| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/SQLInput.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/java/sql/SQLFeatureNotSupportedException.html)   [**NEXT CLASS**](http://docs.google.com/java/sql/SQLIntegrityConstraintViolationException.html) | [**FRAMES**](http://docs.google.com/index.html?java/sql/SQLInput.html)    [**NO FRAMES**](http://docs.google.com/SQLInput.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | CONSTR | [METHOD](#3znysh7) | DETAIL: FIELD | CONSTR | [METHOD](#2et92p0) |

## **java.sql**

Interface SQLInput

**All Known Implementing Classes:** [SQLInputImpl](http://docs.google.com/javax/sql/rowset/serial/SQLInputImpl.html)

public interface **SQLInput**

An input stream that contains a stream of values representing an instance of an SQL structured type or an SQL distinct type. This interface, used only for custom mapping, is used by the driver behind the scenes, and a programmer never directly invokes SQLInput methods. The *reader* methods (readLong, readBytes, and so on) provide a way for an implementation of the SQLData interface to read the values in an SQLInput object. And as described in SQLData, calls to reader methods must be made in the order that their corresponding attributes appear in the SQL definition of the type. The method wasNull is used to determine whether the last value read was SQL NULL.

When the method getObject is called with an object of a class implementing the interface SQLData, the JDBC driver calls the method SQLData.getSQLType to determine the SQL type of the user-defined type (UDT) being custom mapped. The driver creates an instance of SQLInput, populating it with the attributes of the UDT. The driver then passes the input stream to the method SQLData.readSQL, which in turn calls the SQLInput reader methods in its implementation for reading the attributes from the input stream.

**Since:** 1.2

| **Method Summary** | |
| --- | --- |
| [Array](http://docs.google.com/java/sql/Array.html) | [**readArray**](http://docs.google.com/java/sql/SQLInput.html#readArray())()            Reads an SQL ARRAY value from the stream and returns it as an Array object in the Java programming language. |
| [InputStream](http://docs.google.com/java/io/InputStream.html) | [**readAsciiStream**](http://docs.google.com/java/sql/SQLInput.html#readAsciiStream())()            Reads the next attribute in the stream and returns it as a stream of ASCII characters. |
| [BigDecimal](http://docs.google.com/java/math/BigDecimal.html) | [**readBigDecimal**](http://docs.google.com/java/sql/SQLInput.html#readBigDecimal())()            Reads the next attribute in the stream and returns it as a java.math.BigDecimal object in the Java programming language. |
| [InputStream](http://docs.google.com/java/io/InputStream.html) | [**readBinaryStream**](http://docs.google.com/java/sql/SQLInput.html#readBinaryStream())()            Reads the next attribute in the stream and returns it as a stream of uninterpreted bytes. |
| [Blob](http://docs.google.com/java/sql/Blob.html) | [**readBlob**](http://docs.google.com/java/sql/SQLInput.html#readBlob())()            Reads an SQL BLOB value from the stream and returns it as a Blob object in the Java programming language. |
| boolean | [**readBoolean**](http://docs.google.com/java/sql/SQLInput.html#readBoolean())()            Reads the next attribute in the stream and returns it as a boolean in the Java programming language. |
| byte | [**readByte**](http://docs.google.com/java/sql/SQLInput.html#readByte())()            Reads the next attribute in the stream and returns it as a byte in the Java programming language. |
| byte[] | [**readBytes**](http://docs.google.com/java/sql/SQLInput.html#readBytes())()            Reads the next attribute in the stream and returns it as an array of bytes in the Java programming language. |
| [Reader](http://docs.google.com/java/io/Reader.html) | [**readCharacterStream**](http://docs.google.com/java/sql/SQLInput.html#readCharacterStream())()            Reads the next attribute in the stream and returns it as a stream of Unicode characters. |
| [Clob](http://docs.google.com/java/sql/Clob.html) | [**readClob**](http://docs.google.com/java/sql/SQLInput.html#readClob())()            Reads an SQL CLOB value from the stream and returns it as a Clob object in the Java programming language. |
| [Date](http://docs.google.com/java/sql/Date.html) | [**readDate**](http://docs.google.com/java/sql/SQLInput.html#readDate())()            Reads the next attribute in the stream and returns it as a java.sql.Date object. |
| double | [**readDouble**](http://docs.google.com/java/sql/SQLInput.html#readDouble())()            Reads the next attribute in the stream and returns it as a double in the Java programming language. |
| float | [**readFloat**](http://docs.google.com/java/sql/SQLInput.html#readFloat())()            Reads the next attribute in the stream and returns it as a float in the Java programming language. |
| int | [**readInt**](http://docs.google.com/java/sql/SQLInput.html#readInt())()            Reads the next attribute in the stream and returns it as an int in the Java programming language. |
| long | [**readLong**](http://docs.google.com/java/sql/SQLInput.html#readLong())()            Reads the next attribute in the stream and returns it as a long in the Java programming language. |
| [NClob](http://docs.google.com/java/sql/NClob.html) | [**readNClob**](http://docs.google.com/java/sql/SQLInput.html#readNClob())()            Reads an SQL NCLOB value from the stream and returns it as a NClob object in the Java programming language. |
| [String](http://docs.google.com/java/lang/String.html) | [**readNString**](http://docs.google.com/java/sql/SQLInput.html#readNString())()            Reads the next attribute in the stream and returns it as a String in the Java programming language. |
| [Object](http://docs.google.com/java/lang/Object.html) | [**readObject**](http://docs.google.com/java/sql/SQLInput.html#readObject())()            Reads the datum at the head of the stream and returns it as an Object in the Java programming language. |
| [Ref](http://docs.google.com/java/sql/Ref.html) | [**readRef**](http://docs.google.com/java/sql/SQLInput.html#readRef())()            Reads an SQL REF value from the stream and returns it as a Ref object in the Java programming language. |
| [RowId](http://docs.google.com/java/sql/RowId.html) | [**readRowId**](http://docs.google.com/java/sql/SQLInput.html#readRowId())()            Reads an SQL ROWID value from the stream and returns it as a RowId object in the Java programming language. |
| short | [**readShort**](http://docs.google.com/java/sql/SQLInput.html#readShort())()            Reads the next attribute in the stream and returns it as a short in the Java programming language. |
| [SQLXML](http://docs.google.com/java/sql/SQLXML.html) | [**readSQLXML**](http://docs.google.com/java/sql/SQLInput.html#readSQLXML())()            Reads an SQL XML value from the stream and returns it as a SQLXML object in the Java programming language. |
| [String](http://docs.google.com/java/lang/String.html) | [**readString**](http://docs.google.com/java/sql/SQLInput.html#readString())()            Reads the next attribute in the stream and returns it as a String in the Java programming language. |
| [Time](http://docs.google.com/java/sql/Time.html) | [**readTime**](http://docs.google.com/java/sql/SQLInput.html#readTime())()            Reads the next attribute in the stream and returns it as a java.sql.Time object. |
| [Timestamp](http://docs.google.com/java/sql/Timestamp.html) | [**readTimestamp**](http://docs.google.com/java/sql/SQLInput.html#readTimestamp())()            Reads the next attribute in the stream and returns it as a java.sql.Timestamp object. |
| [URL](http://docs.google.com/java/net/URL.html) | [**readURL**](http://docs.google.com/java/sql/SQLInput.html#readURL())()            Reads an SQL DATALINK value from the stream and returns it as a java.net.URL object in the Java programming language. |
| boolean | [**wasNull**](http://docs.google.com/java/sql/SQLInput.html#wasNull())()            Retrieves whether the last value read was SQL NULL. |

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

### readString

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

Reads the next attribute in the stream and returns it as a String in the Java programming language.

**Returns:**the attribute; if the value is SQL NULL, returns null **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.2

### readBoolean

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

Reads the next attribute in the stream and returns it as a boolean in the Java programming language.

**Returns:**the attribute; if the value is SQL NULL, returns false **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.2

### readByte

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

Reads the next attribute in the stream and returns it as a byte in the Java programming language.

**Returns:**the attribute; if the value is SQL NULL, returns 0 **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.2

### readShort

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

Reads the next attribute in the stream and returns it as a short in the Java programming language.

**Returns:**the attribute; if the value is SQL NULL, returns 0 **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.2

### readInt

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

Reads the next attribute in the stream and returns it as an int in the Java programming language.

**Returns:**the attribute; if the value is SQL NULL, returns 0 **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.2

### readLong

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

Reads the next attribute in the stream and returns it as a long in the Java programming language.

**Returns:**the attribute; if the value is SQL NULL, returns 0 **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.2

### readFloat

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

Reads the next attribute in the stream and returns it as a float in the Java programming language.

**Returns:**the attribute; if the value is SQL NULL, returns 0 **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.2

### readDouble

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

Reads the next attribute in the stream and returns it as a double in the Java programming language.

**Returns:**the attribute; if the value is SQL NULL, returns 0 **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.2

### readBigDecimal

[BigDecimal](http://docs.google.com/java/math/BigDecimal.html) **readBigDecimal**()  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Reads the next attribute in the stream and returns it as a java.math.BigDecimal object in the Java programming language.

**Returns:**the attribute; if the value is SQL NULL, returns null **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.2

### readBytes

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

Reads the next attribute in the stream and returns it as an array of bytes in the Java programming language.

**Returns:**the attribute; if the value is SQL NULL, returns null **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.2

### readDate

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

Reads the next attribute in the stream and returns it as a java.sql.Date object.

**Returns:**the attribute; if the value is SQL NULL, returns null **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.2

### readTime

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

Reads the next attribute in the stream and returns it as a java.sql.Time object.

**Returns:**the attribute; if the value is SQL NULL, returns null **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.2

### readTimestamp

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

Reads the next attribute in the stream and returns it as a java.sql.Timestamp object.

**Returns:**the attribute; if the value is SQL NULL, returns null **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.2

### readCharacterStream

[Reader](http://docs.google.com/java/io/Reader.html) **readCharacterStream**()  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Reads the next attribute in the stream and returns it as a stream of Unicode characters.

**Returns:**the attribute; if the value is SQL NULL, returns null **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.2

### readAsciiStream

[InputStream](http://docs.google.com/java/io/InputStream.html) **readAsciiStream**()  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Reads the next attribute in the stream and returns it as a stream of ASCII characters.

**Returns:**the attribute; if the value is SQL NULL, returns null **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.2

### readBinaryStream

[InputStream](http://docs.google.com/java/io/InputStream.html) **readBinaryStream**()  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Reads the next attribute in the stream and returns it as a stream of uninterpreted bytes.

**Returns:**the attribute; if the value is SQL NULL, returns null **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.2

### readObject

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

Reads the datum at the head of the stream and returns it as an Object in the Java programming language. The actual type of the object returned is determined by the default type mapping, and any customizations present in this stream's type map.

A type map is registered with the stream by the JDBC driver before the stream is passed to the application.

When the datum at the head of the stream is an SQL NULL, the method returns null. If the datum is an SQL structured or distinct type, it determines the SQL type of the datum at the head of the stream. If the stream's type map has an entry for that SQL type, the driver constructs an object of the appropriate class and calls the method SQLData.readSQL on that object, which reads additional data from the stream, using the protocol described for that method.

**Returns:**the datum at the head of the stream as an Object in the Java programming language;null if the datum is SQL NULL **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.2

### readRef

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

Reads an SQL REF value from the stream and returns it as a Ref object in the Java programming language.

**Returns:**a Ref object representing the SQL REF value at the head of the stream; null if the value read is SQL NULL **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.2

### readBlob

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

Reads an SQL BLOB value from the stream and returns it as a Blob object in the Java programming language.

**Returns:**a Blob object representing data of the SQL BLOB value at the head of the stream; null if the value read is SQL NULL **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.2

### readClob

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

Reads an SQL CLOB value from the stream and returns it as a Clob object in the Java programming language.

**Returns:**a Clob object representing data of the SQL CLOB value at the head of the stream; null if the value read is SQL NULL **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.2

### readArray

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

Reads an SQL ARRAY value from the stream and returns it as an Array object in the Java programming language.

**Returns:**an Array object representing data of the SQL ARRAY value at the head of the stream; null if the value read is SQL NULL **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.2

### wasNull

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

Retrieves whether the last value read was SQL NULL.

**Returns:**true if the most recently read SQL value was SQL NULL; false otherwise **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.2

### readURL

[URL](http://docs.google.com/java/net/URL.html) **readURL**()  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Reads an SQL DATALINK value from the stream and returns it as a java.net.URL object in the Java programming language.

**Returns:**a java.net.URL object. **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs, or if a URL is malformed [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.4

### readNClob

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

Reads an SQL NCLOB value from the stream and returns it as a NClob object in the Java programming language.

**Returns:**a NClob object representing data of the SQL NCLOB value at the head of the stream; null if the value read is SQL NULL **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.6

### readNString

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

Reads the next attribute in the stream and returns it as a String in the Java programming language. It is intended for use when accessing NCHAR,NVARCHAR and LONGNVARCHAR columns.

**Returns:**the attribute; if the value is SQL NULL, returns null **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.6

### readSQLXML

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

Reads an SQL XML value from the stream and returns it as a SQLXML object in the Java programming language.

**Returns:**a SQLXML object representing data of the SQL XML value at the head of the stream; null if the value read is SQL NULL **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.6

### readRowId

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

Reads an SQL ROWID value from the stream and returns it as a RowId object in the Java programming language.

**Returns:**a RowId object representing data of the SQL ROWID value at the head of the stream; null if the value read is SQL NULL **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.6

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/SQLInput.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/java/sql/SQLFeatureNotSupportedException.html)   [**NEXT CLASS**](http://docs.google.com/java/sql/SQLIntegrityConstraintViolationException.html) | [**FRAMES**](http://docs.google.com/index.html?java/sql/SQLInput.html)    [**NO FRAMES**](http://docs.google.com/SQLInput.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | CONSTR | [METHOD](#3znysh7) | DETAIL: FIELD | CONSTR | [METHOD](#2et92p0) |

[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).