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

## **java.beans**

Interface PropertyEditor

**All Known Implementing Classes:** [PropertyEditorSupport](http://docs.google.com/java/beans/PropertyEditorSupport.html)

public interface **PropertyEditor**

A PropertyEditor class provides support for GUIs that want to allow users to edit a property value of a given type.

PropertyEditor supports a variety of different kinds of ways of displaying and updating property values. Most PropertyEditors will only need to support a subset of the different options available in this API.

Simple PropertyEditors may only support the getAsText and setAsText methods and need not support (say) paintValue or getCustomEditor. More complex types may be unable to support getAsText and setAsText but will instead support paintValue and getCustomEditor.

Every propertyEditor must support one or more of the three simple display styles. Thus it can either (1) support isPaintable or (2) both return a non-null String[] from getTags() and return a non-null value from getAsText or (3) simply return a non-null String from getAsText().

Every property editor must support a call on setValue when the argument object is of the type for which this is the corresponding propertyEditor. In addition, each property editor must either support a custom editor, or support setAsText.

Each PropertyEditor should have a null constructor.

| **Method Summary** | |
| --- | --- |
| void | [**addPropertyChangeListener**](http://docs.google.com/java/beans/PropertyEditor.html#addPropertyChangeListener(java.beans.PropertyChangeListener))([PropertyChangeListener](http://docs.google.com/java/beans/PropertyChangeListener.html) listener)            Register a listener for the PropertyChange event. |
| [String](http://docs.google.com/java/lang/String.html) | [**getAsText**](http://docs.google.com/java/beans/PropertyEditor.html#getAsText())()            Gets the property value as text. |
| [Component](http://docs.google.com/java/awt/Component.html) | [**getCustomEditor**](http://docs.google.com/java/beans/PropertyEditor.html#getCustomEditor())()            A PropertyEditor may choose to make available a full custom Component that edits its property value. |
| [String](http://docs.google.com/java/lang/String.html) | [**getJavaInitializationString**](http://docs.google.com/java/beans/PropertyEditor.html#getJavaInitializationString())()            Returns a fragment of Java code that can be used to set a property to match the editors current state. |
| [String](http://docs.google.com/java/lang/String.html)[] | [**getTags**](http://docs.google.com/java/beans/PropertyEditor.html#getTags())()            If the property value must be one of a set of known tagged values, then this method should return an array of the tags. |
| [Object](http://docs.google.com/java/lang/Object.html) | [**getValue**](http://docs.google.com/java/beans/PropertyEditor.html#getValue())()            Gets the property value. |
| boolean | [**isPaintable**](http://docs.google.com/java/beans/PropertyEditor.html#isPaintable())()            Determines whether this property editor is paintable. |
| void | [**paintValue**](http://docs.google.com/java/beans/PropertyEditor.html#paintValue(java.awt.Graphics,%20java.awt.Rectangle))([Graphics](http://docs.google.com/java/awt/Graphics.html) gfx, [Rectangle](http://docs.google.com/java/awt/Rectangle.html) box)            Paint a representation of the value into a given area of screen real estate. |
| void | [**removePropertyChangeListener**](http://docs.google.com/java/beans/PropertyEditor.html#removePropertyChangeListener(java.beans.PropertyChangeListener))([PropertyChangeListener](http://docs.google.com/java/beans/PropertyChangeListener.html) listener)            Remove a listener for the PropertyChange event. |
| void | [**setAsText**](http://docs.google.com/java/beans/PropertyEditor.html#setAsText(java.lang.String))([String](http://docs.google.com/java/lang/String.html) text)            Set the property value by parsing a given String. |
| void | [**setValue**](http://docs.google.com/java/beans/PropertyEditor.html#setValue(java.lang.Object))([Object](http://docs.google.com/java/lang/Object.html) value)            Set (or change) the object that is to be edited. |
| boolean | [**supportsCustomEditor**](http://docs.google.com/java/beans/PropertyEditor.html#supportsCustomEditor())()            Determines whether this property editor supports a custom editor. |

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

### setValue

void **setValue**([Object](http://docs.google.com/java/lang/Object.html) value)

Set (or change) the object that is to be edited. Primitive types such as "int" must be wrapped as the corresponding object type such as "java.lang.Integer".

**Parameters:**value - The new target object to be edited. Note that this object should not be modified by the PropertyEditor, rather the PropertyEditor should create a new object to hold any modified value.

### getValue

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

Gets the property value.

**Returns:**The value of the property. Primitive types such as "int" will be wrapped as the corresponding object type such as "java.lang.Integer".

### isPaintable

boolean **isPaintable**()

Determines whether this property editor is paintable.

**Returns:**True if the class will honor the paintValue method.

### paintValue

void **paintValue**([Graphics](http://docs.google.com/java/awt/Graphics.html) gfx,  
 [Rectangle](http://docs.google.com/java/awt/Rectangle.html) box)

Paint a representation of the value into a given area of screen real estate. Note that the propertyEditor is responsible for doing its own clipping so that it fits into the given rectangle.

If the PropertyEditor doesn't honor paint requests (see isPaintable) this method should be a silent noop.

The given Graphics object will have the default font, color, etc of the parent container. The PropertyEditor may change graphics attributes such as font and color and doesn't need to restore the old values.

**Parameters:**gfx - Graphics object to paint into.box - Rectangle within graphics object into which we should paint.

### getJavaInitializationString

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

Returns a fragment of Java code that can be used to set a property to match the editors current state. This method is intended for use when generating Java code to reflect changes made through the property editor.

The code fragment should be context free and must be a legal Java expression as specified by the JLS.

Specifically, if the expression represents a computation then all classes and static members should be fully qualified. This rule applies to constructors, static methods and non primitive arguments.

Caution should be used when evaluating the expression as it may throw exceptions. In particular, code generators must ensure that generated code will compile in the presence of an expression that can throw checked exceptions.

Example results are:

* Primitive expresssion: 2
* Class constructor: new java.awt.Color(127,127,34)
* Static field: java.awt.Color.orange
* Static method: javax.swing.Box.createRigidArea(new java.awt.Dimension(0, 5))

**Returns:**a fragment of Java code representing an initializer for the current value. It should not contain a semi-colon (';') to end the expression.

### getAsText

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

Gets the property value as text.

**Returns:**The property value as a human editable string.

Returns null if the value can't be expressed as an editable string.

If a non-null value is returned, then the PropertyEditor should be prepared to parse that string back in setAsText().

### setAsText

void **setAsText**([String](http://docs.google.com/java/lang/String.html) text)  
 throws [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html)

Set the property value by parsing a given String. May raise java.lang.IllegalArgumentException if either the String is badly formatted or if this kind of property can't be expressed as text.

**Parameters:**text - The string to be parsed. **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html)

### getTags

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

If the property value must be one of a set of known tagged values, then this method should return an array of the tags. This can be used to represent (for example) enum values. If a PropertyEditor supports tags, then it should support the use of setAsText with a tag value as a way of setting the value and the use of getAsText to identify the current value.

**Returns:**The tag values for this property. May be null if this property cannot be represented as a tagged value.

### getCustomEditor

[Component](http://docs.google.com/java/awt/Component.html) **getCustomEditor**()

A PropertyEditor may choose to make available a full custom Component that edits its property value. It is the responsibility of the PropertyEditor to hook itself up to its editor Component itself and to report property value changes by firing a PropertyChange event.

The higher-level code that calls getCustomEditor may either embed the Component in some larger property sheet, or it may put it in its own individual dialog, or ...

**Returns:**A java.awt.Component that will allow a human to directly edit the current property value. May be null if this is not supported.

### supportsCustomEditor

boolean **supportsCustomEditor**()

Determines whether this property editor supports a custom editor.

**Returns:**True if the propertyEditor can provide a custom editor.

### addPropertyChangeListener

void **addPropertyChangeListener**([PropertyChangeListener](http://docs.google.com/java/beans/PropertyChangeListener.html) listener)

Register a listener for the PropertyChange event. When a PropertyEditor changes its value it should fire a PropertyChange event on all registered PropertyChangeListeners, specifying the null value for the property name and itself as the source.

**Parameters:**listener - An object to be invoked when a PropertyChange event is fired.

### removePropertyChangeListener

void **removePropertyChangeListener**([PropertyChangeListener](http://docs.google.com/java/beans/PropertyChangeListener.html) listener)

Remove a listener for the PropertyChange event.

**Parameters:**listener - The PropertyChange listener to be removed.

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

[Submit a bug or feature](http://bugs.sun.com/services/bugreport/index.jsp)

For further API reference and developer documentation, see [Java SE Developer Documentation](http://docs.google.com/webnotes/devdocs-vs-specs.html). That documentation contains more detailed, developer-targeted descriptions, with conceptual overviews, definitions of terms, workarounds, and working code examples.

Copyright 2006 Sun Microsystems, Inc. All rights reserved. Use is subject to [license terms](http://docs.google.com/legal/license.html). Also see the [documentation redistribution policy](http://java.sun.com/docs/redist.html).