| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/SerialClob.html) | [**Tree**](http://docs.google.com/package-tree.html) | [**Deprecated**](http://docs.google.com/deprecated-list.html) | [**Index**](http://docs.google.com/index-files/index-1.html) | [**Help**](http://docs.google.com/help-doc.html) | | --- | --- | --- | --- | --- | --- | --- | --- | | | ***Java™ Platform***  ***Standard Ed. 6*** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [**PREV CLASS**](http://docs.google.com/javax/sql/rowset/serial/SerialBlob.html)   [**NEXT CLASS**](http://docs.google.com/javax/sql/rowset/serial/SerialDatalink.html) | [**FRAMES**](http://docs.google.com/index.html?javax/sql/rowset/serial/SerialClob.html)    [**NO FRAMES**](http://docs.google.com/SerialClob.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | [CONSTR](#3znysh7) | [METHOD](#2et92p0) | DETAIL: FIELD | [CONSTR](#3dy6vkm) | [METHOD](#2s8eyo1) |

## **javax.sql.rowset.serial**

Class SerialClob

[java.lang.Object](http://docs.google.com/java/lang/Object.html)  
 **javax.sql.rowset.serial.SerialClob**

**All Implemented Interfaces:** [Serializable](http://docs.google.com/java/io/Serializable.html), [Cloneable](http://docs.google.com/java/lang/Cloneable.html), [Clob](http://docs.google.com/java/sql/Clob.html)

public class **SerialClob**extends [Object](http://docs.google.com/java/lang/Object.html)implements [Clob](http://docs.google.com/java/sql/Clob.html), [Serializable](http://docs.google.com/java/io/Serializable.html), [Cloneable](http://docs.google.com/java/lang/Cloneable.html)

A serialized mapping in the Java programming language of an SQL CLOB value.

The SerialClob class provides a constructor for creating an instance from a Clob object. Note that the Clob object should have brought the SQL CLOB value's data over to the client before a SerialClob object is constructed from it. The data of an SQL CLOB value can be materialized on the client as a stream of Unicode characters.

SerialClob methods make it possible to get a substring from a SerialClob object or to locate the start of a pattern of characters.

**See Also:**[Serialized Form](http://docs.google.com/serialized-form.html#javax.sql.rowset.serial.SerialClob)

| **Constructor Summary** | |
| --- | --- |
| [**SerialClob**](http://docs.google.com/javax/sql/rowset/serial/SerialClob.html#SerialClob(char%5B%5D))(char[] ch)            Constructs a SerialClob object that is a serialized version of the given char array. |
| [**SerialClob**](http://docs.google.com/javax/sql/rowset/serial/SerialClob.html#SerialClob(java.sql.Clob))([Clob](http://docs.google.com/java/sql/Clob.html) clob)            Constructs a SerialClob object that is a serialized version of the given Clob object. |

| **Method Summary** | |
| --- | --- |
| void | [**free**](http://docs.google.com/javax/sql/rowset/serial/SerialClob.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/javax/sql/rowset/serial/SerialClob.html#getAsciiStream())()            Retrieves the CLOB value designated by this SerialClob object as an ascii stream. |
| [Reader](http://docs.google.com/java/io/Reader.html) | [**getCharacterStream**](http://docs.google.com/javax/sql/rowset/serial/SerialClob.html#getCharacterStream())()            Returns this SerialClob object's data as a stream of Unicode characters. |
| [Reader](http://docs.google.com/java/io/Reader.html) | [**getCharacterStream**](http://docs.google.com/javax/sql/rowset/serial/SerialClob.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/javax/sql/rowset/serial/SerialClob.html#getSubString(long,%20int))(long pos, int length)            Returns a copy of the substring contained in this SerialClob object, starting at the given position and continuing for the specified number or characters. |
| long | [**length**](http://docs.google.com/javax/sql/rowset/serial/SerialClob.html#length())()            Retrieves the number of characters in this SerialClob object's array of characters. |
| long | [**position**](http://docs.google.com/javax/sql/rowset/serial/SerialClob.html#position(java.sql.Clob,%20long))([Clob](http://docs.google.com/java/sql/Clob.html) searchStr, long start)            Returns the position in this SerialClob object where the given Clob signature begins, starting the search at the specified position. |
| long | [**position**](http://docs.google.com/javax/sql/rowset/serial/SerialClob.html#position(java.lang.String,%20long))([String](http://docs.google.com/java/lang/String.html) searchStr, long start)            Returns the position in this SerialClob object where the given String object begins, starting the search at the specified position. |
| [OutputStream](http://docs.google.com/java/io/OutputStream.html) | [**setAsciiStream**](http://docs.google.com/javax/sql/rowset/serial/SerialClob.html#setAsciiStream(long))(long pos)            Retrieves a stream to be used to write Ascii characters to the CLOB value that this SerialClob object represents, starting at position pos. |
| [Writer](http://docs.google.com/java/io/Writer.html) | [**setCharacterStream**](http://docs.google.com/javax/sql/rowset/serial/SerialClob.html#setCharacterStream(long))(long pos)            Retrieves a stream to be used to write a stream of Unicode characters to the CLOB value that this SerialClob object represents, at position pos. |
| int | [**setString**](http://docs.google.com/javax/sql/rowset/serial/SerialClob.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 SerialClob object represents, at the position pos. |
| int | [**setString**](http://docs.google.com/javax/sql/rowset/serial/SerialClob.html#setString(long,%20java.lang.String,%20int,%20int))(long pos, [String](http://docs.google.com/java/lang/String.html) str, int offset, int length)            Writes len characters of str, starting at character offset, to the CLOB value that this Clob represents. |
| void | [**truncate**](http://docs.google.com/javax/sql/rowset/serial/SerialClob.html#truncate(long))(long length)            Truncates the CLOB value that this SerialClob object represents so that it has a length of len characters. |

| **Methods inherited from class java.lang.**[**Object**](http://docs.google.com/java/lang/Object.html) |
| --- |
| [clone](http://docs.google.com/java/lang/Object.html#clone()), [equals](http://docs.google.com/java/lang/Object.html#equals(java.lang.Object)), [finalize](http://docs.google.com/java/lang/Object.html#finalize()), [getClass](http://docs.google.com/java/lang/Object.html#getClass()), [hashCode](http://docs.google.com/java/lang/Object.html#hashCode()), [notify](http://docs.google.com/java/lang/Object.html#notify()), [notifyAll](http://docs.google.com/java/lang/Object.html#notifyAll()), [toString](http://docs.google.com/java/lang/Object.html#toString()), [wait](http://docs.google.com/java/lang/Object.html#wait()), [wait](http://docs.google.com/java/lang/Object.html#wait(long)), [wait](http://docs.google.com/java/lang/Object.html#wait(long,%20int)) |

| **Constructor Detail** |
| --- |

### SerialClob

public **SerialClob**(char[] ch)  
 throws [SerialException](http://docs.google.com/javax/sql/rowset/serial/SerialException.html),  
 [SQLException](http://docs.google.com/java/sql/SQLException.html)

Constructs a SerialClob object that is a serialized version of the given char array.

The new SerialClob object is initialized with the data from the char array, thus allowing disconnected RowSet objects to establish a serialized Clob object without touching the data source.

**Parameters:**ch - the char array representing the Clob object to be serialized **Throws:** [SerialException](http://docs.google.com/javax/sql/rowset/serial/SerialException.html) - if an error occurs during serialization [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a SQL error occurs

### SerialClob

public **SerialClob**([Clob](http://docs.google.com/java/sql/Clob.html) clob)  
 throws [SerialException](http://docs.google.com/javax/sql/rowset/serial/SerialException.html),  
 [SQLException](http://docs.google.com/java/sql/SQLException.html)

Constructs a SerialClob object that is a serialized version of the given Clob object.

The new SerialClob object is initialized with the data from the Clob object; therefore, the Clob object should have previously brought the SQL CLOB value's data over to the client from the database. Otherwise, the new SerialClob object object will contain no data.

Note: The Clob object supplied to this constructor cannot return null for the Clob.getCharacterStream() and Clob.getAsciiStream methods. This SerialClob constructor cannot serialize a Clob object in this instance and will throw an SQLException object.

**Parameters:**clob - the Clob object from which this SerialClob object is to be constructed; cannot be null **Throws:** [SerialException](http://docs.google.com/javax/sql/rowset/serial/SerialException.html) - if an error occurs during serialization [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a SQL error occurs in capturing the CLOB; if the Clob object is a null; or if both the Clob.getCharacterStream() and Clob.getAsciiStream() methods on the Clob return a null**See Also:**[Clob](http://docs.google.com/java/sql/Clob.html)

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

### length

public long **length**()  
 throws [SerialException](http://docs.google.com/javax/sql/rowset/serial/SerialException.html)

Retrieves the number of characters in this SerialClob object's array of characters.

**Specified by:**[length](http://docs.google.com/java/sql/Clob.html#length()) in interface [Clob](http://docs.google.com/java/sql/Clob.html) **Returns:**a long indicating the length in characters of this SerialClob object's array of character **Throws:** [SerialException](http://docs.google.com/javax/sql/rowset/serial/SerialException.html) - if an error occurs

### getCharacterStream

public [Reader](http://docs.google.com/java/io/Reader.html) **getCharacterStream**()  
 throws [SerialException](http://docs.google.com/javax/sql/rowset/serial/SerialException.html)

Returns this SerialClob object's data as a stream of Unicode characters. Unlike the related method, getAsciiStream, a stream is produced regardless of whether the SerialClob object was created with a Clob object or a char array.

**Specified by:**[getCharacterStream](http://docs.google.com/java/sql/Clob.html#getCharacterStream()) in interface [Clob](http://docs.google.com/java/sql/Clob.html) **Returns:**a java.io.Reader object containing this SerialClob object's data **Throws:** [SerialException](http://docs.google.com/javax/sql/rowset/serial/SerialException.html) - if an error occurs**See Also:**[Clob.setCharacterStream(long)](http://docs.google.com/java/sql/Clob.html#setCharacterStream(long))

### getAsciiStream

public [InputStream](http://docs.google.com/java/io/InputStream.html) **getAsciiStream**()  
 throws [SerialException](http://docs.google.com/javax/sql/rowset/serial/SerialException.html),  
 [SQLException](http://docs.google.com/java/sql/SQLException.html)

Retrieves the CLOB value designated by this SerialClob object as an ascii stream. This method forwards the getAsciiStream call to the underlying Clob object in the event that this SerialClob object is instantiated with a Clob object. If this SerialClob object is instantiated with a char array, a SerialException object is thrown.

**Specified by:**[getAsciiStream](http://docs.google.com/java/sql/Clob.html#getAsciiStream()) in interface [Clob](http://docs.google.com/java/sql/Clob.html) **Returns:**a java.io.InputStream object containing this SerialClob object's data **Throws:** [SerialException](http://docs.google.com/javax/sql/rowset/serial/SerialException.html) - if this SerialClob object was not instantiated with a Clob object [SQLException](http://docs.google.com/java/sql/SQLException.html) - if there is an error accessing the CLOB value represented by the Clob object that was used to create this SerialClob object**See Also:**[Clob.setAsciiStream(long)](http://docs.google.com/java/sql/Clob.html#setAsciiStream(long))

### getSubString

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

Returns a copy of the substring contained in this SerialClob object, starting at the given position and continuing for the specified number or characters.

**Specified by:**[getSubString](http://docs.google.com/java/sql/Clob.html#getSubString(long,%20int)) in interface [Clob](http://docs.google.com/java/sql/Clob.html) **Parameters:**pos - the position of the first character in the substring to be copied; the first character of the SerialClob object is at position 1; must not be less than 1, and the sum of the starting position and the length of the substring must be less than the length of this SerialClob objectlength - the number of characters in the substring to be returned; must not be greater than the length of this SerialClob object, and the sum of the starting position and the length of the substring must be less than the length of this SerialClob object **Returns:**a String object containing a substring of this SerialClob object beginning at the given position and containing the specified number of consecutive characters **Throws:** [SerialException](http://docs.google.com/javax/sql/rowset/serial/SerialException.html) - if either of the arguments is out of bounds

### position

public long **position**([String](http://docs.google.com/java/lang/String.html) searchStr,  
 long start)  
 throws [SerialException](http://docs.google.com/javax/sql/rowset/serial/SerialException.html),  
 [SQLException](http://docs.google.com/java/sql/SQLException.html)

Returns the position in this SerialClob object where the given String object begins, starting the search at the specified position. This method returns -1 if the pattern is not found.

**Specified by:**[position](http://docs.google.com/java/sql/Clob.html#position(java.lang.String,%20long)) in interface [Clob](http://docs.google.com/java/sql/Clob.html) **Parameters:**searchStr - the String object for which to searchstart - the position in this SerialClob object at which to start the search; the first position is 1; must not be less than 1 nor greater than the length of this SerialClob object **Returns:**the position at which the given String object begins, starting the search at the specified position; -1 if the given String object is not found or the starting position is out of bounds; position numbering for the return value starts at 1 **Throws:** [SerialException](http://docs.google.com/javax/sql/rowset/serial/SerialException.html) - if an error occurs locating the String signature [SQLException](http://docs.google.com/java/sql/SQLException.html) - if there is an error accessing the Blob value from the database.

### position

public long **position**([Clob](http://docs.google.com/java/sql/Clob.html) searchStr,  
 long start)  
 throws [SerialException](http://docs.google.com/javax/sql/rowset/serial/SerialException.html),  
 [SQLException](http://docs.google.com/java/sql/SQLException.html)

Returns the position in this SerialClob object where the given Clob signature begins, starting the search at the specified position. This method returns -1 if the pattern is not found.

**Specified by:**[position](http://docs.google.com/java/sql/Clob.html#position(java.sql.Clob,%20long)) in interface [Clob](http://docs.google.com/java/sql/Clob.html) **Parameters:**searchStr - the Clob object for which to searchstart - the position in this SerialClob object at which to begin the search; the first position is 1; must not be less than 1 nor greater than the length of this SerialClob object **Returns:**the position at which the given Clob object begins in this SerialClob object, at or after the specified starting position **Throws:** [SerialException](http://docs.google.com/javax/sql/rowset/serial/SerialException.html) - if an error occurs locating the Clob signature [SQLException](http://docs.google.com/java/sql/SQLException.html) - if there is an error accessing the Blob value from the database

### setString

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

Writes the given Java String to the CLOB value that this SerialClob object represents, at the position pos.

**Specified by:**[setString](http://docs.google.com/java/sql/Clob.html#setString(long,%20java.lang.String)) in interface [Clob](http://docs.google.com/java/sql/Clob.html) **Parameters:**pos - the position at which to start writing to the CLOB value that this SerialClob object represents; the first position is 1; must not be less than 1 nor greater than the length of this SerialClob objectstr - the string to be written to the CLOB value that this SerialClob object represents **Returns:**the number of characters written **Throws:** [SerialException](http://docs.google.com/javax/sql/rowset/serial/SerialException.html) - if there is an error accessing the CLOB value; if an invalid position is set; if an invalid offset value is set; if number of bytes to be written is greater than the SerialClob length; or the combined values of the length and offset is greater than the Clob buffer

### setString

public int **setString**(long pos,  
 [String](http://docs.google.com/java/lang/String.html) str,  
 int offset,  
 int length)  
 throws [SerialException](http://docs.google.com/javax/sql/rowset/serial/SerialException.html)

Writes len characters of str, starting at character offset, to the CLOB value that this Clob represents.

**Specified by:**[setString](http://docs.google.com/java/sql/Clob.html#setString(long,%20java.lang.String,%20int,%20int)) in interface [Clob](http://docs.google.com/java/sql/Clob.html) **Parameters:**pos - the position at which to start writing to the CLOB value that this SerialClob object represents; the first position is 1; must not be less than 1 nor greater than the length of this SerialClob objectstr - 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 writtenlength - the number of characters to be written **Returns:**the number of characters written **Throws:** [SerialException](http://docs.google.com/javax/sql/rowset/serial/SerialException.html) - if there is an error accessing the CLOB value; if an invalid position is set; if an invalid offset value is set; if number of bytes to be written is greater than the SerialClob length; or the combined values of the length and offset is greater than the Clob buffer

### setAsciiStream

public [OutputStream](http://docs.google.com/java/io/OutputStream.html) **setAsciiStream**(long pos)  
 throws [SerialException](http://docs.google.com/javax/sql/rowset/serial/SerialException.html),  
 [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 SerialClob object represents, starting at position pos. This method forwards the setAsciiStream() call to the underlying Clob object in the event that this SerialClob object is instantiated with a Clob object. If this SerialClob object is instantiated with a char array, a SerialException object is thrown.

**Specified by:**[setAsciiStream](http://docs.google.com/java/sql/Clob.html#setAsciiStream(long)) in interface [Clob](http://docs.google.com/java/sql/Clob.html) **Parameters:**pos - the position at which to start writing to the CLOB object **Returns:**the stream to which ASCII encoded characters can be written **Throws:** [SerialException](http://docs.google.com/javax/sql/rowset/serial/SerialException.html) - if SerialClob is not instantiated with a Clob object that supports setAsciiStream [SQLException](http://docs.google.com/java/sql/SQLException.html) - if there is an error accessing the CLOB value**See Also:**[getAsciiStream()](http://docs.google.com/javax/sql/rowset/serial/SerialClob.html#getAsciiStream())

### setCharacterStream

public [Writer](http://docs.google.com/java/io/Writer.html) **setCharacterStream**(long pos)  
 throws [SerialException](http://docs.google.com/javax/sql/rowset/serial/SerialException.html),  
 [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 SerialClob object represents, at position pos. This method forwards the setCharacterStream() call to the underlying Clob object in the event that this SerialClob object is instantiated with a Clob object. If this SerialClob object is instantiated with a char array, a SerialException is thrown.

**Specified by:**[setCharacterStream](http://docs.google.com/java/sql/Clob.html#setCharacterStream(long)) in interface [Clob](http://docs.google.com/java/sql/Clob.html) **Parameters:**pos - the position at which to start writing to the CLOB value **Returns:**a stream to which Unicode encoded characters can be written **Throws:** [SerialException](http://docs.google.com/javax/sql/rowset/serial/SerialException.html) - if the SerialClob is not instantiated with a Clob object that supports setCharacterStream [SQLException](http://docs.google.com/java/sql/SQLException.html) - if there is an error accessing the CLOB value**See Also:**[getCharacterStream()](http://docs.google.com/javax/sql/rowset/serial/SerialClob.html#getCharacterStream())

### truncate

public void **truncate**(long length)  
 throws [SerialException](http://docs.google.com/javax/sql/rowset/serial/SerialException.html)

Truncates the CLOB value that this SerialClob object represents so that it has a length of len characters.

Truncating a SerialClob object to length 0 has the effect of clearing its contents.

**Specified by:**[truncate](http://docs.google.com/java/sql/Clob.html#truncate(long)) in interface [Clob](http://docs.google.com/java/sql/Clob.html) **Parameters:**length - the length, in bytes, 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 [SerialException](http://docs.google.com/javax/sql/rowset/serial/SerialException.html)

### getCharacterStream

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

**Description copied from interface:** [**Clob**](http://docs.google.com/java/sql/Clob.html#getCharacterStream(long,%20long)) Returns a Reader object that contains a partial Clob value, starting with the character specified by pos, which is length characters in length.

**Specified by:**[getCharacterStream](http://docs.google.com/java/sql/Clob.html#getCharacterStream(long,%20long)) in interface [Clob](http://docs.google.com/java/sql/Clob.html) **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

### free

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

**Description copied from interface:** [**Clob**](http://docs.google.com/java/sql/Clob.html#free()) 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.

**Specified by:**[free](http://docs.google.com/java/sql/Clob.html#free()) in interface [Clob](http://docs.google.com/java/sql/Clob.html) **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

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/SerialClob.html) | [**Tree**](http://docs.google.com/package-tree.html) | [**Deprecated**](http://docs.google.com/deprecated-list.html) | [**Index**](http://docs.google.com/index-files/index-1.html) | [**Help**](http://docs.google.com/help-doc.html) | | --- | --- | --- | --- | --- | --- | --- | --- | | | ***Java™ Platform***  ***Standard Ed. 6*** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [**PREV CLASS**](http://docs.google.com/javax/sql/rowset/serial/SerialBlob.html)   [**NEXT CLASS**](http://docs.google.com/javax/sql/rowset/serial/SerialDatalink.html) | [**FRAMES**](http://docs.google.com/index.html?javax/sql/rowset/serial/SerialClob.html)    [**NO FRAMES**](http://docs.google.com/SerialClob.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | [CONSTR](#3znysh7) | [METHOD](#2et92p0) | DETAIL: FIELD | [CONSTR](#3dy6vkm) | [METHOD](#2s8eyo1) |

[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).