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

public interface **Predicate**

The standard interface that provides the framework for all FilteredRowSet objects to describe their filters.

### 1.0 Background

The Predicate interface is a standard interface that applications can implement to define the filter they wish to apply to a a FilteredRowSet object. A FilteredRowSet object consumes implementations of this interface and enforces the constraints defined in the implementation of the method evaluate. A FilteredRowSet object enforces the filter constraints in a bi-directional manner: It outputs only rows that are within the constraints of the filter; and conversely, it inserts, modifies, or updates only rows that are within the constraints of the filter.

### 2.0 Implementation Guidelines

In order to supply a predicate for the FilteredRowSet. this interface must be implemented. At this time, the JDBC RowSet Implementations (JSR-114) does not specify any standard filters definitions. By specifying a standard means and mechanism for a range of filters to be defined and deployed with both the reference and vendor implementations of the FilteredRowSet interface, this allows for a flexible and application motivated implementations of Predicate to emerge.

A sample implementation would look something like this:

public class Range implements Predicate {  
  
 private Object lo[];  
 private Object hi[];  
 private int idx[];  
  
 public Range(Object[] lo, Object[] hi, int[] idx) {  
 this.lo = lo;  
 this.hi = hi;  
 this.idx = idx;  
 }  
  
 public boolean evaluate(RowSet rs) {  
 CachedRowSet crs = (CachedRowSet)rs;  
 boolean bool1,bool2;   
   
 // Check the present row determine if it lies  
 // within the filtering criteria.   
   
 for (int i = 0; i < idx.length; i++) {  
   
 if ((rs.getObject(idx[i]) >= lo[i]) &&   
 (rs.getObject(idx[i]) >= hi[i]) {   
 bool1 = true; // within filter constraints  
 } else {  
 bool2 = true; // outside of filter constraints   
 }   
 }  
   
 if (bool2) {  
 return false;  
 } else {  
 return true;  
 }  
 }

The example above implements a simple range predicate. Note, that implementations should but are not required to provider String and integer index based constructors to provide for JDBC RowSet Implementation applications that use both column identification conventions.

| **Method Summary** | |
| --- | --- |
| boolean | [**evaluate**](http://docs.google.com/javax/sql/rowset/Predicate.html#evaluate(java.lang.Object,%20int))([Object](http://docs.google.com/java/lang/Object.html) value, int column)            This method is called by a FilteredRowSet object to check whether the value lies between the filtering criterion (or criteria if multiple constraints exist) set using the setFilter() method. |
| boolean | [**evaluate**](http://docs.google.com/javax/sql/rowset/Predicate.html#evaluate(java.lang.Object,%20java.lang.String))([Object](http://docs.google.com/java/lang/Object.html) value, [String](http://docs.google.com/java/lang/String.html) columnName)            This method is called by the FilteredRowSet object to check whether the value lies between the filtering criteria set using the setFilter method. |
| boolean | [**evaluate**](http://docs.google.com/javax/sql/rowset/Predicate.html#evaluate(javax.sql.RowSet))([RowSet](http://docs.google.com/javax/sql/RowSet.html) rs)            This method is typically called a FilteredRowSet object internal methods (not public) that control the RowSet object's cursor moving from row to the next. |

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

### evaluate

boolean **evaluate**([RowSet](http://docs.google.com/javax/sql/RowSet.html) rs)

This method is typically called a FilteredRowSet object internal methods (not public) that control the RowSet object's cursor moving from row to the next. In addition, if this internal method moves the cursor onto a row that has been deleted, the internal method will continue to ove the cursor until a valid row is found.

**Returns:**true if there are more rows in the filter; false otherwise

### evaluate

boolean **evaluate**([Object](http://docs.google.com/java/lang/Object.html) value,  
 int column)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

This method is called by a FilteredRowSet object to check whether the value lies between the filtering criterion (or criteria if multiple constraints exist) set using the setFilter() method.

The FilteredRowSet object will use this method internally while inserting new rows to a FilteredRowSet instance.

**Parameters:**value - An Object value which needs to be checked, whether it can be part of this FilterRowSet object.column - a int object that must match the SQL index of a column in this RowSet object. This must have been passed to Predicate as one of the columns for filtering while initializing a Predicate **Returns:**true ifrow value lies within the filter; false otherwise **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if the column is not part of filtering criteria

### evaluate

boolean **evaluate**([Object](http://docs.google.com/java/lang/Object.html) value,  
 [String](http://docs.google.com/java/lang/String.html) columnName)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

This method is called by the FilteredRowSet object to check whether the value lies between the filtering criteria set using the setFilter method.

The FilteredRowSet object will use this method internally while inserting new rows to a FilteredRowSet instance.

**Parameters:**value - An Object value which needs to be checked, whether it can be part of this FilterRowSet.columnName - a String object that must match the SQL name of a column in this RowSet, ignoring case. This must have been passed to Predicate as one of the columns for filtering while initializing a Predicate **Returns:**true if value lies within the filter; false otherwise **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if the column is not part of filtering criteria

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