public XMLAttr(java.lang.String name, java.lang.String prefix, java.lang.String ns, java.lang.String v)`
Namespace support

Methods

cloneNode(boolean)

`public` org.w3c.dom.Node `cloneNode(boolean deep)`
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.

Specified By

org.w3c.dom.Node.cloneNode(boolean) in interface org.w3c.dom.Node

Overrides

cloneNode(boolean) in class XMLNode

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()

`public java.lang.String getExpandedName()`
Get the fully resolved Name for this attribute

Specified By

getExpandedName() in interface NSName

Returns

the fully resolved Name

getLocalName()

`public java.lang.String getLocalName()`
Get the local Name for this attribute

Specified By

getLocalName() in interface NSName

Returns

the local Name

getName()

```
public java.lang.String getName()
```
Gets the attribute name.

Specified By

org.w3c.dom.Attr.getName() in interface org.w3c.dom.Attr

Returns

attribute name

getNamespace()

```
public java.lang.String getNamespace()
```
Get the resolved Namespace for this attribute

Specified By

getNamespace() in interface NSName

Returns

the resolved Namespace

getNodeValue()

```
public java.lang.String getNodeValue()
```
Gets the value of this node, depending on its type

Specified By

org.w3c.dom.Node.getNodeValue() in interface org.w3c.dom.Node

Overrides

getNodeValue() in class XMLNode

Returns

Value of this node

Throws

org.w3c.dom.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.

getParentNode()

`public` org.w3c.dom.Node `getParentNode()`
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`.

Specified By

org.w3c.dom.Node.getParentNode() in interface org.w3c.dom.Node

Overrides

getParentNode() in class XMLNode

Returns

The parent of this node

getPrefix()

`public java.lang.String getPrefix()`
Get the namespace prefix for this attribute

Specified By

getPrefix() in interface NSName

Returns

the namespace prefix

getQualifiedName()

`public java.lang.String getQualifiedName()`
Gets the qualified name for this attribute

Specified By

getQualifiedName() in interface NSName

Returns

the qualified name

getSpecified()

`public boolean getSpecified()`
Returns true if the attribute was specified explicity in the element

Specified By

org.w3c.dom.Attr.getSpecified() in interface org.w3c.dom.Attr

Returns

`true`, if the attribute was specified explicitly, `false`, if it was not

getValue()

`public java.lang.String getValue()`
Gets the attribute value.

Specified By

org.w3c.dom.Attr.getValue() in interface org.w3c.dom.Attr

Returns

attribute value

setNodeValue(String)

`public void setNodeValue(java.lang.String nodeValue)`
Sets the value of this node, depending on its type

Specified By

org.w3c.dom.Node.setNodeValue(String) in interface org.w3c.dom.Node

Overrides

setNodeValue(String) in class XMLNode

Throws

org.w3c.dom.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)

`public void setValue(java.lang.String arg)`
Sets the value.

Specified By

org.w3c.dom.Attr.setValue(String) in interface org.w3c.dom.Attr

Parameters

`arg` - Value to set

XMLCDATA

Syntax

public class XMLCDATA extends XMLText implements org.w3c.dom.CDATASection, java.io.Serializable

```
java.lang.Object
  |
  +--XMLNode
        |
        +--oracle.xml.parser.v2.CharData
              |
              +--XMLText
                    |
                    +--oracle.xml.parser.v2.XMLCDATA
```

All Implemented Interfaces

org.w3c.dom.CDATASection, org.w3c.dom.CharacterData, java.lang.Cloneable, org.w3c.dom.Node, java.io.Serializable, org.w3c.dom.Text, oracle.xml.parser.v2.XMLConstants

Description

This class implements the DOM CDATASection interface.

See Also

org.w3c.dom.CDATASection, NodeFactory, setNodeFactory(NodeFactory)

Member Summary

Constructors

XMLCDATA(String)	Creates a CDATA node having the given name and text.

Inherited Member Summary

Fields inherited from class XMLNode

Inherited Member Summary

AMP, ASTERISK, ATTRDECL, cANY, cATTLIST, cCDATA, cCDATAEND, cCDATASTART, cCOMMENTEND, cCOMMENTSTART, cDECCREF, cDECLSTART, cDOCTYPE, cELEMENT, cEMPTY, cEMPTYTAGEND, cENCODING, cENDTAGSTART, cENTITIES, cENTITY, cFIXED, cHEXCREF, cID, cIDREF, cIDREFS, cIGNORE, cIMPLIED, cINCLUDE, cNDATA, cNMTOKEN, cNMTOKENS, cNOTATION, COLON, COMMA, cPIEND, cPISTART, cPUBLIC, cREQUIRED, cSTANDALONE, cSYSTEM, cVERSION, cXML, DOUBLEQUOTE, ELEMENTDECL, EOF, EQ, ERROR, FATAL_ERROR, FDIGIT, FLETTER, FMISCNAME, FSTARTNAME, FWHITESPACE, HASH, ICOUNT, ISTART, LEFTSQB, LPAREN, nameCDATA, nameCOMMENT, nameDOCUMENT, nameDOCUMENTFRAGMENT, nameENCODING, nameNameSpace, nameSpaceSeparator, nameSTANDALONE, nameTEXT, nameVERSION, nameXML, nameXMLLang, nameXMLNamespace, nameXMLNSNamespace, nameXMLSpace, nameXSLPI, NONVALIDATING, OR, PERCENT, PLUS, QMARK, QUOTE, RIGHTSQB, RPAREN, SEMICOLON, SLASH, TAGEND, TAGSTART, VALIDATING, WARNING

Fields inherited from interface Node

ATTRIBUTE_NODE, CDATA_SECTION_NODE, COMMENT_NODE, DOCUMENT_FRAGMENT_NODE, DOCUMENT_NODE, DOCUMENT_TYPE_NODE, ELEMENT_NODE, ENTITY_NODE, ENTITY_REFERENCE_NODE, NOTATION_NODE, PROCESSING_INSTRUCTION_NODE, TEXT_NODE

Fields inherited from interface oracle.xml.parser.v2.XMLConstants

AMP, ASTERISK, cANY, cATTLIST, cCDATA, cCDATAEND, cCDATASTART, cCOMMENTEND, cCOMMENTSTART, cDECCREF, cDECLSTART, cDOCTYPE, cELEMENT, cEMPTY, cEMPTYTAGEND, cENCODING, cENDTAGSTART, cENTITIES, cENTITY, cFIXED, cHEXCREF, cID, cIDREF, cIDREFS, cIGNORE, cIMPLIED, cINCLUDE, cNDATA, cNMTOKEN, cNMTOKENS, cNOTATION, COLON, COMMA, cPIEND, cPISTART, cPUBLIC, cREQUIRED, cSTANDALONE, cSYSTEM, cVERSION, cXML, DOUBLEQUOTE, EOF, EQ, ERROR, FATAL_ERROR, FDIGIT, FLETTER, FMISCNAME, FSTARTNAME, FWHITESPACE, HASH, ICOUNT, ISTART, LEFTSQB, LPAREN, nameCDATA, nameCOMMENT, nameDOCUMENT, nameDOCUMENTFRAGMENT, nameENCODING, nameNameSpace, nameSpaceSeparator, nameSTANDALONE, nameTEXT, nameVERSION, nameXML, nameXMLLang, nameXMLNamespace, nameXMLNSNamespace, nameXMLSpace, nameXSLPI, NONVALIDATING, OR, PERCENT, PLUS, QMARK, QUOTE, RIGHTSQB, RPAREN, SEMICOLON, SLASH, TAGEND, TAGSTART, VALIDATING, WARNING

Methods inherited from class XMLText

appendData, deleteData, getData, getLength, getNodeValue(), insertData, replaceData, setData, setNodeValue, splitText(int), substringData

Methods inherited from class oracle.xml.parser.v2.CharData

appendData, deleteData, getData, getLength, insertData, replaceData, setData, setNodeValue, substringData

Methods inherited from class XMLNode

appendChild(Node), cloneNode(boolean), getAttributes(), getChildNodes(), getFirstChild(), getLastChild(), getNextSibling(), getNodeName(), getNodeType(), getOwnerDocument(), getParentNode(), getPreviousSibling(), hasChildNodes(), insertBefore(Node, Node), print(OutputStream), print(OutputStream, String), print(PrintWriter), removeChild(Node), replaceChild(Node, Node), selectNodes(String, NSResolver), selectSingleNode(String, NSResolver), transformNode(XSLStylesheet), valueOf(String, NSResolver)

Methods inherited from class java.lang.Object

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

Methods inherited from interface Text

Inherited Member Summary

splitText(int)

Methods inherited from interface CharacterData

appendData(String), deleteData(int, int), getData(), getLength(), insertData(int, String), replaceData(int, int, String), setData(String), substringData(int, int)

Methods inherited from interface Node

appendChild(Node), cloneNode(boolean), getAttributes(), getChildNodes(), getFirstChild(), getLastChild(), getNextSibling(), getNodeName(), getNodeType(), getNodeValue(), getOwnerDocument(), getParentNode(), getPreviousSibling(), hasChildNodes(), insertBefore(Node, Node), removeChild(Node), replaceChild(Node, Node), setNodeValue(String)

Constructor

XMLCDATA(String)

```
public XMLCDATA(java.lang.String text)
```

Creates a CDATA node having the given name and text.

Parameters

`text` - Text of the node

XMLComment

Syntax

```
public class XMLComment extends oracle.xml.parser.v2.CharData implements
org.w3c.dom.Comment, java.io.Serializable

java.lang.Object
  |
  +--XMLNode
        |
        +--oracle.xml.parser.v2.CharData
              |
              +--oracle.xml.parser.v2.XMLComment
```

All Implemented Interfaces

org.w3c.dom.CharacterData, java.lang.Cloneable, org.w3c.dom.Comment, org.w3c.dom.Node, java.io.Serializable, oracle.xml.parser.v2.XMLConstants

Description

This class implements the DOM Comment interface.

See Also

org.w3c.dom.Comment, NodeFactory, setNodeFactory(NodeFactory)

Member Summary

Constructors

XMLComment(String)	Creates a new Comment node.

Inherited Member Summary

Fields inherited from class XMLNode

Inherited Member Summary

AMP, ASTERISK, ATTRDECL, cANY, cATTLIST, cCDATA, cCDATAEND, cCDATASTART, cCOMMENTEND, cCOMMENTSTART, cDECCREF, cDECLSTART, cDOCTYPE, cELEMENT, cEMPTY, cEMPTYTAGEND, cENCODING, cENDTAGSTART, cENTITIES, cENTITY, cFIXED, cHEXCREF, cID, cIDREF, cIDREFS, cIGNORE, cIMPLIED, cINCLUDE, cNDATA, cNMTOKEN, cNMTOKENS, cNOTATION, COLON, COMMA, cPIEND, cPISTART, cPUBLIC, cREQUIRED, cSTANDALONE, cSYSTEM, cVERSION, cXML, DOUBLEQUOTE, ELEMENTDECL, EOF, EQ, ERROR, FATAL_ERROR, FDIGIT, FLETTER, FMISCNAME, FSTARTNAME, FWHITESPACE, HASH, ICOUNT, ISTART, LEFTSQB, LPAREN, nameCDATA, nameCOMMENT, nameDOCUMENT, nameDOCUMENTFRAGMENT, nameENCODING, nameNameSpace, nameSpaceSeparator, nameSTANDALONE, nameTEXT, nameVERSION, nameXML, nameXMLLang, nameXMLNamespace, nameXMLNSNamespace, nameXMLSpace, nameXSLPI, NONVALIDATING, OR, PERCENT, PLUS, QMARK, QUOTE, RIGHTSQB, RPAREN, SEMICOLON, SLASH, TAGEND, TAGSTART, VALIDATING, WARNING

Fields inherited from interface Node

ATTRIBUTE_NODE, CDATA_SECTION_NODE, COMMENT_NODE, DOCUMENT_FRAGMENT_NODE, DOCUMENT_NODE, DOCUMENT_TYPE_NODE, ELEMENT_NODE, ENTITY_NODE, ENTITY_REFERENCE_NODE, NOTATION_NODE, PROCESSING_INSTRUCTION_NODE, TEXT_NODE

Fields inherited from interface oracle.xml.parser.v2.XMLConstants

AMP, ASTERISK, cANY, cATTLIST, cCDATA, cCDATAEND, cCDATASTART, cCOMMENTEND, cCOMMENTSTART, cDECCREF, cDECLSTART, cDOCTYPE, cELEMENT, cEMPTY, cEMPTYTAGEND, cENCODING, cENDTAGSTART, cENTITIES, cENTITY, cFIXED, cHEXCREF, cID, cIDREF, cIDREFS, cIGNORE, cIMPLIED, cINCLUDE, cNDATA, cNMTOKEN, cNMTOKENS, cNOTATION, COLON, COMMA, cPIEND, cPISTART, cPUBLIC, cREQUIRED, cSTANDALONE, cSYSTEM, cVERSION, cXML, DOUBLEQUOTE, EOF, EQ, ERROR, FATAL_ERROR, FDIGIT, FLETTER, FMISCNAME, FSTARTNAME, FWHITESPACE, HASH, ICOUNT, ISTART, LEFTSQB, LPAREN, nameCDATA, nameCOMMENT, nameDOCUMENT, nameDOCUMENTFRAGMENT, nameENCODING, nameNameSpace, nameSpaceSeparator, nameSTANDALONE, nameTEXT, nameVERSION, nameXML, nameXMLLang, nameXMLNamespace, nameXMLNSNamespace, nameXMLSpace, nameXSLPI, NONVALIDATING, OR, PERCENT, PLUS, QMARK, QUOTE, RIGHTSQB, RPAREN, SEMICOLON, SLASH, TAGEND, TAGSTART, VALIDATING, WARNING

Methods inherited from class oracle.xml.parser.v2.CharData

appendData, deleteData, getData, getLength, insertData, replaceData, setData, setNodeValue, substringData

Methods inherited from class XMLNode

appendChild(Node), cloneNode(boolean), getAttributes(), getChildNodes(), getFirstChild(), getLastChild(), getNextSibling(), getNodeName(), getNodeType(), getNodeValue(), getOwnerDocument(), getParentNode(), getPreviousSibling(), hasChildNodes(), insertBefore(Node, Node), print(OutputStream), print(OutputStream, String), print(PrintWriter), removeChild(Node), replaceChild(Node, Node), selectNodes(String, NSResolver), selectSingleNode(String, NSResolver), transformNode(XSLStylesheet), valueOf(String, NSResolver)

Methods inherited from class java.lang.Object

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

Methods inherited from interface CharacterData

appendData(String), deleteData(int, int), getData(), getLength(), insertData(int, String), replaceData(int, int, String), setData(String), substringData(int, int)

Methods inherited from interface Node

Inherited Member Summary

appendChild(Node), cloneNode(boolean), getAttributes(), getChildNodes(), getFirstChild(), getLastChild(), getNextSibling(), getNodeName(), getNodeType(), getNodeValue(), getOwnerDocument(), getParentNode(), getPreviousSibling(), hasChildNodes(), insertBefore(Node, Node), removeChild(Node), replaceChild(Node, Node), setNodeValue(String)

Constructor

XMLComment(String)

```
public XMLComment(java.lang.String text)
```

Creates a new Comment node.

Parameters

`text` - Text of the comment node

XMLDocument

Syntax

```
public class XMLDocument extends XMLNode implements org.w3c.dom.Document,
java.io.Serializable

java.lang.Object
  |
  +--XMLNode
        |
        +--oracle.xml.parser.v2.XMLDocument
```

All Implemented Interfaces

`java.lang.Cloneable`, org.w3c.dom.Document, org.w3c.dom.Node, `java.io.Serializable`, `oracle.xml.parser.v2.XMLConstants`

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.

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.

Member Summary

Constructors	
XMLDocument()	Creates an empty document.
Methods	
cloneNode(boolean)	Returns a duplicate of this document node.
createAttribute(String)	Creates an `Attr` of the given name.
createCDATASection(String)	Creates a `CDATASection` node whose value is the specified string.
createComment(String)	Creates a `Comment` node given the specified string.
createDocumentFragment()	Creates an empty `DocumentFragment` object.
createElement(String)	Creates an element of the type specified.

Member Summary	
createEntityReference(String)	Creates an EntityReference object.
createProcessingInstruction(String, String)	Creates a `ProcessingInstruction` node given the specified name and data strings.
createTextNode(String)	Creates a `Text` node given the specified string.
expectedElements(Element)	Returns vector of element names that can be appended to the element.
getDoctype()	The Document Type Declaration (DTD) associated with this document.
getDocumentElement()	This is a convenience attribute that allows direct access to the child node that is the root element of the document.
getElementsByTagName(String)	Returns a `NodeList` of all the `Element`s with a given tag name in the order in which they would be encountered in a preorder traversal of the `Document` tree.
getEncoding()	Retrieves the character encoding information.
getImplementation()	The `DOMImplementation` object that handles this document.
getOwnerDocument()	The `Document` object associated with this node.
getStandalone()	Retrieves the standalone information.
getVersion()	Retrieves the version information.
print(OutputStream)	Writes the contents of this document to the given output stream.
print(OutputStream, String)	Writes the contents of this document to the given output stream.
print(PrintWriter)	Writes the contents of this document to the given output stream.
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.
replaceChild(Node, Node)	Replaces the child node `oldChild` with `newChild` in the list of children, and returns the `oldChild` node.
setEncoding(String)	Sets the character encoding for output.
setLocale(Locale)	Sets the locale for error reporting
setStandalone(String)	Sets the standalone information stored in the <?xml ...?> tag.
setVersion(String)	Sets the version number stored in the <?xml ...?> tag.
validateElementContent(Element)	Validates the content of a element node.

Inherited Member Summary

Fields inherited from class XMLNode

AMP, ASTERISK, ATTRDECL, cANY, cATTLIST, cCDATA, cCDATAEND, cCDATASTART, cCOMMENTEND, cCOMMENTSTART, cDECCREF, cDECLSTART, cDOCTYPE, cELEMENT, cEMPTY, cEMPTYTAGEND, cENCODING, cENDTAGSTART, cENTITIES, cENTITY, cFIXED, cHEXCREF, cID, cIDREF, cIDREFS, cIGNORE, cIMPLIED, cINCLUDE, cNDATA, cNMTOKEN, cNMTOKENS, cNOTATION, COLON, COMMA, cPIEND, cPISTART, cPUBLIC, cREQUIRED, cSTANDALONE, cSYSTEM, cVERSION, cXML, DOUBLEQUOTE, ELEMENTDECL, EOF, EQ, ERROR, FATAL_ERROR, FDIGIT, FLETTER, FMISCNAME, FSTARTNAME, FWHITESPACE, HASH, ICOUNT, ISTART, LEFTSQB, LPAREN, nameCDATA, nameCOMMENT, nameDOCUMENT, nameDOCUMENTFRAGMENT, nameENCODING, nameNameSpace, nameSpaceSeparator, nameSTANDALONE, nameTEXT, nameVERSION, nameXML, nameXMLLang, nameXMLNamespace, nameXMLNSNamespace, nameXMLSpace, nameXSLPI, NONVALIDATING, OR, PERCENT, PLUS, QMARK, QUOTE, RIGHTSQB, RPAREN, SEMICOLON, SLASH, TAGEND, TAGSTART, VALIDATING, WARNING

Fields inherited from interface Node

ATTRIBUTE_NODE, CDATA_SECTION_NODE, COMMENT_NODE, DOCUMENT_FRAGMENT_NODE, DOCUMENT_NODE, DOCUMENT_TYPE_NODE, ELEMENT_NODE, ENTITY_NODE, ENTITY_REFERENCE_NODE, NOTATION_NODE, PROCESSING_INSTRUCTION_NODE, TEXT_NODE

Fields inherited from interface oracle.xml.parser.v2.XMLConstants

AMP, ASTERISK, cANY, cATTLIST, cCDATA, cCDATAEND, cCDATASTART, cCOMMENTEND, cCOMMENTSTART, cDECCREF, cDECLSTART, cDOCTYPE, cELEMENT, cEMPTY, cEMPTYTAGEND, cENCODING, cENDTAGSTART, cENTITIES, cENTITY, cFIXED, cHEXCREF, cID, cIDREF, cIDREFS, cIGNORE, cIMPLIED, cINCLUDE, cNDATA, cNMTOKEN, cNMTOKENS, cNOTATION, COLON, COMMA, cPIEND, cPISTART, cPUBLIC, cREQUIRED, cSTANDALONE, cSYSTEM, cVERSION, cXML, DOUBLEQUOTE, EOF, EQ, ERROR, FATAL_ERROR, FDIGIT, FLETTER, FMISCNAME, FSTARTNAME, FWHITESPACE, HASH, ICOUNT, ISTART, LEFTSQB, LPAREN, nameCDATA, nameCOMMENT, nameDOCUMENT, nameDOCUMENTFRAGMENT, nameENCODING, nameNameSpace, nameSpaceSeparator, nameSTANDALONE, nameTEXT, nameVERSION, nameXML, nameXMLLang, nameXMLNamespace, nameXMLNSNamespace, nameXMLSpace, nameXSLPI, NONVALIDATING, OR, PERCENT, PLUS, QMARK, QUOTE, RIGHTSQB, RPAREN, SEMICOLON, SLASH, TAGEND, TAGSTART, VALIDATING, WARNING

Methods inherited from class XMLNode

appendChild(Node), getAttributes(), getChildNodes(), getFirstChild(), getLastChild(), getNextSibling(), getNodeName(), getNodeType(), getNodeValue(), getParentNode(), getPreviousSibling(), hasChildNodes(), insertBefore(Node, Node), removeChild(Node), selectNodes(String, NSResolver), selectSingleNode(String, NSResolver), setNodeValue(String), transformNode(XSLStylesheet), valueOf(String, NSResolver)

Methods inherited from class java.lang.Object

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

Methods inherited from interface Node

appendChild(Node), getAttributes(), getChildNodes(), getFirstChild(), getLastChild(), `getNextSibling()`, getNodeName(), getNodeType(), getNodeValue(), getParentNode(), getPreviousSibling(), hasChildNodes(), insertBefore(Node, Node), removeChild(Node), setNodeValue(String)

Constructor

XMLDocument()

```
public XMLDocument()
```
Creates an empty document.

Methods

cloneNode(boolean)

```
public org.w3c.dom.Node cloneNode(boolean deep)
```
Returns a duplicate of this document node.

Specified By

org.w3c.dom.Node.cloneNode(boolean) in interface org.w3c.dom.Node

Overrides

cloneNode(boolean) in class XMLNode

Parameters

`deep` - If `true`, recursively clone the subtree under the document; if `false`, clone only the document itself

Returns

The duplicate document node.

createAttribute(String)

```
public org.w3c.dom.Attr createAttribute(java.lang.String name)
```
Creates an `Attr` of the given name. Note that the `Attr` instance can then be set on an `Element` using the `setAttribute` method.

Specified By

org.w3c.dom.Document.createAttribute(String) in interface org.w3c.dom.Document

Parameters

`name` - The name of the attribute.

Returns

A new `Attr` object.

Throws

org.w3c.dom.DOMException - INVALID_CHARACTER_ERR: Raised if the specified name contains an invalid character.

createCDATASection(String)

`public` org.w3c.dom.CDATASection `createCDATASection(java.lang.String data)`
Creates a `CDATASection` node whose value is the specified string.

Specified By

org.w3c.dom.Document.createCDATASection(String) in interface org.w3c.dom.Document

Parameters

`data` - The data for the `CDATASection` contents.

Returns

The new `CDATASection` object.

Throws

org.w3c.dom.DOMException - A `DOMException` could be thrown.

createComment(String)

`public` org.w3c.dom.Comment `createComment(java.lang.String data)`
Creates a `Comment` node given the specified string.

Specified By

org.w3c.dom.Document.createComment(String) in interface org.w3c.dom.Document

Parameters

`data` - The data for the node.

Returns

The new `Comment` object.

createDocumentFragment()

`public` org.w3c.dom.DocumentFragment `createDocumentFragment()`
Creates an empty `DocumentFragment` object.

Specified By

org.w3c.dom.Document.createDocumentFragment() in interface org.w3c.dom.Document

Returns

A new `DocumentFragment`.

createElement(String)

`public` org.w3c.dom.Element `createElement(java.lang.String tagName)`
Creates an element of the type specified. Note that the instance returned implements the Element interface, so attributes can be specified directly on the returned object.

Specified By

org.w3c.dom.Document.createElement(String) in interface org.w3c.dom.Document

Parameters

`tagName` - The name of the element type to instantiate. The name is treated as case-sensitive.

Returns

A new `Element` object.

Throws

org.w3c.dom.DOMException - INVALID_CHARACTER_ERR: Raised if the specified name contains an invalid character.

createEntityReference(String)

`public` org.w3c.dom.EntityReference `createEntityReference(java.lang.String name)`
Creates an EntityReference object.

Specified By

org.w3c.dom.Document.createEntityReference(String) in interface org.w3c.dom.Document

Parameters

`name` - The name of the entity to reference.

Returns

The new `EntityReference` object.

Throws

org.w3c.dom.DOMException - INVALID_CHARACTER_ERR: Raised if the specified name contains an invalid character.

createProcessingInstruction(String, String)

`public` org.w3c.dom.ProcessingInstruction
`createProcessingInstruction(java.lang.String target, java.lang.String data)`
Creates a `ProcessingInstruction` node given the specified name and data strings.

Specified By

org.w3c.dom.Document.createProcessingInstruction(String, String) in interface org.w3c.dom.Document

Parameters

`target` - The target part of the processing instruction.

`data` - The data for the node.

Returns

The new `ProcessingInstruction` object.

Throws

org.w3c.dom.DOMException - INVALID_CHARACTER_ERR: Raised if an invalid character is specified.

createTextNode(String)

`public` org.w3c.dom.Text `createTextNode(java.lang.String data)`
Creates a `Text` node given the specified string.

Specified By

org.w3c.dom.Document.createTextNode(String) in interface org.w3c.dom.Document

Parameters

`data` - The data for the node.

Returns

The new `Text` object.

expectedElements(Element)

`public java.util.Vector expectedElements(`org.w3c.dom.Element `e)`
Returns vector of element names that can be appended to the element.

Parameters

`e` - Element

Returns

Vector of names

getDoctype()

`public` org.w3c.dom.DocumentType `getDoctype()`
The Document Type Declaration (DTD) associated with this document. For XML documents without a DTD, this returns `null`. Note that the DOM Level 1 specification does not support editing the DTD.

Specified By

org.w3c.dom.Document.getDoctype() in interface org.w3c.dom.Document

Returns

The associated DTD

See Also

org.w3c.dom.DocumentType

getDocumentElement()

`public` org.w3c.dom.Element `getDocumentElement()`

This is a convenience attribute that allows direct access to the child node that is the root element of the document.

Specified By

org.w3c.dom.Document.getDocumentElement() in interface org.w3c.dom.Document

Returns

The root element

getElementsByTagName(String)

`public` org.w3c.dom.NodeList `getElementsByTagName(java.lang.String tagname)`
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.

Specified By

org.w3c.dom.Document.getElementsByTagName(String) in interface org.w3c.dom.Document

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`.

getEncoding()

`public final java.lang.String getEncoding()`
Retrieves the character encoding information.

Returns

the encoding information stored in the <?xml ...?> tag or the user-defined output encoding if it has been more recently set.

getImplementation()

`public` org.w3c.dom.DOMImplementation `getImplementation()`
The `DOMImplementation` object that handles this document. A DOM application may use objects from multiple implementations.

Specified By

org.w3c.dom.Document.getImplementation() in interface org.w3c.dom.Document

Returns

The associated DOM implementation.

getOwnerDocument()

public org.w3c.dom.Document getOwnerDocument()
The Document object associated with this node. Since this node is a Document this is null.

Specified By

org.w3c.dom.Node.getOwnerDocument() in interface org.w3c.dom.Node

Overrides

getOwnerDocument() in class XMLNode

Returns

null

getStandalone()

public final java.lang.String getStandalone()
Retrieves the standalone information.

Returns

the standalone attribute stored in the <?xml ...?> tag.

getVersion()

public final java.lang.String getVersion()
Retrieves the version information.

Returns

the version number stored in the <?xml ...?> tag.

print(OutputStream)

public void print(java.io.OutputStream out)

Writes the contents of this document to the given output stream.

Overrides

print(OutputStream) in class XMLNode

Parameters

`out` - `OutputStream` to write to

Throws

`IOException` - if an error occurs

print(OutputStream, String)

`public void print(java.io.OutputStream out, java.lang.String enc)`
Writes the contents of this document to the given output stream.

Overrides

print(OutputStream, String) in class XMLNode

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)

`public void print(java.io.PrintWriter out)`
Writes the contents of this document to the given output stream.

Overrides

print(PrintWriter) in class XMLNode

Parameters

`out` - `PrintWriter` to write to

Throws

`IOException` - if an error occurs

printExternalDTD(OutputStream)

`public void printExternalDTD(java.io.OutputStream out)`
Writes the contents of this document to the given output stream.

Parameters

`out` - `OutputStream` to write to

Throws

`IOException` - if an error occurs

printExternalDTD(OutputStream, String)

`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)

`public void printExternalDTD(java.io.PrintWriter out)`
Writes the contents of this document to the given output stream.

Parameters

`out` - `PrintWriter` to write to

Throws

`IOException` - if an error occurs

replaceChild(Node, Node)

`public` org.w3c.dom.Node `replaceChild`(org.w3c.dom.Node `newChild`,

org.w3c.dom.Node oldChild)
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. This is an override of the XMLNode.removeChild method

Specified By

org.w3c.dom.Node.replaceChild(Node, Node) in interface org.w3c.dom.Node

Overrides

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

org.w3c.dom.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.

setEncoding(String)

public final void setEncoding(java.lang.String encoding)
Sets the character encoding for output. 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

setLocale(Locale)

public final void setLocale(java.util.Locale locale)
Sets the locale for error reporting

Parameters

`locale` - Locale for error reporting.

setStandalone(String)

`public final void setStandalone(java.lang.String value)`
Sets the standalone information stored in the <?xml ...?> tag.

Parameters

`value` - The attribute value ('yes' or 'no').

setVersion(String)

`public final void setVersion(java.lang.String version)`
Sets the version number stored in the <?xml ...?> tag.

Parameters

`version` - The version information to set.

validateElementContent(Element)

`public boolean validateElementContent(org.w3c.dom.Element e)`
Validates the content of a element node.

Parameters

`e` - Element to be validated

Returns

True if valid, else false

XMLDocumentFragment

Syntax

```
public class XMLDocumentFragment extends XMLNode implements
org.w3c.dom.DocumentFragment, java.io.Serializable

java.lang.Object
  |
  +--XMLNode
        |
        +--oracle.xml.parser.v2.XMLDocumentFragment
```

All Implemented Interfaces

java.lang.Cloneable, org.w3c.dom.DocumentFragment, org.w3c.dom.Node, java.io.Serializable, oracle.xml.parser.v2.XMLConstants

Description

This class implements the DOM DocumentFragment interface.

See Also

org.w3c.dom.DocumentFragment, NodeFactory, setNodeFactory(NodeFactory)

Member Summary

Constructors	
XMLDocumentFragment()	Creates an empty document fragment
Methods	
getParentNode()	Gets the parent of this node

Inherited Member Summary

Fields inherited from class XMLNode

Inherited Member Summary

AMP, ASTERISK, ATTRDECL, cANY, cATTLIST, cCDATA, cCDATAEND, cCDATASTART, cCOMMENTEND, cCOMMENTSTART, cDECCREF, cDECLSTART, cDOCTYPE, cELEMENT, cEMPTY, cEMPTYTAGEND, cENCODING, cENDTAGSTART, cENTITIES, cENTITY, cFIXED, cHEXCREF, cID, cIDREF, cIDREFS, cIGNORE, cIMPLIED, cINCLUDE, cNDATA, cNMTOKEN, cNMTOKENS, cNOTATION, COLON, COMMA, cPIEND, cPISTART, cPUBLIC, cREQUIRED, cSTANDALONE, cSYSTEM, cVERSION, cXML, DOUBLEQUOTE, ELEMENTDECL, EOF, EQ, ERROR, FATAL_ERROR, FDIGIT, FLETTER, FMISCNAME, FSTARTNAME, FWHITESPACE, HASH, ICOUNT, ISTART, LEFTSQB, LPAREN, nameCDATA, nameCOMMENT, nameDOCUMENT, nameDOCUMENTFRAGMENT, nameENCODING, nameNameSpace, nameSpaceSeparator, nameSTANDALONE, nameTEXT, nameVERSION, nameXML, nameXMLLang, nameXMLNamespace, nameXMLNSNamespace, nameXMLSpace, nameXSLPI, NONVALIDATING, OR, PERCENT, PLUS, QMARK, QUOTE, RIGHTSQB, RPAREN, SEMICOLON, SLASH, TAGEND, TAGSTART, VALIDATING, WARNING

Fields inherited from interface Node

ATTRIBUTE_NODE, CDATA_SECTION_NODE, COMMENT_NODE, DOCUMENT_FRAGMENT_NODE, DOCUMENT_NODE, DOCUMENT_TYPE_NODE, ELEMENT_NODE, ENTITY_NODE, ENTITY_REFERENCE_NODE, NOTATION_NODE, PROCESSING_INSTRUCTION_NODE, TEXT_NODE

Fields inherited from interface oracle.xml.parser.v2.XMLConstants

AMP, ASTERISK, cANY, cATTLIST, cCDATA, cCDATAEND, cCDATASTART, cCOMMENTEND, cCOMMENTSTART, cDECCREF, cDECLSTART, cDOCTYPE, cELEMENT, cEMPTY, cEMPTYTAGEND, cENCODING, cENDTAGSTART, cENTITIES, cENTITY, cFIXED, cHEXCREF, cID, cIDREF, cIDREFS, cIGNORE, cIMPLIED, cINCLUDE, cNDATA, cNMTOKEN, cNMTOKENS, cNOTATION, COLON, COMMA, cPIEND, cPISTART, cPUBLIC, cREQUIRED, cSTANDALONE, cSYSTEM, cVERSION, cXML, DOUBLEQUOTE, EOF, EQ, ERROR, FATAL_ERROR, FDIGIT, FLETTER, FMISCNAME, FSTARTNAME, FWHITESPACE, HASH, ICOUNT, ISTART, LEFTSQB, LPAREN, nameCDATA, nameCOMMENT, nameDOCUMENT, nameDOCUMENTFRAGMENT, nameENCODING, nameNameSpace, nameSpaceSeparator, nameSTANDALONE, nameTEXT, nameVERSION, nameXML, nameXMLLang, nameXMLNamespace, nameXMLNSNamespace, nameXMLSpace, nameXSLPI, NONVALIDATING, OR, PERCENT, PLUS, QMARK, QUOTE, RIGHTSQB, RPAREN, SEMICOLON, SLASH, TAGEND, TAGSTART, VALIDATING, WARNING

Methods inherited from class XMLNode

appendChild(Node), cloneNode(boolean), getAttributes(), getChildNodes(), getFirstChild(), getLastChild(), getNextSibling(), getNodeName(), getNodeType(), getNodeValue(), getOwnerDocument(), getPreviousSibling(), hasChildNodes(), insertBefore(Node, Node), print(OutputStream), print(OutputStream, String), print(PrintWriter), removeChild(Node), replaceChild(Node, Node), selectNodes(String, NSResolver), selectSingleNode(String, NSResolver), setNodeValue(String), transformNode(XSLStylesheet), valueOf(String, NSResolver)

Methods inherited from class java.lang.Object

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

Methods inherited from interface Node

appendChild(Node), cloneNode(boolean), getAttributes(), getChildNodes(), getFirstChild(), getLastChild(), getNextSibling(), getNodeName(), getNodeType(), getNodeValue(), getOwnerDocument(), getPreviousSibling(), hasChildNodes(), insertBefore(Node, Node), removeChild(Node), replaceChild(Node, Node), setNodeValue(String)

Constructor

XMLDocumentFragment()

```
public XMLDocumentFragment()
```
Creates an empty document fragment

Methods

getParentNode()

```
public org.w3c.dom.Node getParentNode()
```
Gets the parent of this node

Specified By

org.w3c.dom.Node.getParentNode() in interface org.w3c.dom.Node

Overrides

getParentNode() in class XMLNode

Returns

The parent of this node (always null)

XMLDocumentHandler

Syntax

`public interface XMLDocumentHandler extends` org.xml.sax.DocumentHandler

All Superinterfaces

`org.xml.sax.DocumentHandler`

All Known Implementing Classes

`Package oracle.xml.parser.v2`

Description

This interface extends the `org.xml.sax.DocumentHandler` interface. SAX Applications requiring Namespace support must implement this interface and register with the SAX Parser via `Parser.setDocumentHandler()`.

Member Summary

Methods

cDATASection(char[ ], int, int)	Receive notification of a CDATA Section.
comment(String)	Receive notification of a comment.
endDoctype()	Receive notification of end of the DTD.
endElement(NSName)	Receive notification of the end of an element.
setDoctype(DTD)	Receive notification of a DTD (Document Type node).
setTextDecl(String, String)	Receive notification of a Text XML Declaration.
setXMLDecl(String, String, String)	Receive notification of a XML Declaration.
startElement(NSName, SAXAttrList)	Receive notification of the beginning of an element.

Inherited Member Summary

Methods inherited from interface DocumentHandler

characters(char[], int, int), endDocument(), endElement(String), ignorableWhitespace(char[], int, int), processingInstruction(String, String), setDocumentLocator(Locator), startDocument(), startElement(String, AttributeList)

Methods

cDATASection(char[ ], int, int)

```
public void cDATASection(char[] ch, int start, int length)
```
Receive notification of a CDATA Section.

The Parser will invoke this method once for each CDATA Section found.

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)

```
public void comment(java.lang.String data)
```
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.

Parameters

`data` - The comment data, or null if none was supplied.

Throws

org.xml.sax.SAXException - Any SAX exception, possibly wrapping another exception.

endDoctype()

```
public void endDoctype()
```
Receive notification of end of the DTD.

Throws

org.xml.sax.SAXException - Any SAX exception, possibly wrapping another exception.

endElement(NSName)

```
public void endElement(NSName elem)
```

Receive notification of the end of an element. The SAX parser will invoke this method at the end of every element in the XML document; there will be a corresponding startElement() event for every endElement() event (even when the element is empty).

By implementing this method instead of `org.xml.sax.DocumentHandler.endElement`, SAX Applications can get the Namespace support provided by `NSName`.

Parameters

`elem` - NSName object

Throws

org.xml.sax.SAXException - A `SAXException` could be thrown.

See Also: org.xml.sax.DocumentHandler.endElement(String)

setDoctype(DTD)

```
public void setDoctype(DTD dtd)
```

Receive notification of a DTD (Document Type node).

The Parser will invoke this method after calling startDocument to register the DTD used.

Parameters

`DTD` - The DTD node

Throws

org.xml.sax.SAXException - Any SAX exception, possibly wrapping another exception.

setTextDecl(String, String)

```
public void setTextDecl(java.lang.String version, java.lang.String encoding)
```

Receive notification of a Text XML Declaration.

The Parser will invoke this method once for each text XML Decl

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)

```
public void setXMLDecl(java.lang.String version, java.lang.String standalone,
java.lang.String encoding)
```
Receive notification of a XML Declaration.

The Parser will invoke this method once for XML Decl

Parameters

`version` - The version number

`standalone` - The standalone value (or null, if not specifed)

`encoding` - The encoding name (or null, if not specifed)

Throws

org.xml.sax.SAXException - Any SAX exception, possibly wrapping another exception.

startElement(NSName, SAXAttrList)

```
public void startElement(NSName elem, SAXAttrList attrlist)
```
Receive notification of the beginning of an element. The Parser will invoke this method at the beginning of every element in the XML document; there will be a corresponding endElement() event for every startElement() event (even when the element is empty). All of the element's content will be reported, inorder, before the corresponding endElement() event.

By implementing this method instead of `org.xml.sax.DocumentHandler.startElement`, SAX Applications can get the Namespace support provided by `NSName` and `SAXAttrList`.

Parameters

`elem` - NSName object

`attrlist` - SAXAttrList for the element

Throws

org.xml.sax.SAXException - A `SAXException` could be thrown.

See Also org.xml.sax.DocumentHandler.startElement(String, AttributeList):

XMLElement

Syntax

```
public class XMLElement extends XMLNode implements org.w3c.dom.Element,
java.io.Serializable, NSName, NSResolver

java.lang.Object
  |
  +--XMLNode
        |
        +--oracle.xml.parser.v2.XMLElement
```

All Implemented Interfaces

ava.lang.Cloneable, org.w3c.dom.Element, org.w3c.dom.Node, NSName, NSResolver, java.io.Serializable, oracle.xml.parser.v2.XMLConstants

Description

This class implements the DOM `Element` interface. Elements are created by the XML parser using the default NodeFactory or the user defined NodeFactory if registered using setNodeFactoty() method.

See Also: org.w3c.dom.Element, XMLParser, NodeFactory, setNodeFactory(NodeFactory):

Member Summary

Constructors	
XMLElement(String)	Creates an element with the given name
XMLElement(String, String, String)	Creates an element with the given name, prefix, and namespace
Methods	
checkNamespace(String, String)	Returns if the element belongs to the namespace specified.
cloneNode(boolean)	Returns a duplicate of this node, i.e., serves as a generic copy constructor for nodes.
getAttribute(String)	Retrieves an attribute value by name.
getAttributeNode(String)	Retrieves an `Attr` node by name.

Member Summary

getAttributes()	A `NamedNodeMap` containing the attributes of this node (if it is an `Element`) or `null` otherwise.
getChildrenByTagName(String)	Returns a `NodeList` of all immediate children with a given tag name,
getChildrenByTagName(String, String)	Returns a `NodeList` of all immediate children with a given tag name and namespace
getElementsByTagName(String)	Returns a `NodeList` of all descendant elements with a given tag name, in the order in which they would be encountered in a preorder traversal of the `Element` tree.
getElementsByTagName(String, String)	Returns a `NodeList` of all descendant elements with a given tag name, and namespace in the order in which they would be encountered in a preorder traversal of the `Element` tree.
getExpandedName()	Get the fully resolved name for this element.
getLocalName()	Get the local Name for this element.
getNamespace()	Get the resolved Namespace for this element.
getPrefix()	Get the namespace prefix for this element.
getQualifiedName()	Get the qualified name for this element.
getTagName()	Gets the name of the element.
normalize()	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.
removeAttribute(String)	Removes an attribute by name.
removeAttributeNode(Attr)	Removes the specified attribute.
resolveNamespacePrefix(String)	Given a namespace prefix, find the namespace definition in scope in this element.
setAttribute(String, String)	Adds a new attribute.
setAttributeNode(Attr)	Adds a new attribute.

Inherited Member Summary

Fields inherited from class XMLNode

Inherited Member Summary

AMP, ASTERISK, ATTRDECL, cANY, cATTLIST, cCDATA, cCDATAEND, cCDATASTART, cCOMMENTEND, cCOMMENTSTART, cDECCREF, cDECLSTART, cDOCTYPE, cELEMENT, cEMPTY, cEMPTYTAGEND, cENCODING, cENDTAGSTART, cENTITIES, cENTITY, cFIXED, cHEXCREF, cID, cIDREF, cIDREFS, cIGNORE, cIMPLIED, cINCLUDE, cNDATA, cNMTOKEN, cNMTOKENS, cNOTATION, COLON, COMMA, cPIEND, cPISTART, cPUBLIC, cREQUIRED, cSTANDALONE, cSYSTEM, cVERSION, cXML, DOUBLEQUOTE, ELEMENTDECL, EOF, EQ, ERROR, FATAL_ERROR, FDIGIT, FLETTER, FMISCNAME, FSTARTNAME, FWHITESPACE, HASH, ICOUNT, ISTART, LEFTSQB, LPAREN, nameCDATA, nameCOMMENT, nameDOCUMENT, nameDOCUMENTFRAGMENT, nameENCODING, nameNameSpace, nameSpaceSeparator, nameSTANDALONE, nameTEXT, nameVERSION, nameXML, nameXMLLang, nameXMLNamespace, nameXMLNSNamespace, nameXMLSpace, nameXSLPI, NONVALIDATING, OR, PERCENT, PLUS, QMARK, QUOTE, RIGHTSQB, RPAREN, SEMICOLON, SLASH, TAGEND, TAGSTART, VALIDATING, WARNING

Fields inherited from interface Node

ATTRIBUTE_NODE, CDATA_SECTION_NODE, COMMENT_NODE, DOCUMENT_FRAGMENT_NODE, DOCUMENT_NODE, DOCUMENT_TYPE_NODE, ELEMENT_NODE, ENTITY_NODE, ENTITY_REFERENCE_NODE, NOTATION_NODE, PROCESSING_INSTRUCTION_NODE, TEXT_NODE

Fields inherited from interface oracle.xml.parser.v2.XMLConstants

AMP, ASTERISK, cANY, cATTLIST, cCDATA, cCDATAEND, cCDATASTART, cCOMMENTEND, cCOMMENTSTART, cDECCREF, cDECLSTART, cDOCTYPE, cELEMENT, cEMPTY, cEMPTYTAGEND, cENCODING, cENDTAGSTART, cENTITIES, cENTITY, cFIXED, cHEXCREF, cID, cIDREF, cIDREFS, cIGNORE, cIMPLIED, cINCLUDE, cNDATA, cNMTOKEN, cNMTOKENS, cNOTATION, COLON, COMMA, cPIEND, cPISTART, cPUBLIC, cREQUIRED, cSTANDALONE, cSYSTEM, cVERSION, cXML, DOUBLEQUOTE, EOF, EQ, ERROR, FATAL_ERROR, FDIGIT, FLETTER, FMISCNAME, FSTARTNAME, FWHITESPACE, HASH, ICOUNT, ISTART, LEFTSQB, LPAREN, nameCDATA, nameCOMMENT, nameDOCUMENT, nameDOCUMENTFRAGMENT, nameENCODING, nameNameSpace, nameSpaceSeparator, nameSTANDALONE, nameTEXT, nameVERSION, nameXML, nameXMLLang, nameXMLNamespace, nameXMLNSNamespace, nameXMLSpace, nameXSLPI, NONVALIDATING, OR, PERCENT, PLUS, QMARK, QUOTE, RIGHTSQB, RPAREN, SEMICOLON, SLASH, TAGEND, TAGSTART, VALIDATING, WARNING

Methods inherited from class XMLNode

appendChild(Node), getChildNodes(), getFirstChild(), getLastChild(), getNextSibling(), getNodeName(), getNodeType(), getNodeValue(), getOwnerDocument(), getParentNode(), getPreviousSibling(), hasChildNodes(), insertBefore(Node, Node), print(OutputStream), print(OutputStream, String), print(PrintWriter), removeChild(Node), replaceChild(Node, Node), selectNodes(String, NSResolver), selectSingleNode(String, NSResolver), setNodeValue(String), transformNode(XSLStylesheet), valueOf(String, NSResolver)

Methods inherited from class java.lang.Object

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

Methods inherited from interface Node

appendChild(Node), getChildNodes(), getFirstChild(), getLastChild(), getNextSibling( ), getNodeName(), getNodeType(), getNodeValue(), getOwnerDocument(), getParentNode(), getPreviousSibling(), hasChildNodes(), insertBefore(Node, Node), removeChild(Node), replaceChild(Node, Node), setNodeValue(String)

Constructor

XMLElement(String)

```
public  XMLElement(java.lang.String tag)
```
Creates an element with the given name

XMLElement(String, String, String)

```
public  XMLElement(java.lang.String name, java.lang.String prefix,
java.lang.String namespace)
```
Creates an element with the given name, prefix, and namespace

Methods

checkNamespace(String, String)

```
public boolean checkNamespace(java.lang.String localname, java.lang.String ns)
```
Returns if the element belongs to the namespace specified.

Parameters

`ns` - Expanded namespace string

Returns

true - if the element belongs to the namespace

cloneNode(boolean)

```
public org.w3c.dom.Node cloneNode(boolean deep)
```
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.

Specified By

org.w3c.dom.Node.cloneNode(boolean) in interface org.w3c.dom.Node

Specified By

org.w3c.dom.Node.cloneNode(boolean) in interface org.w3c.dom.Node

Overrides

cloneNode(boolean) in class XMLNode

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)

`public java.lang.String getAttribute(java.lang.String name)`
Retrieves an attribute value by name.

Specified By

org.w3c.dom.Element.getAttribute(String) in interface org.w3c.dom.Element

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)

`public` org.w3c.dom.Attr `getAttributeNode(java.lang.String name)`
Retrieves an `Attr` node by name.

Specified By

org.w3c.dom.Element.getAttributeNode(String) in interface org.w3c.dom.Element

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.

getAttributes()

`public` org.w3c.dom.NamedNodeMap `getAttributes()`
A `NamedNodeMap` containing the attributes of this node (if it is an `Element`) or `null` otherwise.

Specified By

org.w3c.dom.Node.getAttributes() in interface org.w3c.dom.Node

Overrides

getAttributes() in class XMLNode

Returns

The list of attributes of this element

getChildrenByTagName(String)

`public` org.w3c.dom.NodeList `getChildrenByTagName(java.lang.String name)`
Returns a `NodeList` of all immediate children with a given tag name,

Parameters

`name` - The name of the tag to match on.

Returns

A list of matching children

getChildrenByTagName(String, String)

`public` org.w3c.dom.NodeList `getChildrenByTagName(java.lang.String name, java.lang.String ns)`
Returns a `NodeList` of all immediate children with a given tag name and namespace

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)

`public` org.w3c.dom.NodeList `getElementsByTagName(java.lang.String name)`
Returns a `NodeList` of all descendant elements with a given tag name, in the order in which they would be encountered in a preorder traversal of the `Element` tree.

Specified By

org.w3c.dom.Element.getElementsByTagName(String) in interface org.w3c.dom.Element

Parameters

`name` - The name of the tag to match on. The special value "*" matches all tags.

Returns

A list of matching `Element` nodes.

getElementsByTagName(String, String)

`public` org.w3c.dom.NodeList `getElementsByTagName(java.lang.String name, java.lang.String ns)`
Returns a `NodeList` of all descendant elements with a given tag name, and namespace in the order in which they would be encountered in a preorder traversal of the `Element` tree.

Parameters

`name` - The name of the tag to match on. The special value "*" matches all tags. (should be local name)

`ns` - The namespace of the elements

Returns

A list of matching `Element` nodes.

getExpandedName()

`public java.lang.String getExpandedName()`
Get the fully resolved name for this element.

Specified By
getExpandedName() in interface NSName

Returns
the fully resolved name

getLocalName()

```
public java.lang.String getLocalName()
```
Get the local Name for this element.

Specified By
getLocalName() in interface NSName

Returns
the local Name

getNamespace()

```
public java.lang.String getNamespace()
```
Get the resolved Namespace for this element.

Specified By
getNamespace() in interface NSName

Returns
the resolved Namespace

getPrefix()

```
public java.lang.String getPrefix()
```
Get the namespace prefix for this element.

Specified By
getPrefix() in interface NSName

Returns
the namespace prefix

getQualifiedName()

```
public java.lang.String getQualifiedName()
```
Get the qualified name for this element.

Specified By

getQualifiedName() in interface NSName

Returns

the qualified name

getTagName()

```
public java.lang.String getTagName()
```
Gets the name of the element. 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.

Specified By

org.w3c.dom.Element.getTagName() in interface org.w3c.dom.Element

Returns

The element name

normalize()

```
public void normalize()
```
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 doc1ument tree structure are to be used.

Specified By

org.w3c.dom.Element.normalize() in interface org.w3c.dom.Element

removeAttribute(String)

`public void removeAttribute(java.lang.String name)`
Removes an attribute by name. If the removed attribute has a default value it is immediately replaced.

Specified By

org.w3c.dom.Element.removeAttribute(String) in interface org.w3c.dom.Element

Parameters

`name` - The name of the attribute to remove.

Throws

org.w3c.dom.DOMException - NO_MODIFICATION_ALLOWED_ERR: Raised if this node is readonly.

removeAttributeNode(Attr)

`public` org.w3c.dom.Attr `removeAttributeNode`(org.w3c.dom.Attr `oldAttr`)
Removes the specified attribute.

Specified By

org.w3c.dom.Element.removeAttributeNode(Attr) in interface org.w3c.dom.Element

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

org.w3c.dom.DOMException - NO_MODIFICATION_ALLOWED_ERR: Raised if this node is readonly. NOT_FOUND_ERR: Raised if `oldAttr` is not an attribute of the element.

resolveNamespacePrefix(String)

`public java.lang.String resolveNamespacePrefix(java.lang.String prefix)`
Given a namespace prefix, find the namespace definition in scope in this element.

Specified By

resolveNamespacePrefix(String) in interface NSResolver

Parameters

`prefix` - Namespace prefix to be resolved

Returns

the resolved Namespace (null, if prefix could not be resolved)

setAttribute(String, String)

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

Specified By

org.w3c.dom.Element.setAttribute(String, String) in interface org.w3c.dom.Element

Parameters

`name` - The name of the attribute to create or alter.

`value` - Value to set in string form.

Throws

org.w3c.dom.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)

`public` org.w3c.dom.Attr `setAttributeNode`(org.w3c.dom.Attr `newAttr`)
Adds a new attribute. If an attribute with that name is already present in the element, it is replaced by the new one.

Specified By

org.w3c.dom.Element.setAttributeNode(Attr) in interface org.w3c.dom.Element

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

org.w3c.dom.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.

XMLEntityReference

Syntax

public class XMLEntityReference extends XMLNode implements org.w3c.dom.EntityReference, oracle.xml.parser.v2.XMLConstants, java.lang.Cloneable, java.io.Serializable

```
java.lang.Object
  |
  +--XMLNode
        |
        +--oracle.xml.parser.v2.XMLEntityReference
```

All Implemented Interfaces

java.lang.Cloneable, org.w3c.dom.EntityReference, org.w3c.dom.Node, java.io.Serializable, oracle.xml.parser.v2.XMLConstants

Description

Member Summary

Constructors

XMLEntityReference(String)

Inherited Member Summary

Fields inherited from class XMLNode

AMP, ASTERISK, ATTRDECL, cANY, cATTLIST, cCDATA, cCDATAEND, cCDATASTART, cCOMMENTEND, cCOMMENTSTART, cDECCREF, cDECLSTART, cDOCTYPE, cELEMENT, cEMPTY, cEMPTYTAGEND, cENCODING, cENDTAGSTART, cENTITIES, cENTITY, cFIXED, cHEXCREF, cID, cIDREF, cIDREFS, cIGNORE, cIMPLIED, cINCLUDE, cNDATA, cNMTOKEN, cNMTOKENS, cNOTATION, COLON, COMMA, cPIEND, cPISTART, cPUBLIC, cREQUIRED, cSTANDALONE, cSYSTEM, cVERSION, cXML, DOUBLEQUOTE, ELEMENTDECL, EOF, EQ, ERROR, FATAL_ERROR, FDIGIT, FLETTER, FMISCNAME, FSTARTNAME, FWHITESPACE, HASH, ICOUNT, ISTART, LEFTSQB, LPAREN, nameCDATA, nameCOMMENT, nameDOCUMENT, nameDOCUMENTFRAGMENT, nameENCODING, nameNameSpace, nameSpaceSeparator, nameSTANDALONE, nameTEXT, nameVERSION, nameXML, nameXMLLang, nameXMLNamespace, nameXMLNSNamespace, nameXMLSpace, nameXSLPI, NONVALIDATING, OR, PERCENT, PLUS, QMARK, QUOTE, RIGHTSQB, RPAREN, SEMICOLON, SLASH, TAGEND, TAGSTART, VALIDATING, WARNING

Fields inherited from interface Node

Inherited Member Summary

ATTRIBUTE_NODE, CDATA_SECTION_NODE, COMMENT_NODE, DOCUMENT_FRAGMENT_NODE, DOCUMENT_NODE, DOCUMENT_TYPE_NODE, ELEMENT_NODE, ENTITY_NODE, ENTITY_REFERENCE_NODE, NOTATION_NODE, PROCESSING_INSTRUCTION_NODE, TEXT_NODE

Fields inherited from interface oracle.xml.parser.v2.XMLConstants

AMP, ASTERISK, cANY, cATTLIST, cCDATA, cCDATAEND, cCDATASTART, cCOMMENTEND, cCOMMENTSTART, cDECCREF, cDECLSTART, cDOCTYPE, cELEMENT, cEMPTY, cEMPTYTAGEND, cENCODING, cENDTAGSTART, cENTITIES, cENTITY, cFIXED, cHEXCREF, cID, cIDREF, cIDREFS, cIGNORE, cIMPLIED, cINCLUDE, cNDATA, cNMTOKEN, cNMTOKENS, cNOTATION, COLON, COMMA, cPIEND, cPISTART, cPUBLIC, cREQUIRED, cSTANDALONE, cSYSTEM, cVERSION, cXML, DOUBLEQUOTE, EOF, EQ, ERROR, FATAL_ERROR, FDIGIT, FLETTER, FMISCNAME, FSTARTNAME, FWHITESPACE, HASH, ICOUNT, ISTART, LEFTSQB, LPAREN, nameCDATA, nameCOMMENT, nameDOCUMENT, nameDOCUMENTFRAGMENT, nameENCODING, nameNameSpace, nameSpaceSeparator, nameSTANDALONE, nameTEXT, nameVERSION, nameXML, nameXMLLang, nameXMLNamespace, nameXMLNSNamespace, nameXMLSpace, nameXSLPI, NONVALIDATING, OR, PERCENT, PLUS, QMARK, QUOTE, RIGHTSQB, RPAREN, SEMICOLON, SLASH, TAGEND, TAGSTART, VALIDATING, WARNING

Methods inherited from class XMLNode

appendChild(Node), cloneNode(boolean), getAttributes(), getChildNodes(), getFirstChild(), getLastChild(), getNextSibling(), getNodeName(), getNodeType(), getNodeValue(), getOwnerDocument(), getParentNode(), getPreviousSibling(), hasChildNodes(), insertBefore(Node, Node), print(OutputStream), print(OutputStream, String), print(PrintWriter), removeChild(Node), replaceChild(Node, Node), selectNodes(String, NSResolver), selectSingleNode(String, NSResolver), setNodeValue(String), transformNode(XSLStylesheet), valueOf(String, NSResolver)

Methods inherited from class java.lang.Object

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

Methods inherited from interface Node

appendChild(Node), cloneNode(boolean), getAttributes(), getChildNodes(), getFirstChild(), getLastChild(), getNextSibling(), getNodeName(), getNodeType(), getNodeValue(), getOwnerDocument(), getParentNode(), getPreviousSibling(), hasChildNodes(), insertBefore(Node, Node), removeChild(Node), replaceChild(Node, Node), setNodeValue(String)

Constructor

XMLEntityReference(String)

```
public XMLEntityReference(java.lang.String tag)
```

XMLNode

Syntax

public class XMLNode extends java.lang.Object implements org.w3c.dom.Node, oracle.xml.parser.v2.XMLConstants, java.lang.Cloneable, java.io.Serializable

```
java.lang.Object
  |
  +--oracle.xml.parser.v2.XMLNode
```

Direct Known Subclasses

AttrDecl, oracle.xml.parser.v2.CharData, DTD, ElementDecl, XMLAttr, XMLDocument, XMLDocumentFragment, XMLElement, XMLEntityReference

All Implemented Interfaces

java.lang.Cloneable, org.w3c.dom.Node, java.io.Serializable, oracle.xml.parser.v2.XMLConstants

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.

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.

Member Summary

Fields	
ATTRDECL	A attribute declaration node
ELEMENTDECL	An element declaration node.
Methods	
appendChild(Node)	Adds the node newChild to the end of the list of children of this node.
cloneNode(boolean)	Returns a duplicate of this node, i.e., serves as a generic copy constructor for nodes.

Member Summary	
getAttributes()	Gets a `NamedNodeMap` containing the attributes of this node (if it is an `Element`) or `null` otherwise.
getChildNodes()	Gets a `NodeList` that contains all children of this node.
getFirstChild()	Gets the first child of this node.
getLastChild()	Gets the last child of this node.
getNextSibling()	Gets The node immediately following this node.
getNodeName()	Gets the name of this node, depending on its type
getNodeType()	Gets a code representing the type of the underlying object
getNodeValue()	Gets the value of this node, depending on its type
getOwnerDocument()	Gets the `Document` object associated with this node.
getParentNode()	Gets the parent of this node.
getPreviousSibling()	Gets the node immediately preceding this node.
hasChildNodes()	This is a convenience method to allow easy determination of whether a node has any children.
insertBefore(Node, Node)	Inserts the node `newChild` before the existing child node `refChild`.
print(OutputStream)	Writes the contents of this node to the given output stream.
print(OutputStream, String)	Writes the contents of this node to the given output stream.
print(PrintWriter)	Writes the contents of this node using the given print writer.
removeChild(Node)	Removes the child node indicated by `oldChild` from the list of children, and returns it.
replaceChild(Node, Node)	Replaces the child node `oldChild` with `newChild` in the list of children, and returns the `oldChild` node.
selectNodes(String, NSResolver)	Selects nodes from the tree which match the given pattern
selectSingleNode(String, NSResolver)	Selects the first node from the tree that matches the given pattern
setNodeValue(String)	Sets the value of this node, depending on its type
transformNode(XSLStylesheet)	Transforms a node in the tree using the given stylesheet
valueOf(String, NSResolver)	Selects the value of the first node from the tree that matches the given pattern

Inherited Member Summary

Fields inherited from interface Node

ATTRIBUTE_NODE, CDATA_SECTION_NODE, COMMENT_NODE, DOCUMENT_FRAGMENT_NODE, DOCUMENT_NODE, DOCUMENT_TYPE_NODE, ELEMENT_NODE, ENTITY_NODE, ENTITY_REFERENCE_NODE, NOTATION_NODE, PROCESSING_INSTRUCTION_NODE, TEXT_NODE

Fields inherited from interface oracle.xml.parser.v2.XMLConstants

AMP, ASTERISK, cANY, cATTLIST, cCDATA, cCDATAEND, cCDATASTART, cCOMMENTEND, cCOMMENTSTART, cDECCREF, cDECLSTART, cDOCTYPE, cELEMENT, cEMPTY, cEMPTYTAGEND, cENCODING, cENDTAGSTART, cENTITIES, cENTITY, cFIXED, cHEXCREF, cID, cIDREF, cIDREFS, cIGNORE, cIMPLIED, cINCLUDE, cNDATA, cNMTOKEN, cNMTOKENS, cNOTATION, COLON, COMMA, cPIEND, cPISTART, cPUBLIC, cREQUIRED, cSTANDALONE, cSYSTEM, cVERSION, cXML, DOUBLEQUOTE, EOF, EQ, ERROR, FATAL_ERROR, FDIGIT, FLETTER, FMISCNAME, FSTARTNAME, FWHITESPACE, HASH, ICOUNT, ISTART, LEFTSQB, LPAREN, nameCDATA, nameCOMMENT, nameDOCUMENT, nameDOCUMENTFRAGMENT, nameENCODING, nameNameSpace, nameSpaceSeparator, nameSTANDALONE, nameTEXT, nameVERSION, nameXML, nameXMLLang, nameXMLNamespace, nameXMLNSNamespace, nameXMLSpace, nameXSLPI, NONVALIDATING, OR, PERCENT, PLUS, QMARK, QUOTE, RIGHTSQB, RPAREN, SEMICOLON, SLASH, TAGEND, TAGSTART, VALIDATING, WARNING

Methods inherited from class java.lang.Object

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

Fields

ATTRDECL

`public static final short ATTRDECL`
A attribute declaration node

ELEMENTDECL

`public static final short ELEMENTDECL`
An element declaration node.

Methods

appendChild(Node)

`public` org.w3c.dom.Node `appendChild`(org.w3c.dom.Node `newChild`)
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.

Specified By

org.w3c.dom.Node.appendChild(Node) in interface org.w3c.dom.Node

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

org.w3c.dom.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)

public org.w3c.dom.Node cloneNode(boolean deep)
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.

Specified By

org.w3c.dom.Node.cloneNode(boolean) in interface org.w3c.dom.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.

getAttributes()

public org.w3c.dom.NamedNodeMap getAttributes()
Gets a NamedNodeMap containing the attributes of this node (if it is an Element) or null otherwise.

Specified By

org.w3c.dom.Node.getAttributes() in interface org.w3c.dom.Node

Returns

the attributes of this node

getChildNodes()

`public` org.w3c.dom.NodeList `getChildNodes()`
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.

Specified By

org.w3c.dom.Node.getChildNodes() in interface org.w3c.dom.Node

Returns

The children of this node

getFirstChild()

`public` org.w3c.dom.Node `getFirstChild()`
Gets the first child of this node. If there is no such node, this returns `null`.

Specified By

org.w3c.dom.Node.getFirstChild() in interface org.w3c.dom.Node

Returns

The first child of this node

getLastChild()

`public` org.w3c.dom.Node `getLastChild()`
Gets the last child of this node. If there is no such node, this returns `null`.

Specified By

org.w3c.dom.Node.getLastChild() in interface org.w3c.dom.Node

Returns

The last child of this node

getNextSibling()

`public` org.w3c.dom.Node `getNextSibling()`
Gets The node immediately following this node. If there is no such node, this returns `null`.

Specified By

org.w3c.dom.Node.getNextSibling() in interface org.w3c.dom.Node

Returns

the next node

getNodeName()

`public java.lang.String getNodeName()`
Gets the name of this node, depending on its type

Specified By

org.w3c.dom.Node.getNodeName() in interface org.w3c.dom.Node

Returns

Name of this node

getNodeType()

`public short getNodeType()`
Gets a code representing the type of the underlying object

Specified By

org.w3c.dom.Node.getNodeType() in interface org.w3c.dom.Node

Returns

type of the node

getNodeValue()

`public java.lang.String getNodeValue()`

Gets the value of this node, depending on its type

Specified By

org.w3c.dom.Node.getNodeValue() in interface org.w3c.dom.Node

Returns

Value of this node

Throws

org.w3c.dom.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()

`public` org.w3c.dom.Document `getOwnerDocument()`
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`.

Specified By

org.w3c.dom.Node.getOwnerDocument() in interface org.w3c.dom.Node

Returns

The document associated with this node

getParentNode()

`public` org.w3c.dom.Node `getParentNode()`
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`.

Specified By

org.w3c.dom.Node.getParentNode() in interface org.w3c.dom.Node

Returns

The parent of this node

getPreviousSibling()

public org.w3c.dom.Node getPreviousSibling()
Gets the node immediately preceding this node. If there is no such node, this returns null.

Specified By

org.w3c.dom.Node.getPreviousSibling() in interface org.w3c.dom.Node

Returns

the previous node

hasChildNodes()

public boolean hasChildNodes()
This is a convenience method to allow easy determination of whether a node has any children.

Specified By

org.w3c.dom.Node.hasChildNodes() in interface org.w3c.dom.Node

Returns

true if the node has any children, false if the node has no children.

insertBefore(Node, Node)

public org.w3c.dom.Node insertBefore(org.w3c.dom.Node newChild, org.w3c.dom.Node refChild)
Inserts the node newChild before the existing child node refChild. 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.

Specified By

org.w3c.dom.Node.insertBefore(Node, Node) in interface org.w3c.dom.Node

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

org.w3c.dom.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)

```
public void print(java.io.OutputStream out)
```

Writes the contents of this node to the given output stream.

Parameters

`out` - `OutputStream` to write to

Throws

`IOException` - if an error occurs

print(OutputStream, String)

```
public void print(java.io.OutputStream out, java.lang.String enc)
```

Writes the contents of this node 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

print(PrintWriter)

```
public void print(java.io.PrintWriter out)
```

Writes the contents of this node using the given print writer.

Parameters

`out` - `PrintWriter` to use

Throws

`IOException` - if an error occurs

removeChild(Node)

`public` org.w3c.dom.Node `removeChild`(org.w3c.dom.Node `oldChild`)
Removes the child node indicated by `oldChild` from the list of children, and returns it.

Specified By

org.w3c.dom.Node.removeChild(Node) in interface org.w3c.dom.Node

Parameters

`oldChild` - The node being removed.

Returns

The node removed.

Throws

org.w3c.dom.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)

`public` org.w3c.dom.Node `replaceChild`(org.w3c.dom.Node `newChild`, org.w3c.dom.Node `oldChild`)
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.

Specified By

org.w3c.dom.Node.replaceChild(Node, Node) in interface org.w3c.dom.Node

Parameters

`newChild` - The new node to put in the child list.

`oldChild` - The node being replaced in the list.

Returns

The node replaced.

Throws

org.w3c.dom.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.

selectNodes(String, NSResolver)

`public org.w3c.dom.NodeList selectNodes(java.lang.String pattern, NSResolver nsr)`
Selects nodes from the tree which match the given pattern

Parameters

`pattern` - XSL pattern to match

`nsr` - NSResolver to resolve any prefixes that occur in given pattern

Returns

a list of matching nodes

Throws

XSLException - Raised if there is an error while doing the match

selectSingleNode(String, NSResolver)

`public org.w3c.dom.Node selectSingleNode(java.lang.String pattern, NSResolver nsr)`
Selects the first node from the tree that matches the given pattern

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

setNodeValue(String)

`public void setNodeValue(java.lang.String nodeValue)`
Sets the value of this node, depending on its type

Specified By
org.w3c.dom.Node.setNodeValue(String) in interface org.w3c.dom.Node

Throws
org.w3c.dom.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.

transformNode(XSLStylesheet)

`public org.w3c.dom.DocumentFragment transformNode(XSLStylesheet xsl)`
Transforms a node in the tree using the given stylesheet

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, NSResolver)

`public java.lang.String valueOf(java.lang.String pattern, NSResolver nsr)`
Selects the value of the first node from the tree that matches the given pattern

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

XMLParseException

Syntax

```
public class XMLParseException extends org.xml.sax.SAXParseException

java.lang.Object
  |
  +--java.lang.Throwable
        |
        +--java.lang.Exception
              |
              +--org.xml.sax.SAXException
                    |
                    +--org.xml.sax.SAXParseException
                          |
                          +--oracle.xml.parser.v2.XMLParseException
```

All Implemented Interfaces

```
java.io.Serializable
```

Description

Indicates that a parsing exception occurred while processing an XML document

Member Summary

Fields	
ERROR	Code for non-fatal error
FATAL_ERROR	Code for fatal error
WARNING	Code for warning
Constructors	
XMLParseException(String, String, String, int, int, int)	
Methods	
getColumnNumber(int)	Get the column number of error at specified index
getException(int)	Get the exception (if exists) that occured in error at specified index
getLineNumber(int)	Get the line number of error at specified index

Member Summary	
getMessage(int)	Get the error message at specified index
getMessageType(int)	Get the type of the error message at specified index
getNumMessages()	Return the total number of errors / warnings found during parsing
getPublicId(int)	Get the public ID of input when error at specified index occured
getSystemId(int)	Get the system ID of input when error at specified index occured

Inherited Member Summary

Methods inherited from interface SAXParseException

getColumnNumber(), getLineNumber(), getPublicId(), getSystemId()

Methods inherited from interface SAXException

getException(), getMessage(), toString()

Methods inherited from class java.lang.Throwable

fillInStackTrace, getLocalizedMessage, printStackTrace, printStackTrace, printStackTrace

Methods inherited from class java.lang.Object

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

Fields

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

Constructor

XMLParseException(String, String, String, int, int, int)

```
public  XMLParseException(java.lang.String mesg, java.lang.String pubId,
java.lang.String sysId, int line, int col, int type)
```

Methods

getColumnNumber(int)

```
public int getColumnNumber(int i)
```
Get the column number of error at specified index

Returns

The column number

getException(int)

```
public java.lang.Exception getException(int i)
```
Get the exception (if exists) that occured in error at specified index

Returns

The exception

getLineNumber(int)

```
public int getLineNumber(int i)
```
Get the line number of error at specified index

Returns

The line number

getMessage(int)

```
public java.lang.String getMessage(int i)
```
Get the error message at specified index

Returns

The error message

getMessageType(int)

```
public int getMessageType(int i)
```
Get the type of the error message at specified index

Returns

The error message type

getNumMessages()

```
public int getNumMessages()
```
Return the total number of errors / warnings found during parsing

Returns

The number of errors / warnings

getPublicId(int)

```
public java.lang.String getPublicId(int i)
```
Get the public ID of input when error at specified index occured

Returns

The public ID

getSystemId(int)

```
public java.lang.String getSystemId(int i)
```
Get the system ID of input when error at specified index occured

Returns

The system ID

XMLParser

Syntax

```
public abstract class XMLParser extends java.lang.Object implements
oracle.xml.parser.v2.XMLConstants

java.lang.Object
  |
  +--oracle.xml.parser.v2.XMLParser
```

Direct Known Subclasses

DOMParser, SAXParser

All Implemented Interfaces

oracle.xml.parser.v2.XMLConstants

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).

Member Summary

Methods	
getReleaseVersion()	Returns the release version of the Oracle XML Parser
getValidationMode()	Returns the validation mode
parse(InputSource)	Parses the XML from given input source
parse(InputStream)	Parses the XML from given input stream.
parse(Reader)	Parses the XML from given input stream.
parse(String)	Parses the XML from the URL indicated
parse(URL)	Parses the XML document pointed to by the given URL and creates the corresponding XML document hierarchy.
setBaseURL(URL)	Set the base URL for loading external enitites and DTDs.
setDoctype(DTD)	Set the DTD

Member Summary	
setLocale(Locale)	Applications can use this to set the locale for error reporting.
setPreserveWhitespace(boolean)	Set the white space preserving mode
setValidationMode(boolean)	Set the validation mode

Inherited Member Summary

Fields inherited from interface oracle.xml.parser.v2.XMLConstants

AMP, ASTERISK, cANY, cATTLIST, cCDATA, cCDATAEND, cCDATASTART, cCOMMENTEND, cCOMMENTSTART, cDECCREF, cDECLSTART, cDOCTYPE, cELEMENT, cEMPTY, cEMPTYTAGEND, cENCODING, cENDTAGSTART, cENTITIES, cENTITY, cFIXED, cHEXCREF, cID, cIDREF, cIDREFS, cIGNORE, cIMPLIED, cINCLUDE, cNDATA, cNMTOKEN, cNMTOKENS, cNOTATION, COLON, COMMA, cPIEND, cPISTART, cPUBLIC, cREQUIRED, cSTANDALONE, cSYSTEM, cVERSION, cXML, DOUBLEQUOTE, EOF, EQ, ERROR, FATAL_ERROR, FDIGIT, FLETTER, FMISCNAME, FSTARTNAME, FWHITESPACE, HASH, ICOUNT, ISTART, LEFTSQB, LPAREN, nameCDATA, nameCOMMENT, nameDOCUMENT, nameDOCUMENTFRAGMENT, nameENCODING, nameNameSpace, nameSpaceSeparator, nameSTANDALONE, nameTEXT, nameVERSION, nameXML, nameXMLLang, nameXMLNamespace, nameXMLNSNamespace, nameXMLSpace, nameXSLPI, NONVALIDATING, OR, PERCENT, PLUS, QMARK, QUOTE, RIGHTSQB, RPAREN, SEMICOLON, SLASH, TAGEND, TAGSTART, VALIDATING, WARNING

Methods inherited from class java.lang.Object

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

Methods

getReleaseVersion()

`public static java.lang.String getReleaseVersion()`
Returns the release version of the Oracle XML Parser

Returns

the release version string

getValidationMode()

`public boolean getValidationMode()`
Returns the validation mode

Returns

`true` if the XML parser is validating `false` if not

parse(InputSource)

`public final void parse(org.xml.sax.InputSource in)`
Parses the XML from given input source

Parameters

`in` - the `org.xml.sax.InputSouce` to parse

Throws

XMLParseException - if syntax or other error encountered.

org.xml.sax.SAXException - Any SAX exception, possibly wrapping another exception.

`IOException` - IO Error.

parse(InputStream)

`public final 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.

org.xml.sax.SAXException - Any SAX exception, possibly wrapping another exception.

`IOException` - IO Error.

See Also: setBaseURL(URL):

parse(Reader)

`public final 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.

org.xml.sax.SAXException - Any SAX exception, possibly wrapping another exception.

`IOException` - IO Error.

See Also

setBaseURL(URL)

parse(String)

`public final 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.

org.xml.sax.SAXException - Any SAX exception, possibly wrapping another exception.

`IOException` - IO Error.

parse(URL)

`public final 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.

org.xml.sax.SAXException - Any SAX exception, possibly wrapping another exception.

`IOException` - IO Error.

setBaseURL(URL)

```
public void setBaseURL(java.net.URL url)
```
Set the base URL for loading external enitites and DTDs. This method should to be called if the parse(InputStream) is used to parse the XML Document

Parameters

`url` - The base URL

setDoctype(DTD)

```
public void setDoctype(DTD dtd)
```
Set the DTD

Parameters

`dtd` - `DTD` to set and used while parsing

setLocale(Locale)

```
public void setLocale(java.util.Locale locale)
```
Applications can use this to set the locale for error reporting.

Parameters

`locale` - `Locale` to set

Throws

org.xml.sax.SAXException - A `SAXException` could be thrown.

See Also

org.xml.sax.Parser.setLocale(Locale)

setPreserveWhitespace(boolean)

```
public void setPreserveWhitespace(boolean flag)
```
Set the white space preserving mode

Parameters

`flag` - preserving mode

setValidationMode(boolean)

```
public void setValidationMode(boolean yes)
```

Set the validation mode

Parameters

`yes` - determines whether the XML parser should be validating

XMLPI

Syntax

```
public class XMLPI extends oracle.xml.parser.v2.CharData implements
org.w3c.dom.ProcessingInstruction, java.io.Serializable

java.lang.Object
  |
  +--XMLNode
        |
        +--oracle.xml.parser.v2.CharData
              |
              +--oracle.xml.parser.v2.XMLPI
```

All Implemented Interfaces

org.w3c.dom.CharacterData, java.lang.Cloneable, org.w3c.dom.Node, org.w3c.dom.ProcessingInstruction, java.io.Serializable, oracle.xml.parser.v2.XMLConstants

Description

This class implements the DOM Processing Instruction interface.

See Also

org.w3c.dom.ProcessingInstruction, NodeFactory, setNodeFactory(NodeFactory)

Member Summary

Constructors	
XMLPI(String, String)	Creates a new ProcessingInstruction node with the given target and the data.
Methods	
getTarget()	Returns the target of this PI.

Inherited Member Summary

Fields inherited from class XMLNode

Inherited Member Summary

AMP, ASTERISK, ATTRDECL, cANY, cATTLIST, cCDATA, cCDATAEND, cCDATASTART, cCOMMENTEND, cCOMMENTSTART, cDECCREF, cDECLSTART, cDOCTYPE, cELEMENT, cEMPTY, cEMPTYTAGEND, cENCODING, cENDTAGSTART, cENTITIES, cENTITY, cFIXED, cHEXCREF, cID, cIDREF, cIDREFS, cIGNORE, cIMPLIED, cINCLUDE, cNDATA, cNMTOKEN, cNMTOKENS, cNOTATION, COLON, COMMA, cPIEND, cPISTART, cPUBLIC, cREQUIRED, cSTANDALONE, cSYSTEM, cVERSION, cXML, DOUBLEQUOTE, ELEMENTDECL, EOF, EQ, ERROR, FATAL_ERROR, FDIGIT, FLETTER, FMISCNAME, FSTARTNAME, FWHITESPACE, HASH, ICOUNT, ISTART, LEFTSQB, LPAREN, nameCDATA, nameCOMMENT, nameDOCUMENT, nameDOCUMENTFRAGMENT, nameENCODING, nameNameSpace, nameSpaceSeparator, nameSTANDALONE, nameTEXT, nameVERSION, nameXML, nameXMLLang, nameXMLNamespace, nameXMLNSNamespace, nameXMLSpace, nameXSLPI, NONVALIDATING, OR, PERCENT, PLUS, QMARK, QUOTE, RIGHTSQB, RPAREN, SEMICOLON, SLASH, TAGEND, TAGSTART, VALIDATING, WARNING

Fields inherited from interface Node

ATTRIBUTE_NODE, CDATA_SECTION_NODE, COMMENT_NODE, DOCUMENT_FRAGMENT_NODE, DOCUMENT_NODE, DOCUMENT_TYPE_NODE, ELEMENT_NODE, ENTITY_NODE, ENTITY_REFERENCE_NODE, NOTATION_NODE, PROCESSING_INSTRUCTION_NODE, TEXT_NODE

Fields inherited from interface oracle.xml.parser.v2.XMLConstants

AMP, ASTERISK, cANY, cATTLIST, cCDATA, cCDATAEND, cCDATASTART, cCOMMENTEND, cCOMMENTSTART, cDECCREF, cDECLSTART, cDOCTYPE, cELEMENT, cEMPTY, cEMPTYTAGEND, cENCODING, cENDTAGSTART, cENTITIES, cENTITY, cFIXED, cHEXCREF, cID, cIDREF, cIDREFS, cIGNORE, cIMPLIED, cINCLUDE, cNDATA, cNMTOKEN, cNMTOKENS, cNOTATION, COLON, COMMA, cPIEND, cPISTART, cPUBLIC, cREQUIRED, cSTANDALONE, cSYSTEM, cVERSION, cXML, DOUBLEQUOTE, EOF, EQ, ERROR, FATAL_ERROR, FDIGIT, FLETTER, FMISCNAME, FSTARTNAME, FWHITESPACE, HASH, ICOUNT, ISTART, LEFTSQB, LPAREN, nameCDATA, nameCOMMENT, nameDOCUMENT, nameDOCUMENTFRAGMENT, nameENCODING, nameNameSpace, nameSpaceSeparator, nameSTANDALONE, nameTEXT, nameVERSION, nameXML, nameXMLLang, nameXMLNamespace, nameXMLNSNamespace, nameXMLSpace, nameXSLPI, NONVALIDATING, OR, PERCENT, PLUS, QMARK, QUOTE, RIGHTSQB, RPAREN, SEMICOLON, SLASH, TAGEND, TAGSTART, VALIDATING, WARNING

Methods inherited from class oracle.xml.parser.v2.CharData

appendData, deleteData, getData, getLength, insertData, replaceData, setData, setNodeValue, substringData

Methods inherited from class XMLNode

appendChild(Node), cloneNode(boolean), getAttributes(), getChildNodes(), getFirstChild(), getLastChild(), getNextSibling(), getNodeName(), getNodeType(), getNodeValue(), getOwnerDocument(), getParentNode(), getPreviousSibling(), hasChildNodes(), insertBefore(Node, Node), print(OutputStream), print(OutputStream, String), print(PrintWriter), removeChild(Node), replaceChild(Node, Node), selectNodes(String, NSResolver), selectSingleNode(String, NSResolver), transformNode(XSLStylesheet), valueOf(String, NSResolver)

Methods inherited from class java.lang.Object

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

Methods inherited from interface ProcessingInstruction

getData(), setData(String)

Methods inherited from interface Node

Inherited Member Summary

appendChild(Node), cloneNode(boolean), getAttributes(), getChildNodes(), getFirstChild(), getLastChild(), getNextSibling(), getNodeName(), getNodeType(), getNodeValue(), getOwnerDocument(), getParentNode(), getPreviousSibling(), hasChildNodes(), insertBefore(Node, Node), removeChild(Node), replaceChild(Node, Node), setNodeValue(String)

Methods inherited from interface CharacterData

appendData(String), deleteData(int, int), getLength(), insertData(int, String), replaceData(int, int, String), substringData(int, int)

Constructor

XMLPI(String, String)

```
public  XMLPI(java.lang.String target, java.lang.String data)
```

Creates a new ProcessingInstruction node with the given target and the data.

Parameters

`target` - The target of this PI

`data` - The content of this PI

Methods

getTarget()

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

Returns the target of this PI. XML defines this as the first token following markup that begins the processing instruction.

Specified By

org.w3c.dom.ProcessingInstruction.getTarget() in interface org.w3c.dom.ProcessingInstruction

Returns

The target of the PI.

XMLText

Syntax

public class XMLText extends oracle.xml.parser.v2.CharData implements org.w3c.dom.Text, java.io.Serializable

```
java.lang.Object
  |
  +--XMLNode
        |
        +--oracle.xml.parser.v2.CharData
              |
              +--oracle.xml.parser.v2.XMLText
```

Direct Known Subclasses:

XMLCDATA

All Implemented Interfaces

org.w3c.dom.CharacterData, java.lang.Cloneable, org.w3c.dom.Node, java.io.Serializable, org.w3c.dom.Text, oracle.xml.parser.v2.XMLConstants

Description

This class implements the DOM Text interface.

See Also

org.w3c.dom.Text, NodeFactory, setNodeFactory(NodeFactory)

Member Summary

Constructors	
XMLText(String)	
Methods	
getNodeValue()	
splitText(int)	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.

Inherited Member Summary

Fields inherited from class XMLNode

AMP, ASTERISK, ATTRDECL, cANY, cATTLIST, cCDATA, cCDATAEND, cCDATASTART, cCOMMENTEND, cCOMMENTSTART, cDECCREF, cDECLSTART, cDOCTYPE, cELEMENT, cEMPTY, cEMPTYTAGEND, cENCODING, cENDTAGSTART, cENTITIES, cENTITY, cFIXED, cHEXCREF, cID, cIDREF, cIDREFS, cIGNORE, cIMPLIED, cINCLUDE, cNDATA, cNMTOKEN, cNMTOKENS, cNOTATION, COLON, COMMA, cPIEND, cPISTART, cPUBLIC, cREQUIRED, cSTANDALONE, cSYSTEM, cVERSION, cXML, DOUBLEQUOTE, ELEMENTDECL, EOF, EQ, ERROR, FATAL_ERROR, FDIGIT, FLETTER, FMISCNAME, FSTARTNAME, FWHITESPACE, HASH, ICOUNT, ISTART, LEFTSQB, LPAREN, nameCDATA, nameCOMMENT, nameDOCUMENT, nameDOCUMENTFRAGMENT, nameENCODING, nameNameSpace, nameSpaceSeparator, nameSTANDALONE, nameTEXT, nameVERSION, nameXML, nameXMLLang, nameXMLNamespace, nameXMLNSNamespace, nameXMLSpace, nameXSLPI, NONVALIDATING, OR, PERCENT, PLUS, QMARK, QUOTE, RIGHTSQB, RPAREN, SEMICOLON, SLASH, TAGEND, TAGSTART, VALIDATING, WARNING

Fields inherited from interface Node

ATTRIBUTE_NODE, CDATA_SECTION_NODE, COMMENT_NODE, DOCUMENT_FRAGMENT_NODE, DOCUMENT_NODE, DOCUMENT_TYPE_NODE, ELEMENT_NODE, ENTITY_NODE, ENTITY_REFERENCE_NODE, NOTATION_NODE, PROCESSING_INSTRUCTION_NODE, TEXT_NODE

Fields inherited from interface oracle.xml.parser.v2.XMLConstants

AMP, ASTERISK, cANY, cATTLIST, cCDATA, cCDATAEND, cCDATASTART, cCOMMENTEND, cCOMMENTSTART, cDECCREF, cDECLSTART, cDOCTYPE, cELEMENT, cEMPTY, cEMPTYTAGEND, cENCODING, cENDTAGSTART, cENTITIES, cENTITY, cFIXED, cHEXCREF, cID, cIDREF, cIDREFS, cIGNORE, cIMPLIED, cINCLUDE, cNDATA, cNMTOKEN, cNMTOKENS, cNOTATION, COLON, COMMA, cPIEND, cPISTART, cPUBLIC, cREQUIRED, cSTANDALONE, cSYSTEM, cVERSION, cXML, DOUBLEQUOTE, EOF, EQ, ERROR, FATAL_ERROR, FDIGIT, FLETTER, FMISCNAME, FSTARTNAME, FWHITESPACE, HASH, ICOUNT, ISTART, LEFTSQB, LPAREN, nameCDATA, nameCOMMENT, nameDOCUMENT, nameDOCUMENTFRAGMENT, nameENCODING, nameNameSpace, nameSpaceSeparator, nameSTANDALONE, nameTEXT, nameVERSION, nameXML, nameXMLLang, nameXMLNamespace, nameXMLNSNamespace, nameXMLSpace, nameXSLPI, NONVALIDATING, OR, PERCENT, PLUS, QMARK, QUOTE, RIGHTSQB, RPAREN, SEMICOLON, SLASH, TAGEND, TAGSTART, VALIDATING, WARNING

Methods inherited from class oracle.xml.parser.v2.CharData

appendData, deleteData, getData, getLength, insertData, replaceData, setData, setNodeValue, substringData

Methods inherited from class XMLNode

appendChild(Node), cloneNode(boolean), getAttributes(), getChildNodes(), getFirstChild(), getLastChild(), getNextSibling(), getNodeName(), getNodeType(), getOwnerDocument(), getParentNode(), getPreviousSibling(), hasChildNodes(), insertBefore(Node, Node), print(OutputStream), print(OutputStream, String), print(PrintWriter), removeChild(Node), replaceChild(Node, Node), selectNodes(String, NSResolver), selectSingleNode(String, NSResolver), transformNode(XSLStylesheet), valueOf(String, NSResolver)

Methods inherited from class java.lang.Object

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

Methods inherited from interface CharacterData

Inherited Member Summary

appendData(String), deleteData(int, int), getData(), getLength(), insertData(int, String), replaceData(int, int, String), setData(String), substringData(int, int)

Methods inherited from interface Node

appendChild(Node), cloneNode(boolean), getAttributes(), getChildNodes(), getFirstChild(), getLastChild(), getNextSibling(),getNodeName(), getNodeType(), getOwnerDocument(), getParentNode(), getPreviousSibling(), hasChildNodes(), insertBefore(Node, Node), removeChild(Node), replaceChild(Node, Node), setNodeValue(`String`)

Constructor

XMLText(String)

```
public  XMLText(java.lang.String text)
```

Methods

getNodeValue()

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

Specified By

org.w3c.dom.Node.getNodeValue() in interface org.w3c.dom.Node

Overrides

getNodeValue() in class XMLNode

splitText(int)

`public` org.w3c.dom.Text `splitText(int offset)`
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.

Specified By

org.w3c.dom.Text.splitText(int) in interface org.w3c.dom.Text

Parameters

`offset` - Offset at which to split, starting from 0

Returns

New `Text` node

Throws

org.w3c.dom.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.

XMLToken

Syntax

```
public interface XMLToken
```

Description

Basic interface for XMLToken

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.

Member Summary

Fields	
AttListDecl	AttListDecl ::= '<' '!' 'ATTLIST' S Name AttDef* S? '>'
AttName	AttName ::= Name
Attribute	Attribute ::= AttName Eq AttValue
AttValue	AttValue ::= '"' ([^<&"] | Reference)* '"'
CDSect	CDSect ::= CDStart CData CDEnd
CharData	CharData ::= [^<&]* - ([^<&]* ']]>' [^<&]*)

Member Summary	
Comment	Comment ::= '<' '!' '--' ((Char - '-') | ('-' (Char - '-')))* '-->'
DTDName	DTDName ::= name
ElemDeclName	ElemDeclName ::= name
elementdecl	elementdecl ::= '<' '!ELEMENT' S ElemDeclName S contentspec S? '>'
EmptyElemTag	EmptyElemTag ::= '<' STagName (S Attribute)* S? '/' '>'
EntityDecl	EntityDecl ::= '<' '!' ENTITY' S EntityDeclName S EntityDef S? '>'
EntityDeclName	EntityValue ::= "" ([^%&"] | PEReference | Reference)* ""
EntityValue	EntityDeclName ::= Name
ETag	ETag ::= '<' '/' ETagName S? '>'
ETagName	ETagName ::= Name
ExternalID	ExternalID ::= 'SYSTEM' S SystemLiteral
NotationDecl	NotationDecl ::= '<' '!NOTATION' S Name S (ExternalID | PublicID) S? '>'
PI	PI ::= '<' '?' PITarget (S (Char* - (Char* '?>' Char*)))? '?' '>'
PITarget	PITarget ::= Name - (('X' | 'x') ('M' | 'm') ('L' | 'l'))
Reference	Reference ::= EntityRef | CharRef | PEReference
STag	STag ::= '<' STagName (S Attribute)* S? '>'
STagName	STagName ::= Name
TextDecl	TextDecl ::= '<' '?' 'xml' VersionInfo? EncodingDecl S? '?>'
XMLDecl	XMLDecl ::= '<' '?' 'xml' VersionInfo EncodingDecl? SDDecl? S? '?' '>'
Methods	
token(int, String)	The interface call-back method.

Fields

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)* "'"

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 PEDef S? '>'

EntityDef ::= EntityValue | (ExternalID NDataDecl?)

PEDef ::= 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*)))? '?' '>'

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? SDDecl? S? '?' '>'

Methods

token(int, String)

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

The interface call-back method. Receives an XML token and it's corresponding value

Parameters

`token` - The XML token constant as specified in the interface.

`value` - The corresponding substring from the parsed text.

XMLTokenizer

Syntax

```
public class XMLTokenizer extends Package oracle.xml.parser.v2 implements
oracle.xml.parser.v2.XMLConstants

java.lang.Object
  |
  +--org.xml.sax.HandlerBase
        |
        +--Package oracle.xml.parser.v2
              |
              +--oracle.xml.parser.v2.XMLTokenizer
```

All Implemented Interfaces

org.xml.sax.DocumentHandler, org.xml.sax.DTDHandler, org.xml.sax.EntityResolver, org.xml.sax.ErrorHandler, oracle.xml.parser.v2.XMLConstants, XMLDocumentHandler

Description

This class implements an eXtensible Markup Language (XML) 1.0 parser according to the World Wide Web Consortium (W3C) recommendation.

Member Summary

Constructors	
XMLTokenizer()	Creates a new Tokenizer object.
XMLTokenizer(XMLToken)	Creates a new Tokenizer object.
Methods	
parseDocument()	Document ::= Prolog Element Misc*
setErrorHandler(ErrorHandler)	Applications can use this to register a new error event handler.
setErrorStream(OutputStream)	Register a output stream for errors
setToken(int, boolean)	Applications can use this to register a new token for XML tokenizer.
setTokenHandler(XMLToken)	Applications can use this to register a new XML tokenizer event handler.
tokenize(InputSource)	Tokenizes the XML from given input source
tokenize(InputStream)	Tokenizes the XML from given input stream.

Member Summary	
tokenize(Reader)	Tokenizes the XML from given input stream.
tokenize(String)	Tokenizes the XML from the URL indicated
tokenize(URL)	Tokenizes the XML document pointed to by the given URL and creates the corresponding XML document hierarchy.

Inherited Member Summary

Fields inherited from interface oracle.xml.parser.v2.XMLConstants

AMP, ASTERISK, cANY, cATTLIST, cCDATA, cCDATAEND, cCDATASTART, cCOMMENTEND, cCOMMENTSTART, cDECCREF, cDECLSTART, cDOCTYPE, cELEMENT, cEMPTY, cEMPTYTAGEND, cENCODING, cENDTAGSTART, cENTITIES, cENTITY, cFIXED, cHEXCREF, cID, cIDREF, cIDREFS, cIGNORE, cIMPLIED, cINCLUDE, cNDATA, cNMTOKEN, cNMTOKENS, cNOTATION, COLON, COMMA, cPIEND, cPISTART, cPUBLIC, cREQUIRED, cSTANDALONE, cSYSTEM, cVERSION, cXML, DOUBLEQUOTE, EOF, EQ, ERROR, FATAL_ERROR, FDIGIT, FLETTER, FMISCNAME, FSTARTNAME, FWHITESPACE, HASH, ICOUNT, ISTART, LEFTSQB, LPAREN, nameCDATA, nameCOMMENT, nameDOCUMENT, nameDOCUMENTFRAGMENT, nameENCODING, nameNameSpace, nameSpaceSeparator, nameSTANDALONE, nameTEXT, nameVERSION, nameXML, nameXMLLang, nameXMLNamespace, nameXMLNSNamespace, nameXMLSpace, nameXSLPI, NONVALIDATING, OR, PERCENT, PLUS, QMARK, QUOTE, RIGHTSQB, RPAREN, SEMICOLON, SLASH, TAGEND, TAGSTART, VALIDATING, WARNING

Methods inherited from class Package oracle.xml.parser.v2

cDATASection(char[], int, int), comment(String), endDoctype(), endElement(NSName), setDoctype(DTD), setTextDecl(String, String), setXMLDecl(String, String, String), startElement(NSName, SAXAttrList)

Methods inherited from class HandlerBase

characters(char[], int, int), endDocument(), endElement(String), error(SAXParseException), fatalError(SAXParseException), ignorableWhitespace(char[], int, int), notationDecl(String, String, String), processingInstruction(String, String), resolveEntity(String, String), setDocumentLocator(Locator), startDocument(), startElement(String, AttributeList), unparsedEntityDecl(String, String, String, String), warning(SAXParseException)

Methods inherited from class java.lang.Object

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

Methods inherited from interface XMLDocumentHandler

cDATASection(char[ ], int, int), comment(String), endDoctype(), endElement(NSName), setDoctype(DTD), setTextDecl(String, String), setXMLDecl(String, String, String), startElement(NSName, SAXAttrList)

Methods inherited from interface DocumentHandler

characters(char[], int, int), endDocument(), endElement(String), ignorableWhitespace(char[], int, int), processingInstruction(String, String), setDocumentLocator(Locator), startDocument(), startElement(String, AttributeList)

Methods inherited from interface EntityResolver

Inherited Member Summary

resolveEntity(String, String)

Methods inherited from interface DTDHandler

notationDecl(String, String, String), unparsedEntityDecl(String, String, String, String)

Methods inherited from interface ErrorHandler

error(SAXParseException), fatalError(SAXParseException), warning(SAXParseException)

Constructors

XMLTokenizer()

`public XMLTokenizer()`
Creates a new Tokenizer object.

XMLTokenizer(XMLToken)

`public XMLTokenizer(XMLToken handler)`
Creates a new Tokenizer object.

Methods

parseDocument()

`public void parseDocument()`
Document ::= Prolog Element Misc*

setErrorHandler(ErrorHandler)

`public void setErrorHandler(`org.xml.sax.ErrorHandler `handler)`
Applications can use this to register a new error event handler. This replaces any previous setting for error handling.

Parameters

`handler` - `ErrorHandler` being registered

setErrorStream(OutputStream)

`public void setErrorStream(java.io.OutputStream out)`
Register a output stream for errors

setToken(int, boolean)

```
public void setToken(int token, boolean val)
```
Applications can use this to register a new token for XML tokenizer.

Parameters

`token` - `XMLToken` being set

setTokenHandler(XMLToken)

```
public void setTokenHandler(XMLToken handler)
```
Applications can use this to register a new XML tokenizer event handler.

Parameters

`handler` - `XMLToken` being registered

tokenize(InputSource)

```
public final void tokenize(org.xml.sax.InputSource in)
```
Tokenizes the XML from given input source

Parameters

`in` - the `org.xml.sax.InputSouce` to parse

Throws

XMLParseException - if syntax or other error encountered.

org.xml.sax.SAXException - Any SAX exception, possibly wrapping another exception.

`IOException` - IO Error.

tokenize(InputStream)

```
public final void tokenize(java.io.InputStream in)
```
Tokenizes the XML from given input stream.

Parameters

`in` - the `InputStream` containing XML data to parse.

Throws

XMLParseException - if syntax or other error encountered.

org.xml.sax.SAXException - Any SAX exception, possibly wrapping another exception.

`IOException` - IO Error.

See Also

setBaseURL(URL)

tokenize(Reader)

`public final void tokenize(java.io.Reader r)`
Tokenizes the XML from given input stream.

Parameters

`r` - the `Reader` containing XML data to parse.

Throws

XMLParseException - if syntax or other error encountered.

org.xml.sax.SAXException - Any SAX exception, possibly wrapping another exception.

`IOException` - IO Error.

See Also

setBaseURL(URL)

tokenize(String)

`public final void tokenize(java.lang.String in)`
Tokenizes the XML from the URL indicated

Parameters

`in` - the `String` containing the URL to parse from

Throws

XMLParseException - if syntax or other error encountered.

org.xml.sax.SAXException - Any SAX exception, possibly wrapping another exception.

`IOException` - IO Error.

tokenize(URL)

```
public final void tokenize(java.net.URL url)
```

Tokenizes 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.

org.xml.sax.SAXException - Any SAX exception, possibly wrapping another exception.

`IOException` - IO Error.

XSLException

Syntax

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

java.lang.Object
  |
  +--java.lang.Throwable
        |
        +--java.lang.Exception
              |
              +--oracle.xml.parser.v2.XSLException
```

All Implemented Interfaces

java.io.Serializable

Description

Indicates that an exception occurred during XSL tranformation

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

XSLProcessor

Syntax

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

java.lang.Object
  |
  +--oracle.xml.parser.v2.XSLProcessor
```

Description

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.

Member Summary

Constructors

XSLProcessor()	

Methods

processXSL(XSLStylesheet, InputStream, URL)	Transform input XML document using given InputStream and stylesheet.
processXSL(XSLStylesheet, Reader, URL)	Transform input XML document using given Reader and stylesheet.
processXSL(XSLStylesheet, URL, URL)	Transform input XML document using given URL and stylesheet.
processXSL(XSLStylesheet, XMLDocument)	Transform input XML document using given XMLDocument and stylesheet.
processXSL(XSLStylesheet, XMLDocumentFragment)	Transform input XML document using given XMLDocument and stylesheet.
processXSL(XSLStylesheet, XMLDocumentFragment, OutputStream)	Transform input XML using given XMLDocumentFragment and stylesheet.
processXSL(XSLStylesheet, XMLDocumentFragment, PrintWriter)	Transform input XML using given XMLDocumentFragment and stylesheet.
processXSL(XSLStylesheet, XMLDocument, OutputStream)	Transform input XML document using given XMLDocument and stylesheet.

Member Summary	
processXSL(XSLStylesheet, XMLDocument, PrintWriter)	Transform input XML document using given XMLDocument and stylesheet.
setErrorStream(OutputStream)	Creates an output stream for the output of warnings.
setLocale(Locale)	Applications can use this to set the locale for error reporting.
showWarnings(boolean)	Switch to determine whether to output warnings.

Inherited Member Summary
Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor

XSLProcessor()

```
public  XSLProcessor()
```

Methods

processXSL(XSLStylesheet, InputStream, URL)

```
public XMLDocumentFragment processXSL(XSLStylesheet xsl, java.io.InputStream
xml, java.net.URL ref)
```

Transforms input XML document using given InputStream and stylesheet.

Parameters

`xsl` - `XSLStylesheet` to be used for transformation

`xml` - XML input to be transformed (as a java.io.Inputstream)

`ref` - Reference URL to resolve external entities in input xml file

Returns

`XMLDocumentFragment`

Throws

XSLException - on error.

processXSL(XSLStylesheet, Reader, URL)

```
public XMLDocumentFragment processXSL(XSLStylesheet xsl, java.io.Reader xml,
java.net.URL ref)
```
Transform input XML document using given Reader and stylesheet.

Parameters

`xsl` - `XSLStylesheet` to be used for transformation

`xml` - XML input to be transformed (as a java.io.Reader)

`ref` - Reference URL to resolve external entities in input xml file

Returns

`XMLDocumentFragment`

Throws

XSLException - on error.

processXSL(XSLStylesheet, URL, URL)

```
public XMLDocumentFragment processXSL(XSLStylesheet xsl, java.net.URL xml,
java.net.URL ref)
```
Transform input XML document using given URL and stylesheet.

Parameters

`xsl` - `XSLStylesheet` to be used for transformation

`xml` - XML input to be transformed (as a java.net.URL)

`ref` - Reference URL to resolve external entities in input xml file

Returns

`XMLDocumentFragment`

Throws

XSLException - on error.

processXSL(XSLStylesheet, XMLDocument)

```
public XMLDocumentFragment processXSL(XSLStylesheet xsl, XMLDocument xml)
```
Transform input XML document using given XMLDocument and stylesheet.

Parameters

`xsl` - `XSLStylesheet` to be used for transformation

`xml` - XML input to be transformed (as a DOM Tree)

Returns

`XMLDocumentFragment`

Throws

XSLException - on error.

processXSL(XSLStylesheet, XMLDocumentFragment)

```
public XMLDocumentFragment processXSL(XSLStylesheet xsl, XMLDocumentFragment
inp)
```

Transform input XML document using given XMLDocument and stylesheet.

Parameters

`xsl` - `XSLStylesheet` to be used for transformation

`xml` - XML input to be transformed (as a DOM Tree)

Returns

`XMLDocumentFragment`

Throws

XSLException - on error.

processXSL(XSLStylesheet, XMLDocumentFragment, OutputStream)

```
public void processXSL(XSLStylesheet xsl, XMLDocumentFragment xml,
java.io.OutputStream out)
```

Transform input XML using given XMLDocumentFragment and stylesheet.

Parameters

`xsl` - `XSLStylesheet` to be used for transformation

`xml` - XML input to be transformed (as a DOM Tree)

`out` - Outputstream to which the result is printed

Throws

`XSLException,` - IOException on error.

processXSL(XSLStylesheet, XMLDocumentFragment, PrintWriter)

```
public void processXSL(XSLStylesheet xsl, XMLDocumentFragment xml,
java.io.PrintWriter pw)
```

Transform input XML using given XMLDocumentFragment and stylesheet.

Parameters

`xsl` - `XSLStylesheet` to be used for transformation

`xml` - XML input to be transformed (as a DOM Tree)

`pw` - PrintWriter to which the result is printed

Throws

`XSLException,` - IOException on error.

processXSL(XSLStylesheet, XMLDocument, OutputStream)

```
public void processXSL(XSLStylesheet xsl, XMLDocument xml, java.io.OutputStream
out)
```

Transform input XML document using given XMLDocument and stylesheet.

Parameters

`xsl` - `XSLStylesheet` to be used for transformation

`xml` - XML input to be transformed (as a DOM Tree)

`out` - Outputstream to which the result is printed

Throws

`XSLException,` - IOException on error.

processXSL(XSLStylesheet, XMLDocument, PrintWriter)

```
public void processXSL(XSLStylesheet xsl, XMLDocument xml, java.io.PrintWriter
pw)
```

Transform input XML document using given XMLDocument and stylesheet.

Parameters

`xsl` - `XSLStylesheet` to be used for transformation

`xml` - XML input to be transformed (as a DOM Tree)

`pw` - PrintWriter to which the result is printed

Throws

`XSLException,` - IOException on error.

setErrorStream(OutputStream)

`public final void setErrorStream(java.io.OutputStream out)`
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

Parameters

`out` - The output stream to use for errors and warnings

setLocale(Locale)

`public void setLocale(java.util.Locale locale)`
Applications can use this to set the locale for error reporting.

Parameters

`locale` - `Locale` to set

showWarnings(boolean)

`public final void showWarnings(boolean yes)`
Switch to determine whether to output warnings.

Parameters

`yes` - determines whether warnings should be shown By default, warnings are not output

XSLStylesheet

Syntax

```
public class XSLStylesheet extends java.lang.Object implements
oracle.xml.parser.v2.XSLConstants

java.lang.Object
  |
  +--oracle.xml.parser.v2.XSLStylesheet
```

All Implemented Interfaces

oracle.xml.parser.v2.XSLConstants

Description

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.

Member Summary

Constructors	
XSLStylesheet(InputStream, URL)	Constructs an XSLStylesheet using the given Inputstream
XSLStylesheet(Reader, URL)	Constructs an XSLStylesheet using the given Reader
XSLStylesheet(URL, URL)	Constructs an XSLStylesheet using the given URL
XSLStylesheet(XMLDocument, URL)	Constructs an XSLStylesheet using the given XMLDocument
Methods	
setParam(String, String)	Sets the value of a top-level stylesheet parameter.

Inherited Member Summary

Fields inherited from interface oracle.xml.parser.v2.XSLConstants

APPLY_IMPORTS, APPLY_TEMPLATES, ATTRIBUTE, ATTRIBUTE_SET, CALL_TEMPLATE, CHOOSE, COMMENT, COPY, COPY_OF, DISABLEOUTESC, ELEMENT, FOR_EACH, HREF, IF, IMPORT, INCLUDE, KEY, LOCALE, MATCH, MESSAGE, NAME, NEGINFPRIORITY, NUMBER, ORACLE_NAME, ORACLE_URL, OTHERWISE, OUTPUT, PARAM, PARAM_VARIABLE, PI, PRESERVE_SPACE, RESULT_ROOT, SORT, STRIP_SPACE, TEMPLATE, TEXT, USE, USE_ATTRIBUTE_SETS, VALUE_OF, VARIABLE, WHEN, XSL_ROOT, XSLEXTFUNCNS, XSLNAMESPACE, XSLT_SPEC_VERSION

Inherited Member Summary

Methods inherited from class java.lang.Object

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

Constructors

XSLStylesheet(InputStream, URL)

```
public XSLStylesheet(java.io.InputStream xsl, java.net.URL ref)
```
Constructs an XSLStylesheet using the given Inputstream

Parameters

`xsl` - XSL input as an Inputstream

`ref` - Reference URL for include, import and external entities

Throws

XSLException - on error.

XSLStylesheet(Reader, URL)

```
public XSLStylesheet(java.io.Reader xsl, java.net.URL ref)
```
Constructs an XSLStylesheet using the given Reader

Parameters

`xsl` - XSL input as a Reader

`ref` - Reference URL for include, import and external entities

Throws

XSLException - on error.

XSLStylesheet(URL, URL)

```
public XSLStylesheet(java.net.URL xsl, java.net.URL ref)
```
Constructs an XSLStylesheet using the given URL

Parameters

`xsl` - XSL input as a URL

`ref` - Reference URL for include, import and external entities

Throws

XSLException - on error.

XSLStylesheet(XMLDocument, URL)

```
public  XSLStylesheet(XMLDocument xsl, java.net.URL ref)
```

Constructs an XSLStylesheet using the given XMLDocument

Parameters

`xsl` - XSL input as a DOM Tree

`ref` - Reference URL for include, import

Throws

XSLException - on error.

Methods

setParam(String, String)

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

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).

Parameters

`name` - parameter name

`value` - parameter value as an XPath expression

Throws

XSLException - on error

5

Package oracle.AQ.xml

This package contains classes required by the Oracle9*i* Advanced Queuing (AQ) XML Servlet. This servlet is used to access Oracle9*i* AQ via open protocols like HTTP and SMTP using the Internet Data Access Presentation(iDAP).

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 inturn extend the javax.servlet.http.HttpServlet class.

The servlet can be deployed in any Webserver or ServletRunner that implements Javasoft's Servlet 2.0 or Servlet 2.2 interfaces

1. To deploy the AQ Servlet with a webserver that implements Javasoft's Servlet2.0 interfaces, users must define a class that extends the `oracle.AQ.xml.AQxmlServle20` 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.

The servlet can be compiled using JDK 1.1.x or JDK 1.2.x libraries.

3. 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/xschema.jar
$ORACLE HOME/rdbms/jlib/aqapi11.jar
$ORACLE_HOME/rdbms/jlib/jmscommon.jar
$ORACLE_HOME/rdbms/jlib/aqxml.jar
```

```
$ORACLE_HOME/rdbms/jlib/xsu111.jar
$ORACLE_HOME/jis/lib/servlet.jar
```

4. 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/xschema.jar
$ORACLE_HOME/rdbms/jlib/aqapi.jar
$ORACLE_HOME/rdbms/jlib/jmscommon.jar
$ORACLE_HOME/rdbms/jlib/aqxml.jar
$ORACLE_HOME/rdbms/jlib/xsu12.jar
$ORACLE_HOME/jis/lib/servlet.jar
```

Since the servlet uses JDBC OCI drivers to connect to the Oracle9*i* server, it is required that 9*i* Oracle 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 *Oracle Application Developer's guide - Advanced Queuing*

Package Oracle.AQ.xml Description

Class Summary	
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 the 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

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, AQxmlCallback-Context)

```
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 the 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

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

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

Methods

getDBConnection()	Get the JDBC connection that is used to perform this request Users can perform SQL operations using this database connection.
getOverrideAQResponseFlag()	
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

getDBConnection()

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

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()
```

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

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.
setLdapContext(DirContext)	Set the LDAP context for the servlet.
setSessionMaxInactiveTime(int)	Set the maximum time a session can remain inactive.

Member Summary	
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 he 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 IDAP 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

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.

Member Summary	
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(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 he 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 IDAP 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	
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	
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

6

Oracle XML SQL Utility (XSU) Java API

OracleXMLQuery

Syntax

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

java.lang.Object

oracle.xml.sql.query.OracleXMLQuery

Description

The `OracleXMLQuery` class does the generation of XML given a SQL query. Following is a very simple example:

```
import java.sql.*;
import oracle.xml.sql.query.*;
import oracle.jdbc.driver.*;

public class sample
{
  public static void main(String args[]) throws Exception
  {
    DriverManager.registerDriver(new oracle.jdbc.driver.OracleDriver());
    Connection conn =
        DriverManager.getConnection("jdbc:oracle:oci8:scott/tiger@");
    OracleXMLQuery qry =  new OracleXMLQuery(conn, "select * from emp");
    System.out.println(qry.getXMLString());
    conn.close();
  }
}
```

Member Summary

Fields

DTD	The DTD is used to specified that the DTD is to be generated
ERROR_TAG	The ERROR_TAG specifies the default tag name for the ERROR document
MAXROWS_ALL	The MAXROWS_ALL specifies that all rows be included in the result

Member Summary	
NONE	The NONE is used to specified that no DTD is to be generated
ROW_TAG	The ROW_TAG specifies the default tag name for the ROW elements
ROWIDATTR_TAG	The ROWIDATTR_TAG specifies the default tag name for the ROW elements
ROWSET_TAG	The ROWSET_TAG specifies the default tag name for the document
SCHEMA	The SCHEMA is used to specified that no XML schema is to be generated
SKIPROWS_ALL	The SKIPROWS_ALL specifies that all rows be skipped in the result.
Constructors	
OracleXMLQuery(Connection, ResultSet)	Constructor for the OracleXMLQueryObject.
OracleXMLQuery(Connection, String)	Constructor for the OracleXMLQueryObject.
OracleXMLQuery(OracleXMLDataSet)	Constructor for the OracleXMLQueryObject.
Methods	
close()	Close any open resource, created by the OracleXML engine.
getXMLDOM(int)	Transforms the object-relational data, specified in the constructor, into a XML document.
getXMLDOM(Node)	Transforms the object-relational data, specified in the constructor, into XML.
getXMLDOM(Node, int)	Transforms the object-relational data, specified in the constructor, into XML.
getXMLMetaData(int, boolean)	This functions returns the DTD or the XMLSchema for the XML document which would have been generated by a getXML call.
getXMLSAX(ContentHandler)	Transforms the object-relational data, specified in the constructor, into a XML document.
getXMLSchema()	This methods generated the XML Schema(s) corresponding to the specified query.

Member Summary	
getXMLString()	Transforms the object-relational data, specified in the constructor, into a XML document.
getXMLString(int)	Transforms the object-relational data, specified in the constructor, into a XML document.
getXMLString(Node)	Transforms the object-relational data, specified in the constructor, into XML.
getXMLString(Node, int)	Transforms the object-relational data, specified in the constructor, into XML.
keepObjectOpen(boolean)	The default behavior for all the getXML functions which DO NOT TAKE in a ResultSet object is to close the ResultSet object and Statement objects at the end of the call.
removeXSLTParam(String)	Removes the value of a top-level stylesheet parameter.
setCollIdAttrName(String)	Sets the name of the id attribute of the collection element's separator tag.
setDataHeader(Reader, String)	Sets the xml data header.
setDateFormat(String)	Sets the format of the generated dates in the XML doc.
setEncoding(String)	Sets the encoding PI (processing instruction) in the XML doc.
setErrorTag(String)	Sets the tag to be used to enclose the xml error docs.
setException(Exception)	Allows the user to pass in an exception, and have the XSU handle it.
setMaxRows(int)	Sets the max number of rows to be converted to XML.
setMetaHeader(Reader)	Sets the XML meta header.
setRaiseException(boolean)	Tells the XSU to throw the raised exceptions.
setRaiseNoRowsException(boolean)	Tells the XSU to throw or not to throw an OracleXMLNoRowsException in the case when for one reason or another, the XML doc generated is empty.
setRowIdAttrName(String)	Sets the name of the id attribute of the row enclosing tag.
setRowIdAttrValue(String)	Specifies the scalar column whose value is to be assigned to the id attribute of the row enclosing tag.
setRowsetTag(String)	Sets the tag to be used to enclose the xml dataset.

Member Summary	
setRowTag(String)	Sets the tag to be used to enclose the xml element corresponding to a db.
setSkipRows(int)	Sets the number of rows to skip.
setStylesheetHeader(String)	Sets the stylesheet header (i.e.
setStylesheetHeader(String, String)	Sets the stylesheet header (i.e.
setXSLT(Reader, String)	Registers a XSL transform to be applied to generated XML.
setXSLT(String, String)	Registers a XSL transform to be applied to generated XML.
setXSLTParam(String, String)	Sets the value of a top-level stylesheet parameter.
useLowerCaseTagNames()	This will set the case to be lower for all tag names.
useNullAttributeIndicator(boolean)	Specified weather to use an XML attribute to indicate NULLness, or to do it by omitting the inclusion of the particular entity in the XML document.
useTypeForCollElemTag(boolean)	By default the tag name for elements of a collection is the collection's tag name followed by "_item".
useUpperCaseTagNames()	This will set the case to be upper for all tag names.

Inherited Member Summary
Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Fields

DTD

```
public static final int DTD
```

The DTD is used to specified that the DTD is to be generated

ERROR_TAG

```
public static final java.lang.String ERROR_TAG
```

The ERROR_TAG specifies the default tag name for the ERROR document

MAXROWS_ALL

```
public static final int MAXROWS_ALL
```
The MAXROWS_ALL specifies that all rows be included in the result

NONE

```
public static final int NONE
```
The NONE is used to specified that no DTD is to be generated

ROW_TAG

```
public static final java.lang.String ROW_TAG
```
The ROW_TAG specifies the default tag name for the ROW elements

ROWIDATTR_TAG

```
public static final java.lang.String ROWIDATTR_TAG
```
The ROWIDATTR_TAG specifies the default tag name for the ROW elements

ROWSET_TAG

```
public static final java.lang.String ROWSET_TAG
```
The ROWSET_TAG specifies the default tag name for the document

SCHEMA

```
public static final int SCHEMA
```
The SCHEMA is used to specified that no XML schema is to be generated

SKIPROWS_ALL

```
public static final int SKIPROWS_ALL
```
The SKIPROWS_ALL specifies that all rows be skipped in the result.

Constructors

OracleXMLQuery(Connection, ResultSet)

```
public  OracleXMLQuery(java.sql.Connection conn, java.sql.ResultSet rset)
```
Constructor for the OracleXMLQueryObject.

Parameters:

conn - database connection

rset - jdbc result set object

OracleXMLQuery(Connection, String)

`public OracleXMLQuery(java.sql.Connection conn, java.lang.String query)`
Constructor for the OracleXMLQueryObject.

Parameters:

conn - database connection

query - the SQL query string

OracleXMLQuery(OracleXMLDataSet)

`public OracleXMLQuery(oracle.xml.sql.dataset.OracleXMLDataSet dset)`
Constructor for the OracleXMLQueryObject.

Parameters:

conn - database connection

dset - dataset

Methods

close()

`public void close()`
Close any open resource, created by the OracleXML engine. This will not close for instance resultset supplied by the user

getNumRowsProcessed()

`public long getNumRowsProcessed()`
Returns the number of rows processed.

Returns:

number of rows processed.

getXMLDOM()

`public org.w3c.dom.Document getXMLDOM()`
Transforms the object-relational data, specified in the constructor, into a XML document.

Returns:

the DOM representation of the XML document

getXMLDOM(int)

`public org.w3c.dom.Document getXMLDOM(int metaType)`
Transforms the object-relational data, specified in the constructor, into a XML document. The `metaType` 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.

Parameters:

`metaType` - the type of XML metadata (NONE, SCHEMA)

Returns:

the string representation of the XML document

getXMLDOM(Node)

`public org.w3c.dom.Document getXMLDOM(org.w3c.dom.Node root)`
Transforms the object-relational data, specified in the constructor, into XML. If not NULL, the `root` argument, is considered the "root" element of the XML doc.

Parameters:

`root` - root node to which to append the new XML

Returns:

the string representation of the XML document

getXMLDOM(Node, int)

`public org.w3c.dom.Document getXMLDOM(org.w3c.dom.Node root, int metaType)`
Transforms the object-relational data, specified in the constructor, into XML. If not NULL, the `root` argument, is considered the "root" element of the XML doc. The `metaType` 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.

Parameters:

`root` - root node to which to append the new XML

`metaType` - the type of XML metadata (NONE, SCHEMA)

Returns:

the string representation of the XML document

getXMLMetaData(int, boolean)

`public java.lang.String getXMLMetaData(int metaType, boolean withVer)`
This functions returns the DTD or the XMLSchema for the XML document which would have been generated by a getXML call. The "metaType" parameter specifies the type of XML metadata to be generated. The `withVer` parameter specifies if version header is to be generated or not.

Parameters:

`metaType` - XML meta data type to generate (NONE or DTD)

`withVer` - generate the version PI ?

getXMLSAX(ContentHandler)

`public void getXMLSAX(org.xml.sax.ContentHandler sax)`
Transforms the object-relational data, specified in the constructor, into a XML document.

Parameters:

`sax` - ContentHandler object to be registered

getXMLSchema()

`public org.w3c.dom.Document[] getXMLSchema()`
This methods generated the XML Schema(s) corresponding to the specified query.

Returns:

the XML Schema(s)

getXMLString()

`public java.lang.String getXMLString()`
Transforms the object-relational data, specified in the constructor, into a XML document.

Returns:

the string representation of the XML document

getXMLString(int)

`public java.lang.String getXMLString(int metaType)`
Transforms the object-relational data, specified in the constructor, into a XML document. The `metaType` argument is used to specify the type of XML metadata the XSU is to generate along with the XML. Valid values for the `metaType` argument are `NONE` and `DTD` (static fields of this class).

Parameters:

`metaType` - the type of XML metadata (NONE, DTD, or SCHEMA)

Returns:

the string representation of the XML document

getXMLString(Node)

`public java.lang.String getXMLString(org.w3c.dom.Node root)`
Transforms the object-relational data, specified in the constructor, into XML. If not NULL, the `root` argument, is considered the "root" element of the XML doc.

Parameters:

`root` - root node to which to append the new XML

Returns:

the string representation of the XML document

getXMLString(Node, int)

`public java.lang.String getXMLString(org.w3c.dom.Node root, int metaType)`
Transforms the object-relational data, specified in the constructor, into XML. If not NULL, the `root` argument, is considered the "root" element of the XML doc. The `metaType` argument is used to specify the type of XML metadata the XSU is to generate along with the XML. Valid values for the `metaType` argument are `NONE` and `DTD` (static fields of this class). Note that if the `root` argument is non-null, no DTD is generated even if requested.

Parameters:

`root` - root node to which to append the new XML

`metaType` - the type of XML metadata (NONE, DTD, or SCHEMA)

Returns:

the string representation of the XML document

keepObjectOpen(boolean)

`public void keepObjectOpen(boolean alive)`
The default behavior for all the getXML functions which DO NOT TAKE in a ResultSet object is to close the ResultSet object and Statement objects at the end of the call. If you need to use the persistant feature, where by calling getXML repeatedly you get the next set of rows, you need to turn off this behavior by calling this function with value true. i.e. OracleXMLQuery would not close the ResultSet and Statement objects after the getXML calls. You can call the close() function to explicitly close the cursor state.

Parameters:

`alive` - keep object open ?

removeXSLTParam(String)

`public void removeXSLTParam(java.lang.String name)`
Removes the value of a top-level stylesheet parameter. NOTE: if no stylesheet is registered, this method is a no op.

Parameters:

`name` - parameter name

setCollIdAttrName(String)

`public void setCollIdAttrName(java.lang.String attrName)`
Sets the name of the id attribute of the collection element's separator tag. Passing `null` or an empty string for the `tag` results the row id attribute to be omitted.

Parameters:

`attrName` - attribute name

setDataHeader(Reader, String)

`public void setDataHeader(java.io.Reader header, java.lang.String docTag)`
Sets the xml data header. The data header is an XML entity which is appended at the begining of the query-generated xml entity (ie. rowset). The two entities are enclosed by the tag specified via the `docTag` argument. Note that the last data header specified is the one that is used; furthermore, passing in `null` for the `header`, parameter unsets the data header.

Parameters:

`header` - header

`tag` - tag used to enclose the data header and the rowset

setDateFormat(String)

```
public void setDateFormat(java.lang.String mask)
```

Sets the format of the generated dates in the XML doc. The syntax of the date format patern (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.

Parameters:

`mask` - the date mask

setEncoding(String)

```
public void setEncoding(java.lang.String enc)
```

Sets the encoding PI (processing instruction) in the XML doc. If `null` or an empty string are specified as the encoding, then the default characterset is specified in the encoding PI.

Parameters:

`enc` - characterset encoding of the XML doc

setErrorTag(String)

```
public void setErrorTag(java.lang.String tag)
```

Sets the tag to be used to enclose the xml error docs.

Parameters:

`tag` - tag name

setException(Exception)

```
public void setException(java.lang.Exception e)
```

Allows the user to pass in an exception, and have the XSU handle it.

Parameters:

`e` - the exception to be processed by the XSU.

setMaxRows(int)

`public void setMaxRows(int rows)`
Sets the max number of rows to be converted to XML. By default there is no max set. To explicitly specify no max see MAXROWS_ALL.

Parameters:

`rows` - max number of rows to generate

setMetaHeader(Reader)

`public void setMetaHeader(java.io.Reader header)`
Sets the XML meta header. When set, the header is inserted at the begining of the metadata part (DTD or XMLSchema) of each XML document generated by this object. Note that the last meta header specified is the one that is used; furthermore, passing in `null` for the `header`, parameter unsets the meta header.

Parameters:

`header` - header

setRaiseException(boolean)

`public void setRaiseException(boolean flag)`
Tells the XSU 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 out of the exception's message.

Parameters:

`flag` - throw raised exceptions?

setRaiseNoRowsException(boolean)

`public void setRaiseNoRowsException(boolean flag)`
Tells the XSU to throw or not to throw an OracleXMLNoRowsException in the case when for one reason or another, the XML doc generated is empty. By default, the exception is not thrown.

Parameters:

`flag` - throw OracleXMLNoRowsException if no data found?

setRowIdAttrName(String)

```
public void setRowIdAttrName(java.lang.String attrName)
```

Sets the name of the id attribute of the row enclosing tag. Passing `null` or an empty string for the `tag` results the row id attribute to be omitted.

Parameters:

`attrName` - attribute name

setRowIdAttrValue(String)

```
public void setRowIdAttrValue(java.lang.String colName)
```

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 for the `colName` results the row id attribute being assigned the row count value (i.e. 0, 1, 2 and so on).

Parameters:

`colName` - column whose value is to be assigned to the row id attr

setRowsetTag(String)

```
public void setRowsetTag(java.lang.String tag)
```

Sets the tag to be used to enclose the xml dataset.

Parameters:

`tag` - tag name

setRowTag(String)

```
public void setRowTag(java.lang.String tag)
```

Sets the tag to be used to enclose the xml element corresponding to a db. record.

Parameters:

`tag` - tag name

setSkipRows(int)

```
public void setSkipRows(int rows)
```

Sets the number of rows to skip. By default 0 rows are skipped. To skip all the rows use SKIPROWS_ALL.

Parameters:

rows - number of rows to skip

setStylesheetHeader(String)

`public void setStylesheetHeader(java.lang.String uri)`
Sets the stylesheet header (i.e. stylesheet processing instructions) in the generated XML doc. Note: Passing `null` for the `uri` argument will unset the stylesheet header and the stylesheet type.

Parameters:

`uri` - stylesheet URI

setStylesheetHeader(String, String)

`public void setStylesheetHeader(java.lang.String uri, java.lang.String type)`
Sets the stylesheet header (i.e. stylesheet processing instructions) in the generated XML doc. Note: Passing `null` for the `uri` argument will unset the stylesheet header and the stylesheet type.

Parameters:

`uri` - stylesheet URI

`type` - stylesheet type; defaults to 'text/xsl'

setXSLT(Reader, String)

`public void setXSLT(java.io.Reader stylesheet, java.lang.String ref)`
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.

Parameters:

`stylesheet` - the stylesheet

`ref` - URL for include, import and external entities

setXSLT(String, String)

`public void setXSLT(java.lang.String stylesheet, java.lang.String ref)`
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.

Parameters:

`stylesheet` - the stylesheet URI

`ref` - URL for include, import and external entities

setXSLTParam(String, String)

```
public void setXSLTParam(java.lang.String name, java.lang.String value)
```
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). NOTE: if no stylesheet is registered, this method is a no op.

Parameters:

`name` - parameter name

`value` - parameter value as an XPATH expression

useLowerCaseTagNames()

```
public void useLowerCaseTagNames()
```
This will set the case to be lower for all tag names. Note, make this call after all the desired tags have been set.

useNullAttributeIndicator(boolean)

```
public void useNullAttributeIndicator(boolean flag)
```
Specified weather to use an XML attribute to indicate NULLness, or to do it by omitting the inclusion of the particular entity in the XML document.

Parameters:

`flag` - use attribute to indicate null?

useTypeForCollElemTag(boolean)

```
public void useTypeForCollElemTag(boolean flag)
```
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 of `true`, tells the XSU to use the collection element's type name as the collection element tag name.

Parameters:

`flag` - use the coll. elem. type as its tag name?

useUpperCaseTagNames()

```
public void useUpperCaseTagNames()
```

This will set the case to be upper for all tag names. Note, make this call after all the desired tags have been set.

OracleXMLSave

Syntax

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

java.lang.Object
  |
  +--oracle.xml.sql.dml.OracleXMLSave
```

Description

OracleXMLSave - this 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. A typical sequence might look like

```
OracleXMLSave sav = new OracleXMLSave(conn, "emp");
// insert processing
sav.insertXML(xmlDoc);
-or-
// Update processing
 String[]tempArr = new String[2];
 tempArr[0] = "EMPNO";   // set the empno as the key for updates..
 sav.setKeyColumnList(tempArray);
sav.updateXML(xmlDoc);
-or-
sav.deleteXML(xmlDoc);
sav.close();
```

A lot of 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.

Member Summary

Fields	
DATE_FORMAT	The date format for use in setDateFormat
DEFAULT_BATCH_SIZE	default insert batch size is 17
Constructors	
OracleXMLSave(Connection, String)	The public constructor for the Save class.
Methods	
cleanLobList()	
close()	It closes/deallocates all the context associated with this object.
deleteXML(Document)	Deletes the rows in the table based on the XML document
deleteXML(InputStream)	Deletes the rows in the table based on the XML document
deleteXML(Reader)	Deletes the rows in the table based on the XML document
deleteXML(String)	Deletes the rows in the table based on the XML document
deleteXML(URL)	Deletes rows from a specified table based on the element values in the supplied XML document.
finalize()	
getURL(String)	Given a file name or a URL it return a URL object.
insertXML(Document)	
insertXML(InputStream)	
insertXML(Reader)	
insertXML(String)	
insertXML(URL)	Inserts an XML document from a specified URL into the specified table, By default, the insert routine 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.
removeXSLTParam(String)	Removes the value of a top-level stylesheet parameter.
setBatchSize(int)	This call changes the batch size used during DML operations.

Member Summary	
setCommitBatch(int)	Sets the commit batch size.
setDateFormat(String)	Describes to the XSU the format of the dates in the XML document.
setIgnoreCase(boolean)	The XSU does mapping of XML elements to database columns/attrs.
setKeyColumnList(String[])	Sets the list of columns to be used for identifying a particular row in the database table during update or delete.
setRowTag(String)	Names the tag used in the XML doc., to enclose the XML elements corresponding to each row value.
setUpdateColumnList(String[])	Set the column values to be updated.
setXSLT(Reader, String)	Registers a XSL transform to be applied to generated XML.
setXSLT(String, String)	Registers a XSL transform to be applied to generated XML.
setXSLTParam(String, String)	Sets the value of a top-level stylesheet parameter.
updateXML(Document)	Updates the table given the XML document in a DOM tree form
updateXML(InputStream)	Updates the table given the XML document in a stream form
updateXML(Reader)	Updates the table given the XML document in a stream form
updateXML(String)	Updates the table given the XML document in a string form
updateXML(URL)	Updates the columns in a database table, based on the element values in the supplied XML document.

Inherited Member Summary

Methods inherited from class java.lang.Object

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

Fields

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

Constructors

OracleXMLSave(Connection, String)

```
public  OracleXMLSave(java.sql.Connection oconn, java.lang.String tabName)
```
The public constructor for the Save class.

Parameters:

`oconn` - Connection object (connection to the database)

`tableName` - The name of the table that should be updated

Methods

cleanLobList()

```
public void cleanLobList()
```

close()

```
public void close()
```
It closes/deallocates all the context associated with this object.

deleteXML(Document)

```
public int deleteXML(org.w3c.dom.Document doc)
```
Deletes the rows in the table based on the XML document

Parameters:

`xmlDoc` - The XML document in DOM form

Returns:

The number of XML ROW elements processed.

See Also:

deleteXML(URL)

deleteXML(InputStream)

```
public int deleteXML(java.io.InputStream xmlStream)
```

Deletes the rows in the table based on the XML document

Parameters:

`xmlDoc` - The XML document in Stream form

Returns:

The number of XML ROW elements processed.

See Also:

deleteXML(URL)

deleteXML(Reader)

```
public int deleteXML(java.io.Reader xmlStream)
```

Deletes the rows in the table based on the XML document

Parameters:

`xmlDoc` - The XML document in Stream form

Returns:

The number of XML ROW elements processed.

See Also:

deleteXML(URL)

deleteXML(String)

```
public int deleteXML(java.lang.String xmlDoc)
```

Deletes the rows in the table based on the XML document

Parameters:

`xmlDoc` - The XML document in String form

Returns:

The number of XML ROW elements processed.

See Also:

deleteXML(URL)

deleteXML(URL)

```
public int deleteXML(java.net.URL url)
```

Deletes rows from a specified table based on the element values in the supplied XML document. 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() you can set the list of columns that must be matched to identify the row to be deleted and ignore the other elements This is an effficient method for deleting more than one row in the table, since the delete statement is cached and batching can be employed. If not, a new delete statement has to be created for each ROW element in the input document.

For example, consider the employee table emp,

TABLE emp(empno NUMBER, ename VARCHAR2(20), hiredate DATE);

Now, you want to delete the rows in the table using an XML document and you want to identify the row based on the EMPNO value. You can send in an XML document containing the EMPNO value and call the deleteXML routine with the setKeyColumnList() containing the EMPNO string.

OracleXMLSave save = new OracleXMLSave(conn,"emp");

save.deleteXML(xmlDoc); // Deletes rows by matching all columns with the // element values in the input document.

String[] deleteArray = new String[1];

deleteArray[0] = "EMPNO";

save.setKeyColumnList(deleteArray); // set the key columns

save.deleteXML(xmlDoc); // only deletes rows by matching the EMPNO values

Parameters:

`url` - The URL to the document to use to delete the rows in the table

Returns:

The number of XML row elements processed. Note that 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.

finalize()

```
protected void finalize()
```

Overrides:

java.lang.Object.finalize() in class java.lang.Object

getURL(String)

```
public static java.net.URL getURL(java.lang.String target)
```

Given a file name or a URL it return a URL object. If the argument passed is not in the valid URL format (e.g. http://.. or file://) then this method tried to fix the argument by pre-pending "file://" to the argument. If a `null` or an empty string are passed to it, null is returned.

Parameters:

`target` - file name or URL string

Returns:

the URL object identifiying the `target entity`

insertXML(Document)

```
public int insertXML(org.w3c.dom.Document doc)
```

insertXML(InputStream)

```
public int insertXML(java.io.InputStream xmlStream)
```

insertXML(Reader)

```
public int insertXML(java.io.Reader xmlStream)
```

insertXML(String)

```
public int insertXML(java.lang.String xmlDoc)
```

insertXML(URL)

```
public int insertXML(java.net.URL url)
```

Inserts an XML document from a specified URL into the specified table, By default, the insert routine 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 setting the list of columns to insert using the setUpdateColumnList() you can restrict the insert to only insert values into those columns and let the default values for other columns to be inserted. That is no null value would be inserted for the rest of the columns Use setKeyColumnList() to set the list of all key column. Use setUpdateColumnList() to set the list of columns to update.

For example, consider the employee table emp,

TABLE emp(empno NUMBER, ename VARCHAR2(20), hiredate DATE);

Now, assume that you want to insert the table using an XML document OracleXMLSave

save = new OracleXMLSave(conn,"emp");

save.insertXML(xmlDoc); // xmlDoc supplied..

save.close();

If you want to insert values only in to EMPNO, HIREDATE and SALARY and let the default values handle the rest of the columns,

String insArray = new String[3];

insArrary[0] = "EMPNO";

insArray[1] = "HIREDATE";

insArray[2] = "SALARY";

save.setUpdateColumnList(insArray);

save.insertXML(xmlDoc); // will only insert values into EMPNO, HIREDATE // and SALARY columns

Parameters:

`url` - The URL to the document to use to insert rows into the table

Returns:

The number of rows inserted.

removeXSLTParam(String)

```
public void removeXSLTParam(java.lang.String name)
```
Removes the value of a top-level stylesheet parameter. NOTE: if no stylesheet is registered, this method is a no op.

Parameters:

`name` - parameter name

setBatchSize(int)

```
public void setBatchSize(int size)
```
This call changes the batch size used during DML operations. When performing inserts, updates or deletes, it is better to batch the operations so that the database can execute it in one shot rather than as separate statements. The flip side is that more memory is needed to hold all the bind values before the operation is done. Note that when batching is used, the commits occur only in terms of batches. So if one of the statement inside a batch fails, the whole batch is rolled back. If this behaviour is unaccepatable, then set the batch size to 1. The default batch size is DEFAULT_BATCH_SIZE;

Parameters:

`size` - The batch size to use for all DML

setCommitBatch(int)

```
public void setCommitBatch(int size)
```
Sets the commit batch size. The commit batch size refers to the number or records inserted after which a commit should follow. Note that if commitBatch is < 1 or the session is in "auto-commit" mode then the XSU does not make any explicit commit's. By default the commit-batch size is 0.

Parameters:

`size` - commit batch size

setDateFormat(String)

```
public void setDateFormat(java.lang.String mask)
```
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 patern (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, results the use of the default mask -- OracleXMLSave.DATE_FORMAT.

Parameters:

`mask` - the date mask

setIgnoreCase(boolean)

```
public void setIgnoreCase(boolean ignore)
```

The XSU does mapping of XML elements to database columns/attrs. based on the element names (xml tags). This function tells the XSU to do this match case insensitive. This resetting of case may affect the metadata caching that is done when creating the Save object.

Parameters:

`flag` - ignore tag case in the XML doc? 0-false 1-true

setKeyColumnList(String[])

```
public void setKeyColumnList(java.lang.String[] keyColNames)
```

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

Parameters:

`keyColNames` - The names of the list of columns that are used as keys

setRowTag(String)

```
public void setRowTag(java.lang.String rowTag)
```

Names the tag used in the XML doc., 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.

Parameters:

`tag` - tag name

setUpdateColumnList(String[])

```
public void setUpdateColumnList(java.lang.String[] updColNames)
```
Set the column values to be updated. This is only valid for inserts and updates, and is ignored for 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.

Parameters:

`updColNames` - The string list of columns to be updated

setXSLT(Reader, String)

```
public void setXSLT(java.io.Reader stylesheet, java.lang.String ref)
```
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.

Parameters:

`stylesheet` - the stylesheet

`ref` - URL for include, import and external entities

setXSLT(String, String)

```
public void setXSLT(java.lang.String stylesheet, java.lang.String ref)
```
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.

Parameters:

`stylesheet` - the stylesheet URI

`ref` - URL for include, import and external entities

setXSLTParam(String, String)

```
public void setXSLTParam(java.lang.String name, java.lang.String value)
```
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). NOTE: if no stylesheet is registered, this method is a no op.

Parameters:

name - parameter name

value - parameter value as an XPATH expression

updateXML(Document)

```
public int updateXML(org.w3c.dom.Document doc)
```
Updates the table given the XML document in a DOM tree form

Parameters:

xmlDoc - The DOM tree form of the XML document

Returns:

The number of XML elements processed

See Also:

updateXML(URL)

updateXML(InputStream)

```
public int updateXML(java.io.InputStream xmlStream)
```
Updates the table given the XML document in a stream form

Parameters:

xmlDoc - The stream form of the XML document

Returns:

The number of XML elements processed

See Also:

updateXML(URL)

updateXML(Reader)

```
public int updateXML(java.io.Reader xmlStream)
```
Updates the table given the XML document in a stream form

Parameters:

xmlDoc - The stream form of the XML document

Returns:

The number of XML elements processed

See Also:

updateXML(URL)

updateXML(String)

```
public int updateXML(java.lang.String xmlDoc)
```

Updates the table given the XML document in a string form

Parameters:

`xmlDoc` - The string form of the XML document

Returns:

The number of XML elements processed

See Also:

updateXML(URL)

updateXML(URL)

```
public int updateXML(java.net.URL url)
```

Updates the columns in a database table, based on the element values in the supplied XML document. 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 and then updates 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.

You can also supply a list of columns to update - this will make the utility update only those columns and ignore any other element present in the XML document. This is a very efficient method, since if there are more than one row present in the input XML document, the update statement itself is cached and batching is done. If not, then we need to create a new update statement for each row of the input document.

Use setKeyColumnList() to set the list of all key column.

Use setUpdateColumnList() to set the list of columns to update.

For example, consider the employee table emp,

TABLE emp(empno NUMBER, ename VARCHAR2(20), hiredate DATE);

Now, assume that you want to update the table using an XML document and you want to use the values of the element EMPNO to match the right row and then only update the HIREDATE column.

You can send in an XML document containing the HIREDATE value and the EMPNO value and call the updateXML routine with the setKeyColumnList() containing the EMPNO string.

```
OracleXMLSave save = new OracleXMLSave(conn,"emp");

String[] keyArray = new String[1];

keyArray[0] = "EMPNO"; // Set EMPNO as key column

save.setKeyColumnList(keyArray);  // Set the key column names

String[] updArray = new String[1];

updArray[0] = "HIREDATE"; // SEt hiredate as column to update

save.setUpdateColumnList(updArray);

save.updateXML(xmlDoc); // xmlDoc supplied..

save.close();
```

Parameters:

`url` - The URL to the document to use to update the table

Returns:

The number of XML row elements processed. Note that this may or may not be equal to the number of database rows modified based on whether the rows selected through the XML document uniquely identified the rows in the table.

OracleXMLSQLException

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 Known Subclasses:

OracleXMLSQLNoRowsException

All Implemented Interfaces:

java.io.Serializable

Description

Member Summary

Constructors

OracleXMLSQLException(Exception)

OracleXMLSQLException(Exception, String)

OracleXMLSQLException(String)

OracleXMLSQLException(String, Exception)

OracleXMLSQLException(String, Exception, String)

OracleXMLSQLException(String, int)

OracleXMLSQLException(String, int, String)

OracleXMLSQLException(String, String)

Methods

Member Summary	
getErrorCode()	
getParentException()	returns the original exception, if there was one; otherwise, it returns null
getXMLErrorString()	prints the XML error string with the given error tag name
getXMLSQLErrorString()	prints the SQL parameters as well in the error message
setErrorTag(String)	Sets the error tag name which is then used by getXMLErrorString and getXMLSQLErrorString, to generate xml error reports

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

Constructors

OracleXMLSQLException(Exception)

```
public  OracleXMLSQLException(java.lang.Exception e)
```

OracleXMLSQLException(Exception, String)

```
public  OracleXMLSQLException(java.lang.Exception e, java.lang.String
errorTagName)
```

OracleXMLSQLException(String)

```
public  OracleXMLSQLException(java.lang.String message)
```

OracleXMLSQLException(String, Exception)

```
public  OracleXMLSQLException(java.lang.String message, java.lang.Exception e)
```

OracleXMLSQLException(String, Exception, String)

```
public  OracleXMLSQLException(java.lang.String message, java.lang.Exception e,
java.lang.String errorTagName)
```

OracleXMLSQLException(String, int)

```
public  OracleXMLSQLException(java.lang.String message, int errorCode)
```

OracleXMLSQLException(String, int, String)

```
public  OracleXMLSQLException(java.lang.String message, int errorCode,
java.lang.String errorTagName)
```

OracleXMLSQLException(String, String)

```
public  OracleXMLSQLException(java.lang.String message, java.lang.String
errorTagName)
```

Methods

getErrorCode()

```
public int getErrorCode()
```

getParentException()

```
public java.lang.Exception getParentException()
```
returns the original exception, if there was one; otherwise, it returns null

getXMLErrorString()

```
public java.lang.String getXMLErrorString()
```
prints the XML error string with the given error tag name

getXMLSQLErrorString()

```
public java.lang.String getXMLSQLErrorString()
```
prints the SQL parameters as well in the error message

setErrorTag(String)

```
public void setErrorTag(java.lang.String tagName)
```
Sets the error tag name which is then used by getXMLErrorString and getXMLSQLErrorString, to generate xml error reports

OracleXMLSQLNoRowsException

Syntax

```
public class OracleXMLSQLNoRowsException extends OracleXMLSQLException

java.lang.Object
  |
  +--java.lang.Throwable
        |
        +--java.lang.Exception
              |
              +--java.lang.RuntimeException
                    |
                    +--OracleXMLSQLException
                          |
                          +--oracle.xml.sql.OracleXMLSQLNoRowsException
```

All Implemented Interfaces:

java.io.Serializable

Description

Member Summary

Constructors

OracleXMLSQLNoRowsException()

OracleXMLSQLNoRowsException(String)

Inherited Member Summary

Methods inherited from interface OracleXMLSQLException

getErrorCode(), getParentException(), getXMLErrorString(), getXMLSQLErrorString(), setErrorTag(String)

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

Constructors

OracleXMLSQLNoRowsException()

```
public  OracleXMLSQLNoRowsException()
```

OracleXMLSQLNoRowsException(String)

```
public  OracleXMLSQLNoRowsException(java.lang.String errorTag)
```

7

Package oracle.XML.parser.schema

XMLSchema

```
java.lang.Object
  |
  +---oracle.xml.parser.schema.XSDConstants
          |
          +---oracle.xml.parser.schema.XSDNode
                  |
                  +---oracle.xml.parser.schema.XMLSchema
```

```
public class XMLSchema
extends XSDNode
```

XMLSchema class. Sets top-level XMLSchema document declarations & definitions plus schema location and schema target namespace. XMLSchema objects are created by XSDBuilder as a result of processing XMLSchema documents. They are used by XSDParser for instance XML documents validation and by XSDBuilder as imported schemas.

Constructor Index

`XMLSchema()`
XMLSchema constructor.

`XMLSchema(int)`
XMLSchema constructor.

Constructors

XMLSchema

```
public XMLSchema() throws XSDException
```

XMLSchema constructor.

XMLSchema

```
public XMLSchema(int n) throws XSDException
```

XMLSchema constructor.

Parameters

`n - Initial size of schemanode set.`

XSDBuilder

```
java.lang.Object
   |
   +---oracle.xml.parser.schema.XSDConstants
           |
           +---oracle.xml.parser.schema.XSDBuilder
```

```
public class XSDBuilder
extends XSDConstants
implements ObjectBuilder
```

Builds an XMLSchema object from XMLSchema document. XMLSchema object is a set of objects (Infoset items) corresponding to top-level schema declrations & 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 & definitions are registered as a current schema infoset items. Finaly, top-level tree elements (infoset items) are 'Schema' parsed. The result 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 containg cardinality info (min/maxOccurs).

Constructor Index

`XSDBuilder()`
XSDBuilder constructor

Method Index

`build(InputStream, URL)`
Build an XMLSchema object

`build(Reader, URL)`
Build an XMLSchema object

`build(String)`
Build an XMLSchema object

`build(String, String)`
Build an XMLSchema object

`build(String, URL)`
Build an XMLSchema object

`build(URL)`
Build an XMLSchema object

`build(XMLDocument, URL)`
Build XMLSchema from XML document

`getObject()`
Returns the schema object.

`setError(XMLError)`
Sets XMLError object.

`setLocale(Locale)`
Sets locale for error reporting.

Constructors

XSDBuilder

`public XSDBuilder() throws XSDException`
XSDBuilder constructor

Methods

setError

public void setError(XMLError er)

Sets XMLError object.

Parameters

er - XMLError object

setLocale

```
public void setLocale(Locale locale)
```

Sets locale for error reporting.

Parameters

locale - Locale object

getObject

public Object getObject()

Returns the schema object.

Returns

XMLSchema object.

build

```
public Object build(String sysId) throws Exception
```

Build an XMLSchema object

Parameters

sysId - Schema location

Returns

Object - XMLSchema

Throws

An Exception is thrown if Builder fails to build an XMLSchema object.

build

```
public Object build(InputStream in, URL baseurl) throws Exception
```

Build an XMLSchema object

Parameters

```
in - Inputstream of Schema
baseurl - URL used to resolve any relative refs.
```

Returns
Object - XMLSchema

Throws
An Exception is thrown if Builder fails to build an XMLSchema object.

build
```
public Object build(Reader r, URL baseurl) throws Exception
```
Build an XMLSchema object

Parameters
```
r - Reader of Schema
baseurl - URL used to resolve any relative refs.
```

Returns
Object - XMLSchema

Throws
An Exception is thrown if Builder fails to build an XMLSchema object.

build
```
public Object build(URL schemaurl) throws Exception
```
Build an XMLSchema object

Parameters
```
url - URL of Schema
```

Returns
Object - XMLSchema

Throws
An Exception is thrown if Builder fails to build an XMLSchema object.

build
```
public Object build(XMLDocument schemaDoc) throws Exception
```
Build XMLSchema from XML document

Parameters

schemaDoc - XMLDocument
baseurl - URL used to resolve any relative refs.

Returns

Object - XMLSchema

Throws

An Exception is thrown if Builder fails to build an XMLSchema object.

build

public Object build(String ns, String sysid) throws Exception
Build an XMLSchema object

Parameters

ns - Schema target namespace used to validate targetNamespace
sysId - Schema location

Returns

Object XMLSchema

Throws

An Exception is thrown if Builder fails to build an XMLSchema object.

build

public Object build(String ns, URL sysid) throws Exception
Build an XMLSchema object

Parameters

ns - Schema target namespace used to validate targetNamespace
sysId - URL Schema location

Returns

Object XMLSchema

Throws

An Exception is thrown if Builder fails to build an XMLSchema object.

XSDException

```
java.lang.Object
   |
   +---java.lang.Throwable
           |
           +---java.lang.Exception
                   |
                   +---oracle.xml.parser.schema.XSDException
```

```
public class XSDException
extends Exception
```
Indicates that an exception occurred during XMLSchema validation

Method Index

`getMessage()`
Overrride getMessage, in order to construct error message from error id, and error params

`getMessage(XMLError)`
Get localized message based on the XMLError sent as parameter

Methods

`getMessage`
`public String getMessage()`
Overrride getMessage, in order to construct error message from error id, and error params

Overrides

getMessage in class Throwable

getMessage

`public String getMessage(XMLError err)Get localized message based on the XMLError sent as parameter`

Parameters

`err - XMLError class used to get the error message`

8

Package oracle.xml.xsql

This package is also known as the XSQL Servlet.

Description

Class Summary	
Interfaces	
XSQLActionHandler	Interface that must be implemented by all XSQL Action Element Handlers
XSQLPageRequest	Interface representing a request for an XSQL Page
Classes	
Res	
XSQLActionHandlerImpl	Base Implementation of XSQLActionHandler that can be extended to create your own custom handlers.
XSQLCommandLine	Command-line Utility to process XSQL Pages.
XSQLDiagnostic	
XSQLHttpUtil	
XSQLPageRequestImpl	Base implementation of the XSQLPageRequest interface that case be used to derive new kinds of page request implementations.
XSQLParserHelper	Common XML Parsing Routines
XSQLRequest	Programmatically process a request for an XSQL Page.
XSQLServlet	Servlet to enable HTTP GET-ing of and POST-ing to XSQL Pages
XSQLServletPageRequest	Implementation of XSQLPageRequest for Servlet-based XSQL Page requests.
XSQLStylesheetProcessor	XSLT Stylesheet Processing Engine
XSQLUtil	

Res

Syntax

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

java.lang.Object
  |
  +--oracle.xml.xsql.Res
```

Description

Member Summary

Fields

CANTREAD_XSQL

CANTREAD_XSQL_MSG

CLASSNOTFOUND

CLASSNOTFOUND_MSG

CONN_FILE

CONN_FILE_MSG

ERR_OUTPUT

ERR_OUTPUT_MSG

ERRORINCLUDING

ERRORINCLUDING_MSG

ERRORLOADINGURL

ERRORLOADINGURL_MSG

ERRORREADINGPARAM

ERRORREADINGPARAM_MSG

FATAL_SHEETPOOL

FATAL_SHEETPOOL_MSG

ILLFORMEDXMLPARAMVAL

ILLFORMEDXMLPARAMVAL_MSG

Member Summary

ILLFORMEDXMLRESOURCE

ILLFORMEDXMLRESOURCE_MSG

INSTANTIATIONERR

INSTANTIATIONERR_MSG

INVALID_URI

INVALID_URI_MSG

INVALIDURL

INVALIDURL_MSG

MISSING_ARGS

MISSING_ARGS_MSG

MISSING_ATTR

MISSING_ATTR_MSG

NAMED_CONN

NAMED_CONN_MSG

NO_CONN

NO_CONN_DEF

NO_CONN_DEF_MSG

NO_CONN_MSG

NO_XSQL_FILE

NO_XSQL_FILE_MSG

NOFUNCTIONNAME

NOFUNCTIONNAME_MSG

NOPOSTEDXML

NOPOSTEDXML_MSG

NOQUERYSUPPLIED

NOQUERYSUPPLIED_MSG

NOTANACTIONHANDLER

NOTANACTIONHANDLER_MSG

Member Summary

NULLPARAM

NULLPARAM_MSG

OWAXMLMALFORMED

OWAXMLMALFORMED_MSG

POSTEDXML_ERR

POSTEDXML_ERR_MSG

UNHANDLED_ERR

UNHANDLED_ERR_MSG

UNHANDLED_ERR_XSL_PR

UNHANDLED_ERR_XSL_PR_MSG

UNHANDLED_ERR_XSL_RD

UNHANDLED_ERR_XSL_RD_MSG

XML_INS_ERR

XML_INS_ERR_MSG

XML_PARSE

XML_PARSE_MSG

XML_SQL_ERR

XML_SQL_ERR_MSG

XSL_ERRORS

XSL_ERRORS_MSG

XSL_NOFILE

XSL_NOFILE_MSG

XSL_PARSE

XSL_PARSE_MSG

XSLNOTFOUND

XSLNOTFOUND_MSG

Constructors

Res()

Member Summary
Methods
format(int, String)
getString(int)

Inherited Member Summary
Methods inherited from class java.lang.Object
equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Fields

CANTREAD_XSQL

```
public static final int CANTREAD_XSQL
```

CANTREAD_XSQL_MSG

```
public static final java.lang.String CANTREAD_XSQL_MSG
```

CLASSNOTFOUND

```
public static final int CLASSNOTFOUND
```

CLASSNOTFOUND_MSG

```
public static final java.lang.String CLASSNOTFOUND_MSG
```

CONN_FILE

```
public static final int CONN_FILE
```

CONN_FILE_MSG

```
public static final java.lang.String CONN_FILE_MSG
```

ERR_OUTPUT

```
public static final int ERR_OUTPUT
```

ERR_OUTPUT_MSG

```
public static final java.lang.String ERR_OUTPUT_MSG
```

ERRORINCLUDING

```
public static final int ERRORINCLUDING
```

ERRORINCLUDING_MSG

```
public static final java.lang.String ERRORINCLUDING_MSG
```

ERRORLOADINGURL

```
public static final int ERRORLOADINGURL
```

ERRORLOADINGURL_MSG

```
public static final java.lang.String ERRORLOADINGURL_MSG
```

ERRORREADINGPARAM

```
public static final int ERRORREADINGPARAM
```

ERRORREADINGPARAM_MSG

```
public static final java.lang.String ERRORREADINGPARAM_MSG
```

FATAL_SHEETPOOL

```
public static final int FATAL_SHEETPOOL
```

FATAL_SHEETPOOL_MSG

```
public static final java.lang.String FATAL_SHEETPOOL_MSG
```

ILLFORMEDXMLPARAMVAL

```
public static final int ILLFORMEDXMLPARAMVAL
```

ILLFORMEDXMLPARAMVAL_MSG

```
public static final java.lang.String ILLFORMEDXMLPARAMVAL_MSG
```

ILLFORMEDXMLRESOURCE

```
public static final int ILLFORMEDXMLRESOURCE
```

ILLFORMEDXMLRESOURCE_MSG

```
public static final java.lang.String ILLFORMEDXMLRESOURCE_MSG
```

INSTANTIATIONERR

```
public static final int INSTANTIATIONERR
```

INSTANTIATIONERR_MSG

```
public static final java.lang.String INSTANTIATIONERR_MSG
```

INVALID_URI

```
public static final int INVALID_URI
```

INVALID_URI_MSG

```
public static final java.lang.String INVALID_URI_MSG
```

INVALIDURL

```
public static final int INVALIDURL
```

INVALIDURL_MSG

```
public static final java.lang.String INVALIDURL_MSG
```

MISSING_ARGS

```
public static final int MISSING_ARGS
```

MISSING_ARGS_MSG

```
public static final java.lang.String MISSING_ARGS_MSG
```

MISSING_ATTR

```
public static final int MISSING_ATTR
```

MISSING_ATTR_MSG

```
public static final java.lang.String MISSING_ATTR_MSG
```

NAMED_CONN

```
public static final int NAMED_CONN
```

NAMED_CONN_MSG

```
public static final java.lang.String NAMED_CONN_MSG
```

NO_CONN

```
public static final int NO_CONN
```

NO_CONN_DEF

```
public static final int NO_CONN_DEF
```

NO_CONN_DEF_MSG

```
public static final java.lang.String NO_CONN_DEF_MSG
```

NO_CONN_MSG

```
public static final java.lang.String NO_CONN_MSG
```

NO_XSQL_FILE

```
public static final int NO_XSQL_FILE
```

NO_XSQL_FILE_MSG

```
public static final java.lang.String NO_XSQL_FILE_MSG
```

NOFUNCTIONNAME

```
public static final int NOFUNCTIONNAME
```

NOFUNCTIONNAME_MSG

```
public static final java.lang.String NOFUNCTIONNAME_MSG
```

NOPOSTEDXML

```
public static final int NOPOSTEDXML
```

NOPOSTEDXML_MSG

```
public static final java.lang.String NOPOSTEDXML_MSG
```

NOQUERYSUPPLIED

```
public static final int NOQUERYSUPPLIED
```

NOQUERYSUPPLIED_MSG

```
public static final java.lang.String NOQUERYSUPPLIED_MSG
```

NOTANACTIONHANDLER

```
public static final int NOTANACTIONHANDLER
```

NOTANACTIONHANDLER_MSG

```
public static final java.lang.String NOTANACTIONHANDLER_MSG
```

NULLPARAM

```
public static final int NULLPARAM
```

NULLPARAM_MSG

```
public static final java.lang.String NULLPARAM_MSG
```

OWAXMLMALFORMED

```
public static final int OWAXMLMALFORMED
```

OWAXMLMALFORMED_MSG

```
public static final java.lang.String OWAXMLMALFORMED_MSG
```

POSTEDXML_ERR

```
public static final int POSTEDXML_ERR
```

POSTEDXML_ERR_MSG

```
public static final java.lang.String POSTEDXML_ERR_MSG
```

UNHANDLED_ERR

```
public static final int UNHANDLED_ERR
```

UNHANDLED_ERR_MSG

```
public static final java.lang.String UNHANDLED_ERR_MSG
```

UNHANDLED_ERR_XSL_PR

```
public static final int UNHANDLED_ERR_XSL_PR
```

UNHANDLED_ERR_XSL_PR_MSG

```
public static final java.lang.String UNHANDLED_ERR_XSL_PR_MSG
```

UNHANDLED_ERR_XSL_RD

```
public static final int UNHANDLED_ERR_XSL_RD
```

UNHANDLED_ERR_XSL_RD_MSG

```
public static final java.lang.String UNHANDLED_ERR_XSL_RD_MSG
```

XML_INS_ERR

```
public static final int XML_INS_ERR
```

XML_INS_ERR_MSG

```
public static final java.lang.String XML_INS_ERR_MSG
```

XML_PARSE

```
public static final int XML_PARSE
```

XML_PARSE_MSG

```
public static final java.lang.String XML_PARSE_MSG
```

XML_SQL_ERR

```
public static final int XML_SQL_ERR
```

XML_SQL_ERR_MSG

```
public static final java.lang.String XML_SQL_ERR_MSG
```

XSL_ERRORS

```
public static final int XSL_ERRORS
```

XSL_ERRORS_MSG

```
public static final java.lang.String XSL_ERRORS_MSG
```

XSL_NOFILE

```
public static final int XSL_NOFILE
```

XSL_NOFILE_MSG

```
public static final java.lang.String XSL_NOFILE_MSG
```

XSL_PARSE

```
public static final int XSL_PARSE
```

XSL_PARSE_MSG

```
public static final java.lang.String XSL_PARSE_MSG
```

XSLNOTFOUND

```
public static final int XSLNOTFOUND
```

XSLNOTFOUND_MSG

```
public static final java.lang.String XSLNOTFOUND_MSG
```

Constructors

Res()

```
public  Res()
```

Methods

format(int, String)

```
public static java.lang.String format(int id, java.lang.String s)
```

getString(int)

```
public static java.lang.String getString(int id)
```

XSQLActionHandler

Syntax

```
public interface XSQLActionHandler
```

All Known Implementing Classes:

XSQLActionHandlerImpl

Description

Interface that must be implemented by all XSQL Action Element Handlers

Upon encountering an XSQL Action Element of the form <xsql:xxxx> in an XSQL page, the XSQL Page Processor invokes the associated XSQL Action Handler by:

1. Constructing an instance of the handler using the no-args constructor

Invoking the XSQL Action Handler's handleAction() method

NOTE: conn parameter can be null if no connection specified for the XSQL page being processed.

Member Summary	
Methods	
handleAction(Node)	Handle the action, typically by executing some code and appending new child DOM nodes to the rootNode.
init(XSQLPageRequest, Element)	Initialize the Action Handler

Methods

handleAction(Node)

```
public void handleAction(oracle.xml.xsql.Node rootNode)
```

Handle the action, typically by executing some code and appending new child DOM nodes to the rootNode.

The XSQL Page Processor replaces the action element in the XSQL Page being processed with the document fragment of nodes that your handleAction method appends to the rootNode.

Parameters:

rootNode - Root node of generated document fragment

init(XSQLPageRequest, Element)

```
public void init(XSQLPageRequest env, oracle.xml.xsql.Element e)
```

Initialize the Action Handler

Parameters:

env - XSQLPageRequest object

e - DOM element representing the Action Element being handled

XSQLActionHandlerImpl

Syntax

```
public abstract class XSQLActionHandlerImpl extends java.lang.Object implements
XSQLActionHandler

java.lang.Object
  |
  +--oracle.xml.xsql.XSQLActionHandlerImpl
```

All Implemented Interfaces:

XSQLActionHandler

Description

Base Implementation of XSQLActionHandler that can be extended to create your own custom handlers.

Includes a set of useful helper methods.

NOTE: If you extend this class and override the init() method, make sure to call:

```
super.init(env,e);
```

Member Summary

Constructors

XSQLActionHandlerImpl()

Methods

init(XSQLPageRequest, Element)

Inherited Member Summary

Methods inherited from class java.lang.Object

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

Methods inherited from interface XSQLActionHandler

handleAction(Node)

Constructors

XSQLActionHandlerImpl()

```
public  XSQLActionHandlerImpl()
```

Methods

init(XSQLPageRequest, Element)

```
public void init(XSQLPageRequest env, oracle.xml.xsql.Element e)
```

Specified By:

init(XSQLPageRequest, Element) in interface XSQLActionHandler

XSQLCommandLine

Syntax

```
public final class XSQLCommandLine extends java.lang.Object

java.lang.Object
  |
  +--oracle.xml.xsql.XSQLCommandLine
```

Description

Command-line Utility to process XSQL Pages.

Member Summary

Constructors

XSQLCommandLine()

Methods

main(String[])

Inherited Member Summary

Methods inherited from class java.lang.Object

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

Constructors

XSQLCommandLine()

```
public  XSQLCommandLine()
```

Methods

main(String[])

```
public static void main(java.lang.String[] args)
```

XSQLDiagnostic

Syntax

```
public final class XSQLDiagnostic extends java.lang.Object

java.lang.Object
  |
  +--oracle.xml.xsql.XSQLDiagnostic
```

Description

Member Summary
Constructors
XSQLDiagnostic(String)
Methods
debugPrintToFile(String, String)
msg(String)

Inherited Member Summary
Methods inherited from class java.lang.Object
equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructors

XSQLDiagnostic(String)

```
public  XSQLDiagnostic(java.lang.String diagFile)
```

Methods

debugPrintToFile(String, String)

```
public static void debugPrintToFile(java.lang.String msg, java.lang.String
filename)
```

msg(String)

```
public void msg(java.lang.String text)
```

XSQLHttpUtil

Syntax

```
public final class XSQLHttpUtil extends java.lang.Object

java.lang.Object
  |
  +--oracle.xml.xsql.XSQLHttpUtil
```

Description

Member Summary
Constructors
XSQLHttpUtil()
Methods
HttpRequestAsXMLDocument(HttpServletRequest, String)
XL(String, String)

Inherited Member Summary
Methods inherited from class java.lang.Object
equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructors

XSQLHttpUtil()

```
public  XSQLHttpUtil()
```

Methods

HttpRequestAsXMLDocument(HttpServletRequest, String)

```
public static oracle.xml.xsql.XMLDocument
HttpRequestAsXMLDocument(oracle.xml.xsql.HttpServletRequest req,
```

```
java.lang.String XSQLPageEncoding)
```

XL(String, String)

```
public static java.lang.String XL(java.lang.String s, java.lang.String enc)
```

XSQLPageRequest

Syntax

```
public interface XSQLPageRequest
```

All Known Implementing Classes:

XSQLPageRequestImpl

Description

Interface representing a request for an XSQL Page

Member Summary

Methods

createNestedRequest(URL, Dictionary)	Returns an instance of a nested Request
getConnectionName()	Returns the name of the connection being used for this request May be null if no connection set / in-use.
getErrorWriter()	Returns a PrintWriter to print out errors processing this request
getJDBCConnection()	Gets the JDBC connection being used for this request (can be null)
getPageEncoding()	Returns encoding of source XSQL Page associated with this request
getParameter(String)	Returns the value of the requested parameter
getPostedDocument()	Returns the content of Posted XML for this request as an XML Document
getRequestParamsAsXMLDocument()	Returns the content of a Request parameters as an XML Document
getRequestType()	Returns a string identifying the type of page request being made.
getSourceDocumentURI()	Returns a String representation of the requested document's URI
getStylesheetParameter(String)	Gets a stylesheet parameter by name
getStylesheetParameters()	Gets an enumeration of stylesheet parameter names

Member Summary	
getStylesheetURI()	Returns the URI of the stylesheet to be used to process the result.
getUserAgent()	Returns a String identifier of the requesting program
getWriter()	Returns a PrintWriter used for writing out the results of a page request
getXSQLConnection()	Gets the XSQLConnection Object being used for this request Might be null.
isIncludedRequest()	Returns true if this request is being included in another.
isOracleDriver()	Returns true if the current connection uses the Oracle JDBC Driver
printedErrorHeader()	Returns the state of whether an Error Header has been printed
requestProcessed()	Allows Page Request to Perform end-of-request processing
setConnectionName(String)	Sets the connection name to use for this request
setContentType(String)	Sets the content type of the resulting page
setIncludingRequest(XSQLPageRequest)	Sets the Including Page Request object for this request.
setPageEncoding(String)	Sets encoding of source XSQL page associated with this request.
setPageParam(String, String)	Sets a dynamic page parameter value.
setPostedDocument(Document)	Allows programmatic setting of the Posted Document
setPrintedErrorHeader(boolean)	Sets whether an Error Header has been printed
setStylesheetParameter(String, String)	Sets the value of a parameter to be passed to the associated stylesheet
setStylesheetURI(String)	Sets the URI of the stylesheet to be used to process the result.
translateURL(String)	Returns a string representing an absolute URL resolved relative to the base URI for this request.
useConnectionPooling()	Returns true if connection pooling is desired for this request

Member Summary	
useHTMLErrors()	Returns true if HTML-formatted error messages are desired for this request

Methods

createNestedRequest(URL, Dictionary)

```
public XSQLPageRequest createNestedRequest(java.net.URL pageurl,
java.util.Dictionary params)
```
Returns an instance of a nested Request

getConnectionName()

```
public java.lang.String getConnectionName()
```
Returns the name of the connection being used for this request May be null if no connection set / in-use.

getErrorWriter()

```
public java.io.PrintWriter getErrorWriter()
```
Returns a PrintWriter to print out errors processing this request

getJDBCConnection()

```
public java.sql.Connection getJDBCConnection()
```
Gets the JDBC connection being used for this request (can be null)

getPageEncoding()

```
public java.lang.String getPageEncoding()
```
Returns encoding of source XSQL Page associated with this request

getParameter(String)

```
public java.lang.String getParameter(java.lang.String name)
```
Returns the value of the requested parameter

Parameters:

name - the name of the parameter

getPostedDocument()

```
public oracle.xml.xsql.Document getPostedDocument()
```
Returns the content of Posted XML for this request as an XML Document

getRequestParamsAsXMLDocument()

```
public oracle.xml.xsql.Document getRequestParamsAsXMLDocument()
```
Returns the content of a Request parameters as an XML Document

getRequestType()

```
public java.lang.String getRequestType()
```
Returns a string identifying the type of page request being made.

getSourceDocumentURI()

```
public java.lang.String getSourceDocumentURI()
```
Returns a String representation of the requested document's URI

getStylesheetParameter(String)

```
public java.lang.String getStylesheetParameter(java.lang.String name)
```
Gets a stylesheet parameter by name

getStylesheetParameters()

```
public java.util.Enumeration getStylesheetParameters()
```
Gets an enumeration of stylesheet parameter names

getStylesheetURI()

```
public java.lang.String getStylesheetURI()
```
Returns the URI of the stylesheet to be used to process the result.

getUserAgent()

```
public java.lang.String getUserAgent()
```
Returns a String identifier of the requesting program

getWriter()

```
public java.io.PrintWriter getWriter()
```
Returns a PrintWriter used for writing out the results of a page request

getXSQLConnection()

```
public oracle.xml.xsql.XSQLConnection getXSQLConnection()
```
Gets the XSQLConnection Object being used for this request Might be null.

isIncludedRequest()

```
public boolean isIncludedRequest()
```
Returns true if this request is being included in another.

isOracleDriver()

```
public boolean isOracleDriver()
```
Returns true if the current connection uses the Oracle JDBC Driver

printedErrorHeader()

```
public boolean printedErrorHeader()
```
Returns the state of whether an Error Header has been printed

requestProcessed()

```
public void requestProcessed()
```
Allows Page Request to Perform end-of-request processing

setConnectionName(String)

```
public void setConnectionName(java.lang.String connName)
```
Sets the connection name to use for this request

setContentType(String)

```
public void setContentType(java.lang.String mimetype)
```
Sets the content type of the resulting page

setIncludingRequest(XSQLPageRequest)

```
public void setIncludingRequest(XSQLPageRequest includingEnv)
```
Sets the Including Page Request object for this request.

setPageEncoding(String)

```
public void setPageEncoding(java.lang.String enc)
```
Sets encoding of source XSQL page associated with this request.

setPageParam(String, String)

```
public void setPageParam(java.lang.String name, java.lang.String value)
```
Sets a dynamic page parameter value.

setPostedDocument(Document)

```
public void setPostedDocument(oracle.xml.xsql.Document doc)
```
Allows programmatic setting of the Posted Document

setPrintedErrorHeader(boolean)

```
public void setPrintedErrorHeader(boolean yes)
```
Sets whether an Error Header has been printed

setStylesheetParameter(String, String)

```
public void setStylesheetParameter(java.lang.String name, java.lang.String
value)
```
Sets the value of a parameter to be passed to the associated stylesheet

setStylesheetURI(String)

```
public void setStylesheetURI(java.lang.String uri)
```
Sets the URI of the stylesheet to be used to process the result.

translateURL(String)

```
public java.lang.String translateURL(java.lang.String url)
```
Returns a string representing an absolute URL resolved relative to the base URI for this request.

useConnectionPooling()

```
public boolean useConnectionPooling()
```
Returns true if connection pooling is desired for this request

useHTMLErrors()

```
public boolean useHTMLErrors()
```
Returns true if HTML-formatted error messages are desired for this request

setRequestObject

```
void setRequestObject(java.lang.String name, java.lang.Object obj)
```
Sets a request-scope object

getRequestObject

```
java.lang.Object getRequestObject(java.lang.String name)
```

Gets a request-scope object

XSQLPageRequestImpl

Syntax

```
public abstract class XSQLPageRequestImpl extends java.lang.Object implements
XSQLPageRequest

java.lang.Object
  |
  +--oracle.xml.xsql.XSQLPageRequestImpl
```

Direct Known Subclasses:

XSQLServletPageRequest

All Implemented Interfaces:

XSQLPageRequest

Description

Base implementation of the XSQLPageRequest interface that case be used to derive new kinds of page request implementations.

Member Summary

Constructors

XSQLPageRequestImpl()

XSQLPageRequestImpl(Hashtable)

XSQLPageRequestImpl(String, Hashtable)

Methods

getConnectionName()	Returns the name of the connection being used for this request May be null if no connection set / in-use.
getErrorWriter()	
getJDBCConnection()	Gets the JDBC connection being used for this request (can be null)
getPageEncoding()	Returns encoding of source XSQL Page associated with this request

Member Summary	
getParameter(String)	
getPostedDocument()	
getRequestParamsAsXMLDocument()	
getSourceDocumentURI()	
getStylesheetParameter(String)	Gets a stylesheet parameter by name
getStylesheetParameters()	Gets an enumeration of stylesheet parameter names
getStylesheetURI()	
getUserAgent()	
getWriter()	
getXSQLConnection()	Gets the XSQLConnection Object being used for this request Might be null.
isIncludedRequest()	Returns true if this request is being included in another.
isOracleDriver()	Returns true if the current connection uses the Oracle JDBC Driver
printedErrorHeader()	
requestProcessed()	Allows Page Request to Perform end-of-request processing
setConnectionName(String)	Sets the connection being used for this request (can be null)
setContentType(String)	
setIncludingRequest(XSQLPageRequest)	Sets the Including Page Request object for this request.
setPageEncoding(String)	Associates an XSQL Page with the request
setPageParam(String, String)	Sets a dynamic page parameter value.
setPostedDocument(Document)	
setPrintedErrorHeader(boolean)	
setStylesheetParameter(String, String)	
setStylesheetURI(String)	
translateURL(String)	

Member Summary
useConnectionPooling()
useHTMLErrors()

Inherited Member Summary
Methods inherited from class java.lang.Object
equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait
Methods inherited from interface XSQLPageRequest
createNestedRequest(URL, Dictionary), getRequestType()

Constructors

XSQLPageRequestImpl()

```
public  XSQLPageRequestImpl()
```

XSQLPageRequestImpl(Hashtable)

```
public  XSQLPageRequestImpl(java.util.Hashtable parameters)
```

XSQLPageRequestImpl(String, Hashtable)

```
public  XSQLPageRequestImpl(java.lang.String pageurl, java.util.Hashtable
parameters)
```

Methods

getConnectionName()

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

Returns the name of the connection being used for this request May be null if no connection set / in-use.

Specified By:

getConnectionName() in interface XSQLPageRequest

getErrorWriter()

```
public java.io.PrintWriter getErrorWriter()
```

Specified By:

getErrorWriter() in interface XSQLPageRequest

getJDBCConnection()

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

Gets the JDBC connection being used for this request (can be null)

Specified By:

getJDBCConnection() in interface XSQLPageRequest

getPageEncoding()

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

Returns encoding of source XSQL Page associated with this request

Specified By:

getPageEncoding() in interface XSQLPageRequest

getParameter(String)

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

Specified By:

getParameter(String) in interface XSQLPageRequest

getPostedDocument()

```
public oracle.xml.xsql.Document getPostedDocument()
```

Specified By:

getPostedDocument() in interface XSQLPageRequest

getRequestParamsAsXMLDocument()

```
public oracle.xml.xsql.Document getRequestParamsAsXMLDocument()
```

Specified By:

getRequestParamsAsXMLDocument() in interface XSQLPageRequest

getSourceDocumentURI()

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

Specified By:

getSourceDocumentURI() in interface XSQLPageRequest

getStylesheetParameter(String)

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

Gets a stylesheet parameter by name

Specified By:

getStylesheetParameter(String) in interface XSQLPageRequest

getStylesheetParameters()

```
public java.util.Enumeration getStylesheetParameters()
```

Gets an enumeration of stylesheet parameter names

Specified By:

getStylesheetParameters() in interface XSQLPageRequest

getStylesheetURI()

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

Specified By:

getStylesheetURI() in interface XSQLPageRequest

getUserAgent()

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

Specified By:

getUserAgent() in interface XSQLPageRequest

getWriter()

```
public java.io.PrintWriter getWriter()
```

Specified By:

getWriter() in interface XSQLPageRequest

getXSQLConnection()

```
public oracle.xml.xsql.XSQLConnection getXSQLConnection()
```

Gets the XSQLConnection Object being used for this request Might be null.

Specified By:

getXSQLConnection() in interface XSQLPageRequest

isIncludedRequest()

```
public boolean isIncludedRequest()
```

Returns true if this request is being included in another.

Specified By:

isIncludedRequest() in interface XSQLPageRequest

isOracleDriver()

```
public boolean isOracleDriver()
```

Returns true if the current connection uses the Oracle JDBC Driver

Specified By:

isOracleDriver() in interface XSQLPageRequest

printedErrorHeader()

```
public boolean printedErrorHeader()
```

Specified By:

printedErrorHeader() in interface XSQLPageRequest

requestProcessed()

```
public void requestProcessed()
```

Allows Page Request to Perform end-of-request processing

Specified By:

requestProcessed() in interface XSQLPageRequest

setConnectionName(String)

`public void setConnectionName(java.lang.String connName)`
Sets the connection being used for this request (can be null)

Specified By:

setConnectionName(String) in interface XSQLPageRequest

setContentType(String)

`public void setContentType(java.lang.String mimetype)`

Specified By:

setContentType(String) in interface XSQLPageRequest

setIncludingRequest(XSQLPageRequest)

`public void setIncludingRequest(XSQLPageRequest includingEnv)`
Sets the Including Page Request object for this request.

Specified By:

setIncludingRequest(XSQLPageRequest) in interface XSQLPageRequest

setPageEncoding(String)

`public void setPageEncoding(java.lang.String enc)`
Associates an XSQL Page with the request

Specified By:

setPageEncoding(String) in interface XSQLPageRequest

setPageParam(String, String)

`public void setPageParam(java.lang.String name, java.lang.String value)`
Sets a dynamic page parameter value.

Specified By:

setPageParam(String, String) in interface XSQLPageRequest

setPostedDocument(Document)

```
public void setPostedDocument(oracle.xml.xsql.Document doc)
```

Specified By:

setPostedDocument(Document) in interface XSQLPageRequest

setPrintedErrorHeader(boolean)

```
public void setPrintedErrorHeader(boolean yes)
```

Specified By:

setPrintedErrorHeader(boolean) in interface XSQLPageRequest

setStylesheetParameter(String, String)

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

Specified By:

setStylesheetParameter(String, String) in interface XSQLPageRequest

setStylesheetURI(String)

```
public void setStylesheetURI(java.lang.String uri)
```

Specified By:

setStylesheetURI(String) in interface XSQLPageRequest

translateURL(String)

```
public java.lang.String translateURL(java.lang.String filename)
```

Specified By:

translateURL(String) in interface XSQLPageRequest

useConnectionPooling()

```
public boolean useConnectionPooling()
```

Specified By:

useConnectionPooling() in interface XSQLPageRequest

useHTMLErrors()

```
public boolean useHTMLErrors()
```

Specified By:

useHTMLErrors() in interface XSQLPageRequest

setRequestObject

```
void setRequestObject(java.lang.String name, java.lang.Object obj)
```

Sets a request-scope object

getRequestObject

```
java.lang.Object getRequestObject(java.lang.String name)
```

Gets a request-scope object

XSQLParserHelper

Syntax

```
public final class XSQLParserHelper extends java.lang.Object

java.lang.Object
  |
  +--oracle.xml.xsql.XSQLParserHelper
```

Description

Common XML Parsing Routines

Member Summary

Constructors

XSQLParserHelper()

Methods

newDocument()

parse(InputStream, URL, PrintWriter)

parse(Reader, PrintWriter)

parse(URL, PrintWriter)

parseFromString(StringBuffer, PrintWriter)

parseFromString(String, PrintWriter)

print(Document, PrintWriter)

Inherited Member Summary

Methods inherited from class java.lang.Object

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

Constructors

XSQLParserHelper()

```
public  XSQLParserHelper()
```

Methods

newDocument()

```
public static oracle.xml.xsql.Document newDocument()
```

parse(InputStream, URL, PrintWriter)

```
public static oracle.xml.xsql.Document parse(java.io.InputStream is,
java.net.URL baseUrl, java.io.PrintWriter errorWriter)
```

parse(Reader, PrintWriter)

```
public static oracle.xml.xsql.Document parse(java.io.Reader r,
java.io.PrintWriter errorWriter)
```

parse(URL, PrintWriter)

```
public static oracle.xml.xsql.Document parse(java.net.URL url,
java.io.PrintWriter errorWriter)
```

parseFromString(StringBuffer, PrintWriter)

```
public static oracle.xml.xsql.Document parseFromString(java.lang.StringBuffer
xmlString, java.io.PrintWriter errorWriter)
```

parseFromString(String, PrintWriter)

```
public static oracle.xml.xsql.Document parseFromString(java.lang.String
xmlString, java.io.PrintWriter errorWriter)
```

print(Document, PrintWriter)

```
public static void print(oracle.xml.xsql.Document d, java.io.PrintWriter out)
```

XSQLRequest

Syntax

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

java.lang.Object
  |
  +--oracle.xml.xsql.XSQLRequest
```

Description

Programmatically process a request for an XSQL Page.

Member Summary

Constructors	
XSQLRequest(String)	Create a Request for an XSQL Page
XSQLRequest(String, XSQLPageRequest)	Create a Request for an XSQL Page
XSQLRequest(URL)	Create a Request for an XSQL Page
XSQLRequest(oracle.xml.parser.v2.XMLDocument page, java.net.URL baseURL)	Create a Request for an XSQL Page
XSQLRequest(XSQLConnectionManagerFactory fact, java.lang.String url)	Create a Request for an XSQL Page using a custom connection manager factory
XSQLRequest(XSQLConnectionManagerFactory fact, java.net.URL url)	Create a Request for an XSQL Page with a custom connection manager factory.
XSQLRequest(XSQLConnectionManagerFactory fact, oracle.xml.parser.v2.XMLDocument page, java.net.URL baseURL)	Create a Request for an XSQL Page
XSQLRequest(URL, XSQLPageRequest)	Create a Request for an XSQL Page
Methods	

Member Summary	
process()	Process the request, writing output / errors to System.out / System.err
process(Dictionary)	Process the request, writing output / errors to System.out / System.err
process(Dictionary, PrintWriter, PrintWriter)	Process the request, writing output / errors to respective PrintWriters
process(PrintWriter, PrintWriter)	Process the request, writing output / errors to respective PrintWriters
processToXML()	Process the request, writing output / errors to System.out / System.err
processToXML(Dictionary)	Process the request, writing output / errors to System.out / System.err
processToXML(Dictionary, PrintWriter)	Process the request, writing output / errors to respective PrintWriters
processToXML(PrintWriter)	Process the request, writing errors to respective PrintWriters
setPostedDocument(Document)	Programmatically set an XML Document to be treated the same as if it were posted as part of the request.

Inherited Member Summary

Methods inherited from class java.lang.Object

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

Constructors

XSQLRequest(String)

```
public  XSQLRequest(java.lang.String url)
```
Create a Request for an XSQL Page

Parameters:

`url` - String representation of an URL to an XSQL Page

XSQLRequest(String, XSQLPageRequest)

```
public  XSQLRequest(java.lang.String url, XSQLPageRequest env)
```
Create a Request for an XSQL Page

Parameters:

`url` - String representation of an URL to an XSQL Page

`env` - Calling XSQLPageRequest environment

XSQLRequest(URL)

```
public  XSQLRequest(java.net.URL url)
```
Create a Request for an XSQL Page

Parameters:

`url` - URL to an XSQL Page

XSQLRequest(URL, XSQLPageRequest)

```
public  XSQLRequest(java.net.URL url, XSQLPageRequest env)
```
Create a Request for an XSQL Page

Parameters:

`url` - URL to an XSQL Page

`env` - Calling XSQLPageRequest environment

Methods

process()

```
public void process()
```

Process the request, writing output / errors to System.out / System.err

process(Dictionary)

```
public void process(java.util.Dictionary params)
```
Process the request, writing output / errors to System.out / System.err

Parameters:

`params` - Dictionary (e.g. Hashtable) with XSQL Page parameters

process(Dictionary, PrintWriter, PrintWriter)

```
public void process(java.util.Dictionary params, java.io.PrintWriter out,
java.io.PrintWriter err)
```
Process the request, writing output / errors to respective PrintWriters

Parameters:

`params` - Dictionary (e.g. Hashtable) with XSQL Page parameters

`out` - PrintWriter to use to write the resulting page results

`err` - PrintWriter to use to write the resulting page errors

process(PrintWriter, PrintWriter)

```
public void process(java.io.PrintWriter out, java.io.PrintWriter err)
```
Process the request, writing output / errors to respective PrintWriters

Parameters:

`out` - PrintWriter to use to write the resulting page results

`err` - PrintWriter to use to write the resulting page errors

processToXML()

```
public org.w3c.dom.Document processToXML()
```
Process the request, writing output / errors to System.out / System.err

processToXML(Dictionary)

```
public org.w3c.dom.Document processToXML(java.util.Dictionary params)
```
Process the request, writing output / errors to System.out / System.err

Parameters:

params - Dictionary (e.g. Hashtable) with XSQL Page parameters

processToXML(Dictionary, PrintWriter)

```
public org.w3c.dom.Document processToXML(java.util.Dictionary params,
java.io.PrintWriter err)
```

Process the request, writing output/errors to respective PrintWriters

Parameters:

params - Dictionary (e.g. Hashtable) with XSQL Page parameters

err - PrintWriter to use to write the resulting page errors

processToXML(PrintWriter)

```
public org.w3c.dom.Document processToXML(java.io.PrintWriter err)
```

Process the request, writing errors to respective PrintWriters

Parameters:

err - PrintWriter to use to write the resulting page errors

setPostedDocument(Document)

```
public void setPostedDocument(org.w3c.dom.Document postedDoc)
```

Programmatically set an XML Document to be treated the same as if it were posted as part of the request.

Parameters:

postedDoc - DOM Document

XSQLRequestObjectListener

Interface that an object created by an action handler can implement to be notified when the current page request processing is completed. It has a single method:

```
void pageProcessingCompleted()
```

Objects that implement this interface and which are added to the current request context using XSQLPageRequest::setRequestObject() will be notified when the page processing of the outermost page is completed.

XSQLServlet

Syntax

```
public final class XSQLServlet
```

`oracle.xml.xsql.XSQLServlet`

Description

Servlet to enable HTTP GET-ing of and POST-ing to XSQL Pages

Member Summary

Constructors

XSQLServlet()

Methods

doGet(HttpServletRequest, HttpServletResponse)

doPost(HttpServletRequest, HttpServletResponse)

getServletInfo()

init(ServletConfig)

inJServ()

Constructors

XSQLServlet()

```
public  XSQLServlet()
```

Methods

doGet(HttpServletRequest, HttpServletResponse)

```
public void doGet(oracle.xml.xsql.HttpServletRequest request,
oracle.xml.xsql.HttpServletResponse response)
```

doPost(HttpServletRequest, HttpServletResponse)

```
public void doPost(oracle.xml.xsql.HttpServletRequest request,
oracle.xml.xsql.HttpServletResponse response)
```

getServletInfo()

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

init(ServletConfig)

```
public void init(oracle.xml.xsql.ServletConfig config)
```

inJServ()

```
public static boolean inJServ()
```

XSQLServletPageRequest

Syntax

```
public final class XSQLServletPageRequest extends XSQLPageRequestImpl

java.lang.Object
  |
  +--XSQLPageRequestImpl
        |
        +--oracle.xml.xsql.XSQLServletPageRequest
```

All Implemented Interfaces:

XSQLPageRequest

Description

Implementation of XSQLPageRequest for Servlet-based XSQL Page requests.

Member Summary

Constructors

XSQLServletPageRequest(HttpServletRequest, HttpServletResponse)

Methods

createNestedRequest(URL, Dictionary)	Returns an instance of a nested Request
getCookie(String)	
getHttpServletRequest()	Get the HttpServletRequest that initiated this XSQL Page Request.
getHttpServletResponse()	Get the HttpServletResponse that is associated with this XSQL Page Request
getParameter(String)	Use HTTP Parameters as the source of parameters instead
getPostedDocument()	
getRequestParamsAsXMLDocument()	

Member Summary
getRequestType()
getUserAgent()
setContentType(String)
setPageEncoding(String)
translateURL(String)
useHTMLErrors()

Inherited Member Summary

Methods inherited from class XSQLPageRequestImpl

getConnectionName(), getErrorWriter(), getJDBCConnection(), getPageEncoding(), getSourceDocumentURI(), getStylesheetParameter(String), getStylesheetParameters(), getStylesheetURI(), getWriter(), getXSQLConnection(), isIncludedRequest(), isOracleDriver(), printedErrorHeader(), requestProcessed(), setConnectionName(String), setIncludingRequest(XSQLPageRequest), setPageParam(String, String), setPostedDocument(Document), setPrintedErrorHeader(boolean), setStylesheetParameter(String, String), setStylesheetURI(String), useConnectionPooling()

Methods inherited from class java.lang.Object

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

Methods inherited from interface XSQLPageRequest

getConnectionName(), getErrorWriter(), getJDBCConnection(), getPageEncoding(), getSourceDocumentURI(), getStylesheetParameter(String), getStylesheetParameters(), getStylesheetURI(), getWriter(), getXSQLConnection(), isIncludedRequest(), isOracleDriver(), printedErrorHeader(), requestProcessed(), setConnectionName(String), setIncludingRequest(XSQLPageRequest), setPageParam(String, String), setPostedDocument(Document), setPrintedErrorHeader(boolean), setStylesheetParameter(String, String), setStylesheetURI(String), useConnectionPooling()

Constructors

XSQLServletPageRequest(HttpServletRequest, HttpServletResponse)

```
public  XSQLServletPageRequest(oracle.xml.xsql.HttpServletRequest req,
oracle.xml.xsql.HttpServletResponse resp)
```

Methods

createNestedRequest(URL, Dictionary)

```
public XSQLPageRequest createNestedRequest(java.net.URL pageurl,
java.util.Dictionary params)
```
Returns an instance of a nested Request

getCookie(String)

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

getHttpServletRequest()

```
public oracle.xml.xsql.HttpServletRequest getHttpServletRequest()
```
Get the HttpServletRequest that initiated this XSQL Page Request.

getHttpServletResponse()

```
public oracle.xml.xsql.HttpServletResponse getHttpServletResponse()
```
Get the HttpServletResponse that is associated with this XSQL Page Request

getParameter(String)

```
public java.lang.String getParameter(java.lang.String name)
```
Use HTTP Parameters as the source of parameters instead

Overrides:

getParameter(String) in class XSQLPageRequestImpl

getPostedDocument()

```
public oracle.xml.xsql.Document getPostedDocument()
```

Overrides:

getPostedDocument() in class XSQLPageRequestImpl

getRequestParamsAsXMLDocument()

```
public oracle.xml.xsql.Document getRequestParamsAsXMLDocument()
```

Overrides:

getRequestParamsAsXMLDocument() in class XSQLPageRequestImpl

getRequestType()

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

getUserAgent()

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

Overrides:

getUserAgent() in class XSQLPageRequestImpl

setContentType(String)

```
public void setContentType(java.lang.String mimetype)
```

Overrides:

setContentType(String) in class XSQLPageRequestImpl

setPageEncoding(String)

```
public void setPageEncoding(java.lang.String enc)
```

Overrides:

setPageEncoding(String) in class XSQLPageRequestImpl

translateURL(String)

```
public java.lang.String translateURL(java.lang.String path)
```

Overrides:

translateURL(String) in class XSQLPageRequestImpl

useHTMLErrors()

```
public boolean useHTMLErrors()
```

Overrides:

useHTMLErrors() in class XSQLPageRequestImpl

XSQLStylesheetProcessor

Syntax

```
public final class XSQLStylesheetProcessor extends java.lang.Object

java.lang.Object
  |
  +--oracle.xml.xsql.XSQLStylesheetProcessor
```

Description

XSLT Stylesheet Processing Engine

Member Summary

Constructors

XSQLStylesheetProcessor()

Methods

processToDocument(Document, String, XSQLPageRequest)

processToWriter(Document, String, XSQLPageRequest)

Inherited Member Summary

Methods inherited from class java.lang.Object

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

Constructors

XSQLStylesheetProcessor()

```
public  XSQLStylesheetProcessor()
```

Methods

processToDocument(Document, String, XSQLPageRequest)

```
public static oracle.xml.xsql.Document
processToDocument(oracle.xml.xsql.Document xml, java.lang.String xslURI,
XSQLPageRequest env)
```

processToWriter(Document, String, XSQLPageRequest)

```
public static void processToWriter(oracle.xml.xsql.Document xml,
java.lang.String xslURI, XSQLPageRequest env)
```

getServletContext()

```
javax.servlet.ServletContext getServletContext()
```

Gets the Http Servlet Context

XSQLUtil

Syntax

```
public final class XSQLUtil extends java.lang.Object

java.lang.Object
  |
  +--oracle.xml.xsql.XSQLUtil
```

Description

Member Summary

Constructors

XSQLUtil()

Methods

DictionaryOfParamsAsXMLDocument(Dictionary)

safeURLAsString(URL)

select(Document, String)

select(Element, String)

select(XMLDocument, String)

select(XMLElement, String)

selectFirst(Document, String)

selectFirst(Element, String)

selectFirst(XMLDocument, String)

selectFirst(XMLElement, String)

stringParamValue(Object)

translate(String, String)

translate(URL, String)

valueOf(Element, String)

valueOf(Node, String)

valueOf(XMLElement, String)

Member Summary
valueOf(XMLNode, String)
XL(String, String)

Inherited Member Summary
Methods inherited from class java.lang.Object
equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructors

XSQLUtil()

```
public  XSQLUtil()
```

Methods

DictionaryOfParamsAsXMLDocument(Dictionary)

```
public static oracle.xml.xsql.XMLDocument
DictionaryOfParamsAsXMLDocument(java.util.Dictionary dict)
```

safeURLAsString(URL)

```
public static java.lang.String safeURLAsString(java.net.URL u)
```

select(Document, String)

```
public static oracle.xml.xsql.NodeList select(oracle.xml.xsql.Document d,
java.lang.String pattern)
```

select(Element, String)

```
public static oracle.xml.xsql.NodeList select(oracle.xml.xsql.Element n,
java.lang.String pattern)
```

select(XMLDocument, String)

```
public static oracle.xml.xsql.NodeList select(oracle.xml.xsql.XMLDocument d,
java.lang.String pattern)
```

select(XMLElement, String)

```
public static oracle.xml.xsql.NodeList select(oracle.xml.xsql.XMLElement n,
java.lang.String pattern)
```

selectFirst(Document, String)

```
public static oracle.xml.xsql.Node selectFirst(oracle.xml.xsql.Document d,
java.lang.String pattern)
```

selectFirst(Element, String)

```
public static oracle.xml.xsql.Node selectFirst(oracle.xml.xsql.Element n,
java.lang.String pattern)
```

selectFirst(XMLDocument, String)

```
public static oracle.xml.xsql.Node selectFirst(oracle.xml.xsql.XMLDocument d,
java.lang.String pattern)
```

selectFirst(XMLElement, String)

```
public static oracle.xml.xsql.Node selectFirst(oracle.xml.xsql.XMLElement n,
java.lang.String pattern)
```

stringParamValue(Object)

```
public static java.lang.String stringParamValue(java.lang.Object val)
```

translate(String, String)

```
public static java.lang.String translate(java.lang.String u, java.lang.String
path)
```

translate(URL, String)

```
public static java.lang.String translate(java.net.URL u, java.lang.String path)
```

valueOf(Element, String)

```
public static java.lang.String valueOf(oracle.xml.xsql.Element n,
java.lang.String pattern)
```

valueOf(Node, String)

```
public static java.lang.String valueOf(oracle.xml.xsql.Node n, java.lang.String
pattern)
```

valueOf(XMLElement, String)

```
public static java.lang.String valueOf(oracle.xml.xsql.XMLElement n,
java.lang.String pattern)
```

valueOf(XMLNode, String)

```
public static java.lang.String valueOf(oracle.xml.xsql.XMLNode n,
java.lang.String pattern)
```

XL(String, String)

```
public static java.lang.String XL(java.lang.String s, java.lang.String enc)
```

XSQLRequestObjectListener

Interface that an object created by an action handler can implement to be notified when the current page request processing is completed.

It has a single method:

```
void pageProcessingCompleted()
```

Objects that implement this interface and which are added to the current request context using XSQLPageRequest::setRequestObject() will be notified when the page processing of the outermost page is

```
completed.
```

The following constructors were added to oracle.xml.xsql.XSQLRequest class:

XSQLRequest(oracle.xml.parser.v2.XMLDocument page, java.net.URL baseURL)
Create a Request for an XSQL Page

XSQLRequest(XSQLConnectionManagerFactory fact, java.lang.String url)
Create a Request for an XSQL Page using a custom connection manager factory

XSQLRequest(XSQLConnectionManagerFactory fact, java.net.URL url)
Create a Request for an XSQL Page with a custom connection manager factory.

XSQLRequest(XSQLConnectionManagerFactory fact, oracle.xml.parser.v2.XMLDocument page, java.net.URL baseURL)
Create a Request for an XSQL Page

==========

The following are new interfaces

==========

---

public interface XSQLConnectionManager

One of two interfaces that must be implemented to override the built-in connection manager implementation. The XSQL Page Processor asks the XSQLConnectionManagerFactory associated with each request to create() an instance of an XSQLConnectionManager to service the current request.

In multithreaded environments, the implementation of XSQLConnectionManager must insure that an XSQLConnection instance returned by getConnection() is not used by another thread until it has been released by the XSQL Page Processor after the call to releaseConnection().

Method Summary

XSQLConnection getConnection(java.lang.String connName, XSQLPageRequest env)

void releaseConnection(XSQLConnection theConn, XSQLPageRequest env)

---

p
public interface XSQLConnectionManagerFactory

One of two interfaces that must be implemented to override the built-in connection manager implementation. The XSQL Page Processor asks the XSQLConnectionManagerFactory associated with each request to create() an instance of an XSQLConnectionManager to service the current request.

Method Summary

XSQLConnectionManager create()

---

public interface XSQLDocumentSerializer

Interface that must be implemented by all XSQL Serializers which serialize an XSQL data page as an XML Document to a PrintWriter.

Upon encountering a serializer="XXX" pseudo-attribute in an <?xml-stylesheet?> processing instruction, the XSQL Page Processor invokes the associated serializer by:

- Constructing an instance of the serializer using the no-args constructor
- Invoking the XSQL document serializer's serialize() method

NOTE: An implementation of XSQLDocumentSerializer is expected to do the following actions.

- First, call env.setContentType() to set the content type

- Then, call env.getWriter() to get the Writer to write to

If the serializer throws an unhandled exception, the XSQL Page Processor will format the stacktrace.

See oracle.xml.xsql.src.serializers.XSQLSampleSerializer for an example.

Method Summary
void serialize(org.w3c.dom.Document doc, XSQLPageRequest env)

9

Package oracle.xml.classgen

CGDocument

Syntax

```
public abstract class CGDocument extends oracle.xml.classgen.CGNode

java.lang.Object
  |
  +--oracle.xml.classgen.CGNode
        |
        +--oracle.xml.classgen.CGDocument
```

Description

Serves as the base document class for the DTD compiler generated classes

Constructors

CGDocument()

```
protected  CGDocument()
```

CGDocument(String, DTD)

```
protected  CGDocument(java.lang.String doctype, oracle.xml.parser.v2.DTD dtd)
```

Constructor for the Root element of the DTD

Parameters:

`doctype` - Name of the root Element of the DTD

`dtd` - The DTD used to generate the classes

Methods

print(OutputStream)

```
protected void print(java.io.OutputStream out)
```

Prints the constructed XML Document

Parameters:

`out` - Output stream to which the document will be printed

Throws:

`InvalidContentException` - Throw exception if the document's content do not match the grammer specified by DTD (The validation mode should be set to TRUE)

See Also:

`oracle.xml.classgen.ClassGenerator`

print(OutputStream, String)

`protected void print(java.io.OutputStream out, java.lang.String enc)`
Prints the constructed XML Document

Parameters:

`out` - Output stream to which the document will be printed

`enc` - Encoding of the output stream

Throws:

`InvalidContentException` - Throw exception if the document's content do not match the grammer specified by DTD (The validation mode should be set to TRUE)

See Also:

`oracle.xml.classgen.ClassGenerator`

CGNode

Syntax

```
public abstract class CGNode extends java.lang.Object
```

```
java.lang.Object
  |
  +--oracle.xml.classgen.CGNode
```

Direct Known Subclasses:

CGDocument

Description

Serves as the base class for nodes generated by the DTD compiler

Fields

isValidating

```
protected boolean isValidating
```

Boolean to indicate the validating mode

Constructors

CGNode()

CGNode(String)

```
protected  CGNode(java.lang.String elementName)
```

Constructor for the Elements of the DOM Tree

Parameters:

`elementName` - Name of the element

Methods

addCDATASection(String)

```
protected void addCDATASection(java.lang.String theData)
```
Adds CDATA Section to the Element

Parameters:

`theData` - Text to be added as CDATA Section to the element

Throws:

`InvalidContentException` - Thrown if `theData` has illegal characters (validation must be set to TRUE)

See Also:

`oracle.xml.classgen.ClassGenerator`

addData(String)

```
protected void addData(java.lang.String theData)
```
Adds PCDATA to the Element

Parameters:

`theData` - Text to be added to the element

Throws:

`InvalidContentException` - Thrown if `theData` has illegal characters (validation must be set to TRUE)

See Also:

`oracle.xml.classgen.ClassGenerator`

addNode(CGNode)

```
protected void addNode(CGNode theNode)
```
Adds a node as a child to the element

Parameters:

`theNode` - The node to be added as child

Throws:

`InvalidContentException` - Thrown if the Node cannot be added as child as per Content Model of the element (validation must be set to TRUE)

See Also:

`oracle.xml.classgen.ClassGenerator`

getCGDocument()

`protected CGDocument getCGDocument()`
Gets the base document (root Element)

Returns:

The base CGDocument

getDTDNode()

`protected abstract oracle.xml.parser.v2.DTD getDTDNode()`
Gets the static DTD from the base document

Returns:

DTD stored in base CGDocument

setAttribute(String, String)

`protected void setAttribute(java.lang.String attName, java.lang.String value)`
Sets the value of the Attribute

Parameters:

`attName` - Name of the attribute

`value` - Value of the attribute

setDocument(CGDocument)

`public void setDocument(CGDocument d)`
Sets the base document (root Element)

Parameters:

`d` - Base CGDocument

storeID(String, String)

```
protected void storeID(java.lang.String attName, java.lang.String id)
```
Store this value for an ID identifier, so that we can later verify IDREF values

Parameters:

`attName` - Name of the ID Attribute

`id` - Value of the ID

storeIDREF(String, String)

```
protected void storeIDREF(java.lang.String attName, java.lang.String idref)
```
Store this value for an IDREF identifier, so that we can later verify, if an corresponding ID was defined.

Parameters:

`attName` - Name of the IDREF Attribute

`idref` - Value of the IDREF

validateContent()

```
protected void validateContent()
```
Checks if the content of the element is valid as per the Content Model specified in DTD

Returns:

True if content is valid, else false

validEntity(String)

```
protected boolean validEntity(java.lang.String entity)
```
Checks if the ENTITY identifier is valid

Parameters:

`name` - value of the Entity Attribute

Returns:

True if Entity is valid, else false

validID(String)

```
protected boolean validID(java.lang.String name)
```
Checks if the ID identifier is valid

Parameters:

`name` - value of the ID Attribute

Returns:

True if ID is valid, else false

validNMTOKEN(String)

```
protected boolean validNMTOKEN(java.lang.String name)
```
Checks if the NMTOKEN identifier is valid

Parameters:

`name` - value of the Nmtoken Attribute

Returns:

True if Nmtoken is valid, else false

CGXSDElement

Syntax

```
public abstract class CGXSDElement
```

oracle.xml.classgen.CGXSDElement

Description

This class serves as the base class for the all the generated classes corresponding to the XML Schema generated by Schema Class Generator

Fields

type

```
protected java.lang.Object type
```

Constructors

CGXSDElement()

```
public  CGXSDElement()
```

Methods

addAttribute(String, String)

```
protected void addAttribute(java.lang.String attName, java.lang.String attValue)
```

Add the attribute of a given node to the hashtable.

Parameters:

`attName` - the attribute name

`attValue` - the attribute value

addElement(Object)

```
protected void addElement(java.lang.Object elem)
```

Add the elements of a given element node to the vector correspondig to the elements.

Parameters:

`elem` - the object which needs to be added

getAttributes()

```
public java.util.Hashtable getAttributes()
```

Return the attributes

Returns:

attributes the hashtable containing attribute name and value

getChildElements()

```
public java.util.Vector getChildElements()
```

Get the vector having all the local elements

Returns:

elemChild vector

getNodeValue()

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

Return the node value

getType()

```
public java.lang.Object getType()
```

Return the type

print(XMLOutputStream)

```
public void print(oracle.xml.parser.v2.XMLOutputStream out)
```

Print an element node

Parameters:

`out` - the stream where the output is printed

printAttributes(XMLOutputStream, String)

```
public void printAttributes(oracle.xml.parser.v2.XMLOutputStream out,
```

java.lang.String name)
Print an attribute node

Parameters:

out - the stream where the output is printed

name - the attribute name

setNodeValue(String)

protected void setNodeValue(java.lang.String value)
Set the node value of an element

Parameters:

value - the node value

DTDClassGenerator

Syntax

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

java.lang.Object
  |
  +--oracle.xml.classgen.DTDClassGenerator
```

Description

This class is used by the DTD compiler to generate classes

Constructors

DTDClassGenerator()

`public DTDClassGenerator()`
Default constructor for DTDClassGenerator.

Methods

generate(DTD, String)

`public void generate(oracle.xml.parser.v2.DTD dtd, java.lang.String doctype)`
Traverses the DTD with element `doctype` as root and generates Java classes

Parameters:

`DTD` - The DTD used to generate the classes

`doctype` - Name of the root Element

printDocumentMethods()

`protected void printDocumentMethods()`

setGenerateComments(boolean)

`public void setGenerateComments(boolean comments)`

Switch to determine whether to generate java doc comments Default - TRUE

Parameters:

`comments` - boolean flag

setJavaPackage(Vector)

`public void setJavaPackage(java.util.Vector packageName)`
Sets the package for the classes generated Default - No package

Parameters:

`packageName` - Name of the package

setOutputDirectory(String)

`public void setOutputDirectory(java.lang.String dir)`
Sets the output directory where the java source code for the DTD are generated. Default - current directory

Parameters:

`dir` - Output directory

setSerializationMode(boolean)

`public void setSerializationMode(boolean yes)`
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. Default - FALSE (DTD is saved a text file)

Parameters:

`yes` - boolean flag

setValidationMode(boolean)

`public void setValidationMode(boolean yes)`
Switch to determine whether the classes generated should validate the XML Document being constructed Default - TRUE

Parameters:

`yes` - boolean flag

InvalidContentException

Syntax

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

java.lang.Object
  |
  +--java.lang.Throwable
        |
        +--java.lang.Exception
              |
              +--oracle.xml.classgen.InvalidContentException
```

All Implemented Interfaces:

```
java.io.Serializable
```

Description

Definition of InvalidContentException thrown by dtdcompiler classes

Constructors

InvalidContentException()

```
public  InvalidContentException()
```

InvalidContentException(String)

```
public  InvalidContentException(java.lang.String s)
```

oracg

Syntax

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

java.lang.Object
  |
  +--oracle.xml.classgen.oracg
```

Description

The oracg class provides a command-line interface to generate java classes corresponding to the DTD or XML Schema java oracle.xml.classgen.oracg options are: -h Prints the help message text -d The input file is a DTD file or DTD based XML file -s The input file is a Schema file or Schema based XML file -o The directory name where java source is generated -p The package name(s) of the generated java classes. -c Generate comments for the generated java source code.

Constructors

oracg()

```
public  oracg()
```

Default constructor for oracg

Methods

main(String[])

```
public static void main(java.lang.String[] args)
```

The main method

SchemaClassGenerator

Syntax

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

java.lang.Object
  |
  +--oracle.xml.classgen.SchemaClassGenerator
```

Description

This class generates the classes corresponding to an XML Schema.

Constructors

SchemaClassGenerator()

`public SchemaClassGenerator()`
Default empty constructor for Schema Class Generator

SchemaClassGenerator(String)

`public SchemaClassGenerator(java.lang.String fileName)`
The constructor for Schema Class Generator

Parameters:

`fileName` - the input XML Schema

Methods

generate(XMLSchema)

`public void generate(oracle.xml.parser.schema.XMLSchema schema)`
Generate the Schema classes corresponding to top level elements, simpleType elements and complexType elements by calling createSchemaClass on each of these nodes.

Parameters:

`XML` - Schema object

Throws:

setGenerateComments(boolean)

`public void setGenerateComments(boolean comments)`
Switch to determine whether to generate java doc comments The default setting is true

Parameters:

`comments` - boolean flag

setJavaPackage(XMLSchema, Vector)

`public void setJavaPackage(oracle.xml.parser.schema.XMLSchema schema, java.util.Vector pkgName)`
Sets the Java package names corresponding to Namespaces. The Namespaces defined in the schema are queried. The Number of namespaces defined in the schema should match the number of package names given by the user through command line else an error is thrown. For each namespace, a user-defined package name is assigned.

Parameters:

`schema` - XMLSchema

`pkgName` - A vector containing user defined pacake names given through command line.

setOutputDirectory(String)

`public void setOutputDirectory(java.lang.String dir)`
Sets the output directory where the java source code for the Schema class are generated. Default - current directory

Parameters:

`dir` - Output directory

10

Transviewer Beans

Package oracle.xml.async

Class Summary	
Interfaces	
DOMBuilderErrorListener	This interface must be implemented in order to receive notifications when error is found during parsing.
DOMBuilderListener	This interface must be implemented in order to receive notifications about events during the asyncronous parsing.
XSLTransformerErrorListener	This interface must be implemented in order to receive notifications about error events during the asynchronous transformation.
XSLTransformerListener	

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 DOMBuilderListener 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.InputSouce` 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.InputSouce 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 enitites and DTDs. This method should to 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 asyncronous 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:

<code>i</code> - - 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 avialable 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 asyncronously, calling this method immediately after processXSL will result in holding the control until the result is avialable.

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

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 resulted HTML in scrollable swing panel. This bean has tree buffers: XML, XSL and result buffer. The bean API allow 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 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

getXslCLOBTableName()

```
public java.lang.String getXslCLOBTableName()
```
Get XSL CLOB table name

Returns:

XSL CLOB table name

getXslFileName()

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

Transfroms 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()
```

Transfroms the content of the XML buffer by applying the stylesheet from the XSL buffer.

Package oracle.xml.dbviewer

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.srcviewer

XMLSourceView

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. The default color / 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()

Package oracle.xml.srcviewer

XMLSourceViewBeanInfo

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

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 that is a CLOB object. This 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, list a CLOB table content and also add, replace or delete text documents in this CLOB tables.

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 successfull

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 successfull

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 successfull

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 successfull

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 successfull

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 successfull

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 8i 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

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 the following XML DOM nodes: `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 preffered size.

Overrides:

javax.swing.JComponent.getPreferredSize() in class javax.swing.JComponent

Returns:

The Dimension object containing the XMLTreeView prefered 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()

```
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

11

Class AppCtxManager

AppCtxManager

Syntax

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

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

Description

AppCtxManager - manages which user classes are allowed to administer the application context

Member Summary	
Methods	
clearContext(AppCtxPermit, String, String, String)	checks the AppCtxPermit Object and lets the user do a clear Context
createAppCtxPermit()	returns the AppCtxPermit Object
setContext(AppCtxPermit, String, String, String, String, String)	checks the AppCtxPermit Object and lets the user do a set Contex

Methods

clearContext(AppCtxPermit, String, String, String)

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

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

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

Returns

None

createAppCtxPermit()

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. The AppCtxPermit Object becomes the TRUST point for the administration the HR Context.

`public static AppCtxPermit createAppCtxPermit()`
This method returns the AppCtxPermit Object

Parameters:

`None`

Returns

AppCtxPermitObject

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

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)

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

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

Index

Symbols

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

X