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

## **javax.sql.rowset**

Interface Joinable

**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), [WebRowSet](http://docs.google.com/javax/sql/rowset/WebRowSet.html)

public interface **Joinable**

### 1.0 Background

The Joinable interface provides the methods for getting and setting a match column, which is the basis for forming the SQL JOIN formed by adding RowSet objects to a JoinRowSet object.

Any standard RowSet implementation **may** implement the Joinable interface in order to be added to a JoinRowSet object. Implementing this interface gives a RowSet object the ability to use Joinable methods, which set, retrieve, and get information about match columns. An application may add a RowSet object that has not implemented the Joinable interface to a JoinRowSet object, but to do so it must use one of the JoinRowSet.addRowSet methods that takes both a RowSet object and a match column or an array of RowSet objects and an array of match columns.

To get access to the methods in the Joinable interface, a RowSet object implements at least one of the five standard RowSet interfaces and also implements the Joinable interface. In addition, most RowSet objects extend the BaseRowSet class. For example:

class MyRowSetImpl extends BaseRowSet implements CachedRowSet, Joinable {  
 :  
 :  
 }

### 2.0 Usage Guidelines

The methods in the Joinable interface allow a RowSet object to set a match column, retrieve a match column, or unset a match column, which is the column upon which an SQL JOIN can be based. An instance of a class that implements these methods can be added to a JoinRowSet object to allow an SQL JOIN relationship to be established.

CachedRowSet crs = new MyRowSetImpl();  
 crs.populate((ResultSet)rs);  
 (Joinable)crs.setMatchColumnIndex(1);  
  
 JoinRowSet jrs = new JoinRowSetImpl();  
 jrs.addRowSet(crs);

In the previous example, *crs* is a CachedRowSet object that has emplemented the Joinable interface. In the following example, *crs2* has not, so it must supply the match column as an argument to the addRowSet method. This example assumes that column 1 is the match column.

CachedRowSet crs2 = new MyRowSetImpl();  
 crs2.populate((ResultSet)rs);  
   
 JoinRowSet jrs2 = new JoinRowSetImpl();  
 jrs2.addRowSet(crs2, 1);

The JoinRowSet interface makes it possible to get data from one or more RowSet objects consolidated into one table without having to incur the expense of creating a connection to a database. It is therefore ideally suited for use by disconnected RowSet objects. Nevertheless, any RowSet object **may** implement this interface regardless of whether it is connected or disconnected. Note that a JdbcRowSet object, being always connected to its data source, can become part of an SQL JOIN directly without having to become part of a JoinRowSet object.

### 3.0 Managing Multiple Match Columns

The index array passed into the setMatchColumn methods indicates how many match columns are being set (the length of the array) in addition to which columns will be used for the match. For example:

int[] i = {1, 2, 4, 7}; // indicates four match columns, with column  
 // indexes 1, 2, 4, 7 participating in the JOIN.  
 Joinable.setMatchColumn(i);

Subsequent match columns may be added as follows to a different Joinable object (a RowSet object that has implemented the Joinable interface).

int[] w = {3, 2, 5, 3};  
 Joinable2.setMatchColumn(w);

When an application adds two or more RowSet objects to a JoinRowSet object, the order of the indexes in the array is particularly important. Each index of the array maps directly to the corresponding index of the previously added RowSet object. If overlap or underlap occurs, the match column data is maintained in the event an additional Joinable RowSet is added and needs to relate to the match column data. Therefore, applications can set multiple match columns in any order, but this order has a direct effect on the outcome of the SQL JOIN.

This assertion applies in exactly the same manner when column names are used rather than column indexes to indicate match columns.

**See Also:**[JoinRowSet](http://docs.google.com/javax/sql/rowset/JoinRowSet.html)

| **Method Summary** | |
| --- | --- |
| int[] | [**getMatchColumnIndexes**](http://docs.google.com/javax/sql/rowset/Joinable.html#getMatchColumnIndexes())()            Retrieves the indexes of the match columns that were set for this RowSet object with the method setMatchColumn(int[] columnIdxes). |
| [String](http://docs.google.com/java/lang/String.html)[] | [**getMatchColumnNames**](http://docs.google.com/javax/sql/rowset/Joinable.html#getMatchColumnNames())()            Retrieves the names of the match columns that were set for this RowSet object with the method setMatchColumn(String [] columnNames). |
| void | [**setMatchColumn**](http://docs.google.com/javax/sql/rowset/Joinable.html#setMatchColumn(int))(int columnIdx)            Sets the designated column as the match column for this RowSet object. |
| void | [**setMatchColumn**](http://docs.google.com/javax/sql/rowset/Joinable.html#setMatchColumn(int%5B%5D))(int[] columnIdxes)            Sets the designated columns as the match column for this RowSet object. |
| void | [**setMatchColumn**](http://docs.google.com/javax/sql/rowset/Joinable.html#setMatchColumn(java.lang.String))([String](http://docs.google.com/java/lang/String.html) columnName)            Sets the designated column as the match column for this RowSet object. |
| void | [**setMatchColumn**](http://docs.google.com/javax/sql/rowset/Joinable.html#setMatchColumn(java.lang.String%5B%5D))([String](http://docs.google.com/java/lang/String.html)[] columnNames)            Sets the designated columns as the match column for this RowSet object. |
| void | [**unsetMatchColumn**](http://docs.google.com/javax/sql/rowset/Joinable.html#unsetMatchColumn(int))(int columnIdx)            Unsets the designated column as the match column for this RowSet object. |
| void | [**unsetMatchColumn**](http://docs.google.com/javax/sql/rowset/Joinable.html#unsetMatchColumn(int%5B%5D))(int[] columnIdxes)            Unsets the designated columns as the match column for this RowSet object. |
| void | [**unsetMatchColumn**](http://docs.google.com/javax/sql/rowset/Joinable.html#unsetMatchColumn(java.lang.String))([String](http://docs.google.com/java/lang/String.html) columnName)            Unsets the designated column as the match column for this RowSet object. |
| void | [**unsetMatchColumn**](http://docs.google.com/javax/sql/rowset/Joinable.html#unsetMatchColumn(java.lang.String%5B%5D))([String](http://docs.google.com/java/lang/String.html)[] columnName)            Unsets the designated columns as the match columns for this RowSet object. |

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

### setMatchColumn

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

Sets the designated column as the match column for this RowSet object. A JoinRowSet object can now add this RowSet object based on the match column.

Sub-interfaces such as the CachedRowSetTM interface define the method CachedRowSet.setKeyColumns, which allows primary key semantics to be enforced on specific columns. Implementations of the setMatchColumn(int columnIdx) method should ensure that the constraints on the key columns are maintained when a CachedRowSet object sets a primary key column as a match column.

**Parameters:**columnIdx - an int identifying the index of the column to be set as the match column **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if an invalid column index is set**See Also:**[setMatchColumn(int[])](http://docs.google.com/javax/sql/rowset/Joinable.html#setMatchColumn(int%5B%5D)), [unsetMatchColumn(int)](http://docs.google.com/javax/sql/rowset/Joinable.html#unsetMatchColumn(int))

### setMatchColumn

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

Sets the designated columns as the match column for this RowSet object. A JoinRowSet object can now add this RowSet object based on the match column.

**Parameters:**columnIdxes - an array of int identifying the indexes of the columns to be set as the match columns **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if an invalid column index is set**See Also:**[setMatchColumn(int[])](http://docs.google.com/javax/sql/rowset/Joinable.html#setMatchColumn(int%5B%5D)), [unsetMatchColumn(int[])](http://docs.google.com/javax/sql/rowset/Joinable.html#unsetMatchColumn(int%5B%5D))

### setMatchColumn

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

Sets the designated column as the match column for this RowSet object. A JoinRowSet object can now add this RowSet object based on the match column.

Subinterfaces such as the CachedRowSet interface define the method CachedRowSet.setKeyColumns, which allows primary key semantics to be enforced on specific columns. Implementations of the setMatchColumn(String columnIdx) method should ensure that the constraints on the key columns are maintained when a CachedRowSet object sets a primary key column as a match column.

**Parameters:**columnName - a String object giving the name of the column to be set as the match column **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if an invalid column name is set, the column name is a null, or the column name is an empty string**See Also:**[unsetMatchColumn(int)](http://docs.google.com/javax/sql/rowset/Joinable.html#unsetMatchColumn(int)), [setMatchColumn(int[])](http://docs.google.com/javax/sql/rowset/Joinable.html#setMatchColumn(int%5B%5D))

### setMatchColumn

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

Sets the designated columns as the match column for this RowSet object. A JoinRowSet object can now add this RowSet object based on the match column.

**Parameters:**columnNames - an array of String objects giving the names of the column to be set as the match columns **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if an invalid column name is set, the column name is a null, or the column name is an empty string**See Also:**[unsetMatchColumn(int)](http://docs.google.com/javax/sql/rowset/Joinable.html#unsetMatchColumn(int)), [setMatchColumn(int[])](http://docs.google.com/javax/sql/rowset/Joinable.html#setMatchColumn(int%5B%5D))

### getMatchColumnIndexes

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

Retrieves the indexes of the match columns that were set for this RowSet object with the method setMatchColumn(int[] columnIdxes).

**Returns:**an int array identifying the indexes of the columns that were set as the match columns for this RowSet object **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if no match column has been set**See Also:**[setMatchColumn(int)](http://docs.google.com/javax/sql/rowset/Joinable.html#setMatchColumn(int)), [unsetMatchColumn(int)](http://docs.google.com/javax/sql/rowset/Joinable.html#unsetMatchColumn(int))

### getMatchColumnNames

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

Retrieves the names of the match columns that were set for this RowSet object with the method setMatchColumn(String [] columnNames).

**Returns:**an array of String objects giving the names of the columns set as the match columns for this RowSet object **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if no match column has been set**See Also:**[setMatchColumn(int)](http://docs.google.com/javax/sql/rowset/Joinable.html#setMatchColumn(int)), [unsetMatchColumn(int)](http://docs.google.com/javax/sql/rowset/Joinable.html#unsetMatchColumn(int))

### unsetMatchColumn

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

Unsets the designated column as the match column for this RowSet object.

RowSet objects that implement the Joinable interface must ensure that a key-like constraint continues to be enforced until the method CachedRowSet.unsetKeyColumns has been called on the designated column.

**Parameters:**columnIdx - an int that identifies the index of the column that is to be unset as a match column **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if an invalid column index is designated or if the designated column was not previously set as a match column**See Also:**[setMatchColumn(int)](http://docs.google.com/javax/sql/rowset/Joinable.html#setMatchColumn(int))

### unsetMatchColumn

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

Unsets the designated columns as the match column for this RowSet object.

**Parameters:**columnIdxes - an arrary of int that identifies the indexes of the columns that are to be unset as match columns **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if an invalid column index is designated or if the designated column was not previously set as a match column**See Also:**[setMatchColumn(int)](http://docs.google.com/javax/sql/rowset/Joinable.html#setMatchColumn(int))

### unsetMatchColumn

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

Unsets the designated column as the match column for this RowSet object.

RowSet objects that implement the Joinable interface must ensure that a key-like constraint continues to be enforced until the method CachedRowSet.unsetKeyColumns has been called on the designated column.

**Parameters:**columnName - a String object giving the name of the column that is to be unset as a match column **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if an invalid column name is designated or the designated column was not previously set as a match column**See Also:**[setMatchColumn(int)](http://docs.google.com/javax/sql/rowset/Joinable.html#setMatchColumn(int))

### unsetMatchColumn

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

Unsets the designated columns as the match columns for this RowSet object.

**Parameters:**columnName - an array of String objects giving the names of the columns that are to be unset as the match columns **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if an invalid column name is designated or the designated column was not previously set as a match column**See Also:**[setMatchColumn(int)](http://docs.google.com/javax/sql/rowset/Joinable.html#setMatchColumn(int))

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/Joinable.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/JdbcRowSet.html)   [**NEXT CLASS**](http://docs.google.com/javax/sql/rowset/JoinRowSet.html) | [**FRAMES**](http://docs.google.com/index.html?javax/sql/rowset/Joinable.html)    [**NO FRAMES**](http://docs.google.com/Joinable.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).