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

## **java.sql**

Interface Clob

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

public interface **Clob**

The mapping in the JavaTM programming language for the SQL CLOB type. An SQL CLOB is a built-in type that stores a Character Large Object as a column value in a row of a database table. By default drivers implement a Clob object using an SQL locator(CLOB), which means that a Clob object contains a logical pointer to the SQL CLOB data rather than the data itself. A Clob object is valid for the duration of the transaction in which it was created.

The Clob interface provides methods for getting the length of an SQL CLOB (Character Large Object) value, for materializing a CLOB value on the client, and for searching for a substring or CLOB object within a CLOB value. Methods in the interfaces [ResultSet](http://docs.google.com/java/sql/ResultSet.html), [CallableStatement](http://docs.google.com/java/sql/CallableStatement.html), and [PreparedStatement](http://docs.google.com/java/sql/PreparedStatement.html), such as getClob and setClob allow a programmer to access an SQL CLOB value. In addition, this interface has methods for updating a CLOB value.

All methods on the Clob interface must be fully implemented if the JDBC driver supports the data type.

**Since:** 1.2

| **Method Summary** | |
| --- | --- |
| void | [**free**](http://docs.google.com/java/sql/Clob.html#free())()            This method frees the Clob object and releases the resources the resources that it holds. |
| [InputStream](http://docs.google.com/java/io/InputStream.html) | [**getAsciiStream**](http://docs.google.com/java/sql/Clob.html#getAsciiStream())()            Retrieves the CLOB value designated by this Clob object as an ascii stream. |
| [Reader](http://docs.google.com/java/io/Reader.html) | [**getCharacterStream**](http://docs.google.com/java/sql/Clob.html#getCharacterStream())()            Retrieves the CLOB value designated by this Clob object as a java.io.Reader object (or as a stream of characters). |
| [Reader](http://docs.google.com/java/io/Reader.html) | [**getCharacterStream**](http://docs.google.com/java/sql/Clob.html#getCharacterStream(long,%20long))(long pos, long length)            Returns a Reader object that contains a partial Clob value, starting with the character specified by pos, which is length characters in length. |
| [String](http://docs.google.com/java/lang/String.html) | [**getSubString**](http://docs.google.com/java/sql/Clob.html#getSubString(long,%20int))(long pos, int length)            Retrieves a copy of the specified substring in the CLOB value designated by this Clob object. |
| long | [**length**](http://docs.google.com/java/sql/Clob.html#length())()            Retrieves the number of characters in the CLOB value designated by this Clob object. |
| long | [**position**](http://docs.google.com/java/sql/Clob.html#position(java.sql.Clob,%20long))([Clob](http://docs.google.com/java/sql/Clob.html) searchstr, long start)            Retrieves the character position at which the specified Clob object searchstr appears in this Clob object. |
| long | [**position**](http://docs.google.com/java/sql/Clob.html#position(java.lang.String,%20long))([String](http://docs.google.com/java/lang/String.html) searchstr, long start)            Retrieves the character position at which the specified substring searchstr appears in the SQL CLOB value represented by this Clob object. |
| [OutputStream](http://docs.google.com/java/io/OutputStream.html) | [**setAsciiStream**](http://docs.google.com/java/sql/Clob.html#setAsciiStream(long))(long pos)            Retrieves a stream to be used to write Ascii characters to the CLOB value that this Clob object represents, starting at position pos. |
| [Writer](http://docs.google.com/java/io/Writer.html) | [**setCharacterStream**](http://docs.google.com/java/sql/Clob.html#setCharacterStream(long))(long pos)            Retrieves a stream to be used to write a stream of Unicode characters to the CLOB value that this Clob object represents, at position pos. |
| int | [**setString**](http://docs.google.com/java/sql/Clob.html#setString(long,%20java.lang.String))(long pos, [String](http://docs.google.com/java/lang/String.html) str)            Writes the given Java String to the CLOB value that this Clob object designates at the position pos. |
| int | [**setString**](http://docs.google.com/java/sql/Clob.html#setString(long,%20java.lang.String,%20int,%20int))(long pos, [String](http://docs.google.com/java/lang/String.html) str, int offset, int len)            Writes len characters of str, starting at character offset, to the CLOB value that this Clob represents. |
| void | [**truncate**](http://docs.google.com/java/sql/Clob.html#truncate(long))(long len)            Truncates the CLOB value that this Clob designates to have a length of len characters. |

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

### length

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

Retrieves the number of characters in the CLOB value designated by this Clob object.

**Returns:**length of the CLOB in characters **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if there is an error accessing the length of the CLOB value [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.2

### getSubString

[String](http://docs.google.com/java/lang/String.html) **getSubString**(long pos,  
 int length)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Retrieves a copy of the specified substring in the CLOB value designated by this Clob object. The substring begins at position pos and has up to length consecutive characters.

**Parameters:**pos - the first character of the substring to be extracted. The first character is at position 1.length - the number of consecutive characters to be copied; the value for length must be 0 or greater **Returns:**a String that is the specified substring in the CLOB value designated by this Clob object **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if there is an error accessing the CLOB value; if pos is less than 1 or length is less than 0 [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.2

### getCharacterStream

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

Retrieves the CLOB value designated by this Clob object as a java.io.Reader object (or as a stream of characters).

**Returns:**a java.io.Reader object containing the CLOB data **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if there is an error accessing the CLOB value [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.2 **See Also:**[setCharacterStream(long)](http://docs.google.com/java/sql/Clob.html#setCharacterStream(long))

### getAsciiStream

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

Retrieves the CLOB value designated by this Clob object as an ascii stream.

**Returns:**a java.io.InputStream object containing the CLOB data **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if there is an error accessing the CLOB value [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.2 **See Also:**[setAsciiStream(long)](http://docs.google.com/java/sql/Clob.html#setAsciiStream(long))

### position

long **position**([String](http://docs.google.com/java/lang/String.html) searchstr,  
 long start)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Retrieves the character position at which the specified substring searchstr appears in the SQL CLOB value represented by this Clob object. The search begins at position start.

**Parameters:**searchstr - the substring for which to searchstart - the position at which to begin searching; the first position is 1 **Returns:**the position at which the substring appears or -1 if it is not present; the first position is 1 **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if there is an error accessing the CLOB value or if pos is less than 1 [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.2

### position

long **position**([Clob](http://docs.google.com/java/sql/Clob.html) searchstr,  
 long start)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Retrieves the character position at which the specified Clob object searchstr appears in this Clob object. The search begins at position start.

**Parameters:**searchstr - the Clob object for which to searchstart - the position at which to begin searching; the first position is 1 **Returns:**the position at which the Clob object appears or -1 if it is not present; the first position is 1 **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if there is an error accessing the CLOB value or if start is less than 1 [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.2

### setString

int **setString**(long pos,  
 [String](http://docs.google.com/java/lang/String.html) str)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Writes the given Java String to the CLOB value that this Clob object designates at the position pos. The string will overwrite the existing characters in the Clob object starting at the position pos. If the end of the Clob value is reached while writing the given string, then the length of the Clob value will be increased to accomodate the extra characters.

**Note:** If the value specified for pos is greater then the length+1 of the CLOB value then the behavior is undefined. Some JDBC drivers may throw a SQLException while other drivers may support this operation.

**Parameters:**pos - the position at which to start writing to the CLOB value that this Clob object represents; The first position is 1str - the string to be written to the CLOB value that this Clob designates **Returns:**the number of characters written **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if there is an error accessing the CLOB value or if pos is less than 1 [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.4

### setString

int **setString**(long pos,  
 [String](http://docs.google.com/java/lang/String.html) str,  
 int offset,  
 int len)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Writes len characters of str, starting at character offset, to the CLOB value that this Clob represents. The string will overwrite the existing characters in the Clob object starting at the position pos. If the end of the Clob value is reached while writing the given string, then the length of the Clob value will be increased to accomodate the extra characters.

**Note:** If the value specified for pos is greater then the length+1 of the CLOB value then the behavior is undefined. Some JDBC drivers may throw a SQLException while other drivers may support this operation.

**Parameters:**pos - the position at which to start writing to this CLOB object; The first position is 1str - the string to be written to the CLOB value that this Clob object representsoffset - the offset into str to start reading the characters to be writtenlen - the number of characters to be written **Returns:**the number of characters written **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if there is an error accessing the CLOB value or if pos is less than 1 [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.4

### setAsciiStream

[OutputStream](http://docs.google.com/java/io/OutputStream.html) **setAsciiStream**(long pos)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Retrieves a stream to be used to write Ascii characters to the CLOB value that this Clob object represents, starting at position pos. Characters written to the stream will overwrite the existing characters in the Clob object starting at the position pos. If the end of the Clob value is reached while writing characters to the stream, then the length of the Clob value will be increased to accomodate the extra characters.

**Note:** If the value specified for pos is greater then the length+1 of the CLOB value then the behavior is undefined. Some JDBC drivers may throw a SQLException while other drivers may support this operation.

**Parameters:**pos - the position at which to start writing to this CLOB object; The first position is 1 **Returns:**the stream to which ASCII encoded characters can be written **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if there is an error accessing the CLOB value or if pos is less than 1 [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.4 **See Also:**[getAsciiStream()](http://docs.google.com/java/sql/Clob.html#getAsciiStream())

### setCharacterStream

[Writer](http://docs.google.com/java/io/Writer.html) **setCharacterStream**(long pos)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Retrieves a stream to be used to write a stream of Unicode characters to the CLOB value that this Clob object represents, at position pos. Characters written to the stream will overwrite the existing characters in the Clob object starting at the position pos. If the end of the Clob value is reached while writing characters to the stream, then the length of the Clob value will be increased to accomodate the extra characters.

**Note:** If the value specified for pos is greater then the length+1 of the CLOB value then the behavior is undefined. Some JDBC drivers may throw a SQLException while other drivers may support this operation.

**Parameters:**pos - the position at which to start writing to the CLOB value; The first position is 1 **Returns:**a stream to which Unicode encoded characters can be written **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if there is an error accessing the CLOB value or if pos is less than 1 [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.4 **See Also:**[getCharacterStream()](http://docs.google.com/java/sql/Clob.html#getCharacterStream())

### truncate

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

Truncates the CLOB value that this Clob designates to have a length of len characters.

**Note:** If the value specified for pos is greater then the length+1 of the CLOB value then the behavior is undefined. Some JDBC drivers may throw a SQLException while other drivers may support this operation.

**Parameters:**len - the length, in characters, to which the CLOB value should be truncated **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if there is an error accessing the CLOB value or if len is less than 0 [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.4

### free

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

This method frees the Clob object and releases the resources the resources that it holds. The object is invalid once the free method is called.

After free has been called, any attempt to invoke a method other than free will result in a SQLException being thrown. If free is called multiple times, the subsequent calls to free are treated as a no-op.

**Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if an error occurs releasing the Clob's resources [SQLFeatureNotSupportedException](http://docs.google.com/java/sql/SQLFeatureNotSupportedException.html) - if the JDBC driver does not support this method**Since:** 1.6

### getCharacterStream

[Reader](http://docs.google.com/java/io/Reader.html) **getCharacterStream**(long pos,  
 long length)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Returns a Reader object that contains a partial Clob value, starting with the character specified by pos, which is length characters in length.

**Parameters:**pos - the offset to the first character of the partial value to be retrieved. The first character in the Clob is at position 1.length - the length in characters of the partial value to be retrieved. **Returns:**Reader through which the partial Clob value can be read. **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if pos is less than 1 or if pos is greater than the number of characters in the Clob or if pos + length is greater than the number of characters in the Clob [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/Clob.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/ClientInfoStatus.html)   [**NEXT CLASS**](http://docs.google.com/java/sql/Connection.html) | [**FRAMES**](http://docs.google.com/index.html?java/sql/Clob.html)    [**NO FRAMES**](http://docs.google.com/Clob.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).