| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/VetoableChangeSupport.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/java/beans/VetoableChangeListenerProxy.html)   [**NEXT CLASS**](http://docs.google.com/java/beans/Visibility.html) | [**FRAMES**](http://docs.google.com/index.html?java/beans/VetoableChangeSupport.html)    [**NO FRAMES**](http://docs.google.com/VetoableChangeSupport.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | [CONSTR](#3znysh7) | [METHOD](#2et92p0) | DETAIL: FIELD | [CONSTR](#3dy6vkm) | [METHOD](#4d34og8) |

## **java.beans**

Class VetoableChangeSupport

[java.lang.Object](http://docs.google.com/java/lang/Object.html)  
 **java.beans.VetoableChangeSupport**

**All Implemented Interfaces:** [Serializable](http://docs.google.com/java/io/Serializable.html)

public class **VetoableChangeSupport**extends [Object](http://docs.google.com/java/lang/Object.html)implements [Serializable](http://docs.google.com/java/io/Serializable.html)

This is a utility class that can be used by beans that support constrained properties. You can use an instance of this class as a member field of your bean and delegate various work to it. This class is serializable. When it is serialized it will save (and restore) any listeners that are themselves serializable. Any non-serializable listeners will be skipped during serialization.

**See Also:**[Serialized Form](http://docs.google.com/serialized-form.html#java.beans.VetoableChangeSupport)

| **Constructor Summary** | |
| --- | --- |
| [**VetoableChangeSupport**](http://docs.google.com/java/beans/VetoableChangeSupport.html#VetoableChangeSupport(java.lang.Object))([Object](http://docs.google.com/java/lang/Object.html) sourceBean)            Constructs a VetoableChangeSupport object. |

| **Method Summary** | |
| --- | --- |
| void | [**addVetoableChangeListener**](http://docs.google.com/java/beans/VetoableChangeSupport.html#addVetoableChangeListener(java.lang.String,%20java.beans.VetoableChangeListener))([String](http://docs.google.com/java/lang/String.html) propertyName, [VetoableChangeListener](http://docs.google.com/java/beans/VetoableChangeListener.html) listener)            Add a VetoableChangeListener for a specific property. |
| void | [**addVetoableChangeListener**](http://docs.google.com/java/beans/VetoableChangeSupport.html#addVetoableChangeListener(java.beans.VetoableChangeListener))([VetoableChangeListener](http://docs.google.com/java/beans/VetoableChangeListener.html) listener)            Add a VetoableListener to the listener list. |
| void | [**fireVetoableChange**](http://docs.google.com/java/beans/VetoableChangeSupport.html#fireVetoableChange(java.beans.PropertyChangeEvent))([PropertyChangeEvent](http://docs.google.com/java/beans/PropertyChangeEvent.html) evt)            Fire a vetoable property update to any registered listeners. |
| void | [**fireVetoableChange**](http://docs.google.com/java/beans/VetoableChangeSupport.html#fireVetoableChange(java.lang.String,%20boolean,%20boolean))([String](http://docs.google.com/java/lang/String.html) propertyName, boolean oldValue, boolean newValue)            Report a boolean vetoable property update to any registered listeners. |
| void | [**fireVetoableChange**](http://docs.google.com/java/beans/VetoableChangeSupport.html#fireVetoableChange(java.lang.String,%20int,%20int))([String](http://docs.google.com/java/lang/String.html) propertyName, int oldValue, int newValue)            Report a int vetoable property update to any registered listeners. |
| void | [**fireVetoableChange**](http://docs.google.com/java/beans/VetoableChangeSupport.html#fireVetoableChange(java.lang.String,%20java.lang.Object,%20java.lang.Object))([String](http://docs.google.com/java/lang/String.html) propertyName, [Object](http://docs.google.com/java/lang/Object.html) oldValue, [Object](http://docs.google.com/java/lang/Object.html) newValue)            Report a vetoable property update to any registered listeners. |
| [VetoableChangeListener](http://docs.google.com/java/beans/VetoableChangeListener.html)[] | [**getVetoableChangeListeners**](http://docs.google.com/java/beans/VetoableChangeSupport.html#getVetoableChangeListeners())()            Returns the list of VetoableChangeListeners. |
| [VetoableChangeListener](http://docs.google.com/java/beans/VetoableChangeListener.html)[] | [**getVetoableChangeListeners**](http://docs.google.com/java/beans/VetoableChangeSupport.html#getVetoableChangeListeners(java.lang.String))([String](http://docs.google.com/java/lang/String.html) propertyName)            Returns an array of all the listeners which have been associated with the named property. |
| boolean | [**hasListeners**](http://docs.google.com/java/beans/VetoableChangeSupport.html#hasListeners(java.lang.String))([String](http://docs.google.com/java/lang/String.html) propertyName)            Check if there are any listeners for a specific property, including those registered on all properties. |
| void | [**removeVetoableChangeListener**](http://docs.google.com/java/beans/VetoableChangeSupport.html#removeVetoableChangeListener(java.lang.String,%20java.beans.VetoableChangeListener))([String](http://docs.google.com/java/lang/String.html) propertyName, [VetoableChangeListener](http://docs.google.com/java/beans/VetoableChangeListener.html) listener)            Remove a VetoableChangeListener for a specific property. |
| void | [**removeVetoableChangeListener**](http://docs.google.com/java/beans/VetoableChangeSupport.html#removeVetoableChangeListener(java.beans.VetoableChangeListener))([VetoableChangeListener](http://docs.google.com/java/beans/VetoableChangeListener.html) listener)            Remove a VetoableChangeListener from the listener list. |

| **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** |
| --- |

### VetoableChangeSupport

public **VetoableChangeSupport**([Object](http://docs.google.com/java/lang/Object.html) sourceBean)

Constructs a VetoableChangeSupport object.

**Parameters:**sourceBean - The bean to be given as the source for any events.

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

### addVetoableChangeListener

public void **addVetoableChangeListener**([VetoableChangeListener](http://docs.google.com/java/beans/VetoableChangeListener.html) listener)

Add a VetoableListener to the listener list. The listener is registered for all properties. The same listener object may be added more than once, and will be called as many times as it is added. If listener is null, no exception is thrown and no action is taken.

**Parameters:**listener - The VetoableChangeListener to be added

### removeVetoableChangeListener

public void **removeVetoableChangeListener**([VetoableChangeListener](http://docs.google.com/java/beans/VetoableChangeListener.html) listener)

Remove a VetoableChangeListener from the listener list. This removes a VetoableChangeListener that was registered for all properties. If listener was added more than once to the same event source, it will be notified one less time after being removed. If listener is null, or was never added, no exception is thrown and no action is taken.

**Parameters:**listener - The VetoableChangeListener to be removed

### getVetoableChangeListeners

public [VetoableChangeListener](http://docs.google.com/java/beans/VetoableChangeListener.html)[] **getVetoableChangeListeners**()

Returns the list of VetoableChangeListeners. If named vetoable change listeners were added, then VetoableChangeListenerProxy wrappers will returned

**Returns:**List of VetoableChangeListeners and VetoableChangeListenerProxys if named property change listeners were added.**Since:** 1.4

### addVetoableChangeListener

public void **addVetoableChangeListener**([String](http://docs.google.com/java/lang/String.html) propertyName,  
 [VetoableChangeListener](http://docs.google.com/java/beans/VetoableChangeListener.html) listener)

Add a VetoableChangeListener for a specific property. The listener will be invoked only when a call on fireVetoableChange names that specific property. The same listener object may be added more than once. For each property, the listener will be invoked the number of times it was added for that property. If propertyName or listener is null, no exception is thrown and no action is taken.

**Parameters:**propertyName - The name of the property to listen on.listener - The VetoableChangeListener to be added

### removeVetoableChangeListener

public void **removeVetoableChangeListener**([String](http://docs.google.com/java/lang/String.html) propertyName,  
 [VetoableChangeListener](http://docs.google.com/java/beans/VetoableChangeListener.html) listener)

Remove a VetoableChangeListener for a specific property. If listener was added more than once to the same event source for the specified property, it will be notified one less time after being removed. If propertyName is null, no exception is thrown and no action is taken. If listener is null, or was never added for the specified property, no exception is thrown and no action is taken.

**Parameters:**propertyName - The name of the property that was listened on.listener - The VetoableChangeListener to be removed

### getVetoableChangeListeners

public [VetoableChangeListener](http://docs.google.com/java/beans/VetoableChangeListener.html)[] **getVetoableChangeListeners**([String](http://docs.google.com/java/lang/String.html) propertyName)

Returns an array of all the listeners which have been associated with the named property.

**Parameters:**propertyName - The name of the property being listened to **Returns:**all the VetoableChangeListeners associated with the named property. If no such listeners have been added, or if propertyName is null, an empty array is returned.**Since:** 1.4

### fireVetoableChange

public void **fireVetoableChange**([String](http://docs.google.com/java/lang/String.html) propertyName,  
 [Object](http://docs.google.com/java/lang/Object.html) oldValue,  
 [Object](http://docs.google.com/java/lang/Object.html) newValue)  
 throws [PropertyVetoException](http://docs.google.com/java/beans/PropertyVetoException.html)

Report a vetoable property update to any registered listeners. If anyone vetos the change, then fire a new event reverting everyone to the old value and then rethrow the PropertyVetoException.

No event is fired if old and new are equal and non-null.

**Parameters:**propertyName - The programmatic name of the property that is about to change..oldValue - The old value of the property.newValue - The new value of the property. **Throws:** [PropertyVetoException](http://docs.google.com/java/beans/PropertyVetoException.html) - if the recipient wishes the property change to be rolled back.

### fireVetoableChange

public void **fireVetoableChange**([String](http://docs.google.com/java/lang/String.html) propertyName,  
 int oldValue,  
 int newValue)  
 throws [PropertyVetoException](http://docs.google.com/java/beans/PropertyVetoException.html)

Report a int vetoable property update to any registered listeners. No event is fired if old and new are equal.

This is merely a convenience wrapper around the more general fireVetoableChange method that takes Object values.

**Parameters:**propertyName - The programmatic name of the property that is about to change.oldValue - The old value of the property.newValue - The new value of the property. **Throws:** [PropertyVetoException](http://docs.google.com/java/beans/PropertyVetoException.html)

### fireVetoableChange

public void **fireVetoableChange**([String](http://docs.google.com/java/lang/String.html) propertyName,  
 boolean oldValue,  
 boolean newValue)  
 throws [PropertyVetoException](http://docs.google.com/java/beans/PropertyVetoException.html)

Report a boolean vetoable property update to any registered listeners. No event is fired if old and new are equal.

This is merely a convenience wrapper around the more general fireVetoableChange method that takes Object values.

**Parameters:**propertyName - The programmatic name of the property that is about to change.oldValue - The old value of the property.newValue - The new value of the property. **Throws:** [PropertyVetoException](http://docs.google.com/java/beans/PropertyVetoException.html)

### fireVetoableChange

public void **fireVetoableChange**([PropertyChangeEvent](http://docs.google.com/java/beans/PropertyChangeEvent.html) evt)  
 throws [PropertyVetoException](http://docs.google.com/java/beans/PropertyVetoException.html)

Fire a vetoable property update to any registered listeners. If anyone vetos the change, then fire a new event reverting everyone to the old value and then rethrow the PropertyVetoException.

No event is fired if old and new are equal and non-null.

**Parameters:**evt - The PropertyChangeEvent to be fired. **Throws:** [PropertyVetoException](http://docs.google.com/java/beans/PropertyVetoException.html) - if the recipient wishes the property change to be rolled back.

### hasListeners

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

Check if there are any listeners for a specific property, including those registered on all properties. If propertyName is null, only check for listeners registered on all properties.

**Parameters:**propertyName - the property name. **Returns:**true if there are one or more listeners for the given property

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/VetoableChangeSupport.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/java/beans/VetoableChangeListenerProxy.html)   [**NEXT CLASS**](http://docs.google.com/java/beans/Visibility.html) | [**FRAMES**](http://docs.google.com/index.html?java/beans/VetoableChangeSupport.html)    [**NO FRAMES**](http://docs.google.com/VetoableChangeSupport.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | [CONSTR](#3znysh7) | [METHOD](#2et92p0) | DETAIL: FIELD | [CONSTR](#3dy6vkm) | [METHOD](#4d34og8) |

[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).