| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/RowSet.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/PooledConnection.html)   [**NEXT CLASS**](http://docs.google.com/javax/sql/RowSetEvent.html) | [**FRAMES**](http://docs.google.com/index.html?javax/sql/RowSet.html)    [**NO FRAMES**](http://docs.google.com/RowSet.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | CONSTR | [METHOD](#tyjcwt) | DETAIL: FIELD | CONSTR | [METHOD](#4d34og8) |

## **javax.sql**

Interface RowSet

**All Superinterfaces:** [ResultSet](http://docs.google.com/java/sql/ResultSet.html), [Wrapper](http://docs.google.com/java/sql/Wrapper.html) **All Known Subinterfaces:** [CachedRowSet](http://docs.google.com/javax/sql/rowset/CachedRowSet.html), [FilteredRowSet](http://docs.google.com/javax/sql/rowset/FilteredRowSet.html), [JdbcRowSet](http://docs.google.com/javax/sql/rowset/JdbcRowSet.html), [JoinRowSet](http://docs.google.com/javax/sql/rowset/JoinRowSet.html), [SyncResolver](http://docs.google.com/javax/sql/rowset/spi/SyncResolver.html), [WebRowSet](http://docs.google.com/javax/sql/rowset/WebRowSet.html)

public interface **RowSet**extends [ResultSet](http://docs.google.com/java/sql/ResultSet.html)

The interface that adds support to the JDBC API for the JavaBeansTM component model. A rowset, which can be used as a JavaBeans component in a visual Bean development environment, can be created and configured at design time and executed at run time.

The RowSet interface provides a set of JavaBeans properties that allow a RowSet instance to be configured to connect to a JDBC data source and read some data from the data source. A group of setter methods (setInt, setBytes, setString, and so on) provide a way to pass input parameters to a rowset's command property. This command is the SQL query the rowset uses when it gets its data from a relational database, which is generally the case.

The RowSet interface supports JavaBeans events, allowing other components in an application to be notified when an event occurs on a rowset, such as a change in its value.

The RowSet interface is unique in that it is intended to be implemented using the rest of the JDBC API. In other words, a RowSet implementation is a layer of software that executes "on top" of a JDBC driver. Implementations of the RowSet interface can be provided by anyone, including JDBC driver vendors who want to provide a RowSet implementation as part of their JDBC products.

A RowSet object may make a connection with a data source and maintain that connection throughout its life cycle, in which case it is called a *connected* rowset. A rowset may also make a connection with a data source, get data from it, and then close the connection. Such a rowset is called a *disconnected* rowset. A disconnected rowset may make changes to its data while it is disconnected and then send the changes back to the original source of the data, but it must reestablish a connection to do so.

A disconnected rowset may have a reader (a RowSetReader object) and a writer (a RowSetWriter object) associated with it. The reader may be implemented in many different ways to populate a rowset with data, including getting data from a non-relational data source. The writer can also be implemented in many different ways to propagate changes made to the rowset's data back to the underlying data source.

Rowsets are easy to use. The RowSet interface extends the standard java.sql.ResultSet interface. The RowSetMetaData interface extends the java.sql.ResultSetMetaData interface. Thus, developers familiar with the JDBC API will have to learn a minimal number of new APIs to use rowsets. In addition, third-party software tools that work with JDBC ResultSet objects will also easily be made to work with rowsets.

**Since:** 1.4

| **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 | [**addRowSetListener**](http://docs.google.com/javax/sql/RowSet.html#addRowSetListener(javax.sql.RowSetListener))([RowSetListener](http://docs.google.com/javax/sql/RowSetListener.html) listener)            Registers the given listener so that it will be notified of events that occur on this RowSet object. |
| void | [**clearParameters**](http://docs.google.com/javax/sql/RowSet.html#clearParameters())()            Clears the parameters set for this RowSet object's command. |
| void | [**execute**](http://docs.google.com/javax/sql/RowSet.html#execute())()            Fills this RowSet object with data. |
| [String](http://docs.google.com/java/lang/String.html) | [**getCommand**](http://docs.google.com/javax/sql/RowSet.html#getCommand())()            Retrieves this RowSet object's command property. |
| [String](http://docs.google.com/java/lang/String.html) | [**getDataSourceName**](http://docs.google.com/javax/sql/RowSet.html#getDataSourceName())()            Retrieves the logical name that identifies the data source for this RowSet object. |
| boolean | [**getEscapeProcessing**](http://docs.google.com/javax/sql/RowSet.html#getEscapeProcessing())()            Retrieves whether escape processing is enabled for this RowSet object. |
| int | [**getMaxFieldSize**](http://docs.google.com/javax/sql/RowSet.html#getMaxFieldSize())()            Retrieves the maximum number of bytes that may be returned for certain column values. |
| int | [**getMaxRows**](http://docs.google.com/javax/sql/RowSet.html#getMaxRows())()            Retrieves the maximum number of rows that this RowSet object can contain. |
| [String](http://docs.google.com/java/lang/String.html) | [**getPassword**](http://docs.google.com/javax/sql/RowSet.html#getPassword())()            Retrieves the password used to create a database connection. |
| int | [**getQueryTimeout**](http://docs.google.com/javax/sql/RowSet.html#getQueryTimeout())()            Retrieves the maximum number of seconds the driver will wait for a statement to execute. |
| int | [**getTransactionIsolation**](http://docs.google.com/javax/sql/RowSet.html#getTransactionIsolation())()            Retrieves the transaction isolation level set for this RowSet object. |
| [Map](http://docs.google.com/java/util/Map.html)<[String](http://docs.google.com/java/lang/String.html),[Class](http://docs.google.com/java/lang/Class.html)<?>> | [**getTypeMap**](http://docs.google.com/javax/sql/RowSet.html#getTypeMap())()            Retrieves the Map object associated with this RowSet object, which specifies the custom mapping of SQL user-defined types, if any. |
| [String](http://docs.google.com/java/lang/String.html) | [**getUrl**](http://docs.google.com/javax/sql/RowSet.html#getUrl())()            Retrieves the url property this RowSet object will use to create a connection if it uses the DriverManager instead of a DataSource object to establish the connection. |
| [String](http://docs.google.com/java/lang/String.html) | [**getUsername**](http://docs.google.com/javax/sql/RowSet.html#getUsername())()            Retrieves the username used to create a database connection for this RowSet object. |
| boolean | [**isReadOnly**](http://docs.google.com/javax/sql/RowSet.html#isReadOnly())()            Retrieves whether this RowSet object is read-only. |
| void | [**removeRowSetListener**](http://docs.google.com/javax/sql/RowSet.html#removeRowSetListener(javax.sql.RowSetListener))([RowSetListener](http://docs.google.com/javax/sql/RowSetListener.html) listener)            Removes the specified listener from the list of components that will be notified when an event occurs on this RowSet object. |
| void | [**setArray**](http://docs.google.com/javax/sql/RowSet.html#setArray(int,%20java.sql.Array))(int i, [Array](http://docs.google.com/java/sql/Array.html) x)            Sets the designated parameter in this RowSet object's command with the given Array value. |
| void | [**setAsciiStream**](http://docs.google.com/javax/sql/RowSet.html#setAsciiStream(int,%20java.io.InputStream))(int parameterIndex, [InputStream](http://docs.google.com/java/io/InputStream.html) x)            Sets the designated parameter in this RowSet object's command to the given input stream. |
| void | [**setAsciiStream**](http://docs.google.com/javax/sql/RowSet.html#setAsciiStream(int,%20java.io.InputStream,%20int))(int parameterIndex, [InputStream](http://docs.google.com/java/io/InputStream.html) x, int length)            Sets the designated parameter in this RowSet object's command to the given java.io.InputStream value. |
| void | [**setAsciiStream**](http://docs.google.com/javax/sql/RowSet.html#setAsciiStream(java.lang.String,%20java.io.InputStream))([String](http://docs.google.com/java/lang/String.html) parameterName, [InputStream](http://docs.google.com/java/io/InputStream.html) x)            Sets the designated parameter to the given input stream. |
| void | [**setAsciiStream**](http://docs.google.com/javax/sql/RowSet.html#setAsciiStream(java.lang.String,%20java.io.InputStream,%20int))([String](http://docs.google.com/java/lang/String.html) parameterName, [InputStream](http://docs.google.com/java/io/InputStream.html) x, int length)            Sets the designated parameter to the given input stream, which will have the specified number of bytes. |
| void | [**setBigDecimal**](http://docs.google.com/javax/sql/RowSet.html#setBigDecimal(int,%20java.math.BigDecimal))(int parameterIndex, [BigDecimal](http://docs.google.com/java/math/BigDecimal.html) x)            Sets the designated parameter in this RowSet object's command to the given java.math.BigDeciaml value. |
| void | [**setBigDecimal**](http://docs.google.com/javax/sql/RowSet.html#setBigDecimal(java.lang.String,%20java.math.BigDecimal))([String](http://docs.google.com/java/lang/String.html) parameterName, [BigDecimal](http://docs.google.com/java/math/BigDecimal.html) x)            Sets the designated parameter to the given java.math.BigDecimal value. |
| void | [**setBinaryStream**](http://docs.google.com/javax/sql/RowSet.html#setBinaryStream(int,%20java.io.InputStream))(int parameterIndex, [InputStream](http://docs.google.com/java/io/InputStream.html) x)            Sets the designated parameter in this RowSet object's command to the given input stream. |
| void | [**setBinaryStream**](http://docs.google.com/javax/sql/RowSet.html#setBinaryStream(int,%20java.io.InputStream,%20int))(int parameterIndex, [InputStream](http://docs.google.com/java/io/InputStream.html) x, int length)            Sets the designated parameter in this RowSet object's command to the given java.io.InputStream value. |
| void | [**setBinaryStream**](http://docs.google.com/javax/sql/RowSet.html#setBinaryStream(java.lang.String,%20java.io.InputStream))([String](http://docs.google.com/java/lang/String.html) parameterName, [InputStream](http://docs.google.com/java/io/InputStream.html) x)            Sets the designated parameter to the given input stream. |
| void | [**setBinaryStream**](http://docs.google.com/javax/sql/RowSet.html#setBinaryStream(java.lang.String,%20java.io.InputStream,%20int))([String](http://docs.google.com/java/lang/String.html) parameterName, [InputStream](http://docs.google.com/java/io/InputStream.html) x, int length)            Sets the designated parameter to the given input stream, which will have the specified number of bytes. |
| void | [**setBlob**](http://docs.google.com/javax/sql/RowSet.html#setBlob(int,%20java.sql.Blob))(int i, [Blob](http://docs.google.com/java/sql/Blob.html) x)            Sets the designated parameter in this RowSet object's command with the given Blob value. |
| void | [**setBlob**](http://docs.google.com/javax/sql/RowSet.html#setBlob(int,%20java.io.InputStream))(int parameterIndex, [InputStream](http://docs.google.com/java/io/InputStream.html) inputStream)            Sets the designated parameter to a InputStream object. |
| void | [**setBlob**](http://docs.google.com/javax/sql/RowSet.html#setBlob(int,%20java.io.InputStream,%20long))(int parameterIndex, [InputStream](http://docs.google.com/java/io/InputStream.html) inputStream, long length)            Sets the designated parameter to a InputStream object. |
| void | [**setBlob**](http://docs.google.com/javax/sql/RowSet.html#setBlob(java.lang.String,%20java.sql.Blob))([String](http://docs.google.com/java/lang/String.html) parameterName, [Blob](http://docs.google.com/java/sql/Blob.html) x)            Sets the designated parameter to the given java.sql.Blob object. |
| void | [**setBlob**](http://docs.google.com/javax/sql/RowSet.html#setBlob(java.lang.String,%20java.io.InputStream))([String](http://docs.google.com/java/lang/String.html) parameterName, [InputStream](http://docs.google.com/java/io/InputStream.html) inputStream)            Sets the designated parameter to a InputStream object. |
| void | [**setBlob**](http://docs.google.com/javax/sql/RowSet.html#setBlob(java.lang.String,%20java.io.InputStream,%20long))([String](http://docs.google.com/java/lang/String.html) parameterName, [InputStream](http://docs.google.com/java/io/InputStream.html) inputStream, long length)            Sets the designated parameter to a InputStream object. |
| void | [**setBoolean**](http://docs.google.com/javax/sql/RowSet.html#setBoolean(int,%20boolean))(int parameterIndex, boolean x)            Sets the designated parameter in this RowSet object's command to the given Java boolean value. |
| void | [**setBoolean**](http://docs.google.com/javax/sql/RowSet.html#setBoolean(java.lang.String,%20boolean))([String](http://docs.google.com/java/lang/String.html) parameterName, boolean x)            Sets the designated parameter to the given Java boolean value. |
| void | [**setByte**](http://docs.google.com/javax/sql/RowSet.html#setByte(int,%20byte))(int parameterIndex, byte x)            Sets the designated parameter in this RowSet object's command to the given Java byte value. |
| void | [**setByte**](http://docs.google.com/javax/sql/RowSet.html#setByte(java.lang.String,%20byte))([String](http://docs.google.com/java/lang/String.html) parameterName, byte x)            Sets the designated parameter to the given Java byte value. |
| void | [**setBytes**](http://docs.google.com/javax/sql/RowSet.html#setBytes(int,%20byte%5B%5D))(int parameterIndex, byte[] x)            Sets the designated parameter in this RowSet object's command to the given Java array of byte values. |
| void | [**setBytes**](http://docs.google.com/javax/sql/RowSet.html#setBytes(java.lang.String,%20byte%5B%5D))([String](http://docs.google.com/java/lang/String.html) parameterName, byte[] x)            Sets the designated parameter to the given Java array of bytes. |
| void | [**setCharacterStream**](http://docs.google.com/javax/sql/RowSet.html#setCharacterStream(int,%20java.io.Reader))(int parameterIndex, [Reader](http://docs.google.com/java/io/Reader.html) reader)            Sets the designated parameter in this RowSet object's command to the given Reader object. |
| void | [**setCharacterStream**](http://docs.google.com/javax/sql/RowSet.html#setCharacterStream(int,%20java.io.Reader,%20int))(int parameterIndex, [Reader](http://docs.google.com/java/io/Reader.html) reader, int length)            Sets the designated parameter in this RowSet object's command to the given java.io.Reader value. |
| void | [**setCharacterStream**](http://docs.google.com/javax/sql/RowSet.html#setCharacterStream(java.lang.String,%20java.io.Reader))([String](http://docs.google.com/java/lang/String.html) parameterName, [Reader](http://docs.google.com/java/io/Reader.html) reader)            Sets the designated parameter to the given Reader object. |
| void | [**setCharacterStream**](http://docs.google.com/javax/sql/RowSet.html#setCharacterStream(java.lang.String,%20java.io.Reader,%20int))([String](http://docs.google.com/java/lang/String.html) parameterName, [Reader](http://docs.google.com/java/io/Reader.html) reader, int length)            Sets the designated parameter to the given Reader object, which is the given number of characters long. |
| void | [**setClob**](http://docs.google.com/javax/sql/RowSet.html#setClob(int,%20java.sql.Clob))(int i, [Clob](http://docs.google.com/java/sql/Clob.html) x)            Sets the designated parameter in this RowSet object's command with the given Clob value. |
| void | [**setClob**](http://docs.google.com/javax/sql/RowSet.html#setClob(int,%20java.io.Reader))(int parameterIndex, [Reader](http://docs.google.com/java/io/Reader.html) reader)            Sets the designated parameter to a Reader object. |
| void | [**setClob**](http://docs.google.com/javax/sql/RowSet.html#setClob(int,%20java.io.Reader,%20long))(int parameterIndex, [Reader](http://docs.google.com/java/io/Reader.html) reader, long length)            Sets the designated parameter to a Reader object. |
| void | [**setClob**](http://docs.google.com/javax/sql/RowSet.html#setClob(java.lang.String,%20java.sql.Clob))([String](http://docs.google.com/java/lang/String.html) parameterName, [Clob](http://docs.google.com/java/sql/Clob.html) x)            Sets the designated parameter to the given java.sql.Clob object. |
| void | [**setClob**](http://docs.google.com/javax/sql/RowSet.html#setClob(java.lang.String,%20java.io.Reader))([String](http://docs.google.com/java/lang/String.html) parameterName, [Reader](http://docs.google.com/java/io/Reader.html) reader)            Sets the designated parameter to a Reader object. |
| void | [**setClob**](http://docs.google.com/javax/sql/RowSet.html#setClob(java.lang.String,%20java.io.Reader,%20long))([String](http://docs.google.com/java/lang/String.html) parameterName, [Reader](http://docs.google.com/java/io/Reader.html) reader, long length)            Sets the designated parameter to a Reader object. |
| void | [**setCommand**](http://docs.google.com/javax/sql/RowSet.html#setCommand(java.lang.String))([String](http://docs.google.com/java/lang/String.html) cmd)            Sets this RowSet object's command property to the given SQL query. |
| void | [**setConcurrency**](http://docs.google.com/javax/sql/RowSet.html#setConcurrency(int))(int concurrency)            Sets the concurrency of this RowSet object to the given concurrency level. |
| void | [**setDataSourceName**](http://docs.google.com/javax/sql/RowSet.html#setDataSourceName(java.lang.String))([String](http://docs.google.com/java/lang/String.html) name)            Sets the data source name property for this RowSet object to the given String. |
| void | [**setDate**](http://docs.google.com/javax/sql/RowSet.html#setDate(int,%20java.sql.Date))(int parameterIndex, [Date](http://docs.google.com/java/sql/Date.html) x)            Sets the designated parameter in this RowSet object's command to the given java.sql.Date value. |
| void | [**setDate**](http://docs.google.com/javax/sql/RowSet.html#setDate(int,%20java.sql.Date,%20java.util.Calendar))(int parameterIndex, [Date](http://docs.google.com/java/sql/Date.html) x, [Calendar](http://docs.google.com/java/util/Calendar.html) cal)            Sets the designated parameter in this RowSet object's command with the given java.sql.Date value. |
| void | [**setDate**](http://docs.google.com/javax/sql/RowSet.html#setDate(java.lang.String,%20java.sql.Date))([String](http://docs.google.com/java/lang/String.html) parameterName, [Date](http://docs.google.com/java/sql/Date.html) x)            Sets the designated parameter to the given java.sql.Date value using the default time zone of the virtual machine that is running the application. |
| void | [**setDate**](http://docs.google.com/javax/sql/RowSet.html#setDate(java.lang.String,%20java.sql.Date,%20java.util.Calendar))([String](http://docs.google.com/java/lang/String.html) parameterName, [Date](http://docs.google.com/java/sql/Date.html) x, [Calendar](http://docs.google.com/java/util/Calendar.html) cal)            Sets the designated parameter to the given java.sql.Date value, using the given Calendar object. |
| void | [**setDouble**](http://docs.google.com/javax/sql/RowSet.html#setDouble(int,%20double))(int parameterIndex, double x)            Sets the designated parameter in this RowSet object's command to the given Java double value. |
| void | [**setDouble**](http://docs.google.com/javax/sql/RowSet.html#setDouble(java.lang.String,%20double))([String](http://docs.google.com/java/lang/String.html) parameterName, double x)            Sets the designated parameter to the given Java double value. |
| void | [**setEscapeProcessing**](http://docs.google.com/javax/sql/RowSet.html#setEscapeProcessing(boolean))(boolean enable)            Sets escape processing for this RowSet object on or off. |
| void | [**setFloat**](http://docs.google.com/javax/sql/RowSet.html#setFloat(int,%20float))(int parameterIndex, float x)            Sets the designated parameter in this RowSet object's command to the given Java float value. |
| void | [**setFloat**](http://docs.google.com/javax/sql/RowSet.html#setFloat(java.lang.String,%20float))([String](http://docs.google.com/java/lang/String.html) parameterName, float x)            Sets the designated parameter to the given Java float value. |
| void | [**setInt**](http://docs.google.com/javax/sql/RowSet.html#setInt(int,%20int))(int parameterIndex, int x)            Sets the designated parameter in this RowSet object's command to the given Java int value. |
| void | [**setInt**](http://docs.google.com/javax/sql/RowSet.html#setInt(java.lang.String,%20int))([String](http://docs.google.com/java/lang/String.html) parameterName, int x)            Sets the designated parameter to the given Java int value. |
| void | [**setLong**](http://docs.google.com/javax/sql/RowSet.html#setLong(int,%20long))(int parameterIndex, long x)            Sets the designated parameter in this RowSet object's command to the given Java long value. |
| void | [**setLong**](http://docs.google.com/javax/sql/RowSet.html#setLong(java.lang.String,%20long))([String](http://docs.google.com/java/lang/String.html) parameterName, long x)            Sets the designated parameter to the given Java long value. |
| void | [**setMaxFieldSize**](http://docs.google.com/javax/sql/RowSet.html#setMaxFieldSize(int))(int max)            Sets the maximum number of bytes that can be returned for a column value to the given number of bytes. |
| void | [**setMaxRows**](http://docs.google.com/javax/sql/RowSet.html#setMaxRows(int))(int max)            Sets the maximum number of rows that this RowSet object can contain to the specified number. |
| void | [**setNCharacterStream**](http://docs.google.com/javax/sql/RowSet.html#setNCharacterStream(int,%20java.io.Reader))(int parameterIndex, [Reader](http://docs.google.com/java/io/Reader.html) value)            Sets the designated parameter in this RowSet object's command to a Reader object. |
| void | [**setNCharacterStream**](http://docs.google.com/javax/sql/RowSet.html#setNCharacterStream(int,%20java.io.Reader,%20long))(int parameterIndex, [Reader](http://docs.google.com/java/io/Reader.html) value, long length)            Sets the designated parameter to a Reader object. |
| void | [**setNCharacterStream**](http://docs.google.com/javax/sql/RowSet.html#setNCharacterStream(java.lang.String,%20java.io.Reader))([String](http://docs.google.com/java/lang/String.html) parameterName, [Reader](http://docs.google.com/java/io/Reader.html) value)            Sets the designated parameter to a Reader object. |
| void | [**setNCharacterStream**](http://docs.google.com/javax/sql/RowSet.html#setNCharacterStream(java.lang.String,%20java.io.Reader,%20long))([String](http://docs.google.com/java/lang/String.html) parameterName, [Reader](http://docs.google.com/java/io/Reader.html) value, long length)            Sets the designated parameter to a Reader object. |
| void | [**setNClob**](http://docs.google.com/javax/sql/RowSet.html#setNClob(int,%20java.sql.NClob))(int parameterIndex, [NClob](http://docs.google.com/java/sql/NClob.html) value)            Sets the designated parameter to a java.sql.NClob object. |
| void | [**setNClob**](http://docs.google.com/javax/sql/RowSet.html#setNClob(int,%20java.io.Reader))(int parameterIndex, [Reader](http://docs.google.com/java/io/Reader.html) reader)            Sets the designated parameter to a Reader object. |
| void | [**setNClob**](http://docs.google.com/javax/sql/RowSet.html#setNClob(int,%20java.io.Reader,%20long))(int parameterIndex, [Reader](http://docs.google.com/java/io/Reader.html) reader, long length)            Sets the designated parameter to a Reader object. |
| void | [**setNClob**](http://docs.google.com/javax/sql/RowSet.html#setNClob(java.lang.String,%20java.sql.NClob))([String](http://docs.google.com/java/lang/String.html) parameterName, [NClob](http://docs.google.com/java/sql/NClob.html) value)            Sets the designated parameter to a java.sql.NClob object. |
| void | [**setNClob**](http://docs.google.com/javax/sql/RowSet.html#setNClob(java.lang.String,%20java.io.Reader))([String](http://docs.google.com/java/lang/String.html) parameterName, [Reader](http://docs.google.com/java/io/Reader.html) reader)            Sets the designated parameter to a Reader object. |
| void | [**setNClob**](http://docs.google.com/javax/sql/RowSet.html#setNClob(java.lang.String,%20java.io.Reader,%20long))([String](http://docs.google.com/java/lang/String.html) parameterName, [Reader](http://docs.google.com/java/io/Reader.html) reader, long length)            Sets the designated parameter to a Reader object. |
| void | [**setNString**](http://docs.google.com/javax/sql/RowSet.html#setNString(int,%20java.lang.String))(int parameterIndex, [String](http://docs.google.com/java/lang/String.html) value)            Sets the designated paramter to the given String object. |
| void | [**setNString**](http://docs.google.com/javax/sql/RowSet.html#setNString(java.lang.String,%20java.lang.String))([String](http://docs.google.com/java/lang/String.html) parameterName, [String](http://docs.google.com/java/lang/String.html) value)            Sets the designated paramter to the given String object. |
| void | [**setNull**](http://docs.google.com/javax/sql/RowSet.html#setNull(int,%20int))(int parameterIndex, int sqlType)            Sets the designated parameter in this RowSet object's SQL command to SQL NULL. |
| void | [**setNull**](http://docs.google.com/javax/sql/RowSet.html#setNull(int,%20int,%20java.lang.String))(int paramIndex, int sqlType, [String](http://docs.google.com/java/lang/String.html) typeName)            Sets the designated parameter in this RowSet object's SQL command to SQL NULL. |
| void | [**setNull**](http://docs.google.com/javax/sql/RowSet.html#setNull(java.lang.String,%20int))([String](http://docs.google.com/java/lang/String.html) parameterName, int sqlType)            Sets the designated parameter to SQL NULL. |
| void | [**setNull**](http://docs.google.com/javax/sql/RowSet.html#setNull(java.lang.String,%20int,%20java.lang.String))([String](http://docs.google.com/java/lang/String.html) parameterName, int sqlType, [String](http://docs.google.com/java/lang/String.html) typeName)            Sets the designated parameter to SQL NULL. |
| void | [**setObject**](http://docs.google.com/javax/sql/RowSet.html#setObject(int,%20java.lang.Object))(int parameterIndex, [Object](http://docs.google.com/java/lang/Object.html) x)            Sets the designated parameter in this RowSet object's command with a Java Object. |
| void | [**setObject**](http://docs.google.com/javax/sql/RowSet.html#setObject(int,%20java.lang.Object,%20int))(int parameterIndex, [Object](http://docs.google.com/java/lang/Object.html) x, int targetSqlType)            Sets the designated parameter in this RowSet object's command with a Java Object. |
| void | [**setObject**](http://docs.google.com/javax/sql/RowSet.html#setObject(int,%20java.lang.Object,%20int,%20int))(int parameterIndex, [Object](http://docs.google.com/java/lang/Object.html) x, int targetSqlType, int scaleOrLength)            Sets the designated parameter in this RowSet object's command with the given Java Object. |
| void | [**setObject**](http://docs.google.com/javax/sql/RowSet.html#setObject(java.lang.String,%20java.lang.Object))([String](http://docs.google.com/java/lang/String.html) parameterName, [Object](http://docs.google.com/java/lang/Object.html) x)            Sets the value of the designated parameter with the given object. |
| void | [**setObject**](http://docs.google.com/javax/sql/RowSet.html#setObject(java.lang.String,%20java.lang.Object,%20int))([String](http://docs.google.com/java/lang/String.html) parameterName, [Object](http://docs.google.com/java/lang/Object.html) x, int targetSqlType)            Sets the value of the designated parameter with the given object. |
| void | [**setObject**](http://docs.google.com/javax/sql/RowSet.html#setObject(java.lang.String,%20java.lang.Object,%20int,%20int))([String](http://docs.google.com/java/lang/String.html) parameterName, [Object](http://docs.google.com/java/lang/Object.html) x, int targetSqlType, int scale)            Sets the value of the designated parameter with the given object. |
| void | [**setPassword**](http://docs.google.com/javax/sql/RowSet.html#setPassword(java.lang.String))([String](http://docs.google.com/java/lang/String.html) password)            Sets the database password for this RowSet object to the given String. |
| void | [**setQueryTimeout**](http://docs.google.com/javax/sql/RowSet.html#setQueryTimeout(int))(int seconds)            Sets the maximum time the driver will wait for a statement to execute to the given number of seconds. |
| void | [**setReadOnly**](http://docs.google.com/javax/sql/RowSet.html#setReadOnly(boolean))(boolean value)            Sets whether this RowSet object is read-only to the given boolean. |
| void | [**setRef**](http://docs.google.com/javax/sql/RowSet.html#setRef(int,%20java.sql.Ref))(int i, [Ref](http://docs.google.com/java/sql/Ref.html) x)            Sets the designated parameter in this RowSet object's command with the given Ref value. |
| void | [**setRowId**](http://docs.google.com/javax/sql/RowSet.html#setRowId(int,%20java.sql.RowId))(int parameterIndex, [RowId](http://docs.google.com/java/sql/RowId.html) x)            Sets the designated parameter to the given java.sql.RowId object. |
| void | [**setRowId**](http://docs.google.com/javax/sql/RowSet.html#setRowId(java.lang.String,%20java.sql.RowId))([String](http://docs.google.com/java/lang/String.html) parameterName, [RowId](http://docs.google.com/java/sql/RowId.html) x)            Sets the designated parameter to the given java.sql.RowId object. |
| void | [**setShort**](http://docs.google.com/javax/sql/RowSet.html#setShort(int,%20short))(int parameterIndex, short x)            Sets the designated parameter in this RowSet object's command to the given Java short value. |
| void | [**setShort**](http://docs.google.com/javax/sql/RowSet.html#setShort(java.lang.String,%20short))([String](http://docs.google.com/java/lang/String.html) parameterName, short x)            Sets the designated parameter to the given Java short value. |
| void | [**setSQLXML**](http://docs.google.com/javax/sql/RowSet.html#setSQLXML(int,%20java.sql.SQLXML))(int parameterIndex, [SQLXML](http://docs.google.com/java/sql/SQLXML.html) xmlObject)            Sets the designated parameter to the given java.sql.SQLXML object. |
| void | [**setSQLXML**](http://docs.google.com/javax/sql/RowSet.html#setSQLXML(java.lang.String,%20java.sql.SQLXML))([String](http://docs.google.com/java/lang/String.html) parameterName, [SQLXML](http://docs.google.com/java/sql/SQLXML.html) xmlObject)            Sets the designated parameter to the given java.sql.SQLXML object. |
| void | [**setString**](http://docs.google.com/javax/sql/RowSet.html#setString(int,%20java.lang.String))(int parameterIndex, [String](http://docs.google.com/java/lang/String.html) x)            Sets the designated parameter in this RowSet object's command to the given Java String value. |
| void | [**setString**](http://docs.google.com/javax/sql/RowSet.html#setString(java.lang.String,%20java.lang.String))([String](http://docs.google.com/java/lang/String.html) parameterName, [String](http://docs.google.com/java/lang/String.html) x)            Sets the designated parameter to the given Java String value. |
| void | [**setTime**](http://docs.google.com/javax/sql/RowSet.html#setTime(int,%20java.sql.Time))(int parameterIndex, [Time](http://docs.google.com/java/sql/Time.html) x)            Sets the designated parameter in this RowSet object's command to the given java.sql.Time value. |
| void | [**setTime**](http://docs.google.com/javax/sql/RowSet.html#setTime(int,%20java.sql.Time,%20java.util.Calendar))(int parameterIndex, [Time](http://docs.google.com/java/sql/Time.html) x, [Calendar](http://docs.google.com/java/util/Calendar.html) cal)            Sets the designated parameter in this RowSet object's command with the given java.sql.Time value. |
| void | [**setTime**](http://docs.google.com/javax/sql/RowSet.html#setTime(java.lang.String,%20java.sql.Time))([String](http://docs.google.com/java/lang/String.html) parameterName, [Time](http://docs.google.com/java/sql/Time.html) x)            Sets the designated parameter to the given java.sql.Time value. |
| void | [**setTime**](http://docs.google.com/javax/sql/RowSet.html#setTime(java.lang.String,%20java.sql.Time,%20java.util.Calendar))([String](http://docs.google.com/java/lang/String.html) parameterName, [Time](http://docs.google.com/java/sql/Time.html) x, [Calendar](http://docs.google.com/java/util/Calendar.html) cal)            Sets the designated parameter to the given java.sql.Time value, using the given Calendar object. |
| void | [**setTimestamp**](http://docs.google.com/javax/sql/RowSet.html#setTimestamp(int,%20java.sql.Timestamp))(int parameterIndex, [Timestamp](http://docs.google.com/java/sql/Timestamp.html) x)            Sets the designated parameter in this RowSet object's command to the given java.sql.Timestamp value. |
| void | [**setTimestamp**](http://docs.google.com/javax/sql/RowSet.html#setTimestamp(int,%20java.sql.Timestamp,%20java.util.Calendar))(int parameterIndex, [Timestamp](http://docs.google.com/java/sql/Timestamp.html) x, [Calendar](http://docs.google.com/java/util/Calendar.html) cal)            Sets the designated parameter in this RowSet object's command with the given java.sql.Timestamp value. |
| void | [**setTimestamp**](http://docs.google.com/javax/sql/RowSet.html#setTimestamp(java.lang.String,%20java.sql.Timestamp))([String](http://docs.google.com/java/lang/String.html) parameterName, [Timestamp](http://docs.google.com/java/sql/Timestamp.html) x)            Sets the designated parameter to the given java.sql.Timestamp value. |
| void | [**setTimestamp**](http://docs.google.com/javax/sql/RowSet.html#setTimestamp(java.lang.String,%20java.sql.Timestamp,%20java.util.Calendar))([String](http://docs.google.com/java/lang/String.html) parameterName, [Timestamp](http://docs.google.com/java/sql/Timestamp.html) x, [Calendar](http://docs.google.com/java/util/Calendar.html) cal)            Sets the designated parameter to the given java.sql.Timestamp value, using the given Calendar object. |
| void | [**setTransactionIsolation**](http://docs.google.com/javax/sql/RowSet.html#setTransactionIsolation(int))(int level)            Sets the transaction isolation level for this RowSet obejct. |
| void | [**setType**](http://docs.google.com/javax/sql/RowSet.html#setType(int))(int type)            Sets the type of this RowSet object to the given type. |
| void | [**setTypeMap**](http://docs.google.com/javax/sql/RowSet.html#setTypeMap(java.util.Map))([Map](http://docs.google.com/java/util/Map.html)<[String](http://docs.google.com/java/lang/String.html),[Class](http://docs.google.com/java/lang/Class.html)<?>> map)            Installs the given java.util.Map object as the default type map for this RowSet object. |
| void | [**setURL**](http://docs.google.com/javax/sql/RowSet.html#setURL(int,%20java.net.URL))(int parameterIndex, [URL](http://docs.google.com/java/net/URL.html) x)            Sets the designated parameter to the given java.net.URL value. |
| void | [**setUrl**](http://docs.google.com/javax/sql/RowSet.html#setUrl(java.lang.String))([String](http://docs.google.com/java/lang/String.html) url)            Sets the URL this RowSet object will use when it uses the DriverManager to create a connection. |
| void | [**setUsername**](http://docs.google.com/javax/sql/RowSet.html#setUsername(java.lang.String))([String](http://docs.google.com/java/lang/String.html) name)            Sets the username property for this RowSet object to the given 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)) |

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

### getUrl

[String](http://docs.google.com/java/lang/String.html) **getUrl**()  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Retrieves the url property this RowSet object will use to create a connection if it uses the DriverManager instead of a DataSource object to establish the connection. The default value is null.

**Returns:**a string url **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs**See Also:**[setUrl(java.lang.String)](http://docs.google.com/javax/sql/RowSet.html#setUrl(java.lang.String))

### setUrl

void **setUrl**([String](http://docs.google.com/java/lang/String.html) url)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Sets the URL this RowSet object will use when it uses the DriverManager to create a connection. Setting this property is optional. If a URL is used, a JDBC driver that accepts the URL must be loaded before the rowset is used to connect to a database. The rowset will use the URL internally to create a database connection when reading or writing data. Either a URL or a data source name is used to create a connection, whichever was set to non null value most recently.

**Parameters:**url - a string value; may be null **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs**See Also:**[getUrl()](http://docs.google.com/javax/sql/RowSet.html#getUrl())

### getDataSourceName

[String](http://docs.google.com/java/lang/String.html) **getDataSourceName**()

Retrieves the logical name that identifies the data source for this RowSet object.

**Returns:**a data source name**See Also:**[setDataSourceName(java.lang.String)](http://docs.google.com/javax/sql/RowSet.html#setDataSourceName(java.lang.String)), [setUrl(java.lang.String)](http://docs.google.com/javax/sql/RowSet.html#setUrl(java.lang.String))

### setDataSourceName

void **setDataSourceName**([String](http://docs.google.com/java/lang/String.html) name)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Sets the data source name property for this RowSet object to the given String.

The value of the data source name property can be used to do a lookup of a DataSource object that has been registered with a naming service. After being retrieved, the DataSource object can be used to create a connection to the data source that it represents.

**Parameters:**name - the logical name of the data source for this RowSet object; may be null **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs**See Also:**[getDataSourceName()](http://docs.google.com/javax/sql/RowSet.html#getDataSourceName())

### getUsername

[String](http://docs.google.com/java/lang/String.html) **getUsername**()

Retrieves the username used to create a database connection for this RowSet object. The username property is set at run time before calling the method execute. It is not usually part of the serialized state of a RowSet object.

**Returns:**the username property**See Also:**[setUsername(java.lang.String)](http://docs.google.com/javax/sql/RowSet.html#setUsername(java.lang.String))

### setUsername

void **setUsername**([String](http://docs.google.com/java/lang/String.html) name)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Sets the username property for this RowSet object to the given String.

**Parameters:**name - a user name **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs**See Also:**[getUsername()](http://docs.google.com/javax/sql/RowSet.html#getUsername())

### getPassword

[String](http://docs.google.com/java/lang/String.html) **getPassword**()

Retrieves the password used to create a database connection. The password property is set at run time before calling the method execute. It is not usually part of the serialized state of a RowSet object.

**Returns:**the password for making a database connection**See Also:**[setPassword(java.lang.String)](http://docs.google.com/javax/sql/RowSet.html#setPassword(java.lang.String))

### setPassword

void **setPassword**([String](http://docs.google.com/java/lang/String.html) password)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Sets the database password for this RowSet object to the given String.

**Parameters:**password - the password string **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs**See Also:**[getPassword()](http://docs.google.com/javax/sql/RowSet.html#getPassword())

### getTransactionIsolation

int **getTransactionIsolation**()

Retrieves the transaction isolation level set for this RowSet object.

**Returns:**the transaction isolation level; one of Connection.TRANSACTION\_READ\_UNCOMMITTED, Connection.TRANSACTION\_READ\_COMMITTED, Connection.TRANSACTION\_REPEATABLE\_READ, or Connection.TRANSACTION\_SERIALIZABLE**See Also:**[setTransactionIsolation(int)](http://docs.google.com/javax/sql/RowSet.html#setTransactionIsolation(int))

### setTransactionIsolation

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

Sets the transaction isolation level for this RowSet obejct.

**Parameters:**level - the transaction isolation level; one of Connection.TRANSACTION\_READ\_UNCOMMITTED, Connection.TRANSACTION\_READ\_COMMITTED, Connection.TRANSACTION\_REPEATABLE\_READ, or Connection.TRANSACTION\_SERIALIZABLE **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs**See Also:**[getTransactionIsolation()](http://docs.google.com/javax/sql/RowSet.html#getTransactionIsolation())

### getTypeMap

[Map](http://docs.google.com/java/util/Map.html)<[String](http://docs.google.com/java/lang/String.html),[Class](http://docs.google.com/java/lang/Class.html)<?>> **getTypeMap**()  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Retrieves the Map object associated with this RowSet object, which specifies the custom mapping of SQL user-defined types, if any. The default is for the type map to be empty.

**Returns:**a java.util.Map object containing the names of SQL user-defined types and the Java classes to which they are to be mapped **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs**See Also:**[setTypeMap(java.util.Map>)](http://docs.google.com/javax/sql/RowSet.html#setTypeMap(java.util.Map))

### setTypeMap

void **setTypeMap**([Map](http://docs.google.com/java/util/Map.html)<[String](http://docs.google.com/java/lang/String.html),[Class](http://docs.google.com/java/lang/Class.html)<?>> map)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Installs the given java.util.Map object as the default type map for this RowSet object. This type map will be used unless another type map is supplied as a method parameter.

**Parameters:**map - a java.util.Map object containing the names of SQL user-defined types and the Java classes to which they are to be mapped **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs**See Also:**[getTypeMap()](http://docs.google.com/javax/sql/RowSet.html#getTypeMap())

### getCommand

[String](http://docs.google.com/java/lang/String.html) **getCommand**()

Retrieves this RowSet object's command property. The command property contains a command string, which must be an SQL query, that can be executed to fill the rowset with data. The default value is null.

**Returns:**the command string; may be null**See Also:**[setCommand(java.lang.String)](http://docs.google.com/javax/sql/RowSet.html#setCommand(java.lang.String))

### setCommand

void **setCommand**([String](http://docs.google.com/java/lang/String.html) cmd)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Sets this RowSet object's command property to the given SQL query. This property is optional when a rowset gets its data from a data source that does not support commands, such as a spreadsheet.

**Parameters:**cmd - the SQL query that will be used to get the data for this RowSet object; may be null **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs**See Also:**[getCommand()](http://docs.google.com/javax/sql/RowSet.html#getCommand())

### isReadOnly

boolean **isReadOnly**()

Retrieves whether this RowSet object is read-only. If updates are possible, the default is for a rowset to be updatable.

Attempts to update a read-only rowset will result in an SQLException being thrown.

**Returns:**true if this RowSet object is read-only; false if it is updatable**See Also:**[setReadOnly(boolean)](http://docs.google.com/javax/sql/RowSet.html#setReadOnly(boolean))

### setReadOnly

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

Sets whether this RowSet object is read-only to the given boolean.

**Parameters:**value - true if read-only; false if updatable **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs**See Also:**[isReadOnly()](http://docs.google.com/javax/sql/RowSet.html#isReadOnly())

### getMaxFieldSize

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

Retrieves the maximum number of bytes that may be returned for certain column values. This limit applies only to BINARY, VARBINARY, LONGVARBINARYBINARY, CHAR, VARCHAR, LONGVARCHAR, NCHAR and NVARCHAR columns. If the limit is exceeded, the excess data is silently discarded.

**Returns:**the current maximum column size limit; zero means that there is no limit **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs**See Also:**[setMaxFieldSize(int)](http://docs.google.com/javax/sql/RowSet.html#setMaxFieldSize(int))

### setMaxFieldSize

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

Sets the maximum number of bytes that can be returned for a column value to the given number of bytes. This limit applies only to BINARY, VARBINARY, LONGVARBINARYBINARY, CHAR, VARCHAR, LONGVARCHAR, NCHAR and NVARCHAR columns. If the limit is exceeded, the excess data is silently discarded. For maximum portability, use values greater than 256.

**Parameters:**max - the new max column size limit in bytes; zero means unlimited **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs**See Also:**[getMaxFieldSize()](http://docs.google.com/javax/sql/RowSet.html#getMaxFieldSize())

### getMaxRows

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

Retrieves the maximum number of rows that this RowSet object can contain. If the limit is exceeded, the excess rows are silently dropped.

**Returns:**the current maximum number of rows that this RowSet object can contain; zero means unlimited **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs**See Also:**[setMaxRows(int)](http://docs.google.com/javax/sql/RowSet.html#setMaxRows(int))

### setMaxRows

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

Sets the maximum number of rows that this RowSet object can contain to the specified number. If the limit is exceeded, the excess rows are silently dropped.

**Parameters:**max - the new maximum number of rows; zero means unlimited **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs**See Also:**[getMaxRows()](http://docs.google.com/javax/sql/RowSet.html#getMaxRows())

### getEscapeProcessing

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

Retrieves whether escape processing is enabled for this RowSet object. If escape scanning is enabled, which is the default, the driver will do escape substitution before sending an SQL statement to the database.

**Returns:**true if escape processing is enabled; false if it is disabled **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs**See Also:**[setEscapeProcessing(boolean)](http://docs.google.com/javax/sql/RowSet.html#setEscapeProcessing(boolean))

### setEscapeProcessing

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

Sets escape processing for this RowSet object on or off. If escape scanning is on (the default), the driver will do escape substitution before sending an SQL statement to the database.

**Parameters:**enable - true to enable escape processing; false to disable it **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs**See Also:**[getEscapeProcessing()](http://docs.google.com/javax/sql/RowSet.html#getEscapeProcessing())

### getQueryTimeout

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

Retrieves the maximum number of seconds the driver will wait for a statement to execute. If this limit is exceeded, an SQLException is thrown.

**Returns:**the current query timeout limit in seconds; zero means unlimited **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs**See Also:**[setQueryTimeout(int)](http://docs.google.com/javax/sql/RowSet.html#setQueryTimeout(int))

### setQueryTimeout

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

Sets the maximum time the driver will wait for a statement to execute to the given number of seconds. If this limit is exceeded, an SQLException is thrown.

**Parameters:**seconds - the new query timeout limit in seconds; zero means that there is no limit **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs**See Also:**[getQueryTimeout()](http://docs.google.com/javax/sql/RowSet.html#getQueryTimeout())

### setType

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

Sets the type of this RowSet object to the given type. This method is used to change the type of a rowset, which is by default read-only and non-scrollable.

**Parameters:**type - one of the ResultSet constants specifying a type: ResultSet.TYPE\_FORWARD\_ONLY, ResultSet.TYPE\_SCROLL\_INSENSITIVE, or ResultSet.TYPE\_SCROLL\_SENSITIVE **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs**See Also:**[ResultSet.getType()](http://docs.google.com/java/sql/ResultSet.html#getType())

### setConcurrency

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

Sets the concurrency of this RowSet object to the given concurrency level. This method is used to change the concurrency level of a rowset, which is by default ResultSet.CONCUR\_READ\_ONLY

**Parameters:**concurrency - one of the ResultSet constants specifying a concurrency level: ResultSet.CONCUR\_READ\_ONLY or ResultSet.CONCUR\_UPDATABLE **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs**See Also:**[ResultSet.getConcurrency()](http://docs.google.com/java/sql/ResultSet.html#getConcurrency())

### setNull

void **setNull**(int parameterIndex,  
 int sqlType)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Sets the designated parameter in this RowSet object's SQL command to SQL NULL.

**Note:** You must specify the parameter's SQL type.

**Parameters:**parameterIndex - the first parameter is 1, the second is 2, ...sqlType - a SQL type code defined by java.sql.Types **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs

### setNull

void **setNull**([String](http://docs.google.com/java/lang/String.html) parameterName,  
 int sqlType)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Sets the designated parameter to SQL NULL.

**Note:** You must specify the parameter's SQL type.

**Parameters:**parameterName - the name of the parametersqlType - the SQL type code defined in java.sql.Types **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs or this method is called on a closed CallableStatement [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.4

### setNull

void **setNull**(int paramIndex,  
 int sqlType,  
 [String](http://docs.google.com/java/lang/String.html) typeName)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Sets the designated parameter in this RowSet object's SQL command to SQL NULL. This version of the method setNull should be used for SQL user-defined types (UDTs) and REF type parameters. Examples of UDTs include: STRUCT, DISTINCT, JAVA\_OBJECT, and named array types.

**Note:** To be portable, applications must give the SQL type code and the fully qualified SQL type name when specifying a NULL UDT or REF parameter. In the case of a UDT, the name is the type name of the parameter itself. For a REF parameter, the name is the type name of the referenced type. If a JDBC driver does not need the type code or type name information, it may ignore it. Although it is intended for UDT and REF parameters, this method may be used to set a null parameter of any JDBC type. If the parameter does not have a user-defined or REF type, the typeName parameter is ignored.

**Parameters:**paramIndex - the first parameter is 1, the second is 2, ...sqlType - a value from java.sql.TypestypeName - the fully qualified name of an SQL UDT or the type name of the SQL structured type being referenced by a REF type; ignored if the parameter is not a UDT or REF type **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs

### setNull

void **setNull**([String](http://docs.google.com/java/lang/String.html) parameterName,  
 int sqlType,  
 [String](http://docs.google.com/java/lang/String.html) typeName)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Sets the designated parameter to SQL NULL. This version of the method setNull should be used for user-defined types and REF type parameters. Examples of user-defined types include: STRUCT, DISTINCT, JAVA\_OBJECT, and named array types.

**Note:** To be portable, applications must give the SQL type code and the fully-qualified SQL type name when specifying a NULL user-defined or REF parameter. In the case of a user-defined type the name is the type name of the parameter itself. For a REF parameter, the name is the type name of the referenced type. If a JDBC driver does not need the type code or type name information, it may ignore it. Although it is intended for user-defined and Ref parameters, this method may be used to set a null parameter of any JDBC type. If the parameter does not have a user-defined or REF type, the given typeName is ignored.

**Parameters:**parameterName - the name of the parametersqlType - a value from java.sql.TypestypeName - the fully-qualified name of an SQL user-defined type; ignored if the parameter is not a user-defined type or SQL REF value **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs or this method is called on a closed CallableStatement [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.4

### setBoolean

void **setBoolean**(int parameterIndex,  
 boolean x)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Sets the designated parameter in this RowSet object's command to the given Java boolean value. The driver converts this to an SQL BIT value before sending it to the database.

**Parameters:**parameterIndex - the first parameter is 1, the second is 2, ...x - the parameter value **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs

### setBoolean

void **setBoolean**([String](http://docs.google.com/java/lang/String.html) parameterName,  
 boolean x)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Sets the designated parameter to the given Java boolean value. The driver converts this to an SQL BIT or BOOLEAN value when it sends it to the database.

**Parameters:**parameterName - the name of the parameterx - the parameter value **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs or this method is called on a closed CallableStatement [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.4 **See Also:**[ResultSet.getBoolean(int)](http://docs.google.com/java/sql/ResultSet.html#getBoolean(int))

### setByte

void **setByte**(int parameterIndex,  
 byte x)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Sets the designated parameter in this RowSet object's command to the given Java byte value. The driver converts this to an SQL TINYINT value before sending it to the database.

**Parameters:**parameterIndex - the first parameter is 1, the second is 2, ...x - the parameter value **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs

### setByte

void **setByte**([String](http://docs.google.com/java/lang/String.html) parameterName,  
 byte x)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Sets the designated parameter to the given Java byte value. The driver converts this to an SQL TINYINT value when it sends it to the database.

**Parameters:**parameterName - the name of the parameterx - the parameter value **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs or this method is called on a closed CallableStatement [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.4 **See Also:**[ResultSet.getByte(int)](http://docs.google.com/java/sql/ResultSet.html#getByte(int))

### setShort

void **setShort**(int parameterIndex,  
 short x)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Sets the designated parameter in this RowSet object's command to the given Java short value. The driver converts this to an SQL SMALLINT value before sending it to the database.

**Parameters:**parameterIndex - the first parameter is 1, the second is 2, ...x - the parameter value **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs

### setShort

void **setShort**([String](http://docs.google.com/java/lang/String.html) parameterName,  
 short x)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Sets the designated parameter to the given Java short value. The driver converts this to an SQL SMALLINT value when it sends it to the database.

**Parameters:**parameterName - the name of the parameterx - the parameter value **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs or this method is called on a closed CallableStatement [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.4 **See Also:**[ResultSet.getShort(int)](http://docs.google.com/java/sql/ResultSet.html#getShort(int))

### setInt

void **setInt**(int parameterIndex,  
 int x)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Sets the designated parameter in this RowSet object's command to the given Java int value. The driver converts this to an SQL INTEGER value before sending it to the database.

**Parameters:**parameterIndex - the first parameter is 1, the second is 2, ...x - the parameter value **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs

### setInt

void **setInt**([String](http://docs.google.com/java/lang/String.html) parameterName,  
 int x)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Sets the designated parameter to the given Java int value. The driver converts this to an SQL INTEGER value when it sends it to the database.

**Parameters:**parameterName - the name of the parameterx - the parameter value **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs or this method is called on a closed CallableStatement [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.4 **See Also:**[ResultSet.getInt(int)](http://docs.google.com/java/sql/ResultSet.html#getInt(int))

### setLong

void **setLong**(int parameterIndex,  
 long x)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Sets the designated parameter in this RowSet object's command to the given Java long value. The driver converts this to an SQL BIGINT value before sending it to the database.

**Parameters:**parameterIndex - the first parameter is 1, the second is 2, ...x - the parameter value **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs

### setLong

void **setLong**([String](http://docs.google.com/java/lang/String.html) parameterName,  
 long x)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Sets the designated parameter to the given Java long value. The driver converts this to an SQL BIGINT value when it sends it to the database.

**Parameters:**parameterName - the name of the parameterx - the parameter value **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs or this method is called on a closed CallableStatement [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.4 **See Also:**[ResultSet.getLong(int)](http://docs.google.com/java/sql/ResultSet.html#getLong(int))

### setFloat

void **setFloat**(int parameterIndex,  
 float x)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Sets the designated parameter in this RowSet object's command to the given Java float value. The driver converts this to an SQL REAL value before sending it to the database.

**Parameters:**parameterIndex - the first parameter is 1, the second is 2, ...x - the parameter value **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs

### setFloat

void **setFloat**([String](http://docs.google.com/java/lang/String.html) parameterName,  
 float x)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Sets the designated parameter to the given Java float value. The driver converts this to an SQL FLOAT value when it sends it to the database.

**Parameters:**parameterName - the name of the parameterx - the parameter value **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs or this method is called on a closed CallableStatement [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.4 **See Also:**[ResultSet.getFloat(int)](http://docs.google.com/java/sql/ResultSet.html#getFloat(int))

### setDouble

void **setDouble**(int parameterIndex,  
 double x)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Sets the designated parameter in this RowSet object's command to the given Java double value. The driver converts this to an SQL DOUBLE value before sending it to the database.

**Parameters:**parameterIndex - the first parameter is 1, the second is 2, ...x - the parameter value **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs

### setDouble

void **setDouble**([String](http://docs.google.com/java/lang/String.html) parameterName,  
 double x)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Sets the designated parameter to the given Java double value. The driver converts this to an SQL DOUBLE value when it sends it to the database.

**Parameters:**parameterName - the name of the parameterx - the parameter value **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs or this method is called on a closed CallableStatement [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.4 **See Also:**[ResultSet.getDouble(int)](http://docs.google.com/java/sql/ResultSet.html#getDouble(int))

### setBigDecimal

void **setBigDecimal**(int parameterIndex,  
 [BigDecimal](http://docs.google.com/java/math/BigDecimal.html) x)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Sets the designated parameter in this RowSet object's command to the given java.math.BigDeciaml value. The driver converts this to an SQL NUMERIC value before sending it to the database.

**Parameters:**parameterIndex - the first parameter is 1, the second is 2, ...x - the parameter value **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs

### setBigDecimal

void **setBigDecimal**([String](http://docs.google.com/java/lang/String.html) parameterName,  
 [BigDecimal](http://docs.google.com/java/math/BigDecimal.html) x)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Sets the designated parameter to the given java.math.BigDecimal value. The driver converts this to an SQL NUMERIC value when it sends it to the database.

**Parameters:**parameterName - the name of the parameterx - the parameter value **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs or this method is called on a closed CallableStatement [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.4 **See Also:**[ResultSet.getBigDecimal(int, int)](http://docs.google.com/java/sql/ResultSet.html#getBigDecimal(int,%20int))

### setString

void **setString**(int parameterIndex,  
 [String](http://docs.google.com/java/lang/String.html) x)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Sets the designated parameter in this RowSet object's command to the given Java String value. Before sending it to the database, the driver converts this to an SQL VARCHAR or LONGVARCHAR value, depending on the argument's size relative to the driver's limits on VARCHAR values.

**Parameters:**parameterIndex - the first parameter is 1, the second is 2, ...x - the parameter value **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs

### setString

void **setString**([String](http://docs.google.com/java/lang/String.html) parameterName,  
 [String](http://docs.google.com/java/lang/String.html) x)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Sets the designated parameter to the given Java String value. The driver converts this to an SQL VARCHAR or LONGVARCHAR value (depending on the argument's size relative to the driver's limits on VARCHAR values) when it sends it to the database.

**Parameters:**parameterName - the name of the parameterx - the parameter value **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs or this method is called on a closed CallableStatement [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.4 **See Also:**[ResultSet.getString(int)](http://docs.google.com/java/sql/ResultSet.html#getString(int))

### setBytes

void **setBytes**(int parameterIndex,  
 byte[] x)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Sets the designated parameter in this RowSet object's command to the given Java array of byte values. Before sending it to the database, the driver converts this to an SQL VARBINARY or LONGVARBINARY value, depending on the argument's size relative to the driver's limits on VARBINARY values.

**Parameters:**parameterIndex - the first parameter is 1, the second is 2, ...x - the parameter value **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs

### setBytes

void **setBytes**([String](http://docs.google.com/java/lang/String.html) parameterName,  
 byte[] x)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Sets the designated parameter to the given Java array of bytes. The driver converts this to an SQL VARBINARY or LONGVARBINARY (depending on the argument's size relative to the driver's limits on VARBINARY values) when it sends it to the database.

**Parameters:**parameterName - the name of the parameterx - the parameter value **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs or this method is called on a closed CallableStatement [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.4 **See Also:**[ResultSet.getBytes(int)](http://docs.google.com/java/sql/ResultSet.html#getBytes(int))

### setDate

void **setDate**(int parameterIndex,  
 [Date](http://docs.google.com/java/sql/Date.html) x)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Sets the designated parameter in this RowSet object's command to the given java.sql.Date value. The driver converts this to an SQL DATE value before sending it to the database, using the default java.util.Calendar to calculate the date.

**Parameters:**parameterIndex - the first parameter is 1, the second is 2, ...x - the parameter value **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs

### setTime

void **setTime**(int parameterIndex,  
 [Time](http://docs.google.com/java/sql/Time.html) x)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Sets the designated parameter in this RowSet object's command to the given java.sql.Time value. The driver converts this to an SQL TIME value before sending it to the database, using the default java.util.Calendar to calculate it.

**Parameters:**parameterIndex - the first parameter is 1, the second is 2, ...x - the parameter value **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs

### setTimestamp

void **setTimestamp**(int parameterIndex,  
 [Timestamp](http://docs.google.com/java/sql/Timestamp.html) x)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Sets the designated parameter in this RowSet object's command to the given java.sql.Timestamp value. The driver converts this to an SQL TIMESTAMP value before sending it to the database, using the default java.util.Calendar to calculate it.

**Parameters:**parameterIndex - the first parameter is 1, the second is 2, ...x - the parameter value **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs

### setTimestamp

void **setTimestamp**([String](http://docs.google.com/java/lang/String.html) parameterName,  
 [Timestamp](http://docs.google.com/java/sql/Timestamp.html) x)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Sets the designated parameter to the given java.sql.Timestamp value. The driver converts this to an SQL TIMESTAMP value when it sends it to the database.

**Parameters:**parameterName - the name of the parameterx - the parameter value **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs or this method is called on a closed CallableStatement [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.4 **See Also:**[ResultSet.getTimestamp(int)](http://docs.google.com/java/sql/ResultSet.html#getTimestamp(int))

### setAsciiStream

void **setAsciiStream**(int parameterIndex,  
 [InputStream](http://docs.google.com/java/io/InputStream.html) x,  
 int length)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Sets the designated parameter in this RowSet object's command to the given java.io.InputStream value. It may be more practical to send a very large ASCII value via a java.io.InputStream rather than as a LONGVARCHAR parameter. The driver will read the data from the stream as needed until it reaches end-of-file.

**Note:** This stream object can either be a standard Java stream object or your own subclass that implements the standard interface.

**Parameters:**parameterIndex - the first parameter is 1, the second is 2, ...x - the Java input stream that contains the ASCII parameter valuelength - the number of bytes in the stream **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs

### setAsciiStream

void **setAsciiStream**([String](http://docs.google.com/java/lang/String.html) parameterName,  
 [InputStream](http://docs.google.com/java/io/InputStream.html) x,  
 int length)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Sets the designated parameter to the given input stream, which will have the specified number of bytes. When a very large ASCII value is input to a LONGVARCHAR parameter, it may be more practical to send it via a java.io.InputStream. Data will be read from the stream as needed until end-of-file is reached. The JDBC driver will do any necessary conversion from ASCII to the database char format.

**Note:** This stream object can either be a standard Java stream object or your own subclass that implements the standard interface.

**Parameters:**parameterName - the name of the parameterx - the Java input stream that contains the ASCII parameter valuelength - the number of bytes in the stream **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs or this method is called on a closed CallableStatement [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.4

### setBinaryStream

void **setBinaryStream**(int parameterIndex,  
 [InputStream](http://docs.google.com/java/io/InputStream.html) x,  
 int length)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Sets the designated parameter in this RowSet object's command to the given java.io.InputStream value. It may be more practical to send a very large binary value via a java.io.InputStream rather than as a LONGVARBINARY parameter. The driver will read the data from the stream as needed until it reaches end-of-file.

**Note:** This stream object can either be a standard Java stream object or your own subclass that implements the standard interface.

**Parameters:**parameterIndex - the first parameter is 1, the second is 2, ...x - the java input stream which contains the binary parameter valuelength - the number of bytes in the stream **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs

### setBinaryStream

void **setBinaryStream**([String](http://docs.google.com/java/lang/String.html) parameterName,  
 [InputStream](http://docs.google.com/java/io/InputStream.html) x,  
 int length)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Sets the designated parameter to the given input stream, which will have the specified number of bytes. When a very large binary value is input to a LONGVARBINARY parameter, it may be more practical to send it via a java.io.InputStream object. The data will be read from the stream as needed until end-of-file is reached.

**Note:** This stream object can either be a standard Java stream object or your own subclass that implements the standard interface.

**Parameters:**parameterName - the name of the parameterx - the java input stream which contains the binary parameter valuelength - the number of bytes in the stream **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs or this method is called on a closed CallableStatement [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.4

### setCharacterStream

void **setCharacterStream**(int parameterIndex,  
 [Reader](http://docs.google.com/java/io/Reader.html) reader,  
 int length)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Sets the designated parameter in this RowSet object's command to the given java.io.Reader value. It may be more practical to send a very large UNICODE value via a java.io.Reader rather than as a LONGVARCHAR parameter. The driver will read the data from the stream as needed until it reaches end-of-file.

**Note:** This stream object can either be a standard Java stream object or your own subclass that implements the standard interface.

**Parameters:**parameterIndex - the first parameter is 1, the second is 2, ...reader - the Reader object that contains the UNICODE data to be setlength - the number of characters in the stream **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs

### setCharacterStream

void **setCharacterStream**([String](http://docs.google.com/java/lang/String.html) parameterName,  
 [Reader](http://docs.google.com/java/io/Reader.html) reader,  
 int length)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Sets the designated parameter to the given Reader object, which is the given number of characters long. When a very large UNICODE value is input to a LONGVARCHAR parameter, it may be more practical to send it via a java.io.Reader object. The data will be read from the stream as needed until end-of-file is reached. The JDBC driver will do any necessary conversion from UNICODE to the database char format.

**Note:** This stream object can either be a standard Java stream object or your own subclass that implements the standard interface.

**Parameters:**parameterName - the name of the parameterreader - the java.io.Reader object that contains the UNICODE data used as the designated parameterlength - the number of characters in the stream **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs or this method is called on a closed CallableStatement [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.4

### setAsciiStream

void **setAsciiStream**(int parameterIndex,  
 [InputStream](http://docs.google.com/java/io/InputStream.html) x)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Sets the designated parameter in this RowSet object's command to the given input stream. When a very large ASCII value is input to a LONGVARCHAR parameter, it may be more practical to send it via a java.io.InputStream. Data will be read from the stream as needed until end-of-file is reached. The JDBC driver will do any necessary conversion from ASCII to the database char format.

**Note:** This stream object can either be a standard Java stream object or your own subclass that implements the standard interface.

**Note:** Consult your JDBC driver documentation to determine if it might be more efficient to use a version of setAsciiStream which takes a length parameter.

**Parameters:**parameterIndex - the first parameter is 1, the second is 2, ...x - the Java input stream that contains the ASCII parameter value **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs or this method is called on a closed PreparedStatement [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.6

### setAsciiStream

void **setAsciiStream**([String](http://docs.google.com/java/lang/String.html) parameterName,  
 [InputStream](http://docs.google.com/java/io/InputStream.html) x)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Sets the designated parameter to the given input stream. When a very large ASCII value is input to a LONGVARCHAR parameter, it may be more practical to send it via a java.io.InputStream. Data will be read from the stream as needed until end-of-file is reached. The JDBC driver will do any necessary conversion from ASCII to the database char format.

**Note:** This stream object can either be a standard Java stream object or your own subclass that implements the standard interface.

**Note:** Consult your JDBC driver documentation to determine if it might be more efficient to use a version of setAsciiStream which takes a length parameter.

**Parameters:**parameterName - the name of the parameterx - the Java input stream that contains the ASCII parameter value **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs or this method is called on a closed CallableStatement [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.6

### setBinaryStream

void **setBinaryStream**(int parameterIndex,  
 [InputStream](http://docs.google.com/java/io/InputStream.html) x)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Sets the designated parameter in this RowSet object's command to the given input stream. When a very large binary value is input to a LONGVARBINARY parameter, it may be more practical to send it via a java.io.InputStream object. The data will be read from the stream as needed until end-of-file is reached.

**Note:** This stream object can either be a standard Java stream object or your own subclass that implements the standard interface.

**Note:** Consult your JDBC driver documentation to determine if it might be more efficient to use a version of setBinaryStream which takes a length parameter.

**Parameters:**parameterIndex - the first parameter is 1, the second is 2, ...x - the java input stream which contains the binary parameter value **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs or this method is called on a closed PreparedStatement [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.6

### setBinaryStream

void **setBinaryStream**([String](http://docs.google.com/java/lang/String.html) parameterName,  
 [InputStream](http://docs.google.com/java/io/InputStream.html) x)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Sets the designated parameter to the given input stream. When a very large binary value is input to a LONGVARBINARY parameter, it may be more practical to send it via a java.io.InputStream object. The data will be read from the stream as needed until end-of-file is reached.

**Note:** This stream object can either be a standard Java stream object or your own subclass that implements the standard interface.

**Note:** Consult your JDBC driver documentation to determine if it might be more efficient to use a version of setBinaryStream which takes a length parameter.

**Parameters:**parameterName - the name of the parameterx - the java input stream which contains the binary parameter value **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs or this method is called on a closed CallableStatement [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.6

### setCharacterStream

void **setCharacterStream**(int parameterIndex,  
 [Reader](http://docs.google.com/java/io/Reader.html) reader)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Sets the designated parameter in this RowSet object's command to the given Reader object. When a very large UNICODE value is input to a LONGVARCHAR parameter, it may be more practical to send it via a java.io.Reader object. The data will be read from the stream as needed until end-of-file is reached. The JDBC driver will do any necessary conversion from UNICODE to the database char format.

**Note:** This stream object can either be a standard Java stream object or your own subclass that implements the standard interface.

**Note:** Consult your JDBC driver documentation to determine if it might be more efficient to use a version of setCharacterStream which takes a length parameter.

**Parameters:**parameterIndex - the first parameter is 1, the second is 2, ...reader - the java.io.Reader object that contains the Unicode data **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs or this method is called on a closed PreparedStatement [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.6

### setCharacterStream

void **setCharacterStream**([String](http://docs.google.com/java/lang/String.html) parameterName,  
 [Reader](http://docs.google.com/java/io/Reader.html) reader)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Sets the designated parameter to the given Reader object. When a very large UNICODE value is input to a LONGVARCHAR parameter, it may be more practical to send it via a java.io.Reader object. The data will be read from the stream as needed until end-of-file is reached. The JDBC driver will do any necessary conversion from UNICODE to the database char format.

**Note:** This stream object can either be a standard Java stream object or your own subclass that implements the standard interface.

**Note:** Consult your JDBC driver documentation to determine if it might be more efficient to use a version of setCharacterStream which takes a length parameter.

**Parameters:**parameterName - the name of the parameterreader - the java.io.Reader object that contains the Unicode data **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs or this method is called on a closed CallableStatement [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.6

### setNCharacterStream

void **setNCharacterStream**(int parameterIndex,  
 [Reader](http://docs.google.com/java/io/Reader.html) value)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Sets the designated parameter in this RowSet object's command to a Reader object. The Reader reads the data till end-of-file is reached. The driver does the necessary conversion from Java character format to the national character set in the database.

**Note:** This stream object can either be a standard Java stream object or your own subclass that implements the standard interface.

**Note:** Consult your JDBC driver documentation to determine if it might be more efficient to use a version of setNCharacterStream which takes a length parameter.

**Parameters:**parameterIndex - of the first parameter is 1, the second is 2, ...value - the parameter value **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if the driver does not support national character sets; if the driver can detect that a data conversion error could occur ; if a database access error occurs; or this method is called on a closed PreparedStatement [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.6

### setObject

void **setObject**(int parameterIndex,  
 [Object](http://docs.google.com/java/lang/Object.html) x,  
 int targetSqlType,  
 int scaleOrLength)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Sets the designated parameter in this RowSet object's command with the given Java Object. For integral values, the java.lang equivalent objects should be used (for example, an instance of the class Integer for an int). If the second argument is an InputStream then the stream must contain the number of bytes specified by scaleOrLength. If the second argument is a Reader then the reader must contain the number of characters specified \* by scaleOrLength. If these conditions are not true the driver will generate a SQLException when the prepared statement is executed.

The given Java object will be converted to the targetSqlType before being sent to the database.

If the object is of a class implementing SQLData, the rowset should call the method SQLData.writeSQL to write the object to an SQLOutput data stream. If, on the other hand, the object is of a class implementing Ref, Blob, Clob, NClob, Struct, java.net.URL, or Array, the driver should pass it to the database as a value of the corresponding SQL type.

Note that this method may be used to pass datatabase-specific abstract data types.

**Parameters:**parameterIndex - the first parameter is 1, the second is 2, ...x - the object containing the input parameter valuetargetSqlType - the SQL type (as defined in java.sql.Types) to be sent to the database. The scale argument may further qualify this type.scaleOrLength - for java.sql.Types.DECIMAL or java.sql.Types.NUMERIC types, this is the number of digits after the decimal point. For Java Object types InputStream and Reader, this is the length of the data in the stream or reader. For all other types, this value will be ignored. **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs**See Also:**[Types](http://docs.google.com/java/sql/Types.html)

### setObject

void **setObject**([String](http://docs.google.com/java/lang/String.html) parameterName,  
 [Object](http://docs.google.com/java/lang/Object.html) x,  
 int targetSqlType,  
 int scale)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Sets the value of the designated parameter with the given object. The second argument must be an object type; for integral values, the java.lang equivalent objects should be used.

The given Java object will be converted to the given targetSqlType before being sent to the database. If the object has a custom mapping (is of a class implementing the interface SQLData), the JDBC driver should call the method SQLData.writeSQL to write it to the SQL data stream. If, on the other hand, the object is of a class implementing Ref, Blob, Clob, NClob, Struct, java.net.URL, or Array, the driver should pass it to the database as a value of the corresponding SQL type.

Note that this method may be used to pass datatabase- specific abstract data types.

**Parameters:**parameterName - the name of the parameterx - the object containing the input parameter valuetargetSqlType - the SQL type (as defined in java.sql.Types) to be sent to the database. The scale argument may further qualify this type.scale - for java.sql.Types.DECIMAL or java.sql.Types.NUMERIC types, this is the number of digits after the decimal point. For all other types, this value will be ignored. **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs or this method is called on a closed CallableStatement [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if targetSqlType is a ARRAY, BLOB, CLOB, DATALINK, JAVA\_OBJECT, NCHAR, NCLOB, NVARCHAR, LONGNVARCHAR, REF, ROWID, SQLXML or STRUCT data type and the JDBC driver does not support this data type**Since:** 1.4 **See Also:**[Types](http://docs.google.com/java/sql/Types.html), [ResultSet.getObject(int)](http://docs.google.com/java/sql/ResultSet.html#getObject(int))

### setObject

void **setObject**(int parameterIndex,  
 [Object](http://docs.google.com/java/lang/Object.html) x,  
 int targetSqlType)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Sets the designated parameter in this RowSet object's command with a Java Object. For integral values, the java.lang equivalent objects should be used. This method is like setObject above, but the scale used is the scale of the second parameter. Scalar values have a scale of zero. Literal values have the scale present in the literal.

Even though it is supported, it is not recommended that this method be called with floating point input values.

**Parameters:**parameterIndex - the first parameter is 1, the second is 2, ...x - the object containing the input parameter valuetargetSqlType - the SQL type (as defined in java.sql.Types) to be sent to the database **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs

### setObject

void **setObject**([String](http://docs.google.com/java/lang/String.html) parameterName,  
 [Object](http://docs.google.com/java/lang/Object.html) x,  
 int targetSqlType)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Sets the value of the designated parameter with the given object. This method is like the method setObject above, except that it assumes a scale of zero.

**Parameters:**parameterName - the name of the parameterx - the object containing the input parameter valuetargetSqlType - the SQL type (as defined in java.sql.Types) to be sent to the database **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs or this method is called on a closed CallableStatement [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if targetSqlType is a ARRAY, BLOB, CLOB, DATALINK, JAVA\_OBJECT, NCHAR, NCLOB, NVARCHAR, LONGNVARCHAR, REF, ROWID, SQLXML or STRUCT data type and the JDBC driver does not support this data type**Since:** 1.4 **See Also:**[ResultSet.getObject(int)](http://docs.google.com/java/sql/ResultSet.html#getObject(int))

### setObject

void **setObject**([String](http://docs.google.com/java/lang/String.html) parameterName,  
 [Object](http://docs.google.com/java/lang/Object.html) x)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Sets the value of the designated parameter with the given object. The second parameter must be of type Object; therefore, the java.lang equivalent objects should be used for built-in types.

The JDBC specification specifies a standard mapping from Java Object types to SQL types. The given argument will be converted to the corresponding SQL type before being sent to the database.

Note that this method may be used to pass datatabase- specific abstract data types, by using a driver-specific Java type. If the object is of a class implementing the interface SQLData, the JDBC driver should call the method SQLData.writeSQL to write it to the SQL data stream. If, on the other hand, the object is of a class implementing Ref, Blob, Clob, NClob, Struct, java.net.URL, or Array, the driver should pass it to the database as a value of the corresponding SQL type.

This method throws an exception if there is an ambiguity, for example, if the object is of a class implementing more than one of the interfaces named above.

**Parameters:**parameterName - the name of the parameterx - the object containing the input parameter value **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs, this method is called on a closed CallableStatement or if the given Object parameter is ambiguous [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.4 **See Also:**[ResultSet.getObject(int)](http://docs.google.com/java/sql/ResultSet.html#getObject(int))

### setObject

void **setObject**(int parameterIndex,  
 [Object](http://docs.google.com/java/lang/Object.html) x)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Sets the designated parameter in this RowSet object's command with a Java Object. For integral values, the java.lang equivalent objects should be used.

The JDBC specification provides a standard mapping from Java Object types to SQL types. The driver will convert the given Java object to its standard SQL mapping before sending it to the database.

Note that this method may be used to pass datatabase-specific abstract data types by using a driver-specific Java type. If the object is of a class implementing SQLData, the rowset should call the method SQLData.writeSQL to write the object to an SQLOutput data stream. If, on the other hand, the object is of a class implementing Ref, Blob, Clob, NClob, Struct, java.net.URL, or Array, the driver should pass it to the database as a value of the corresponding SQL type.

An exception is thrown if there is an ambiguity, for example, if the object is of a class implementing more than one of these interfaces.

**Parameters:**parameterIndex - The first parameter is 1, the second is 2, ...x - The object containing the input parameter value **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs

### setRef

void **setRef**(int i,  
 [Ref](http://docs.google.com/java/sql/Ref.html) x)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Sets the designated parameter in this RowSet object's command with the given Ref value. The driver will convert this to the appropriate REF(<structured-type>) value.

**Parameters:**i - the first parameter is 1, the second is 2, ...x - an object representing data of an SQL REF type **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs

### setBlob

void **setBlob**(int i,  
 [Blob](http://docs.google.com/java/sql/Blob.html) x)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Sets the designated parameter in this RowSet object's command with the given Blob value. The driver will convert this to the BLOB value that the Blob object represents before sending it to the database.

**Parameters:**i - the first parameter is 1, the second is 2, ...x - an object representing a BLOB **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs

### setBlob

void **setBlob**(int parameterIndex,  
 [InputStream](http://docs.google.com/java/io/InputStream.html) inputStream,  
 long length)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Sets the designated parameter to a InputStream object. The inputstream must contain the number of characters specified by length otherwise a SQLException will be generated when the PreparedStatement is executed. This method differs from the setBinaryStream (int, InputStream, int) method because it informs the driver that the parameter value should be sent to the server as a BLOB. When the setBinaryStream method is used, the driver may have to do extra work to determine whether the parameter data should be sent to the server as a LONGVARBINARY or a BLOB

**Parameters:**parameterIndex - index of the first parameter is 1, the second is 2, ...inputStream - An object that contains the data to set the parameter value to.length - the number of bytes in the parameter data. **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs, this method is called on a closed PreparedStatement, if parameterIndex does not correspond to a parameter marker in the SQL statement, if the length specified is less than zero or if the number of bytes in the inputstream does not match the specfied length. [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.6

### setBlob

void **setBlob**(int parameterIndex,  
 [InputStream](http://docs.google.com/java/io/InputStream.html) inputStream)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Sets the designated parameter to a InputStream object. This method differs from the setBinaryStream (int, InputStream) method because it informs the driver that the parameter value should be sent to the server as a BLOB. When the setBinaryStream method is used, the driver may have to do extra work to determine whether the parameter data should be sent to the server as a LONGVARBINARY or a BLOB

**Note:** Consult your JDBC driver documentation to determine if it might be more efficient to use a version of setBlob which takes a length parameter.

**Parameters:**parameterIndex - index of the first parameter is 1, the second is 2, ...inputStream - An object that contains the data to set the parameter value to. **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs, this method is called on a closed PreparedStatement or if parameterIndex does not correspond to a parameter marker in the SQL statement, [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.6

### setBlob

void **setBlob**([String](http://docs.google.com/java/lang/String.html) parameterName,  
 [InputStream](http://docs.google.com/java/io/InputStream.html) inputStream,  
 long length)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Sets the designated parameter to a InputStream object. The inputstream must contain the number of characters specified by length, otherwise a SQLException will be generated when the CallableStatement is executed. This method differs from the setBinaryStream (int, InputStream, int) method because it informs the driver that the parameter value should be sent to the server as a BLOB. When the setBinaryStream method is used, the driver may have to do extra work to determine whether the parameter data should be sent to the server as a LONGVARBINARY or a BLOB

**Parameters:**parameterName - the name of the parameter to be set the second is 2, ...inputStream - An object that contains the data to set the parameter value to.length - the number of bytes in the parameter data. **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if parameterIndex does not correspond to a parameter marker in the SQL statement, or if the length specified is less than zero; if the number of bytes in the inputstream does not match the specfied length; if a database access error occurs or this method is called on a closed CallableStatement [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.6

### setBlob

void **setBlob**([String](http://docs.google.com/java/lang/String.html) parameterName,  
 [Blob](http://docs.google.com/java/sql/Blob.html) x)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Sets the designated parameter to the given java.sql.Blob object. The driver converts this to an SQL BLOB value when it sends it to the database.

**Parameters:**parameterName - the name of the parameterx - a Blob object that maps an SQL BLOB value **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs or this method is called on a closed CallableStatement [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.6

### setBlob

void **setBlob**([String](http://docs.google.com/java/lang/String.html) parameterName,  
 [InputStream](http://docs.google.com/java/io/InputStream.html) inputStream)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Sets the designated parameter to a InputStream object. This method differs from the setBinaryStream (int, InputStream) method because it informs the driver that the parameter value should be sent to the server as a BLOB. When the setBinaryStream method is used, the driver may have to do extra work to determine whether the parameter data should be send to the server as a LONGVARBINARY or a BLOB

**Note:** Consult your JDBC driver documentation to determine if it might be more efficient to use a version of setBlob which takes a length parameter.

**Parameters:**parameterName - the name of the parameterinputStream - An object that contains the data to set the parameter value to. **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs or this method is called on a closed CallableStatement [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.6

### setClob

void **setClob**(int i,  
 [Clob](http://docs.google.com/java/sql/Clob.html) x)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Sets the designated parameter in this RowSet object's command with the given Clob value. The driver will convert this to the CLOB value that the Clob object represents before sending it to the database.

**Parameters:**i - the first parameter is 1, the second is 2, ...x - an object representing a CLOB **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs

### setClob

void **setClob**(int parameterIndex,  
 [Reader](http://docs.google.com/java/io/Reader.html) reader,  
 long length)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Sets the designated parameter to a Reader object. The reader must contain the number of characters specified by length otherwise a SQLException will be generated when the PreparedStatement is executed. This method differs from the setCharacterStream (int, Reader, int) method because it informs the driver that the parameter value should be sent to the server as a CLOB. When the setCharacterStream method is used, the driver may have to do extra work to determine whether the parameter data should be sent to the server as a LONGVARCHAR or a CLOB

**Parameters:**parameterIndex - index of the first parameter is 1, the second is 2, ...reader - An object that contains the data to set the parameter value to.length - the number of characters in the parameter data. **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs, this method is called on a closed PreparedStatement, if parameterIndex does not correspond to a parameter marker in the SQL statement, or if the length specified is less than zero. [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.6

### setClob

void **setClob**(int parameterIndex,  
 [Reader](http://docs.google.com/java/io/Reader.html) reader)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Sets the designated parameter to a Reader object. This method differs from the setCharacterStream (int, Reader) method because it informs the driver that the parameter value should be sent to the server as a CLOB. When the setCharacterStream method is used, the driver may have to do extra work to determine whether the parameter data should be sent to the server as a LONGVARCHAR or a CLOB

**Note:** Consult your JDBC driver documentation to determine if it might be more efficient to use a version of setClob which takes a length parameter.

**Parameters:**parameterIndex - index of the first parameter is 1, the second is 2, ...reader - An object that contains the data to set the parameter value to. **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs, this method is called on a closed PreparedStatementor if parameterIndex does not correspond to a parameter marker in the SQL statement [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.6

### setClob

void **setClob**([String](http://docs.google.com/java/lang/String.html) parameterName,  
 [Reader](http://docs.google.com/java/io/Reader.html) reader,  
 long length)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Sets the designated parameter to a Reader object. The reader must contain the number of characters specified by length otherwise a SQLException will be generated when the CallableStatement is executed. This method differs from the setCharacterStream (int, Reader, int) method because it informs the driver that the parameter value should be sent to the server as a CLOB. When the setCharacterStream method is used, the driver may have to do extra work to determine whether the parameter data should be send to the server as a LONGVARCHAR or a CLOB

**Parameters:**parameterName - the name of the parameter to be setreader - An object that contains the data to set the parameter value to.length - the number of characters in the parameter data. **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if parameterIndex does not correspond to a parameter marker in the SQL statement; if the length specified is less than zero; a database access error occurs or this method is called on a closed CallableStatement [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.6

### setClob

void **setClob**([String](http://docs.google.com/java/lang/String.html) parameterName,  
 [Clob](http://docs.google.com/java/sql/Clob.html) x)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Sets the designated parameter to the given java.sql.Clob object. The driver converts this to an SQL CLOB value when it sends it to the database.

**Parameters:**parameterName - the name of the parameterx - a Clob object that maps an SQL CLOB value **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs or this method is called on a closed CallableStatement [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.6

### setClob

void **setClob**([String](http://docs.google.com/java/lang/String.html) parameterName,  
 [Reader](http://docs.google.com/java/io/Reader.html) reader)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Sets the designated parameter to a Reader object. This method differs from the setCharacterStream (int, Reader) method because it informs the driver that the parameter value should be sent to the server as a CLOB. When the setCharacterStream method is used, the driver may have to do extra work to determine whether the parameter data should be send to the server as a LONGVARCHAR or a CLOB

**Note:** Consult your JDBC driver documentation to determine if it might be more efficient to use a version of setClob which takes a length parameter.

**Parameters:**parameterName - the name of the parameterreader - An object that contains the data to set the parameter value to. **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs or this method is called on a closed CallableStatement [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.6

### setArray

void **setArray**(int i,  
 [Array](http://docs.google.com/java/sql/Array.html) x)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Sets the designated parameter in this RowSet object's command with the given Array value. The driver will convert this to the ARRAY value that the Array object represents before sending it to the database.

**Parameters:**i - the first parameter is 1, the second is 2, ...x - an object representing an SQL array **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs

### setDate

void **setDate**(int parameterIndex,  
 [Date](http://docs.google.com/java/sql/Date.html) x,  
 [Calendar](http://docs.google.com/java/util/Calendar.html) cal)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Sets the designated parameter in this RowSet object's command with the given java.sql.Date value. The driver will convert this to an SQL DATE value, using the given java.util.Calendar object to calculate the date.

**Parameters:**parameterIndex - the first parameter is 1, the second is 2, ...x - the parameter valuecal - the java.util.Calendar object to use for calculating the date **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs

### setDate

void **setDate**([String](http://docs.google.com/java/lang/String.html) parameterName,  
 [Date](http://docs.google.com/java/sql/Date.html) x)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Sets the designated parameter to the given java.sql.Date value using the default time zone of the virtual machine that is running the application. The driver converts this to an SQL DATE value when it sends it to the database.

**Parameters:**parameterName - the name of the parameterx - the parameter value **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs or this method is called on a closed CallableStatement [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.4 **See Also:**[ResultSet.getDate(int)](http://docs.google.com/java/sql/ResultSet.html#getDate(int))

### setDate

void **setDate**([String](http://docs.google.com/java/lang/String.html) parameterName,  
 [Date](http://docs.google.com/java/sql/Date.html) x,  
 [Calendar](http://docs.google.com/java/util/Calendar.html) cal)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Sets the designated parameter to the given java.sql.Date value, using the given Calendar object. The driver uses the Calendar object to construct an SQL DATE value, which the driver then sends to the database. With a a Calendar object, the driver can calculate the date taking into account a custom timezone. If no Calendar object is specified, the driver uses the default timezone, which is that of the virtual machine running the application.

**Parameters:**parameterName - the name of the parameterx - the parameter valuecal - the Calendar object the driver will use to construct the date **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs or this method is called on a closed CallableStatement [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.4 **See Also:**[ResultSet.getDate(int)](http://docs.google.com/java/sql/ResultSet.html#getDate(int))

### setTime

void **setTime**(int parameterIndex,  
 [Time](http://docs.google.com/java/sql/Time.html) x,  
 [Calendar](http://docs.google.com/java/util/Calendar.html) cal)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Sets the designated parameter in this RowSet object's command with the given java.sql.Time value. The driver will convert this to an SQL TIME value, using the given java.util.Calendar object to calculate it, before sending it to the database.

**Parameters:**parameterIndex - the first parameter is 1, the second is 2, ...x - the parameter valuecal - the java.util.Calendar object to use for calculating the time **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs

### setTime

void **setTime**([String](http://docs.google.com/java/lang/String.html) parameterName,  
 [Time](http://docs.google.com/java/sql/Time.html) x)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Sets the designated parameter to the given java.sql.Time value. The driver converts this to an SQL TIME value when it sends it to the database.

**Parameters:**parameterName - the name of the parameterx - the parameter value **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs or this method is called on a closed CallableStatement [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.4 **See Also:**[ResultSet.getTime(int)](http://docs.google.com/java/sql/ResultSet.html#getTime(int))

### setTime

void **setTime**([String](http://docs.google.com/java/lang/String.html) parameterName,  
 [Time](http://docs.google.com/java/sql/Time.html) x,  
 [Calendar](http://docs.google.com/java/util/Calendar.html) cal)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Sets the designated parameter to the given java.sql.Time value, using the given Calendar object. The driver uses the Calendar object to construct an SQL TIME value, which the driver then sends to the database. With a a Calendar object, the driver can calculate the time taking into account a custom timezone. If no Calendar object is specified, the driver uses the default timezone, which is that of the virtual machine running the application.

**Parameters:**parameterName - the name of the parameterx - the parameter valuecal - the Calendar object the driver will use to construct the time **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs or this method is called on a closed CallableStatement [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.4 **See Also:**[ResultSet.getTime(int)](http://docs.google.com/java/sql/ResultSet.html#getTime(int))

### setTimestamp

void **setTimestamp**(int parameterIndex,  
 [Timestamp](http://docs.google.com/java/sql/Timestamp.html) x,  
 [Calendar](http://docs.google.com/java/util/Calendar.html) cal)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Sets the designated parameter in this RowSet object's command with the given java.sql.Timestamp value. The driver will convert this to an SQL TIMESTAMP value, using the given java.util.Calendar object to calculate it, before sending it to the database.

**Parameters:**parameterIndex - the first parameter is 1, the second is 2, ...x - the parameter valuecal - the java.util.Calendar object to use for calculating the timestamp **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs

### setTimestamp

void **setTimestamp**([String](http://docs.google.com/java/lang/String.html) parameterName,  
 [Timestamp](http://docs.google.com/java/sql/Timestamp.html) x,  
 [Calendar](http://docs.google.com/java/util/Calendar.html) cal)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Sets the designated parameter to the given java.sql.Timestamp value, using the given Calendar object. The driver uses the Calendar object to construct an SQL TIMESTAMP value, which the driver then sends to the database. With a a Calendar object, the driver can calculate the timestamp taking into account a custom timezone. If no Calendar object is specified, the driver uses the default timezone, which is that of the virtual machine running the application.

**Parameters:**parameterName - the name of the parameterx - the parameter valuecal - the Calendar object the driver will use to construct the timestamp **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs or this method is called on a closed CallableStatement [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.4 **See Also:**[ResultSet.getTimestamp(int)](http://docs.google.com/java/sql/ResultSet.html#getTimestamp(int))

### clearParameters

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

Clears the parameters set for this RowSet object's command.

In general, parameter values remain in force for repeated use of a RowSet object. Setting a parameter value automatically clears its previous value. However, in some cases it is useful to immediately release the resources used by the current parameter values, which can be done by calling the method clearParameters.

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

### execute

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

Fills this RowSet object with data.

The execute method may use the following properties to create a connection for reading data: url, data source name, user name, password, transaction isolation, and type map. The execute method may use the following properties to create a statement to execute a command: command, read only, maximum field size, maximum rows, escape processing, and query timeout.

If the required properties have not been set, an exception is thrown. If this method is successful, the current contents of the rowset are discarded and the rowset's metadata is also (re)set. If there are outstanding updates, they are ignored.

If this RowSet object does not maintain a continuous connection with its source of data, it may use a reader (a RowSetReader object) to fill itself with data. In this case, a reader will have been registered with this RowSet object, and the method execute will call on the reader's readData method as part of its implementation.

**Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs or any of the properties necessary for making a connection and creating a statement have not been set

### addRowSetListener

void **addRowSetListener**([RowSetListener](http://docs.google.com/javax/sql/RowSetListener.html) listener)

Registers the given listener so that it will be notified of events that occur on this RowSet object.

**Parameters:**listener - a component that has implemented the RowSetListener interface and wants to be notified when events occur on this RowSet object**See Also:**[removeRowSetListener(javax.sql.RowSetListener)](http://docs.google.com/javax/sql/RowSet.html#removeRowSetListener(javax.sql.RowSetListener))

### removeRowSetListener

void **removeRowSetListener**([RowSetListener](http://docs.google.com/javax/sql/RowSetListener.html) listener)

Removes the specified listener from the list of components that will be notified when an event occurs on this RowSet object.

**Parameters:**listener - a component that has been registered as a listener for this RowSet object**See Also:**[addRowSetListener(javax.sql.RowSetListener)](http://docs.google.com/javax/sql/RowSet.html#addRowSetListener(javax.sql.RowSetListener))

### setSQLXML

void **setSQLXML**(int parameterIndex,  
 [SQLXML](http://docs.google.com/java/sql/SQLXML.html) xmlObject)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Sets the designated parameter to the given java.sql.SQLXML object. The driver converts this to an SQL XML value when it sends it to the database.

**Parameters:**parameterIndex - index of the first parameter is 1, the second is 2, ...xmlObject - a SQLXML object that maps an SQL XML value **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs, this method is called on a closed result set, the java.xml.transform.Result, Writer or OutputStream has not been closed for the SQLXML object or if there is an error processing the XML value. The getCause method of the exception may provide a more detailed exception, for example, if the stream does not contain valid XML.**Since:** 1.6

### setSQLXML

void **setSQLXML**([String](http://docs.google.com/java/lang/String.html) parameterName,  
 [SQLXML](http://docs.google.com/java/sql/SQLXML.html) xmlObject)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Sets the designated parameter to the given java.sql.SQLXML object. The driver converts this to an SQL XML value when it sends it to the database.

**Parameters:**parameterName - the name of the parameterxmlObject - a SQLXML object that maps an SQL XML value **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs, this method is called on a closed result set, the java.xml.transform.Result, Writer or OutputStream has not been closed for the SQLXML object or if there is an error processing the XML value. The getCause method of the exception may provide a more detailed exception, for example, if the stream does not contain valid XML.**Since:** 1.6

### setRowId

void **setRowId**(int parameterIndex,  
 [RowId](http://docs.google.com/java/sql/RowId.html) x)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Sets the designated parameter to the given java.sql.RowId object. The driver converts this to a SQL ROWID value when it sends it to the database

**Parameters:**parameterIndex - the first parameter is 1, the second is 2, ...x - the parameter value **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs**Since:** 1.6

### setRowId

void **setRowId**([String](http://docs.google.com/java/lang/String.html) parameterName,  
 [RowId](http://docs.google.com/java/sql/RowId.html) x)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Sets the designated parameter to the given java.sql.RowId object. The driver converts this to a SQL ROWID when it sends it to the database.

**Parameters:**parameterName - the name of the parameterx - the parameter value **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs**Since:** 1.6

### setNString

void **setNString**(int parameterIndex,  
 [String](http://docs.google.com/java/lang/String.html) value)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Sets the designated paramter to the given String object. The driver converts this to a SQL NCHAR or NVARCHAR or LONGNVARCHAR value (depending on the argument's size relative to the driver's limits on NVARCHAR values) when it sends it to the database.

**Parameters:**parameterIndex - of the first parameter is 1, the second is 2, ...value - the parameter value **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if the driver does not support national character sets; if the driver can detect that a data conversion error could occur ; or if a database access error occurs**Since:** 1.6

### setNString

void **setNString**([String](http://docs.google.com/java/lang/String.html) parameterName,  
 [String](http://docs.google.com/java/lang/String.html) value)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Sets the designated paramter to the given String object. The driver converts this to a SQL NCHAR or NVARCHAR or LONGNVARCHAR

**Parameters:**parameterName - the name of the column to be setvalue - the parameter value **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if the driver does not support national character sets; if the driver can detect that a data conversion error could occur; or if a database access error occurs**Since:** 1.6

### setNCharacterStream

void **setNCharacterStream**(int parameterIndex,  
 [Reader](http://docs.google.com/java/io/Reader.html) value,  
 long length)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Sets the designated parameter to a Reader object. The Reader reads the data till end-of-file is reached. The driver does the necessary conversion from Java character format to the national character set in the database.

**Parameters:**parameterIndex - of the first parameter is 1, the second is 2, ...value - the parameter valuelength - the number of characters in the parameter data. **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if the driver does not support national character sets; if the driver can detect that a data conversion error could occur ; or if a database access error occurs**Since:** 1.6

### setNCharacterStream

void **setNCharacterStream**([String](http://docs.google.com/java/lang/String.html) parameterName,  
 [Reader](http://docs.google.com/java/io/Reader.html) value,  
 long length)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Sets the designated parameter to a Reader object. The Reader reads the data till end-of-file is reached. The driver does the necessary conversion from Java character format to the national character set in the database.

**Parameters:**parameterName - the name of the column to be setvalue - the parameter valuelength - the number of characters in the parameter data. **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if the driver does not support national character sets; if the driver can detect that a data conversion error could occur; or if a database access error occurs**Since:** 1.6

### setNCharacterStream

void **setNCharacterStream**([String](http://docs.google.com/java/lang/String.html) parameterName,  
 [Reader](http://docs.google.com/java/io/Reader.html) value)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Sets the designated parameter to a Reader object. The Reader reads the data till end-of-file is reached. The driver does the necessary conversion from Java character format to the national character set in the database.

**Note:** This stream object can either be a standard Java stream object or your own subclass that implements the standard interface.

**Note:** Consult your JDBC driver documentation to determine if it might be more efficient to use a version of setNCharacterStream which takes a length parameter.

**Parameters:**parameterName - the name of the parametervalue - the parameter value **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if the driver does not support national character sets; if the driver can detect that a data conversion error could occur ; if a database access error occurs; or this method is called on a closed CallableStatement [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.6

### setNClob

void **setNClob**([String](http://docs.google.com/java/lang/String.html) parameterName,  
 [NClob](http://docs.google.com/java/sql/NClob.html) value)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Sets the designated parameter to a java.sql.NClob object. The object implements the java.sql.NClob interface. This NClob object maps to a SQL NCLOB.

**Parameters:**parameterName - the name of the column to be setvalue - the parameter value **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if the driver does not support national character sets; if the driver can detect that a data conversion error could occur; or if a database access error occurs**Since:** 1.6

### setNClob

void **setNClob**([String](http://docs.google.com/java/lang/String.html) parameterName,  
 [Reader](http://docs.google.com/java/io/Reader.html) reader,  
 long length)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Sets the designated parameter to a Reader object. The reader must contain the number of characters specified by length otherwise a SQLException will be generated when the CallableStatement is executed. This method differs from the setCharacterStream (int, Reader, int) method because it informs the driver that the parameter value should be sent to the server as a NCLOB. When the setCharacterStream method is used, the driver may have to do extra work to determine whether the parameter data should be send to the server as a LONGNVARCHAR or a NCLOB

**Parameters:**parameterName - the name of the parameter to be setreader - An object that contains the data to set the parameter value to.length - the number of characters in the parameter data. **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if parameterIndex does not correspond to a parameter marker in the SQL statement; if the length specified is less than zero; if the driver does not support national character sets; if the driver can detect that a data conversion error could occur; if a database access error occurs or this method is called on a closed CallableStatement [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.6

### setNClob

void **setNClob**([String](http://docs.google.com/java/lang/String.html) parameterName,  
 [Reader](http://docs.google.com/java/io/Reader.html) reader)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Sets the designated parameter to a Reader object. This method differs from the setCharacterStream (int, Reader) method because it informs the driver that the parameter value should be sent to the server as a NCLOB. When the setCharacterStream method is used, the driver may have to do extra work to determine whether the parameter data should be send to the server as a LONGNVARCHAR or a NCLOB

**Note:** Consult your JDBC driver documentation to determine if it might be more efficient to use a version of setNClob which takes a length parameter.

**Parameters:**parameterName - the name of the parameterreader - An object that contains the data to set the parameter value to. **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if the driver does not support national character sets; if the driver can detect that a data conversion error could occur; if a database access error occurs or this method is called on a closed CallableStatement [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.6

### setNClob

void **setNClob**(int parameterIndex,  
 [Reader](http://docs.google.com/java/io/Reader.html) reader,  
 long length)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Sets the designated parameter to a Reader object. The reader must contain the number of characters specified by length otherwise a SQLException will be generated when the PreparedStatement is executed. This method differs from the setCharacterStream (int, Reader, int) method because it informs the driver that the parameter value should be sent to the server as a NCLOB. When the setCharacterStream method is used, the driver may have to do extra work to determine whether the parameter data should be sent to the server as a LONGNVARCHAR or a NCLOB

**Parameters:**parameterIndex - index of the first parameter is 1, the second is 2, ...reader - An object that contains the data to set the parameter value to.length - the number of characters in the parameter data. **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if parameterIndex does not correspond to a parameter marker in the SQL statement; if the length specified is less than zero; if the driver does not support national character sets; if the driver can detect that a data conversion error could occur; if a database access error occurs or this method is called on a closed PreparedStatement [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.6

### setNClob

void **setNClob**(int parameterIndex,  
 [NClob](http://docs.google.com/java/sql/NClob.html) value)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Sets the designated parameter to a java.sql.NClob object. The driver converts this to a SQL NCLOB value when it sends it to the database.

**Parameters:**parameterIndex - of the first parameter is 1, the second is 2, ...value - the parameter value **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if the driver does not support national character sets; if the driver can detect that a data conversion error could occur ; or if a database access error occurs**Since:** 1.6

### setNClob

void **setNClob**(int parameterIndex,  
 [Reader](http://docs.google.com/java/io/Reader.html) reader)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Sets the designated parameter to a Reader object. This method differs from the setCharacterStream (int, Reader) method because it informs the driver that the parameter value should be sent to the server as a NCLOB. When the setCharacterStream method is used, the driver may have to do extra work to determine whether the parameter data should be sent to the server as a LONGNVARCHAR or a NCLOB

**Note:** Consult your JDBC driver documentation to determine if it might be more efficient to use a version of setNClob which takes a length parameter.

**Parameters:**parameterIndex - index of the first parameter is 1, the second is 2, ...reader - An object that contains the data to set the parameter value to. **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if parameterIndex does not correspond to a parameter marker in the SQL statement; if the driver does not support national character sets; if the driver can detect that a data conversion error could occur; if a database access error occurs or this method is called on a closed PreparedStatement [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.6

### setURL

void **setURL**(int parameterIndex,  
 [URL](http://docs.google.com/java/net/URL.html) x)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Sets the designated parameter to the given java.net.URL value. The driver converts this to an SQL DATALINK value when it sends it to the database.

**Parameters:**parameterIndex - the first parameter is 1, the second is 2, ...x - the java.net.URL object to be set **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs or this method is called on a closed PreparedStatement [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.4

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/RowSet.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/PooledConnection.html)   [**NEXT CLASS**](http://docs.google.com/javax/sql/RowSetEvent.html) | [**FRAMES**](http://docs.google.com/index.html?javax/sql/RowSet.html)    [**NO FRAMES**](http://docs.google.com/RowSet.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | CONSTR | [METHOD](#tyjcwt) | DETAIL: FIELD | CONSTR | [METHOD](#4d34og8) |

[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).