| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/RowFilter.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/swing/RootPaneContainer.html)   [**NEXT CLASS**](http://docs.google.com/javax/swing/RowFilter.ComparisonType.html) | [**FRAMES**](http://docs.google.com/index.html?javax/swing/RowFilter.html)    [**NO FRAMES**](http://docs.google.com/RowFilter.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: [NESTED](#3znysh7) | FIELD | [CONSTR](#2et92p0) | [METHOD](#tyjcwt) | DETAIL: FIELD | [CONSTR](#1t3h5sf) | [METHOD](#2s8eyo1) |

## **javax.swing**

Class RowFilter<M,I>

[java.lang.Object](http://docs.google.com/java/lang/Object.html)  
 **javax.swing.RowFilter<M,I>**

**Type Parameters:**M - the type of the model; for example PersonModelI - the type of the identifier; when using TableRowSorter this will be Integer

public abstract class **RowFilter<M,I>**extends [Object](http://docs.google.com/java/lang/Object.html)

RowFilter is used to filter out entries from the model so that they are not shown in the view. For example, a RowFilter associated with a JTable might only allow rows that contain a column with a specific string. The meaning of *entry* depends on the component type. For example, when a filter is associated with a JTable, an entry corresponds to a row; when associated with a JTree, an entry corresponds to a node.

Subclasses must override the include method to indicate whether the entry should be shown in the view. The Entry argument can be used to obtain the values in each of the columns in that entry. The following example shows an include method that allows only entries containing one or more values starting with the string "a":

RowFilter<Object,Object> startsWithAFilter = new RowFilter<Object,Object>() {  
 public boolean include(Entry<? extends Object, ? extends Object> entry) {  
 for (int i = entry.getValueCount() - 1; i >= 0; i--) {  
 if (entry.getStringValue(i).startsWith("a")) {  
 // The value starts with "a", include it  
 return true;  
 }  
 }  
 // None of the columns start with "a"; return false so that this  
 // entry is not shown  
 return false;  
 }  
 };

RowFilter has two formal type parameters that allow you to create a RowFilter for a specific model. For example, the following assumes a specific model that is wrapping objects of type Person. Only Persons with an age over 20 will be shown:

RowFilter<PersonModel,Integer> ageFilter = new RowFilter<PersonModel,Integer>() {  
 public boolean include(Entry<? extends PersonModel, ? extends Integer> entry) {  
 PersonModel personModel = entry.getModel();  
 Person person = personModel.getPerson(entry.getIdentifier());  
 if (person.getAge() > 20) {  
 // Returning true indicates this row should be shown.  
 return true;  
 }  
 // Age is <= 20, don't show it.  
 return false;  
 }  
 };  
 PersonModel model = createPersonModel();  
 TableRowSorter<PersonModel> sorter = new TableRowSorter<PersonModel>(model);  
 sorter.setRowFilter(ageFilter);

**Since:** 1.6 **See Also:**[TableRowSorter](http://docs.google.com/javax/swing/table/TableRowSorter.html)

| **Nested Class Summary** | |
| --- | --- |
| static class | [**RowFilter.ComparisonType**](http://docs.google.com/javax/swing/RowFilter.ComparisonType.html)            Enumeration of the possible comparison values supported by some of the default RowFilters. |
| static class | [**RowFilter.Entry**](http://docs.google.com/javax/swing/RowFilter.Entry.html)**<**[**M**](http://docs.google.com/javax/swing/RowFilter.Entry.html)**,**[**I**](http://docs.google.com/javax/swing/RowFilter.Entry.html)**>**            An Entry object is passed to instances of RowFilter, allowing the filter to get the value of the entry's data, and thus to determine whether the entry should be shown. |

| **Constructor Summary** | |
| --- | --- |
| [**RowFilter**](http://docs.google.com/javax/swing/RowFilter.html#RowFilter())() |

| **Method Summary** | |
| --- | --- |
| static   | <M,I> [RowFilter](http://docs.google.com/javax/swing/RowFilter.html)<M,I> | | --- | | [**andFilter**](http://docs.google.com/javax/swing/RowFilter.html#andFilter(java.lang.Iterable))([Iterable](http://docs.google.com/java/lang/Iterable.html)<? extends [RowFilter](http://docs.google.com/javax/swing/RowFilter.html)<? super M,? super I>> filters)            Returns a RowFilter that includes entries if all of the supplied filters include the entry. |
| static   | <M,I> [RowFilter](http://docs.google.com/javax/swing/RowFilter.html)<M,I> | | --- | | [**dateFilter**](http://docs.google.com/javax/swing/RowFilter.html#dateFilter(javax.swing.RowFilter.ComparisonType,%20java.util.Date,%20int...))([RowFilter.ComparisonType](http://docs.google.com/javax/swing/RowFilter.ComparisonType.html) type, [Date](http://docs.google.com/java/util/Date.html) date, int... indices)            Returns a RowFilter that includes entries that have at least one Date value meeting the specified criteria. |
| abstract  boolean | [**include**](http://docs.google.com/javax/swing/RowFilter.html#include(javax.swing.RowFilter.Entry))([RowFilter.Entry](http://docs.google.com/javax/swing/RowFilter.Entry.html)<? extends [M](http://docs.google.com/javax/swing/RowFilter.html),? extends [I](http://docs.google.com/javax/swing/RowFilter.html)> entry)            Returns true if the specified entry should be shown; returns false if the entry should be hidden. |
| static   | <M,I> [RowFilter](http://docs.google.com/javax/swing/RowFilter.html)<M,I> | | --- | | [**notFilter**](http://docs.google.com/javax/swing/RowFilter.html#notFilter(javax.swing.RowFilter))([RowFilter](http://docs.google.com/javax/swing/RowFilter.html)<M,I> filter)            Returns a RowFilter that includes entries if the supplied filter does not include the entry. |
| static   | <M,I> [RowFilter](http://docs.google.com/javax/swing/RowFilter.html)<M,I> | | --- | | [**numberFilter**](http://docs.google.com/javax/swing/RowFilter.html#numberFilter(javax.swing.RowFilter.ComparisonType,%20java.lang.Number,%20int...))([RowFilter.ComparisonType](http://docs.google.com/javax/swing/RowFilter.ComparisonType.html) type, [Number](http://docs.google.com/java/lang/Number.html) number, int... indices)            Returns a RowFilter that includes entries that have at least one Number value meeting the specified criteria. |
| static   | <M,I> [RowFilter](http://docs.google.com/javax/swing/RowFilter.html)<M,I> | | --- | | [**orFilter**](http://docs.google.com/javax/swing/RowFilter.html#orFilter(java.lang.Iterable))([Iterable](http://docs.google.com/java/lang/Iterable.html)<? extends [RowFilter](http://docs.google.com/javax/swing/RowFilter.html)<? super M,? super I>> filters)            Returns a RowFilter that includes entries if any of the supplied filters includes the entry. |
| static   | <M,I> [RowFilter](http://docs.google.com/javax/swing/RowFilter.html)<M,I> | | --- | | [**regexFilter**](http://docs.google.com/javax/swing/RowFilter.html#regexFilter(java.lang.String,%20int...))([String](http://docs.google.com/java/lang/String.html) regex, int... indices)            Returns a RowFilter that uses a regular expression to determine which entries to include. |

| **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** |
| --- |

### RowFilter

public **RowFilter**()

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

### regexFilter

public static <M,I> [RowFilter](http://docs.google.com/javax/swing/RowFilter.html)<M,I> **regexFilter**([String](http://docs.google.com/java/lang/String.html) regex,  
 int... indices)

Returns a RowFilter that uses a regular expression to determine which entries to include. Only entries with at least one matching value are included. For example, the following creates a RowFilter that includes entries with at least one value starting with "a":

RowFilter.regexFilter("^a");

The returned filter uses [Matcher.find()](http://docs.google.com/java/util/regex/Matcher.html#find()) to test for inclusion. To test for exact matches use the characters '^' and '$' to match the beginning and end of the string respectively. For example, "^foo$" includes only rows whose string is exactly "foo" and not, for example, "food". See [Pattern](http://docs.google.com/java/util/regex/Pattern.html) for a complete description of the supported regular-expression constructs.

**Parameters:**regex - the regular expression to filter onindices - the indices of the values to check. If not supplied all values are evaluated **Returns:**a RowFilter implementing the specified criteria **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if regex is null [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if any of the indices are < 0 PatternSyntaxException - if regex is not a valid regular expression.**See Also:**[Pattern](http://docs.google.com/java/util/regex/Pattern.html)

### dateFilter

public static <M,I> [RowFilter](http://docs.google.com/javax/swing/RowFilter.html)<M,I> **dateFilter**([RowFilter.ComparisonType](http://docs.google.com/javax/swing/RowFilter.ComparisonType.html) type,  
 [Date](http://docs.google.com/java/util/Date.html) date,  
 int... indices)

Returns a RowFilter that includes entries that have at least one Date value meeting the specified criteria. For example, the following RowFilter includes only entries with at least one date value after the current date:

RowFilter.dateFilter(ComparisonType.AFTER, new Date());

**Parameters:**type - the type of comparison to performdate - the date to compare againstindices - the indices of the values to check. If not supplied all values are evaluated **Returns:**a RowFilter implementing the specified criteria **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if date is null [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if any of the indices are < 0 or type is null**See Also:**[Calendar](http://docs.google.com/java/util/Calendar.html), [Date](http://docs.google.com/java/util/Date.html)

### numberFilter

public static <M,I> [RowFilter](http://docs.google.com/javax/swing/RowFilter.html)<M,I> **numberFilter**([RowFilter.ComparisonType](http://docs.google.com/javax/swing/RowFilter.ComparisonType.html) type,  
 [Number](http://docs.google.com/java/lang/Number.html) number,  
 int... indices)

Returns a RowFilter that includes entries that have at least one Number value meeting the specified criteria. For example, the following filter will only include entries with at least one number value equal to 10:

RowFilter.numberFilter(ComparisonType.EQUAL, 10);

**Parameters:**type - the type of comparison to performindices - the indices of the values to check. If not supplied all values are evaluated **Returns:**a RowFilter implementing the specified criteria **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if any of the indices are < 0, type is null or number is null

### orFilter

public static <M,I> [RowFilter](http://docs.google.com/javax/swing/RowFilter.html)<M,I> **orFilter**([Iterable](http://docs.google.com/java/lang/Iterable.html)<? extends [RowFilter](http://docs.google.com/javax/swing/RowFilter.html)<? super M,? super I>> filters)

Returns a RowFilter that includes entries if any of the supplied filters includes the entry.

The following example creates a RowFilter that will include any entries containing the string "foo" or the string "bar":

List<RowFilter<Object,Object>> filters = new ArrayList<RowFilter<Object,Object>>(2);  
 filters.add(RowFilter.regexFilter("foo"));  
 filters.add(RowFilter.regexFilter("bar"));  
 RowFilter<Object,Object> fooBarFilter = RowFilter.orFilter(filters);

**Parameters:**filters - the RowFilters to test **Returns:**a RowFilter implementing the specified criteria **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if any of the filters are null [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if filters is null**See Also:**[Arrays.asList(T...)](http://docs.google.com/java/util/Arrays.html#asList(T...))

### andFilter

public static <M,I> [RowFilter](http://docs.google.com/javax/swing/RowFilter.html)<M,I> **andFilter**([Iterable](http://docs.google.com/java/lang/Iterable.html)<? extends [RowFilter](http://docs.google.com/javax/swing/RowFilter.html)<? super M,? super I>> filters)

Returns a RowFilter that includes entries if all of the supplied filters include the entry.

The following example creates a RowFilter that will include any entries containing the string "foo" and the string "bar":

List<RowFilter<Object,Object>> filters = new ArrayList<RowFilter<Object,Object>>(2);  
 filters.add(RowFilter.regexFilter("foo"));  
 filters.add(RowFilter.regexFilter("bar"));  
 RowFilter<Object,Object> fooBarFilter = RowFilter.andFilter(filters);

**Parameters:**filters - the RowFilters to test **Returns:**a RowFilter implementing the specified criteria **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if any of the filters are null [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if filters is null**See Also:**[Arrays.asList(T...)](http://docs.google.com/java/util/Arrays.html#asList(T...))

### notFilter

public static <M,I> [RowFilter](http://docs.google.com/javax/swing/RowFilter.html)<M,I> **notFilter**([RowFilter](http://docs.google.com/javax/swing/RowFilter.html)<M,I> filter)

Returns a RowFilter that includes entries if the supplied filter does not include the entry.

**Parameters:**filter - the RowFilter to negate **Returns:**a RowFilter implementing the specified criteria **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if filter is null

### include

public abstract boolean **include**([RowFilter.Entry](http://docs.google.com/javax/swing/RowFilter.Entry.html)<? extends [M](http://docs.google.com/javax/swing/RowFilter.html),? extends [I](http://docs.google.com/javax/swing/RowFilter.html)> entry)

Returns true if the specified entry should be shown; returns false if the entry should be hidden.

The entry argument is valid only for the duration of the invocation. Using entry after the call returns results in undefined behavior.

**Parameters:**entry - a non-null object that wraps the underlying object from the model **Returns:**true if the entry should be shown

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/RowFilter.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/swing/RootPaneContainer.html)   [**NEXT CLASS**](http://docs.google.com/javax/swing/RowFilter.ComparisonType.html) | [**FRAMES**](http://docs.google.com/index.html?javax/swing/RowFilter.html)    [**NO FRAMES**](http://docs.google.com/RowFilter.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: [NESTED](#3znysh7) | FIELD | [CONSTR](#2et92p0) | [METHOD](#tyjcwt) | DETAIL: FIELD | [CONSTR](#1t3h5sf) | [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).