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

## **javax.swing**

Class InputVerifier

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

public abstract class **InputVerifier**extends [Object](http://docs.google.com/java/lang/Object.html)

The purpose of this class is to help clients support smooth focus navigation through GUIs with text fields. Such GUIs often need to ensure that the text entered by the user is valid (for example, that it's in the proper format) before allowing the user to navigate out of the text field. To do this, clients create a subclass of InputVerifier and, using JComponent's setInputVerifier method, attach an instance of their subclass to the JComponent whose input they want to validate. Before focus is transfered to another Swing component that requests it, the input verifier's shouldYieldFocus method is called. Focus is transfered only if that method returns true.

The following example has two text fields, with the first one expecting the string "pass" to be entered by the user. If that string is entered in the first text field, then the user can advance to the second text field either by clicking in it or by pressing TAB. However, if another string is entered in the first text field, then the user will be unable to transfer focus to the second text field.

import java.awt.\*;  
 import java.util.\*;  
 import java.awt.event.\*;  
 import javax.swing.\*;  
   
 // This program demonstrates the use of the Swing InputVerifier class.  
 // It creates two text fields; the first of the text fields expects the  
 // string "pass" as input, and will allow focus to advance out of it  
 // only after that string is typed in by the user.  
  
 public class VerifierTest extends JFrame {  
 public VerifierTest() {  
 JTextField tf1 = new JTextField ("Type \"pass\" here");  
 getContentPane().add (tf1, BorderLayout.NORTH);  
 tf1.setInputVerifier(new PassVerifier());  
   
 JTextField tf2 = new JTextField ("TextField2");  
 getContentPane().add (tf2, BorderLayout.SOUTH);  
   
 WindowListener l = new WindowAdapter() {  
 public void windowClosing(WindowEvent e) {   
 System.exit(0);   
 }  
 };  
 addWindowListener(l);  
 }  
   
 class PassVerifier extends InputVerifier {  
 public boolean verify(JComponent input) {  
 JTextField tf = (JTextField) input;  
 return "pass".equals(tf.getText());  
 }  
 }  
   
 public static void main(String[] args) {  
 Frame f = new VerifierTest();  
 f.pack();  
 f.setVisible(true);  
 }  
 }

**Since:** 1.3

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

| **Method Summary** | |
| --- | --- |
| boolean | [**shouldYieldFocus**](http://docs.google.com/javax/swing/InputVerifier.html#shouldYieldFocus(javax.swing.JComponent))([JComponent](http://docs.google.com/javax/swing/JComponent.html) input)            Calls verify(input) to ensure that the input is valid. |
| abstract  boolean | [**verify**](http://docs.google.com/javax/swing/InputVerifier.html#verify(javax.swing.JComponent))([JComponent](http://docs.google.com/javax/swing/JComponent.html) input)            Checks whether the JComponent's input is valid. |

| **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** |
| --- |

### InputVerifier

public **InputVerifier**()

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

### verify

public abstract boolean **verify**([JComponent](http://docs.google.com/javax/swing/JComponent.html) input)

Checks whether the JComponent's input is valid. This method should have no side effects. It returns a boolean indicating the status of the argument's input.

**Parameters:**input - the JComponent to verify **Returns:**true when valid, false when invalid**See Also:**[JComponent.setInputVerifier(javax.swing.InputVerifier)](http://docs.google.com/javax/swing/JComponent.html#setInputVerifier(javax.swing.InputVerifier)), [JComponent.getInputVerifier()](http://docs.google.com/javax/swing/JComponent.html#getInputVerifier())

### shouldYieldFocus

public boolean **shouldYieldFocus**([JComponent](http://docs.google.com/javax/swing/JComponent.html) input)

Calls verify(input) to ensure that the input is valid. This method can have side effects. In particular, this method is called when the user attempts to advance focus out of the argument component into another Swing component in this window. If this method returns true, then the focus is transfered normally; if it returns false, then the focus remains in the argument component.

**Parameters:**input - the JComponent to verify **Returns:**true when valid, false when invalid**See Also:**[JComponent.setInputVerifier(javax.swing.InputVerifier)](http://docs.google.com/javax/swing/JComponent.html#setInputVerifier(javax.swing.InputVerifier)), [JComponent.getInputVerifier()](http://docs.google.com/javax/swing/JComponent.html#getInputVerifier())

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/InputVerifier.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/InputMap.html)   [**NEXT CLASS**](http://docs.google.com/javax/swing/InternalFrameFocusTraversalPolicy.html) | [**FRAMES**](http://docs.google.com/index.html?javax/swing/InputVerifier.html)    [**NO FRAMES**](http://docs.google.com/InputVerifier.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).