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

## **javax.swing**

Interface ListSelectionModel

**All Known Implementing Classes:** [DefaultListSelectionModel](http://docs.google.com/javax/swing/DefaultListSelectionModel.html)

public interface **ListSelectionModel**

This interface represents the current state of the selection for any of the components that display a list of values with stable indices. The selection is modeled as a set of intervals, each interval represents a contiguous range of selected list elements. The methods for modifying the set of selected intervals all take a pair of indices, index0 and index1, that represent a closed interval, i.e. the interval includes both index0 and index1.

**See Also:**[DefaultListSelectionModel](http://docs.google.com/javax/swing/DefaultListSelectionModel.html)

| **Field Summary** | |
| --- | --- |
| static int | [**MULTIPLE\_INTERVAL\_SELECTION**](http://docs.google.com/javax/swing/ListSelectionModel.html#MULTIPLE_INTERVAL_SELECTION)            A value for the selectionMode property: select one or more contiguous ranges of indices at a time. |
| static int | [**SINGLE\_INTERVAL\_SELECTION**](http://docs.google.com/javax/swing/ListSelectionModel.html#SINGLE_INTERVAL_SELECTION)            A value for the selectionMode property: select one contiguous range of indices at a time. |
| static int | [**SINGLE\_SELECTION**](http://docs.google.com/javax/swing/ListSelectionModel.html#SINGLE_SELECTION)            A value for the selectionMode property: select one list index at a time. |

| **Method Summary** | |
| --- | --- |
| void | [**addListSelectionListener**](http://docs.google.com/javax/swing/ListSelectionModel.html#addListSelectionListener(javax.swing.event.ListSelectionListener))([ListSelectionListener](http://docs.google.com/javax/swing/event/ListSelectionListener.html) x)            Add a listener to the list that's notified each time a change to the selection occurs. |
| void | [**addSelectionInterval**](http://docs.google.com/javax/swing/ListSelectionModel.html#addSelectionInterval(int,%20int))(int index0, int index1)            Changes the selection to be the set union of the current selection and the indices between index0 and index1 inclusive. |
| void | [**clearSelection**](http://docs.google.com/javax/swing/ListSelectionModel.html#clearSelection())()            Change the selection to the empty set. |
| int | [**getAnchorSelectionIndex**](http://docs.google.com/javax/swing/ListSelectionModel.html#getAnchorSelectionIndex())()            Return the first index argument from the most recent call to setSelectionInterval(), addSelectionInterval() or removeSelectionInterval(). |
| int | [**getLeadSelectionIndex**](http://docs.google.com/javax/swing/ListSelectionModel.html#getLeadSelectionIndex())()            Return the second index argument from the most recent call to setSelectionInterval(), addSelectionInterval() or removeSelectionInterval(). |
| int | [**getMaxSelectionIndex**](http://docs.google.com/javax/swing/ListSelectionModel.html#getMaxSelectionIndex())()            Returns the last selected index or -1 if the selection is empty. |
| int | [**getMinSelectionIndex**](http://docs.google.com/javax/swing/ListSelectionModel.html#getMinSelectionIndex())()            Returns the first selected index or -1 if the selection is empty. |
| int | [**getSelectionMode**](http://docs.google.com/javax/swing/ListSelectionModel.html#getSelectionMode())()            Returns the current selection mode. |
| boolean | [**getValueIsAdjusting**](http://docs.google.com/javax/swing/ListSelectionModel.html#getValueIsAdjusting())()            Returns true if the selection is undergoing a series of changes. |
| void | [**insertIndexInterval**](http://docs.google.com/javax/swing/ListSelectionModel.html#insertIndexInterval(int,%20int,%20boolean))(int index, int length, boolean before)            Insert length indices beginning before/after index. |
| boolean | [**isSelectedIndex**](http://docs.google.com/javax/swing/ListSelectionModel.html#isSelectedIndex(int))(int index)            Returns true if the specified index is selected. |
| boolean | [**isSelectionEmpty**](http://docs.google.com/javax/swing/ListSelectionModel.html#isSelectionEmpty())()            Returns true if no indices are selected. |
| void | [**removeIndexInterval**](http://docs.google.com/javax/swing/ListSelectionModel.html#removeIndexInterval(int,%20int))(int index0, int index1)            Remove the indices in the interval index0,index1 (inclusive) from the selection model. |
| void | [**removeListSelectionListener**](http://docs.google.com/javax/swing/ListSelectionModel.html#removeListSelectionListener(javax.swing.event.ListSelectionListener))([ListSelectionListener](http://docs.google.com/javax/swing/event/ListSelectionListener.html) x)            Remove a listener from the list that's notified each time a change to the selection occurs. |
| void | [**removeSelectionInterval**](http://docs.google.com/javax/swing/ListSelectionModel.html#removeSelectionInterval(int,%20int))(int index0, int index1)            Changes the selection to be the set difference of the current selection and the indices between index0 and index1 inclusive. |
| void | [**setAnchorSelectionIndex**](http://docs.google.com/javax/swing/ListSelectionModel.html#setAnchorSelectionIndex(int))(int index)            Set the anchor selection index. |
| void | [**setLeadSelectionIndex**](http://docs.google.com/javax/swing/ListSelectionModel.html#setLeadSelectionIndex(int))(int index)            Set the lead selection index. |
| void | [**setSelectionInterval**](http://docs.google.com/javax/swing/ListSelectionModel.html#setSelectionInterval(int,%20int))(int index0, int index1)            Changes the selection to be between index0 and index1 inclusive. |
| void | [**setSelectionMode**](http://docs.google.com/javax/swing/ListSelectionModel.html#setSelectionMode(int))(int selectionMode)            Sets the selection mode. |
| void | [**setValueIsAdjusting**](http://docs.google.com/javax/swing/ListSelectionModel.html#setValueIsAdjusting(boolean))(boolean valueIsAdjusting)            Sets the valueIsAdjusting property, which indicates whether or not upcoming selection changes should be considered part of a single change. |

| **Field Detail** |
| --- |

### SINGLE\_SELECTION

static final int **SINGLE\_SELECTION**

A value for the selectionMode property: select one list index at a time.

**See Also:**[setSelectionMode(int)](http://docs.google.com/javax/swing/ListSelectionModel.html#setSelectionMode(int)), [Constant Field Values](http://docs.google.com/constant-values.html#javax.swing.ListSelectionModel.SINGLE_SELECTION)

### SINGLE\_INTERVAL\_SELECTION

static final int **SINGLE\_INTERVAL\_SELECTION**

A value for the selectionMode property: select one contiguous range of indices at a time.

**See Also:**[setSelectionMode(int)](http://docs.google.com/javax/swing/ListSelectionModel.html#setSelectionMode(int)), [Constant Field Values](http://docs.google.com/constant-values.html#javax.swing.ListSelectionModel.SINGLE_INTERVAL_SELECTION)

### MULTIPLE\_INTERVAL\_SELECTION

static final int **MULTIPLE\_INTERVAL\_SELECTION**

A value for the selectionMode property: select one or more contiguous ranges of indices at a time.

**See Also:**[setSelectionMode(int)](http://docs.google.com/javax/swing/ListSelectionModel.html#setSelectionMode(int)), [Constant Field Values](http://docs.google.com/constant-values.html#javax.swing.ListSelectionModel.MULTIPLE_INTERVAL_SELECTION)

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

### setSelectionInterval

void **setSelectionInterval**(int index0,  
 int index1)

Changes the selection to be between index0 and index1 inclusive. index0 doesn't have to be less than or equal to index1.

In SINGLE\_SELECTION selection mode, only the second index is used.

If this represents a change to the current selection, then each ListSelectionListener is notified of the change.

**Parameters:**index0 - one end of the interval.index1 - other end of the interval**See Also:**[addListSelectionListener(javax.swing.event.ListSelectionListener)](http://docs.google.com/javax/swing/ListSelectionModel.html#addListSelectionListener(javax.swing.event.ListSelectionListener))

### addSelectionInterval

void **addSelectionInterval**(int index0,  
 int index1)

Changes the selection to be the set union of the current selection and the indices between index0 and index1 inclusive. index0 doesn't have to be less than or equal to index1.

In SINGLE\_SELECTION selection mode, this is equivalent to calling setSelectionInterval, and only the second index is used. In SINGLE\_INTERVAL\_SELECTION selection mode, this method behaves like setSelectionInterval, unless the given interval is immediately adjacent to or overlaps the existing selection, and can therefore be used to grow the selection.

If this represents a change to the current selection, then each ListSelectionListener is notified of the change.

**Parameters:**index0 - one end of the interval.index1 - other end of the interval**See Also:**[addListSelectionListener(javax.swing.event.ListSelectionListener)](http://docs.google.com/javax/swing/ListSelectionModel.html#addListSelectionListener(javax.swing.event.ListSelectionListener)), [setSelectionInterval(int, int)](http://docs.google.com/javax/swing/ListSelectionModel.html#setSelectionInterval(int,%20int))

### removeSelectionInterval

void **removeSelectionInterval**(int index0,  
 int index1)

Changes the selection to be the set difference of the current selection and the indices between index0 and index1 inclusive. index0 doesn't have to be less than or equal to index1.

In SINGLE\_INTERVAL\_SELECTION selection mode, if the removal would produce two disjoint selections, the removal is extended through the greater end of the selection. For example, if the selection is 0-10 and you supply indices 5,6 (in any order) the resulting selection is 0-4.

If this represents a change to the current selection, then each ListSelectionListener is notified of the change.

**Parameters:**index0 - one end of the interval.index1 - other end of the interval**See Also:**[addListSelectionListener(javax.swing.event.ListSelectionListener)](http://docs.google.com/javax/swing/ListSelectionModel.html#addListSelectionListener(javax.swing.event.ListSelectionListener))

### getMinSelectionIndex

int **getMinSelectionIndex**()

Returns the first selected index or -1 if the selection is empty.

### getMaxSelectionIndex

int **getMaxSelectionIndex**()

Returns the last selected index or -1 if the selection is empty.

### isSelectedIndex

boolean **isSelectedIndex**(int index)

Returns true if the specified index is selected.

### getAnchorSelectionIndex

int **getAnchorSelectionIndex**()

Return the first index argument from the most recent call to setSelectionInterval(), addSelectionInterval() or removeSelectionInterval(). The most recent index0 is considered the "anchor" and the most recent index1 is considered the "lead". Some interfaces display these indices specially, e.g. Windows95 displays the lead index with a dotted yellow outline.

**See Also:**[getLeadSelectionIndex()](http://docs.google.com/javax/swing/ListSelectionModel.html#getLeadSelectionIndex()), [setSelectionInterval(int, int)](http://docs.google.com/javax/swing/ListSelectionModel.html#setSelectionInterval(int,%20int)), [addSelectionInterval(int, int)](http://docs.google.com/javax/swing/ListSelectionModel.html#addSelectionInterval(int,%20int))

### setAnchorSelectionIndex

void **setAnchorSelectionIndex**(int index)

Set the anchor selection index.

**See Also:**[getAnchorSelectionIndex()](http://docs.google.com/javax/swing/ListSelectionModel.html#getAnchorSelectionIndex())

### getLeadSelectionIndex

int **getLeadSelectionIndex**()

Return the second index argument from the most recent call to setSelectionInterval(), addSelectionInterval() or removeSelectionInterval().

**See Also:**[getAnchorSelectionIndex()](http://docs.google.com/javax/swing/ListSelectionModel.html#getAnchorSelectionIndex()), [setSelectionInterval(int, int)](http://docs.google.com/javax/swing/ListSelectionModel.html#setSelectionInterval(int,%20int)), [addSelectionInterval(int, int)](http://docs.google.com/javax/swing/ListSelectionModel.html#addSelectionInterval(int,%20int))

### setLeadSelectionIndex

void **setLeadSelectionIndex**(int index)

Set the lead selection index.

**See Also:**[getLeadSelectionIndex()](http://docs.google.com/javax/swing/ListSelectionModel.html#getLeadSelectionIndex())

### clearSelection

void **clearSelection**()

Change the selection to the empty set. If this represents a change to the current selection then notify each ListSelectionListener.

**See Also:**[addListSelectionListener(javax.swing.event.ListSelectionListener)](http://docs.google.com/javax/swing/ListSelectionModel.html#addListSelectionListener(javax.swing.event.ListSelectionListener))

### isSelectionEmpty

boolean **isSelectionEmpty**()

Returns true if no indices are selected.

### insertIndexInterval

void **insertIndexInterval**(int index,  
 int length,  
 boolean before)

Insert length indices beginning before/after index. This is typically called to sync the selection model with a corresponding change in the data model.

### removeIndexInterval

void **removeIndexInterval**(int index0,  
 int index1)

Remove the indices in the interval index0,index1 (inclusive) from the selection model. This is typically called to sync the selection model width a corresponding change in the data model.

### setValueIsAdjusting

void **setValueIsAdjusting**(boolean valueIsAdjusting)

Sets the valueIsAdjusting property, which indicates whether or not upcoming selection changes should be considered part of a single change. The value of this property is used to initialize the valueIsAdjusting property of the ListSelectionEvents that are generated.

For example, if the selection is being updated in response to a user drag, this property can be set to true when the drag is initiated and set to false when the drag is finished. During the drag, listeners receive events with a valueIsAdjusting property set to true. At the end of the drag, when the change is finalized, listeners receive an event with the value set to false. Listeners can use this pattern if they wish to update only when a change has been finalized.

Setting this property to true begins a series of changes that is to be considered part of a single change. When the property is changed back to false, an event is sent out characterizing the entire selection change (if there was one), with the event's valueIsAdjusting property set to false.

**Parameters:**valueIsAdjusting - the new value of the property**See Also:**[getValueIsAdjusting()](http://docs.google.com/javax/swing/ListSelectionModel.html#getValueIsAdjusting()), [ListSelectionEvent.getValueIsAdjusting()](http://docs.google.com/javax/swing/event/ListSelectionEvent.html#getValueIsAdjusting())

### getValueIsAdjusting

boolean **getValueIsAdjusting**()

Returns true if the selection is undergoing a series of changes.

**Returns:**true if the selection is undergoing a series of changes**See Also:**[setValueIsAdjusting(boolean)](http://docs.google.com/javax/swing/ListSelectionModel.html#setValueIsAdjusting(boolean))

### setSelectionMode

void **setSelectionMode**(int selectionMode)

Sets the selection mode. The following list describes the accepted selection modes:

* ListSelectionModel.SINGLE\_SELECTION - Only one list index can be selected at a time. In this mode, setSelectionInterval and addSelectionInterval are equivalent, both replacing the current selection with the index represented by the second argument (the "lead").
* ListSelectionModel.SINGLE\_INTERVAL\_SELECTION - Only one contiguous interval can be selected at a time. In this mode, addSelectionInterval behaves like setSelectionInterval (replacing the current selection), unless the given interval is immediately adjacent to or overlaps the existing selection, and can therefore be used to grow it.
* ListSelectionModel.MULTIPLE\_INTERVAL\_SELECTION - In this mode, there's no restriction on what can be selected.

**Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if the selection mode isn't one of those allowed**See Also:**[getSelectionMode()](http://docs.google.com/javax/swing/ListSelectionModel.html#getSelectionMode())

### getSelectionMode

int **getSelectionMode**()

Returns the current selection mode.

**Returns:**the current selection mode**See Also:**[setSelectionMode(int)](http://docs.google.com/javax/swing/ListSelectionModel.html#setSelectionMode(int))

### addListSelectionListener

void **addListSelectionListener**([ListSelectionListener](http://docs.google.com/javax/swing/event/ListSelectionListener.html) x)

Add a listener to the list that's notified each time a change to the selection occurs.

**Parameters:**x - the ListSelectionListener**See Also:**[removeListSelectionListener(javax.swing.event.ListSelectionListener)](http://docs.google.com/javax/swing/ListSelectionModel.html#removeListSelectionListener(javax.swing.event.ListSelectionListener)), [setSelectionInterval(int, int)](http://docs.google.com/javax/swing/ListSelectionModel.html#setSelectionInterval(int,%20int)), [addSelectionInterval(int, int)](http://docs.google.com/javax/swing/ListSelectionModel.html#addSelectionInterval(int,%20int)), [removeSelectionInterval(int, int)](http://docs.google.com/javax/swing/ListSelectionModel.html#removeSelectionInterval(int,%20int)), [clearSelection()](http://docs.google.com/javax/swing/ListSelectionModel.html#clearSelection()), [insertIndexInterval(int, int, boolean)](http://docs.google.com/javax/swing/ListSelectionModel.html#insertIndexInterval(int,%20int,%20boolean)), [removeIndexInterval(int, int)](http://docs.google.com/javax/swing/ListSelectionModel.html#removeIndexInterval(int,%20int))

### removeListSelectionListener

void **removeListSelectionListener**([ListSelectionListener](http://docs.google.com/javax/swing/event/ListSelectionListener.html) x)

Remove a listener from the list that's notified each time a change to the selection occurs.

**Parameters:**x - the ListSelectionListener**See Also:**[addListSelectionListener(javax.swing.event.ListSelectionListener)](http://docs.google.com/javax/swing/ListSelectionModel.html#addListSelectionListener(javax.swing.event.ListSelectionListener))

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