| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/JdbcRowSet.html) | [**Tree**](http://docs.google.com/package-tree.html) | [**Deprecated**](http://docs.google.com/deprecated-list.html) | [**Index**](http://docs.google.com/index-files/index-1.html) | [**Help**](http://docs.google.com/help-doc.html) | | --- | --- | --- | --- | --- | --- | --- | --- | | | ***Java™ Platform***  ***Standard Ed. 6*** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [**PREV CLASS**](http://docs.google.com/javax/sql/rowset/FilteredRowSet.html)   [**NEXT CLASS**](http://docs.google.com/javax/sql/rowset/Joinable.html) | [**FRAMES**](http://docs.google.com/index.html?javax/sql/rowset/JdbcRowSet.html)    [**NO FRAMES**](http://docs.google.com/JdbcRowSet.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | CONSTR | [METHOD](#tyjcwt) | DETAIL: FIELD | CONSTR | [METHOD](#17dp8vu) |

## **javax.sql.rowset**

Interface JdbcRowSet

**All Superinterfaces:** [Joinable](http://docs.google.com/javax/sql/rowset/Joinable.html), [ResultSet](http://docs.google.com/java/sql/ResultSet.html), [RowSet](http://docs.google.com/javax/sql/RowSet.html), [Wrapper](http://docs.google.com/java/sql/Wrapper.html)

public interface **JdbcRowSet**extends [RowSet](http://docs.google.com/javax/sql/RowSet.html), [Joinable](http://docs.google.com/javax/sql/rowset/Joinable.html)

The standard interface that all standard implementations of JdbcRowSet must implement.

### 1.0 Overview

A wrapper around a ResultSet object that makes it possible to use the result set as a JavaBeansTM component. Thus, a JdbcRowSet object can be one of the Beans that a tool makes available for composing an application. Because a JdbcRowSet is a connected rowset, that is, it continually maintains its connection to a database using a JDBC technology-enabled driver, it also effectively makes the driver a JavaBeans component.

Because it is always connected to its database, an instance of JdbcRowSet can simply take calls invoked on it and in turn call them on its ResultSet object. As a consequence, a result set can, for example, be a component in a Swing application.

Another advantage of a JdbcRowSet object is that it can be used to make a ResultSet object scrollable and updatable. All RowSet objects are by default scrollable and updatable. If the driver and database being used do not support scrolling and/or updating of result sets, an application can populate a JdbcRowSet object with the data of a ResultSet object and then operate on the JdbcRowSet object as if it were the ResultSet object.

### 2.0 Creating a JdbcRowSet Object

The reference implementation of the JdbcRowSet interface, JdbcRowSetImpl, provides an implementation of the default constructor. A new instance is initialized with default values, which can be set with new values as needed. A new instance is not really functional until its execute method is called. In general, this method does the following:

* establishes a connection with a database
* creates a PreparedStatement object and sets any of its placeholder parameters
* executes the statement to create a ResultSet object

If the execute method is successful, it will set the appropriate private JdbcRowSet fields with the following:

* a Connection object -- the connection between the rowset and the database
* a PreparedStatement object -- the query that produces the result set
* a ResultSet object -- the result set that the rowset's command produced and that is being made, in effect, a JavaBeans component

If these fields have not been set, meaning that the execute method has not executed successfully, no methods other than execute and close may be called on the rowset. All other public methods will throw an exception.

Before calling the execute method, however, the command and properties needed for establishing a connection must be set. The following code fragment creates a JdbcRowSetImpl object, sets the command and connection properties, sets the placeholder parameter, and then invokes the method execute.

JdbcRowSetImpl jrs = new JdbcRowSetImpl();  
 jrs.setCommand("SELECT \* FROM TITLES WHERE TYPE = ?");  
 jrs.setURL("jdbc:myDriver:myAttribute");  
 jrs.setUsername("cervantes");  
 jrs.setPassword("sancho");  
 jrs.setString(1, "BIOGRAPHY");  
 jrs.execute();

The variable jrs now represents an instance of JdbcRowSetImpl that is a thin wrapper around the ResultSet object containing all the rows in the table TITLES where the type of book is biography. At this point, operations called on jrs will affect the rows in the result set, which is effectively a JavaBeans component.

The implementation of the RowSet method execute in the JdbcRowSet reference implementation differs from that in the CachedRowSetTM reference implementation to account for the different requirements of connected and disconnected RowSet objects.

| **Field Summary** | |
| --- | --- |

| **Fields inherited from interface java.sql.**[**ResultSet**](http://docs.google.com/java/sql/ResultSet.html) |
| --- |
| [CLOSE\_CURSORS\_AT\_COMMIT](http://docs.google.com/java/sql/ResultSet.html#CLOSE_CURSORS_AT_COMMIT), [CONCUR\_READ\_ONLY](http://docs.google.com/java/sql/ResultSet.html#CONCUR_READ_ONLY), [CONCUR\_UPDATABLE](http://docs.google.com/java/sql/ResultSet.html#CONCUR_UPDATABLE), [FETCH\_FORWARD](http://docs.google.com/java/sql/ResultSet.html#FETCH_FORWARD), [FETCH\_REVERSE](http://docs.google.com/java/sql/ResultSet.html#FETCH_REVERSE), [FETCH\_UNKNOWN](http://docs.google.com/java/sql/ResultSet.html#FETCH_UNKNOWN), [HOLD\_CURSORS\_OVER\_COMMIT](http://docs.google.com/java/sql/ResultSet.html#HOLD_CURSORS_OVER_COMMIT), [TYPE\_FORWARD\_ONLY](http://docs.google.com/java/sql/ResultSet.html#TYPE_FORWARD_ONLY), [TYPE\_SCROLL\_INSENSITIVE](http://docs.google.com/java/sql/ResultSet.html#TYPE_SCROLL_INSENSITIVE), [TYPE\_SCROLL\_SENSITIVE](http://docs.google.com/java/sql/ResultSet.html#TYPE_SCROLL_SENSITIVE) |

| **Method Summary** | |
| --- | --- |
| void | [**commit**](http://docs.google.com/javax/sql/rowset/JdbcRowSet.html#commit())()            Each JdbcRowSet contains a Connection object from the ResultSet or JDBC properties passed to it's constructors. |
| boolean | [**getAutoCommit**](http://docs.google.com/javax/sql/rowset/JdbcRowSet.html#getAutoCommit())()            Each JdbcRowSet contains a Connection object from the original ResultSet or JDBC properties passed to it. |
| [RowSetWarning](http://docs.google.com/javax/sql/rowset/RowSetWarning.html) | [**getRowSetWarnings**](http://docs.google.com/javax/sql/rowset/JdbcRowSet.html#getRowSetWarnings())()            Retrieves the first warning reported by calls on this JdbcRowSet object. |
| boolean | [**getShowDeleted**](http://docs.google.com/javax/sql/rowset/JdbcRowSet.html#getShowDeleted())()            Retrieves a boolean indicating whether rows marked for deletion appear in the set of current rows. |
| void | [**rollback**](http://docs.google.com/javax/sql/rowset/JdbcRowSet.html#rollback())()            Each JdbcRowSet contains a Connection object from the original ResultSet or JDBC properties passed to it. |
| void | [**rollback**](http://docs.google.com/javax/sql/rowset/JdbcRowSet.html#rollback(java.sql.Savepoint))([Savepoint](http://docs.google.com/java/sql/Savepoint.html) s)            Each JdbcRowSet contains a Connection object from the original ResultSet or JDBC properties passed to it. |
| void | [**setAutoCommit**](http://docs.google.com/javax/sql/rowset/JdbcRowSet.html#setAutoCommit(boolean))(boolean autoCommit)            Each JdbcRowSet contains a Connection object from the original ResultSet or JDBC properties passed to it. |
| void | [**setShowDeleted**](http://docs.google.com/javax/sql/rowset/JdbcRowSet.html#setShowDeleted(boolean))(boolean b)            Sets the property showDeleted to the given boolean value. |

| **Methods inherited from interface javax.sql.**[**RowSet**](http://docs.google.com/javax/sql/RowSet.html) |
| --- |
| [addRowSetListener](http://docs.google.com/javax/sql/RowSet.html#addRowSetListener(javax.sql.RowSetListener)), [clearParameters](http://docs.google.com/javax/sql/RowSet.html#clearParameters()), [execute](http://docs.google.com/javax/sql/RowSet.html#execute()), [getCommand](http://docs.google.com/javax/sql/RowSet.html#getCommand()), [getDataSourceName](http://docs.google.com/javax/sql/RowSet.html#getDataSourceName()), [getEscapeProcessing](http://docs.google.com/javax/sql/RowSet.html#getEscapeProcessing()), [getMaxFieldSize](http://docs.google.com/javax/sql/RowSet.html#getMaxFieldSize()), [getMaxRows](http://docs.google.com/javax/sql/RowSet.html#getMaxRows()), [getPassword](http://docs.google.com/javax/sql/RowSet.html#getPassword()), [getQueryTimeout](http://docs.google.com/javax/sql/RowSet.html#getQueryTimeout()), [getTransactionIsolation](http://docs.google.com/javax/sql/RowSet.html#getTransactionIsolation()), [getTypeMap](http://docs.google.com/javax/sql/RowSet.html#getTypeMap()), [getUrl](http://docs.google.com/javax/sql/RowSet.html#getUrl()), [getUsername](http://docs.google.com/javax/sql/RowSet.html#getUsername()), [isReadOnly](http://docs.google.com/javax/sql/RowSet.html#isReadOnly()), [removeRowSetListener](http://docs.google.com/javax/sql/RowSet.html#removeRowSetListener(javax.sql.RowSetListener)), [setArray](http://docs.google.com/javax/sql/RowSet.html#setArray(int,%20java.sql.Array)), [setAsciiStream](http://docs.google.com/javax/sql/RowSet.html#setAsciiStream(int,%20java.io.InputStream)), [setAsciiStream](http://docs.google.com/javax/sql/RowSet.html#setAsciiStream(int,%20java.io.InputStream,%20int)), [setAsciiStream](http://docs.google.com/javax/sql/RowSet.html#setAsciiStream(java.lang.String,%20java.io.InputStream)), [setAsciiStream](http://docs.google.com/javax/sql/RowSet.html#setAsciiStream(java.lang.String,%20java.io.InputStream,%20int)), [setBigDecimal](http://docs.google.com/javax/sql/RowSet.html#setBigDecimal(int,%20java.math.BigDecimal)), [setBigDecimal](http://docs.google.com/javax/sql/RowSet.html#setBigDecimal(java.lang.String,%20java.math.BigDecimal)), [setBinaryStream](http://docs.google.com/javax/sql/RowSet.html#setBinaryStream(int,%20java.io.InputStream)), [setBinaryStream](http://docs.google.com/javax/sql/RowSet.html#setBinaryStream(int,%20java.io.InputStream,%20int)), [setBinaryStream](http://docs.google.com/javax/sql/RowSet.html#setBinaryStream(java.lang.String,%20java.io.InputStream)), [setBinaryStream](http://docs.google.com/javax/sql/RowSet.html#setBinaryStream(java.lang.String,%20java.io.InputStream,%20int)), [setBlob](http://docs.google.com/javax/sql/RowSet.html#setBlob(int,%20java.sql.Blob)), [setBlob](http://docs.google.com/javax/sql/RowSet.html#setBlob(int,%20java.io.InputStream)), [setBlob](http://docs.google.com/javax/sql/RowSet.html#setBlob(int,%20java.io.InputStream,%20long)), [setBlob](http://docs.google.com/javax/sql/RowSet.html#setBlob(java.lang.String,%20java.sql.Blob)), [setBlob](http://docs.google.com/javax/sql/RowSet.html#setBlob(java.lang.String,%20java.io.InputStream)), [setBlob](http://docs.google.com/javax/sql/RowSet.html#setBlob(java.lang.String,%20java.io.InputStream,%20long)), [setBoolean](http://docs.google.com/javax/sql/RowSet.html#setBoolean(int,%20boolean)), [setBoolean](http://docs.google.com/javax/sql/RowSet.html#setBoolean(java.lang.String,%20boolean)), [setByte](http://docs.google.com/javax/sql/RowSet.html#setByte(int,%20byte)), [setByte](http://docs.google.com/javax/sql/RowSet.html#setByte(java.lang.String,%20byte)), [setBytes](http://docs.google.com/javax/sql/RowSet.html#setBytes(int,%20byte%5B%5D)), [setBytes](http://docs.google.com/javax/sql/RowSet.html#setBytes(java.lang.String,%20byte%5B%5D)), [setCharacterStream](http://docs.google.com/javax/sql/RowSet.html#setCharacterStream(int,%20java.io.Reader)), [setCharacterStream](http://docs.google.com/javax/sql/RowSet.html#setCharacterStream(int,%20java.io.Reader,%20int)), [setCharacterStream](http://docs.google.com/javax/sql/RowSet.html#setCharacterStream(java.lang.String,%20java.io.Reader)), [setCharacterStream](http://docs.google.com/javax/sql/RowSet.html#setCharacterStream(java.lang.String,%20java.io.Reader,%20int)), [setClob](http://docs.google.com/javax/sql/RowSet.html#setClob(int,%20java.sql.Clob)), [setClob](http://docs.google.com/javax/sql/RowSet.html#setClob(int,%20java.io.Reader)), [setClob](http://docs.google.com/javax/sql/RowSet.html#setClob(int,%20java.io.Reader,%20long)), [setClob](http://docs.google.com/javax/sql/RowSet.html#setClob(java.lang.String,%20java.sql.Clob)), [setClob](http://docs.google.com/javax/sql/RowSet.html#setClob(java.lang.String,%20java.io.Reader)), [setClob](http://docs.google.com/javax/sql/RowSet.html#setClob(java.lang.String,%20java.io.Reader,%20long)), [setCommand](http://docs.google.com/javax/sql/RowSet.html#setCommand(java.lang.String)), [setConcurrency](http://docs.google.com/javax/sql/RowSet.html#setConcurrency(int)), [setDataSourceName](http://docs.google.com/javax/sql/RowSet.html#setDataSourceName(java.lang.String)), [setDate](http://docs.google.com/javax/sql/RowSet.html#setDate(int,%20java.sql.Date)), [setDate](http://docs.google.com/javax/sql/RowSet.html#setDate(int,%20java.sql.Date,%20java.util.Calendar)), [setDate](http://docs.google.com/javax/sql/RowSet.html#setDate(java.lang.String,%20java.sql.Date)), [setDate](http://docs.google.com/javax/sql/RowSet.html#setDate(java.lang.String,%20java.sql.Date,%20java.util.Calendar)), [setDouble](http://docs.google.com/javax/sql/RowSet.html#setDouble(int,%20double)), [setDouble](http://docs.google.com/javax/sql/RowSet.html#setDouble(java.lang.String,%20double)), [setEscapeProcessing](http://docs.google.com/javax/sql/RowSet.html#setEscapeProcessing(boolean)), [setFloat](http://docs.google.com/javax/sql/RowSet.html#setFloat(int,%20float)), [setFloat](http://docs.google.com/javax/sql/RowSet.html#setFloat(java.lang.String,%20float)), [setInt](http://docs.google.com/javax/sql/RowSet.html#setInt(int,%20int)), [setInt](http://docs.google.com/javax/sql/RowSet.html#setInt(java.lang.String,%20int)), [setLong](http://docs.google.com/javax/sql/RowSet.html#setLong(int,%20long)), [setLong](http://docs.google.com/javax/sql/RowSet.html#setLong(java.lang.String,%20long)), [setMaxFieldSize](http://docs.google.com/javax/sql/RowSet.html#setMaxFieldSize(int)), [setMaxRows](http://docs.google.com/javax/sql/RowSet.html#setMaxRows(int)), [setNCharacterStream](http://docs.google.com/javax/sql/RowSet.html#setNCharacterStream(int,%20java.io.Reader)), [setNCharacterStream](http://docs.google.com/javax/sql/RowSet.html#setNCharacterStream(int,%20java.io.Reader,%20long)), [setNCharacterStream](http://docs.google.com/javax/sql/RowSet.html#setNCharacterStream(java.lang.String,%20java.io.Reader)), [setNCharacterStream](http://docs.google.com/javax/sql/RowSet.html#setNCharacterStream(java.lang.String,%20java.io.Reader,%20long)), [setNClob](http://docs.google.com/javax/sql/RowSet.html#setNClob(int,%20java.sql.NClob)), [setNClob](http://docs.google.com/javax/sql/RowSet.html#setNClob(int,%20java.io.Reader)), [setNClob](http://docs.google.com/javax/sql/RowSet.html#setNClob(int,%20java.io.Reader,%20long)), [setNClob](http://docs.google.com/javax/sql/RowSet.html#setNClob(java.lang.String,%20java.sql.NClob)), [setNClob](http://docs.google.com/javax/sql/RowSet.html#setNClob(java.lang.String,%20java.io.Reader)), [setNClob](http://docs.google.com/javax/sql/RowSet.html#setNClob(java.lang.String,%20java.io.Reader,%20long)), [setNString](http://docs.google.com/javax/sql/RowSet.html#setNString(int,%20java.lang.String)), [setNString](http://docs.google.com/javax/sql/RowSet.html#setNString(java.lang.String,%20java.lang.String)), [setNull](http://docs.google.com/javax/sql/RowSet.html#setNull(int,%20int)), [setNull](http://docs.google.com/javax/sql/RowSet.html#setNull(int,%20int,%20java.lang.String)), [setNull](http://docs.google.com/javax/sql/RowSet.html#setNull(java.lang.String,%20int)), [setNull](http://docs.google.com/javax/sql/RowSet.html#setNull(java.lang.String,%20int,%20java.lang.String)), [setObject](http://docs.google.com/javax/sql/RowSet.html#setObject(int,%20java.lang.Object)), [setObject](http://docs.google.com/javax/sql/RowSet.html#setObject(int,%20java.lang.Object,%20int)), [setObject](http://docs.google.com/javax/sql/RowSet.html#setObject(int,%20java.lang.Object,%20int,%20int)), [setObject](http://docs.google.com/javax/sql/RowSet.html#setObject(java.lang.String,%20java.lang.Object)), [setObject](http://docs.google.com/javax/sql/RowSet.html#setObject(java.lang.String,%20java.lang.Object,%20int)), [setObject](http://docs.google.com/javax/sql/RowSet.html#setObject(java.lang.String,%20java.lang.Object,%20int,%20int)), [setPassword](http://docs.google.com/javax/sql/RowSet.html#setPassword(java.lang.String)), [setQueryTimeout](http://docs.google.com/javax/sql/RowSet.html#setQueryTimeout(int)), [setReadOnly](http://docs.google.com/javax/sql/RowSet.html#setReadOnly(boolean)), [setRef](http://docs.google.com/javax/sql/RowSet.html#setRef(int,%20java.sql.Ref)), [setRowId](http://docs.google.com/javax/sql/RowSet.html#setRowId(int,%20java.sql.RowId)), [setRowId](http://docs.google.com/javax/sql/RowSet.html#setRowId(java.lang.String,%20java.sql.RowId)), [setShort](http://docs.google.com/javax/sql/RowSet.html#setShort(int,%20short)), [setShort](http://docs.google.com/javax/sql/RowSet.html#setShort(java.lang.String,%20short)), [setSQLXML](http://docs.google.com/javax/sql/RowSet.html#setSQLXML(int,%20java.sql.SQLXML)), [setSQLXML](http://docs.google.com/javax/sql/RowSet.html#setSQLXML(java.lang.String,%20java.sql.SQLXML)), [setString](http://docs.google.com/javax/sql/RowSet.html#setString(int,%20java.lang.String)), [setString](http://docs.google.com/javax/sql/RowSet.html#setString(java.lang.String,%20java.lang.String)), [setTime](http://docs.google.com/javax/sql/RowSet.html#setTime(int,%20java.sql.Time)), [setTime](http://docs.google.com/javax/sql/RowSet.html#setTime(int,%20java.sql.Time,%20java.util.Calendar)), [setTime](http://docs.google.com/javax/sql/RowSet.html#setTime(java.lang.String,%20java.sql.Time)), [setTime](http://docs.google.com/javax/sql/RowSet.html#setTime(java.lang.String,%20java.sql.Time,%20java.util.Calendar)), [setTimestamp](http://docs.google.com/javax/sql/RowSet.html#setTimestamp(int,%20java.sql.Timestamp)), [setTimestamp](http://docs.google.com/javax/sql/RowSet.html#setTimestamp(int,%20java.sql.Timestamp,%20java.util.Calendar)), [setTimestamp](http://docs.google.com/javax/sql/RowSet.html#setTimestamp(java.lang.String,%20java.sql.Timestamp)), [setTimestamp](http://docs.google.com/javax/sql/RowSet.html#setTimestamp(java.lang.String,%20java.sql.Timestamp,%20java.util.Calendar)), [setTransactionIsolation](http://docs.google.com/javax/sql/RowSet.html#setTransactionIsolation(int)), [setType](http://docs.google.com/javax/sql/RowSet.html#setType(int)), [setTypeMap](http://docs.google.com/javax/sql/RowSet.html#setTypeMap(java.util.Map)), [setURL](http://docs.google.com/javax/sql/RowSet.html#setURL(int,%20java.net.URL)), [setUrl](http://docs.google.com/javax/sql/RowSet.html#setUrl(java.lang.String)), [setUsername](http://docs.google.com/javax/sql/RowSet.html#setUsername(java.lang.String)) |

| **Methods inherited from interface java.sql.**[**ResultSet**](http://docs.google.com/java/sql/ResultSet.html) |
| --- |
| [absolute](http://docs.google.com/java/sql/ResultSet.html#absolute(int)), [afterLast](http://docs.google.com/java/sql/ResultSet.html#afterLast()), [beforeFirst](http://docs.google.com/java/sql/ResultSet.html#beforeFirst()), [cancelRowUpdates](http://docs.google.com/java/sql/ResultSet.html#cancelRowUpdates()), [clearWarnings](http://docs.google.com/java/sql/ResultSet.html#clearWarnings()), [close](http://docs.google.com/java/sql/ResultSet.html#close()), [deleteRow](http://docs.google.com/java/sql/ResultSet.html#deleteRow()), [findColumn](http://docs.google.com/java/sql/ResultSet.html#findColumn(java.lang.String)), [first](http://docs.google.com/java/sql/ResultSet.html#first()), [getArray](http://docs.google.com/java/sql/ResultSet.html#getArray(int)), [getArray](http://docs.google.com/java/sql/ResultSet.html#getArray(java.lang.String)), [getAsciiStream](http://docs.google.com/java/sql/ResultSet.html#getAsciiStream(int)), [getAsciiStream](http://docs.google.com/java/sql/ResultSet.html#getAsciiStream(java.lang.String)), [getBigDecimal](http://docs.google.com/java/sql/ResultSet.html#getBigDecimal(int)), [getBigDecimal](http://docs.google.com/java/sql/ResultSet.html#getBigDecimal(int,%20int)), [getBigDecimal](http://docs.google.com/java/sql/ResultSet.html#getBigDecimal(java.lang.String)), [getBigDecimal](http://docs.google.com/java/sql/ResultSet.html#getBigDecimal(java.lang.String,%20int)), [getBinaryStream](http://docs.google.com/java/sql/ResultSet.html#getBinaryStream(int)), [getBinaryStream](http://docs.google.com/java/sql/ResultSet.html#getBinaryStream(java.lang.String)), [getBlob](http://docs.google.com/java/sql/ResultSet.html#getBlob(int)), [getBlob](http://docs.google.com/java/sql/ResultSet.html#getBlob(java.lang.String)), [getBoolean](http://docs.google.com/java/sql/ResultSet.html#getBoolean(int)), [getBoolean](http://docs.google.com/java/sql/ResultSet.html#getBoolean(java.lang.String)), [getByte](http://docs.google.com/java/sql/ResultSet.html#getByte(int)), [getByte](http://docs.google.com/java/sql/ResultSet.html#getByte(java.lang.String)), [getBytes](http://docs.google.com/java/sql/ResultSet.html#getBytes(int)), [getBytes](http://docs.google.com/java/sql/ResultSet.html#getBytes(java.lang.String)), [getCharacterStream](http://docs.google.com/java/sql/ResultSet.html#getCharacterStream(int)), [getCharacterStream](http://docs.google.com/java/sql/ResultSet.html#getCharacterStream(java.lang.String)), [getClob](http://docs.google.com/java/sql/ResultSet.html#getClob(int)), [getClob](http://docs.google.com/java/sql/ResultSet.html#getClob(java.lang.String)), [getConcurrency](http://docs.google.com/java/sql/ResultSet.html#getConcurrency()), [getCursorName](http://docs.google.com/java/sql/ResultSet.html#getCursorName()), [getDate](http://docs.google.com/java/sql/ResultSet.html#getDate(int)), [getDate](http://docs.google.com/java/sql/ResultSet.html#getDate(int,%20java.util.Calendar)), [getDate](http://docs.google.com/java/sql/ResultSet.html#getDate(java.lang.String)), [getDate](http://docs.google.com/java/sql/ResultSet.html#getDate(java.lang.String,%20java.util.Calendar)), [getDouble](http://docs.google.com/java/sql/ResultSet.html#getDouble(int)), [getDouble](http://docs.google.com/java/sql/ResultSet.html#getDouble(java.lang.String)), [getFetchDirection](http://docs.google.com/java/sql/ResultSet.html#getFetchDirection()), [getFetchSize](http://docs.google.com/java/sql/ResultSet.html#getFetchSize()), [getFloat](http://docs.google.com/java/sql/ResultSet.html#getFloat(int)), [getFloat](http://docs.google.com/java/sql/ResultSet.html#getFloat(java.lang.String)), [getHoldability](http://docs.google.com/java/sql/ResultSet.html#getHoldability()), [getInt](http://docs.google.com/java/sql/ResultSet.html#getInt(int)), [getInt](http://docs.google.com/java/sql/ResultSet.html#getInt(java.lang.String)), [getLong](http://docs.google.com/java/sql/ResultSet.html#getLong(int)), [getLong](http://docs.google.com/java/sql/ResultSet.html#getLong(java.lang.String)), [getMetaData](http://docs.google.com/java/sql/ResultSet.html#getMetaData()), [getNCharacterStream](http://docs.google.com/java/sql/ResultSet.html#getNCharacterStream(int)), [getNCharacterStream](http://docs.google.com/java/sql/ResultSet.html#getNCharacterStream(java.lang.String)), [getNClob](http://docs.google.com/java/sql/ResultSet.html#getNClob(int)), [getNClob](http://docs.google.com/java/sql/ResultSet.html#getNClob(java.lang.String)), [getNString](http://docs.google.com/java/sql/ResultSet.html#getNString(int)), [getNString](http://docs.google.com/java/sql/ResultSet.html#getNString(java.lang.String)), [getObject](http://docs.google.com/java/sql/ResultSet.html#getObject(int)), [getObject](http://docs.google.com/java/sql/ResultSet.html#getObject(int,%20java.util.Map)), [getObject](http://docs.google.com/java/sql/ResultSet.html#getObject(java.lang.String)), [getObject](http://docs.google.com/java/sql/ResultSet.html#getObject(java.lang.String,%20java.util.Map)), [getRef](http://docs.google.com/java/sql/ResultSet.html#getRef(int)), [getRef](http://docs.google.com/java/sql/ResultSet.html#getRef(java.lang.String)), [getRow](http://docs.google.com/java/sql/ResultSet.html#getRow()), [getRowId](http://docs.google.com/java/sql/ResultSet.html#getRowId(int)), [getRowId](http://docs.google.com/java/sql/ResultSet.html#getRowId(java.lang.String)), [getShort](http://docs.google.com/java/sql/ResultSet.html#getShort(int)), [getShort](http://docs.google.com/java/sql/ResultSet.html#getShort(java.lang.String)), [getSQLXML](http://docs.google.com/java/sql/ResultSet.html#getSQLXML(int)), [getSQLXML](http://docs.google.com/java/sql/ResultSet.html#getSQLXML(java.lang.String)), [getStatement](http://docs.google.com/java/sql/ResultSet.html#getStatement()), [getString](http://docs.google.com/java/sql/ResultSet.html#getString(int)), [getString](http://docs.google.com/java/sql/ResultSet.html#getString(java.lang.String)), [getTime](http://docs.google.com/java/sql/ResultSet.html#getTime(int)), [getTime](http://docs.google.com/java/sql/ResultSet.html#getTime(int,%20java.util.Calendar)), [getTime](http://docs.google.com/java/sql/ResultSet.html#getTime(java.lang.String)), [getTime](http://docs.google.com/java/sql/ResultSet.html#getTime(java.lang.String,%20java.util.Calendar)), [getTimestamp](http://docs.google.com/java/sql/ResultSet.html#getTimestamp(int)), [getTimestamp](http://docs.google.com/java/sql/ResultSet.html#getTimestamp(int,%20java.util.Calendar)), [getTimestamp](http://docs.google.com/java/sql/ResultSet.html#getTimestamp(java.lang.String)), [getTimestamp](http://docs.google.com/java/sql/ResultSet.html#getTimestamp(java.lang.String,%20java.util.Calendar)), [getType](http://docs.google.com/java/sql/ResultSet.html#getType()), [getUnicodeStream](http://docs.google.com/java/sql/ResultSet.html#getUnicodeStream(int)), [getUnicodeStream](http://docs.google.com/java/sql/ResultSet.html#getUnicodeStream(java.lang.String)), [getURL](http://docs.google.com/java/sql/ResultSet.html#getURL(int)), [getURL](http://docs.google.com/java/sql/ResultSet.html#getURL(java.lang.String)), [getWarnings](http://docs.google.com/java/sql/ResultSet.html#getWarnings()), [insertRow](http://docs.google.com/java/sql/ResultSet.html#insertRow()), [isAfterLast](http://docs.google.com/java/sql/ResultSet.html#isAfterLast()), [isBeforeFirst](http://docs.google.com/java/sql/ResultSet.html#isBeforeFirst()), [isClosed](http://docs.google.com/java/sql/ResultSet.html#isClosed()), [isFirst](http://docs.google.com/java/sql/ResultSet.html#isFirst()), [isLast](http://docs.google.com/java/sql/ResultSet.html#isLast()), [last](http://docs.google.com/java/sql/ResultSet.html#last()), [moveToCurrentRow](http://docs.google.com/java/sql/ResultSet.html#moveToCurrentRow()), [moveToInsertRow](http://docs.google.com/java/sql/ResultSet.html#moveToInsertRow()), [next](http://docs.google.com/java/sql/ResultSet.html#next()), [previous](http://docs.google.com/java/sql/ResultSet.html#previous()), [refreshRow](http://docs.google.com/java/sql/ResultSet.html#refreshRow()), [relative](http://docs.google.com/java/sql/ResultSet.html#relative(int)), [rowDeleted](http://docs.google.com/java/sql/ResultSet.html#rowDeleted()), [rowInserted](http://docs.google.com/java/sql/ResultSet.html#rowInserted()), [rowUpdated](http://docs.google.com/java/sql/ResultSet.html#rowUpdated()), [setFetchDirection](http://docs.google.com/java/sql/ResultSet.html#setFetchDirection(int)), [setFetchSize](http://docs.google.com/java/sql/ResultSet.html#setFetchSize(int)), [updateArray](http://docs.google.com/java/sql/ResultSet.html#updateArray(int,%20java.sql.Array)), [updateArray](http://docs.google.com/java/sql/ResultSet.html#updateArray(java.lang.String,%20java.sql.Array)), [updateAsciiStream](http://docs.google.com/java/sql/ResultSet.html#updateAsciiStream(int,%20java.io.InputStream)), [updateAsciiStream](http://docs.google.com/java/sql/ResultSet.html#updateAsciiStream(int,%20java.io.InputStream,%20int)), [updateAsciiStream](http://docs.google.com/java/sql/ResultSet.html#updateAsciiStream(int,%20java.io.InputStream,%20long)), [updateAsciiStream](http://docs.google.com/java/sql/ResultSet.html#updateAsciiStream(java.lang.String,%20java.io.InputStream)), [updateAsciiStream](http://docs.google.com/java/sql/ResultSet.html#updateAsciiStream(java.lang.String,%20java.io.InputStream,%20int)), [updateAsciiStream](http://docs.google.com/java/sql/ResultSet.html#updateAsciiStream(java.lang.String,%20java.io.InputStream,%20long)), [updateBigDecimal](http://docs.google.com/java/sql/ResultSet.html#updateBigDecimal(int,%20java.math.BigDecimal)), [updateBigDecimal](http://docs.google.com/java/sql/ResultSet.html#updateBigDecimal(java.lang.String,%20java.math.BigDecimal)), [updateBinaryStream](http://docs.google.com/java/sql/ResultSet.html#updateBinaryStream(int,%20java.io.InputStream)), [updateBinaryStream](http://docs.google.com/java/sql/ResultSet.html#updateBinaryStream(int,%20java.io.InputStream,%20int)), [updateBinaryStream](http://docs.google.com/java/sql/ResultSet.html#updateBinaryStream(int,%20java.io.InputStream,%20long)), [updateBinaryStream](http://docs.google.com/java/sql/ResultSet.html#updateBinaryStream(java.lang.String,%20java.io.InputStream)), [updateBinaryStream](http://docs.google.com/java/sql/ResultSet.html#updateBinaryStream(java.lang.String,%20java.io.InputStream,%20int)), [updateBinaryStream](http://docs.google.com/java/sql/ResultSet.html#updateBinaryStream(java.lang.String,%20java.io.InputStream,%20long)), [updateBlob](http://docs.google.com/java/sql/ResultSet.html#updateBlob(int,%20java.sql.Blob)), [updateBlob](http://docs.google.com/java/sql/ResultSet.html#updateBlob(int,%20java.io.InputStream)), [updateBlob](http://docs.google.com/java/sql/ResultSet.html#updateBlob(int,%20java.io.InputStream,%20long)), [updateBlob](http://docs.google.com/java/sql/ResultSet.html#updateBlob(java.lang.String,%20java.sql.Blob)), [updateBlob](http://docs.google.com/java/sql/ResultSet.html#updateBlob(java.lang.String,%20java.io.InputStream)), [updateBlob](http://docs.google.com/java/sql/ResultSet.html#updateBlob(java.lang.String,%20java.io.InputStream,%20long)), [updateBoolean](http://docs.google.com/java/sql/ResultSet.html#updateBoolean(int,%20boolean)), [updateBoolean](http://docs.google.com/java/sql/ResultSet.html#updateBoolean(java.lang.String,%20boolean)), [updateByte](http://docs.google.com/java/sql/ResultSet.html#updateByte(int,%20byte)), [updateByte](http://docs.google.com/java/sql/ResultSet.html#updateByte(java.lang.String,%20byte)), [updateBytes](http://docs.google.com/java/sql/ResultSet.html#updateBytes(int,%20byte%5B%5D)), [updateBytes](http://docs.google.com/java/sql/ResultSet.html#updateBytes(java.lang.String,%20byte%5B%5D)), [updateCharacterStream](http://docs.google.com/java/sql/ResultSet.html#updateCharacterStream(int,%20java.io.Reader)), [updateCharacterStream](http://docs.google.com/java/sql/ResultSet.html#updateCharacterStream(int,%20java.io.Reader,%20int)), [updateCharacterStream](http://docs.google.com/java/sql/ResultSet.html#updateCharacterStream(int,%20java.io.Reader,%20long)), [updateCharacterStream](http://docs.google.com/java/sql/ResultSet.html#updateCharacterStream(java.lang.String,%20java.io.Reader)), [updateCharacterStream](http://docs.google.com/java/sql/ResultSet.html#updateCharacterStream(java.lang.String,%20java.io.Reader,%20int)), [updateCharacterStream](http://docs.google.com/java/sql/ResultSet.html#updateCharacterStream(java.lang.String,%20java.io.Reader,%20long)), [updateClob](http://docs.google.com/java/sql/ResultSet.html#updateClob(int,%20java.sql.Clob)), [updateClob](http://docs.google.com/java/sql/ResultSet.html#updateClob(int,%20java.io.Reader)), [updateClob](http://docs.google.com/java/sql/ResultSet.html#updateClob(int,%20java.io.Reader,%20long)), [updateClob](http://docs.google.com/java/sql/ResultSet.html#updateClob(java.lang.String,%20java.sql.Clob)), [updateClob](http://docs.google.com/java/sql/ResultSet.html#updateClob(java.lang.String,%20java.io.Reader)), [updateClob](http://docs.google.com/java/sql/ResultSet.html#updateClob(java.lang.String,%20java.io.Reader,%20long)), [updateDate](http://docs.google.com/java/sql/ResultSet.html#updateDate(int,%20java.sql.Date)), [updateDate](http://docs.google.com/java/sql/ResultSet.html#updateDate(java.lang.String,%20java.sql.Date)), [updateDouble](http://docs.google.com/java/sql/ResultSet.html#updateDouble(int,%20double)), [updateDouble](http://docs.google.com/java/sql/ResultSet.html#updateDouble(java.lang.String,%20double)), [updateFloat](http://docs.google.com/java/sql/ResultSet.html#updateFloat(int,%20float)), [updateFloat](http://docs.google.com/java/sql/ResultSet.html#updateFloat(java.lang.String,%20float)), [updateInt](http://docs.google.com/java/sql/ResultSet.html#updateInt(int,%20int)), [updateInt](http://docs.google.com/java/sql/ResultSet.html#updateInt(java.lang.String,%20int)), [updateLong](http://docs.google.com/java/sql/ResultSet.html#updateLong(int,%20long)), [updateLong](http://docs.google.com/java/sql/ResultSet.html#updateLong(java.lang.String,%20long)), [updateNCharacterStream](http://docs.google.com/java/sql/ResultSet.html#updateNCharacterStream(int,%20java.io.Reader)), [updateNCharacterStream](http://docs.google.com/java/sql/ResultSet.html#updateNCharacterStream(int,%20java.io.Reader,%20long)), [updateNCharacterStream](http://docs.google.com/java/sql/ResultSet.html#updateNCharacterStream(java.lang.String,%20java.io.Reader)), [updateNCharacterStream](http://docs.google.com/java/sql/ResultSet.html#updateNCharacterStream(java.lang.String,%20java.io.Reader,%20long)), [updateNClob](http://docs.google.com/java/sql/ResultSet.html#updateNClob(int,%20java.sql.NClob)), [updateNClob](http://docs.google.com/java/sql/ResultSet.html#updateNClob(int,%20java.io.Reader)), [updateNClob](http://docs.google.com/java/sql/ResultSet.html#updateNClob(int,%20java.io.Reader,%20long)), [updateNClob](http://docs.google.com/java/sql/ResultSet.html#updateNClob(java.lang.String,%20java.sql.NClob)), [updateNClob](http://docs.google.com/java/sql/ResultSet.html#updateNClob(java.lang.String,%20java.io.Reader)), [updateNClob](http://docs.google.com/java/sql/ResultSet.html#updateNClob(java.lang.String,%20java.io.Reader,%20long)), [updateNString](http://docs.google.com/java/sql/ResultSet.html#updateNString(int,%20java.lang.String)), [updateNString](http://docs.google.com/java/sql/ResultSet.html#updateNString(java.lang.String,%20java.lang.String)), [updateNull](http://docs.google.com/java/sql/ResultSet.html#updateNull(int)), [updateNull](http://docs.google.com/java/sql/ResultSet.html#updateNull(java.lang.String)), [updateObject](http://docs.google.com/java/sql/ResultSet.html#updateObject(int,%20java.lang.Object)), [updateObject](http://docs.google.com/java/sql/ResultSet.html#updateObject(int,%20java.lang.Object,%20int)), [updateObject](http://docs.google.com/java/sql/ResultSet.html#updateObject(java.lang.String,%20java.lang.Object)), [updateObject](http://docs.google.com/java/sql/ResultSet.html#updateObject(java.lang.String,%20java.lang.Object,%20int)), [updateRef](http://docs.google.com/java/sql/ResultSet.html#updateRef(int,%20java.sql.Ref)), [updateRef](http://docs.google.com/java/sql/ResultSet.html#updateRef(java.lang.String,%20java.sql.Ref)), [updateRow](http://docs.google.com/java/sql/ResultSet.html#updateRow()), [updateRowId](http://docs.google.com/java/sql/ResultSet.html#updateRowId(int,%20java.sql.RowId)), [updateRowId](http://docs.google.com/java/sql/ResultSet.html#updateRowId(java.lang.String,%20java.sql.RowId)), [updateShort](http://docs.google.com/java/sql/ResultSet.html#updateShort(int,%20short)), [updateShort](http://docs.google.com/java/sql/ResultSet.html#updateShort(java.lang.String,%20short)), [updateSQLXML](http://docs.google.com/java/sql/ResultSet.html#updateSQLXML(int,%20java.sql.SQLXML)), [updateSQLXML](http://docs.google.com/java/sql/ResultSet.html#updateSQLXML(java.lang.String,%20java.sql.SQLXML)), [updateString](http://docs.google.com/java/sql/ResultSet.html#updateString(int,%20java.lang.String)), [updateString](http://docs.google.com/java/sql/ResultSet.html#updateString(java.lang.String,%20java.lang.String)), [updateTime](http://docs.google.com/java/sql/ResultSet.html#updateTime(int,%20java.sql.Time)), [updateTime](http://docs.google.com/java/sql/ResultSet.html#updateTime(java.lang.String,%20java.sql.Time)), [updateTimestamp](http://docs.google.com/java/sql/ResultSet.html#updateTimestamp(int,%20java.sql.Timestamp)), [updateTimestamp](http://docs.google.com/java/sql/ResultSet.html#updateTimestamp(java.lang.String,%20java.sql.Timestamp)), [wasNull](http://docs.google.com/java/sql/ResultSet.html#wasNull()) |

| **Methods inherited from interface java.sql.**[**Wrapper**](http://docs.google.com/java/sql/Wrapper.html) |
| --- |
| [isWrapperFor](http://docs.google.com/java/sql/Wrapper.html#isWrapperFor(java.lang.Class)), [unwrap](http://docs.google.com/java/sql/Wrapper.html#unwrap(java.lang.Class)) |

| **Methods inherited from interface javax.sql.rowset.**[**Joinable**](http://docs.google.com/javax/sql/rowset/Joinable.html) |
| --- |
| [getMatchColumnIndexes](http://docs.google.com/javax/sql/rowset/Joinable.html#getMatchColumnIndexes()), [getMatchColumnNames](http://docs.google.com/javax/sql/rowset/Joinable.html#getMatchColumnNames()), [setMatchColumn](http://docs.google.com/javax/sql/rowset/Joinable.html#setMatchColumn(int)), [setMatchColumn](http://docs.google.com/javax/sql/rowset/Joinable.html#setMatchColumn(int%5B%5D)), [setMatchColumn](http://docs.google.com/javax/sql/rowset/Joinable.html#setMatchColumn(java.lang.String)), [setMatchColumn](http://docs.google.com/javax/sql/rowset/Joinable.html#setMatchColumn(java.lang.String%5B%5D)), [unsetMatchColumn](http://docs.google.com/javax/sql/rowset/Joinable.html#unsetMatchColumn(int)), [unsetMatchColumn](http://docs.google.com/javax/sql/rowset/Joinable.html#unsetMatchColumn(int%5B%5D)), [unsetMatchColumn](http://docs.google.com/javax/sql/rowset/Joinable.html#unsetMatchColumn(java.lang.String)), [unsetMatchColumn](http://docs.google.com/javax/sql/rowset/Joinable.html#unsetMatchColumn(java.lang.String%5B%5D)) |

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

### getShowDeleted

boolean **getShowDeleted**()  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Retrieves a boolean indicating whether rows marked for deletion appear in the set of current rows. If true is returned, deleted rows are visible with the current rows. If false is returned, rows are not visible with the set of current rows. The default value is false.

Standard rowset implementations may choose to restrict this behavior for security considerations or for certain deployment scenarios. The visibility of deleted rows is implementation-defined and does not represent standard behavior.

Note: Allowing deleted rows to remain visible complicates the behavior of some standard JDBC RowSet implementations methods. However, most rowset users can simply ignore this extra detail because only very specialized applications will likely want to take advantage of this feature.

**Returns:**true if deleted rows are visible; false otherwise **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a rowset implementation is unable to to determine whether rows marked for deletion remain visible**See Also:**[setShowDeleted(boolean)](http://docs.google.com/javax/sql/rowset/JdbcRowSet.html#setShowDeleted(boolean))

### setShowDeleted

void **setShowDeleted**(boolean b)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Sets the property showDeleted to the given boolean value. This property determines whether rows marked for deletion continue to appear in the set of current rows. If the value is set to true, deleted rows are immediately visible with the set of current rows. If the value is set to false, the deleted rows are set as invisible with the current set of rows.

Standard rowset implementations may choose to restrict this behavior for security considerations or for certain deployment scenarios. This is left as implementation-defined and does not represent standard behavior.

**Parameters:**b - true if deleted rows should be shown; false otherwise **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a rowset implementation is unable to to reset whether deleted rows should be visible**See Also:**[getShowDeleted()](http://docs.google.com/javax/sql/rowset/JdbcRowSet.html#getShowDeleted())

### getRowSetWarnings

[RowSetWarning](http://docs.google.com/javax/sql/rowset/RowSetWarning.html) **getRowSetWarnings**()  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Retrieves the first warning reported by calls on this JdbcRowSet object. If a second warning was reported on this JdbcRowSet object, it will be chained to the first warning and can be retrieved by calling the method RowSetWarning.getNextWarning on the first warning. Subsequent warnings on this JdbcRowSet object will be chained to the RowSetWarning objects returned by the method RowSetWarning.getNextWarning. The warning chain is automatically cleared each time a new row is read. This method may not be called on a RowSet object that has been closed; doing so will cause an SQLException to be thrown.

Because it is always connected to its data source, a JdbcRowSet object can rely on the presence of active Statement, Connection, and ResultSet instances. This means that applications can obtain additional SQLWarning notifications by calling the getNextWarning methods that they provide. Disconnected Rowset objects, such as a CachedRowSet object, do not have access to these getNextWarning methods.

**Returns:**the first RowSetWarning object reported on this JdbcRowSet object or null if there are none **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if this method is called on a closed JdbcRowSet object**See Also:**[RowSetWarning](http://docs.google.com/javax/sql/rowset/RowSetWarning.html)

### commit

void **commit**()  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Each JdbcRowSet contains a Connection object from the ResultSet or JDBC properties passed to it's constructors. This method wraps the Connection commit method to allow flexible auto commit or non auto commit transactional control support.

Makes all changes made since the previous commit/rollback permanent and releases any database locks currently held by this Connection object. This method should be used only when auto-commit mode has been disabled.

**Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs or this Connection object within this JdbcRowSet is in auto-commit mode**See Also:**[Connection.setAutoCommit(boolean)](http://docs.google.com/java/sql/Connection.html#setAutoCommit(boolean))

### getAutoCommit

boolean **getAutoCommit**()  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Each JdbcRowSet contains a Connection object from the original ResultSet or JDBC properties passed to it. This method wraps the Connection's getAutoCommit method to allow an application to determine the JdbcRowSet transaction behavior.

Sets this connection's auto-commit mode to the given state. If a connection is in auto-commit mode, then all its SQL statements will be executed and committed as individual transactions. Otherwise, its SQL statements are grouped into transactions that are terminated by a call to either the method commit or the method rollback. By default, new connections are in auto-commit mode.

**Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs**See Also:**[Connection.getAutoCommit()](http://docs.google.com/java/sql/Connection.html#getAutoCommit())

### setAutoCommit

void **setAutoCommit**(boolean autoCommit)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Each JdbcRowSet contains a Connection object from the original ResultSet or JDBC properties passed to it. This method wraps the Connection's getAutoCommit method to allow an application to set the JdbcRowSet transaction behavior.

Sets the current auto-commit mode for this Connection object.

**Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs**See Also:**[Connection.setAutoCommit(boolean)](http://docs.google.com/java/sql/Connection.html#setAutoCommit(boolean))

### rollback

void **rollback**()  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Each JdbcRowSet contains a Connection object from the original ResultSet or JDBC properties passed to it. Undoes all changes made in the current transaction and releases any database locks currently held by this Connection object. This method should be used only when auto-commit mode has been disabled.

**Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs or this Connection object within this JdbcRowSet is in auto-commit mode.**See Also:**[rollback(Savepoint)](http://docs.google.com/javax/sql/rowset/JdbcRowSet.html#rollback(java.sql.Savepoint))

### rollback

void **rollback**([Savepoint](http://docs.google.com/java/sql/Savepoint.html) s)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Each JdbcRowSet contains a Connection object from the original ResultSet or JDBC properties passed to it. Undoes all changes made in the current transaction to the last set savepoint and releases any database locks currently held by this Connection object. This method should be used only when auto-commit mode has been disabled.

**Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs or this Connection object within this JdbcRowSet is in auto-commit mode.**See Also:**[rollback()](http://docs.google.com/javax/sql/rowset/JdbcRowSet.html#rollback())

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/JdbcRowSet.html) | [**Tree**](http://docs.google.com/package-tree.html) | [**Deprecated**](http://docs.google.com/deprecated-list.html) | [**Index**](http://docs.google.com/index-files/index-1.html) | [**Help**](http://docs.google.com/help-doc.html) | | --- | --- | --- | --- | --- | --- | --- | --- | | | ***Java™ Platform***  ***Standard Ed. 6*** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [**PREV CLASS**](http://docs.google.com/javax/sql/rowset/FilteredRowSet.html)   [**NEXT CLASS**](http://docs.google.com/javax/sql/rowset/Joinable.html) | [**FRAMES**](http://docs.google.com/index.html?javax/sql/rowset/JdbcRowSet.html)    [**NO FRAMES**](http://docs.google.com/JdbcRowSet.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | CONSTR | [METHOD](#tyjcwt) | DETAIL: FIELD | CONSTR | [METHOD](#17dp8vu) |

[Submit a bug or feature](http://bugs.sun.com/services/bugreport/index.jsp)

For further API reference and developer documentation, see [Java SE Developer Documentation](http://docs.google.com/webnotes/devdocs-vs-specs.html). That documentation contains more detailed, developer-targeted descriptions, with conceptual overviews, definitions of terms, workarounds, and working code examples.

Copyright 2006 Sun Microsystems, Inc. All rights reserved. Use is subject to [license terms](http://docs.google.com/legal/license.html). Also see the [documentation redistribution policy](http://java.sun.com/docs/redist.html).