| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/DefaultListSelectionModel.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/DefaultListModel.html)   [**NEXT CLASS**](http://docs.google.com/javax/swing/DefaultRowSorter.html) | [**FRAMES**](http://docs.google.com/index.html?javax/swing/DefaultListSelectionModel.html)    [**NO FRAMES**](http://docs.google.com/DefaultListSelectionModel.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#3znysh7) | [CONSTR](#tyjcwt) | [METHOD](#3dy6vkm) | DETAIL: [FIELD](#4d34og8) | [CONSTR](#3rdcrjn) | [METHOD](#lnxbz9) |

## **javax.swing**

Class DefaultListSelectionModel

[java.lang.Object](http://docs.google.com/java/lang/Object.html)  
 **javax.swing.DefaultListSelectionModel**

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

public class **DefaultListSelectionModel**extends [Object](http://docs.google.com/java/lang/Object.html)implements [ListSelectionModel](http://docs.google.com/javax/swing/ListSelectionModel.html), [Cloneable](http://docs.google.com/java/lang/Cloneable.html), [Serializable](http://docs.google.com/java/io/Serializable.html)

Default data model for list selections.

**Warning:** Serialized objects of this class will not be compatible with future Swing releases. The current serialization support is appropriate for short term storage or RMI between applications running the same version of Swing. As of 1.4, support for long term storage of all JavaBeansTM has been added to the java.beans package. Please see [XMLEncoder](http://docs.google.com/java/beans/XMLEncoder.html).

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

| **Field Summary** | |
| --- | --- |
| protected  boolean | [**leadAnchorNotificationEnabled**](http://docs.google.com/javax/swing/DefaultListSelectionModel.html#leadAnchorNotificationEnabled) |
| protected  [EventListenerList](http://docs.google.com/javax/swing/event/EventListenerList.html) | [**listenerList**](http://docs.google.com/javax/swing/DefaultListSelectionModel.html#listenerList) |

| **Fields inherited from interface javax.swing.**[**ListSelectionModel**](http://docs.google.com/javax/swing/ListSelectionModel.html) |
| --- |
| [MULTIPLE\_INTERVAL\_SELECTION](http://docs.google.com/javax/swing/ListSelectionModel.html#MULTIPLE_INTERVAL_SELECTION), [SINGLE\_INTERVAL\_SELECTION](http://docs.google.com/javax/swing/ListSelectionModel.html#SINGLE_INTERVAL_SELECTION), [SINGLE\_SELECTION](http://docs.google.com/javax/swing/ListSelectionModel.html#SINGLE_SELECTION) |

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

| **Method Summary** | |
| --- | --- |
| void | [**addListSelectionListener**](http://docs.google.com/javax/swing/DefaultListSelectionModel.html#addListSelectionListener(javax.swing.event.ListSelectionListener))([ListSelectionListener](http://docs.google.com/javax/swing/event/ListSelectionListener.html) l)            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/DefaultListSelectionModel.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/DefaultListSelectionModel.html#clearSelection())()            Change the selection to the empty set. |
| [Object](http://docs.google.com/java/lang/Object.html) | [**clone**](http://docs.google.com/javax/swing/DefaultListSelectionModel.html#clone())()            Returns a clone of this selection model with the same selection. |
| protected  void | [**fireValueChanged**](http://docs.google.com/javax/swing/DefaultListSelectionModel.html#fireValueChanged(boolean))(boolean isAdjusting)            Notifies listeners that we have ended a series of adjustments. |
| protected  void | [**fireValueChanged**](http://docs.google.com/javax/swing/DefaultListSelectionModel.html#fireValueChanged(int,%20int))(int firstIndex, int lastIndex)            Notifies ListSelectionListeners that the value of the selection, in the closed interval firstIndex, lastIndex, has changed. |
| protected  void | [**fireValueChanged**](http://docs.google.com/javax/swing/DefaultListSelectionModel.html#fireValueChanged(int,%20int,%20boolean))(int firstIndex, int lastIndex, boolean isAdjusting) |
| int | [**getAnchorSelectionIndex**](http://docs.google.com/javax/swing/DefaultListSelectionModel.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/DefaultListSelectionModel.html#getLeadSelectionIndex())()            Return the second index argument from the most recent call to setSelectionInterval(), addSelectionInterval() or removeSelectionInterval(). |
| | <T extends [EventListener](http://docs.google.com/java/util/EventListener.html)>  T[] | | --- | | [**getListeners**](http://docs.google.com/javax/swing/DefaultListSelectionModel.html#getListeners(java.lang.Class))([Class](http://docs.google.com/java/lang/Class.html)<T> listenerType)            Returns an array of all the objects currently registered as *Foo*Listeners upon this model. |
| [ListSelectionListener](http://docs.google.com/javax/swing/event/ListSelectionListener.html)[] | [**getListSelectionListeners**](http://docs.google.com/javax/swing/DefaultListSelectionModel.html#getListSelectionListeners())()            Returns an array of all the list selection listeners registered on this DefaultListSelectionModel. |
| int | [**getMaxSelectionIndex**](http://docs.google.com/javax/swing/DefaultListSelectionModel.html#getMaxSelectionIndex())()            Returns the last selected index or -1 if the selection is empty. |
| int | [**getMinSelectionIndex**](http://docs.google.com/javax/swing/DefaultListSelectionModel.html#getMinSelectionIndex())()            Returns the first selected index or -1 if the selection is empty. |
| int | [**getSelectionMode**](http://docs.google.com/javax/swing/DefaultListSelectionModel.html#getSelectionMode())()            Returns the current selection mode. |
| boolean | [**getValueIsAdjusting**](http://docs.google.com/javax/swing/DefaultListSelectionModel.html#getValueIsAdjusting())()            Returns true if the selection is undergoing a series of changes. |
| void | [**insertIndexInterval**](http://docs.google.com/javax/swing/DefaultListSelectionModel.html#insertIndexInterval(int,%20int,%20boolean))(int index, int length, boolean before)            Insert length indices beginning before/after index. |
| boolean | [**isLeadAnchorNotificationEnabled**](http://docs.google.com/javax/swing/DefaultListSelectionModel.html#isLeadAnchorNotificationEnabled())()            Returns the value of the leadAnchorNotificationEnabled flag. |
| boolean | [**isSelectedIndex**](http://docs.google.com/javax/swing/DefaultListSelectionModel.html#isSelectedIndex(int))(int index)            Returns true if the specified index is selected. |
| boolean | [**isSelectionEmpty**](http://docs.google.com/javax/swing/DefaultListSelectionModel.html#isSelectionEmpty())()            Returns true if no indices are selected. |
| void | [**moveLeadSelectionIndex**](http://docs.google.com/javax/swing/DefaultListSelectionModel.html#moveLeadSelectionIndex(int))(int leadIndex)            Set the lead selection index, leaving all selection values unchanged. |
| void | [**removeIndexInterval**](http://docs.google.com/javax/swing/DefaultListSelectionModel.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/DefaultListSelectionModel.html#removeListSelectionListener(javax.swing.event.ListSelectionListener))([ListSelectionListener](http://docs.google.com/javax/swing/event/ListSelectionListener.html) l)            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/DefaultListSelectionModel.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/DefaultListSelectionModel.html#setAnchorSelectionIndex(int))(int anchorIndex)            Set the anchor selection index, leaving all selection values unchanged. |
| void | [**setLeadAnchorNotificationEnabled**](http://docs.google.com/javax/swing/DefaultListSelectionModel.html#setLeadAnchorNotificationEnabled(boolean))(boolean flag)            Sets the value of the leadAnchorNotificationEnabled flag. |
| void | [**setLeadSelectionIndex**](http://docs.google.com/javax/swing/DefaultListSelectionModel.html#setLeadSelectionIndex(int))(int leadIndex)            Sets the lead selection index, ensuring that values between the anchor and the new lead are either all selected or all deselected. |
| void | [**setSelectionInterval**](http://docs.google.com/javax/swing/DefaultListSelectionModel.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/DefaultListSelectionModel.html#setSelectionMode(int))(int selectionMode)            Sets the selection mode. |
| void | [**setValueIsAdjusting**](http://docs.google.com/javax/swing/DefaultListSelectionModel.html#setValueIsAdjusting(boolean))(boolean isAdjusting)            Sets the valueIsAdjusting property, which indicates whether or not upcoming selection changes should be considered part of a single change. |
| [String](http://docs.google.com/java/lang/String.html) | [**toString**](http://docs.google.com/javax/swing/DefaultListSelectionModel.html#toString())()            Returns a string that displays and identifies this object's properties. |

| **Methods inherited from class java.lang.**[**Object**](http://docs.google.com/java/lang/Object.html) |
| --- |
| [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()), [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)) |

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

### listenerList

protected [EventListenerList](http://docs.google.com/javax/swing/event/EventListenerList.html) **listenerList**

### leadAnchorNotificationEnabled

protected boolean **leadAnchorNotificationEnabled**

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

### DefaultListSelectionModel

public **DefaultListSelectionModel**()

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

### getMinSelectionIndex

public int **getMinSelectionIndex**()

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

**Specified by:**[getMinSelectionIndex](http://docs.google.com/javax/swing/ListSelectionModel.html#getMinSelectionIndex()) in interface [ListSelectionModel](http://docs.google.com/javax/swing/ListSelectionModel.html)

### getMaxSelectionIndex

public int **getMaxSelectionIndex**()

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

**Specified by:**[getMaxSelectionIndex](http://docs.google.com/javax/swing/ListSelectionModel.html#getMaxSelectionIndex()) in interface [ListSelectionModel](http://docs.google.com/javax/swing/ListSelectionModel.html)

### getValueIsAdjusting

public boolean **getValueIsAdjusting**()

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

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

### getSelectionMode

public int **getSelectionMode**()

Returns the current selection mode.

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

### setSelectionMode

public 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.

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

### isSelectedIndex

public boolean **isSelectedIndex**(int index)

Returns true if the specified index is selected.

**Specified by:**[isSelectedIndex](http://docs.google.com/javax/swing/ListSelectionModel.html#isSelectedIndex(int)) in interface [ListSelectionModel](http://docs.google.com/javax/swing/ListSelectionModel.html)

### isSelectionEmpty

public boolean **isSelectionEmpty**()

Returns true if no indices are selected.

**Specified by:**[isSelectionEmpty](http://docs.google.com/javax/swing/ListSelectionModel.html#isSelectionEmpty()) in interface [ListSelectionModel](http://docs.google.com/javax/swing/ListSelectionModel.html)

### addListSelectionListener

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

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

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

### removeListSelectionListener

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

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

**Specified by:**[removeListSelectionListener](http://docs.google.com/javax/swing/ListSelectionModel.html#removeListSelectionListener(javax.swing.event.ListSelectionListener)) in interface [ListSelectionModel](http://docs.google.com/javax/swing/ListSelectionModel.html) **Parameters:**l - the ListSelectionListener**See Also:**[ListSelectionModel.addListSelectionListener(javax.swing.event.ListSelectionListener)](http://docs.google.com/javax/swing/ListSelectionModel.html#addListSelectionListener(javax.swing.event.ListSelectionListener))

### getListSelectionListeners

public [ListSelectionListener](http://docs.google.com/javax/swing/event/ListSelectionListener.html)[] **getListSelectionListeners**()

Returns an array of all the list selection listeners registered on this DefaultListSelectionModel.

**Returns:**all of this model's ListSelectionListeners or an empty array if no list selection listeners are currently registered**Since:** 1.4 **See Also:**[addListSelectionListener(javax.swing.event.ListSelectionListener)](http://docs.google.com/javax/swing/DefaultListSelectionModel.html#addListSelectionListener(javax.swing.event.ListSelectionListener)), [removeListSelectionListener(javax.swing.event.ListSelectionListener)](http://docs.google.com/javax/swing/DefaultListSelectionModel.html#removeListSelectionListener(javax.swing.event.ListSelectionListener))

### fireValueChanged

protected void **fireValueChanged**(boolean isAdjusting)

Notifies listeners that we have ended a series of adjustments.

### fireValueChanged

protected void **fireValueChanged**(int firstIndex,  
 int lastIndex)

Notifies ListSelectionListeners that the value of the selection, in the closed interval firstIndex, lastIndex, has changed.

### fireValueChanged

protected void **fireValueChanged**(int firstIndex,  
 int lastIndex,  
 boolean isAdjusting)

**Parameters:**firstIndex - the first index in the intervallastIndex - the last index in the intervalisAdjusting - true if this is the final change in a series of adjustments**See Also:**[EventListenerList](http://docs.google.com/javax/swing/event/EventListenerList.html)

### getListeners

public <T extends [EventListener](http://docs.google.com/java/util/EventListener.html)> T[] **getListeners**([Class](http://docs.google.com/java/lang/Class.html)<T> listenerType)

Returns an array of all the objects currently registered as *Foo*Listeners upon this model. *Foo*Listeners are registered using the add*Foo*Listener method.

You can specify the listenerType argument with a class literal, such as *Foo*Listener.class. For example, you can query a DefaultListSelectionModel instance m for its list selection listeners with the following code:

ListSelectionListener[] lsls = (ListSelectionListener[])(m.getListeners(ListSelectionListener.class));

If no such listeners exist, this method returns an empty array.

**Parameters:**listenerType - the type of listeners requested; this parameter should specify an interface that descends from java.util.EventListener **Returns:**an array of all objects registered as *Foo*Listeners on this model, or an empty array if no such listeners have been added **Throws:** [ClassCastException](http://docs.google.com/java/lang/ClassCastException.html) - if listenerType doesn't specify a class or interface that implements java.util.EventListener**Since:** 1.3 **See Also:**[getListSelectionListeners()](http://docs.google.com/javax/swing/DefaultListSelectionModel.html#getListSelectionListeners())

### setLeadAnchorNotificationEnabled

public void **setLeadAnchorNotificationEnabled**(boolean flag)

Sets the value of the leadAnchorNotificationEnabled flag.

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

### isLeadAnchorNotificationEnabled

public boolean **isLeadAnchorNotificationEnabled**()

Returns the value of the leadAnchorNotificationEnabled flag. When leadAnchorNotificationEnabled is true the model generates notification events with bounds that cover all the changes to the selection plus the changes to the lead and anchor indices. Setting the flag to false causes a narrowing of the event's bounds to include only the elements that have been selected or deselected since the last change. Either way, the model continues to maintain the lead and anchor variables internally. The default is true.

Note: It is possible for the lead or anchor to be changed without a change to the selection. Notification of these changes is often important, such as when the new lead or anchor needs to be updated in the view. Therefore, caution is urged when changing the default value.

**Returns:**the value of the leadAnchorNotificationEnabled flag**See Also:**[setLeadAnchorNotificationEnabled(boolean)](http://docs.google.com/javax/swing/DefaultListSelectionModel.html#setLeadAnchorNotificationEnabled(boolean))

### clearSelection

public void **clearSelection**()

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

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

### setSelectionInterval

public 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.

If either index is -1, this method does nothing and returns without exception. Otherwise, if either index is less than -1, an IndexOutOfBoundsException is thrown.

**Specified by:**[setSelectionInterval](http://docs.google.com/javax/swing/ListSelectionModel.html#setSelectionInterval(int,%20int)) in interface [ListSelectionModel](http://docs.google.com/javax/swing/ListSelectionModel.html) **Parameters:**index0 - one end of the interval.index1 - other end of the interval **Throws:** [IndexOutOfBoundsException](http://docs.google.com/java/lang/IndexOutOfBoundsException.html) - if either index is less than -1 (and neither index is -1)**See Also:**[addListSelectionListener(javax.swing.event.ListSelectionListener)](http://docs.google.com/javax/swing/DefaultListSelectionModel.html#addListSelectionListener(javax.swing.event.ListSelectionListener))

### addSelectionInterval

public 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.

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 it.

If this represents a change to the current selection, then each ListSelectionListener is notified of the change. Note that index0 doesn't have to be less than or equal to index1.

If either index is -1, this method does nothing and returns without exception. Otherwise, if either index is less than -1, an IndexOutOfBoundsException is thrown.

**Specified by:**[addSelectionInterval](http://docs.google.com/javax/swing/ListSelectionModel.html#addSelectionInterval(int,%20int)) in interface [ListSelectionModel](http://docs.google.com/javax/swing/ListSelectionModel.html) **Parameters:**index0 - one end of the interval.index1 - other end of the interval **Throws:** [IndexOutOfBoundsException](http://docs.google.com/java/lang/IndexOutOfBoundsException.html) - if either index is less than -1 (and neither index is -1)**See Also:**[addListSelectionListener(javax.swing.event.ListSelectionListener)](http://docs.google.com/javax/swing/DefaultListSelectionModel.html#addListSelectionListener(javax.swing.event.ListSelectionListener)), [setSelectionInterval(int, int)](http://docs.google.com/javax/swing/DefaultListSelectionModel.html#setSelectionInterval(int,%20int))

### removeSelectionInterval

public 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.

If either index is -1, this method does nothing and returns without exception. Otherwise, if either index is less than -1, an IndexOutOfBoundsException is thrown.

**Specified by:**[removeSelectionInterval](http://docs.google.com/javax/swing/ListSelectionModel.html#removeSelectionInterval(int,%20int)) in interface [ListSelectionModel](http://docs.google.com/javax/swing/ListSelectionModel.html) **Parameters:**index0 - one end of the intervalindex1 - other end of the interval **Throws:** [IndexOutOfBoundsException](http://docs.google.com/java/lang/IndexOutOfBoundsException.html) - if either index is less than -1 (and neither index is -1)**See Also:**[addListSelectionListener(javax.swing.event.ListSelectionListener)](http://docs.google.com/javax/swing/DefaultListSelectionModel.html#addListSelectionListener(javax.swing.event.ListSelectionListener))

### insertIndexInterval

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

Insert length indices beginning before/after index. If the value at index is itself selected and the selection mode is not SINGLE\_SELECTION, set all of the newly inserted items as selected. Otherwise leave them unselected. This method is typically called to sync the selection model with a corresponding change in the data model.

**Specified by:**[insertIndexInterval](http://docs.google.com/javax/swing/ListSelectionModel.html#insertIndexInterval(int,%20int,%20boolean)) in interface [ListSelectionModel](http://docs.google.com/javax/swing/ListSelectionModel.html)

### removeIndexInterval

public 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. Note that (as always) index0 need not be <= index1.

**Specified by:**[removeIndexInterval](http://docs.google.com/javax/swing/ListSelectionModel.html#removeIndexInterval(int,%20int)) in interface [ListSelectionModel](http://docs.google.com/javax/swing/ListSelectionModel.html)

### setValueIsAdjusting

public void **setValueIsAdjusting**(boolean isAdjusting)

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.

**Specified by:**[setValueIsAdjusting](http://docs.google.com/javax/swing/ListSelectionModel.html#setValueIsAdjusting(boolean)) in interface [ListSelectionModel](http://docs.google.com/javax/swing/ListSelectionModel.html) **Parameters:**isAdjusting - the new value of the property**See Also:**[ListSelectionModel.getValueIsAdjusting()](http://docs.google.com/javax/swing/ListSelectionModel.html#getValueIsAdjusting()), [ListSelectionEvent.getValueIsAdjusting()](http://docs.google.com/javax/swing/event/ListSelectionEvent.html#getValueIsAdjusting())

### toString

public [String](http://docs.google.com/java/lang/String.html) **toString**()

Returns a string that displays and identifies this object's properties.

**Overrides:**[toString](http://docs.google.com/java/lang/Object.html#toString()) in class [Object](http://docs.google.com/java/lang/Object.html) **Returns:**a String representation of this object

### clone

public [Object](http://docs.google.com/java/lang/Object.html) **clone**()  
 throws [CloneNotSupportedException](http://docs.google.com/java/lang/CloneNotSupportedException.html)

Returns a clone of this selection model with the same selection. listenerLists are not duplicated.

**Overrides:**[clone](http://docs.google.com/java/lang/Object.html#clone()) in class [Object](http://docs.google.com/java/lang/Object.html) **Returns:**a clone of this instance. **Throws:** [CloneNotSupportedException](http://docs.google.com/java/lang/CloneNotSupportedException.html) - if the selection model does not both (a) implement the Cloneable interface and (b) define a clone method.**See Also:**[Cloneable](http://docs.google.com/java/lang/Cloneable.html)

### getAnchorSelectionIndex

public 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.

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

### getLeadSelectionIndex

public int **getLeadSelectionIndex**()

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

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

### setAnchorSelectionIndex

public void **setAnchorSelectionIndex**(int anchorIndex)

Set the anchor selection index, leaving all selection values unchanged. If leadAnchorNotificationEnabled is true, send a notification covering the old and new anchor cells.

**Specified by:**[setAnchorSelectionIndex](http://docs.google.com/javax/swing/ListSelectionModel.html#setAnchorSelectionIndex(int)) in interface [ListSelectionModel](http://docs.google.com/javax/swing/ListSelectionModel.html) **See Also:**[getAnchorSelectionIndex()](http://docs.google.com/javax/swing/DefaultListSelectionModel.html#getAnchorSelectionIndex()), [setLeadSelectionIndex(int)](http://docs.google.com/javax/swing/DefaultListSelectionModel.html#setLeadSelectionIndex(int))

### moveLeadSelectionIndex

public void **moveLeadSelectionIndex**(int leadIndex)

Set the lead selection index, leaving all selection values unchanged. If leadAnchorNotificationEnabled is true, send a notification covering the old and new lead cells.

**Parameters:**leadIndex - the new lead selection index**Since:** 1.5 **See Also:**[setAnchorSelectionIndex(int)](http://docs.google.com/javax/swing/DefaultListSelectionModel.html#setAnchorSelectionIndex(int)), [setLeadSelectionIndex(int)](http://docs.google.com/javax/swing/DefaultListSelectionModel.html#setLeadSelectionIndex(int)), [getLeadSelectionIndex()](http://docs.google.com/javax/swing/DefaultListSelectionModel.html#getLeadSelectionIndex())

### setLeadSelectionIndex

public void **setLeadSelectionIndex**(int leadIndex)

Sets the lead selection index, ensuring that values between the anchor and the new lead are either all selected or all deselected. If the value at the anchor index is selected, first clear all the values in the range [anchor, oldLeadIndex], then select all the values values in the range [anchor, newLeadIndex], where oldLeadIndex is the old leadIndex and newLeadIndex is the new one.

If the value at the anchor index is not selected, do the same thing in reverse selecting values in the old range and deslecting values in the new one.

Generate a single event for this change and notify all listeners. For the purposes of generating minimal bounds in this event, do the operation in a single pass; that way the first and last index inside the ListSelectionEvent that is broadcast will refer to cells that actually changed value because of this method. If, instead, this operation were done in two steps the effect on the selection state would be the same but two events would be generated and the bounds around the changed values would be wider, including cells that had been first cleared only to later be set.

This method can be used in the mouseDragged method of a UI class to extend a selection.

**Specified by:**[setLeadSelectionIndex](http://docs.google.com/javax/swing/ListSelectionModel.html#setLeadSelectionIndex(int)) in interface [ListSelectionModel](http://docs.google.com/javax/swing/ListSelectionModel.html) **See Also:**[getLeadSelectionIndex()](http://docs.google.com/javax/swing/DefaultListSelectionModel.html#getLeadSelectionIndex()), [setAnchorSelectionIndex(int)](http://docs.google.com/javax/swing/DefaultListSelectionModel.html#setAnchorSelectionIndex(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/DefaultListSelectionModel.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/DefaultListModel.html)   [**NEXT CLASS**](http://docs.google.com/javax/swing/DefaultRowSorter.html) | [**FRAMES**](http://docs.google.com/index.html?javax/swing/DefaultListSelectionModel.html)    [**NO FRAMES**](http://docs.google.com/DefaultListSelectionModel.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#3znysh7) | [CONSTR](#tyjcwt) | [METHOD](#3dy6vkm) | DETAIL: [FIELD](#4d34og8) | [CONSTR](#3rdcrjn) | [METHOD](#lnxbz9) |

[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).