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

## **javax.swing**

Class LayoutFocusTraversalPolicy

[java.lang.Object](http://docs.google.com/java/lang/Object.html)  
 [java.awt.FocusTraversalPolicy](http://docs.google.com/java/awt/FocusTraversalPolicy.html)  
 [javax.swing.InternalFrameFocusTraversalPolicy](http://docs.google.com/javax/swing/InternalFrameFocusTraversalPolicy.html)  
 [javax.swing.SortingFocusTraversalPolicy](http://docs.google.com/javax/swing/SortingFocusTraversalPolicy.html)  
 **javax.swing.LayoutFocusTraversalPolicy**

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

public class **LayoutFocusTraversalPolicy**extends [SortingFocusTraversalPolicy](http://docs.google.com/javax/swing/SortingFocusTraversalPolicy.html)implements [Serializable](http://docs.google.com/java/io/Serializable.html)

A SortingFocusTraversalPolicy which sorts Components based on their size, position, and orientation. Based on their size and position, Components are roughly categorized into rows and columns. For a Container with horizontal orientation, columns run left-to-right or right-to-left, and rows run top- to-bottom. For a Container with vertical orientation, columns run top-to- bottom and rows run left-to-right or right-to-left. See ComponentOrientation for more information. All columns in a row are fully traversed before proceeding to the next row.

**Since:** 1.4 **See Also:**[ComponentOrientation](http://docs.google.com/java/awt/ComponentOrientation.html)

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

| **Method Summary** | |
| --- | --- |
| protected  boolean | [**accept**](http://docs.google.com/javax/swing/LayoutFocusTraversalPolicy.html#accept(java.awt.Component))([Component](http://docs.google.com/java/awt/Component.html) aComponent)            Determines whether the specified Component is an acceptable choice as the new focus owner. |
| [Component](http://docs.google.com/java/awt/Component.html) | [**getComponentAfter**](http://docs.google.com/javax/swing/LayoutFocusTraversalPolicy.html#getComponentAfter(java.awt.Container,%20java.awt.Component))([Container](http://docs.google.com/java/awt/Container.html) aContainer, [Component](http://docs.google.com/java/awt/Component.html) aComponent)            Returns the Component that should receive the focus after aComponent. |
| [Component](http://docs.google.com/java/awt/Component.html) | [**getComponentBefore**](http://docs.google.com/javax/swing/LayoutFocusTraversalPolicy.html#getComponentBefore(java.awt.Container,%20java.awt.Component))([Container](http://docs.google.com/java/awt/Container.html) aContainer, [Component](http://docs.google.com/java/awt/Component.html) aComponent)            Returns the Component that should receive the focus before aComponent. |
| [Component](http://docs.google.com/java/awt/Component.html) | [**getFirstComponent**](http://docs.google.com/javax/swing/LayoutFocusTraversalPolicy.html#getFirstComponent(java.awt.Container))([Container](http://docs.google.com/java/awt/Container.html) aContainer)            Returns the first Component in the traversal cycle. |
| [Component](http://docs.google.com/java/awt/Component.html) | [**getLastComponent**](http://docs.google.com/javax/swing/LayoutFocusTraversalPolicy.html#getLastComponent(java.awt.Container))([Container](http://docs.google.com/java/awt/Container.html) aContainer)            Returns the last Component in the traversal cycle. |

| **Methods inherited from class javax.swing.**[**SortingFocusTraversalPolicy**](http://docs.google.com/javax/swing/SortingFocusTraversalPolicy.html) |
| --- |
| [getComparator](http://docs.google.com/javax/swing/SortingFocusTraversalPolicy.html#getComparator()), [getDefaultComponent](http://docs.google.com/javax/swing/SortingFocusTraversalPolicy.html#getDefaultComponent(java.awt.Container)), [getImplicitDownCycleTraversal](http://docs.google.com/javax/swing/SortingFocusTraversalPolicy.html#getImplicitDownCycleTraversal()), [setComparator](http://docs.google.com/javax/swing/SortingFocusTraversalPolicy.html#setComparator(java.util.Comparator)), [setImplicitDownCycleTraversal](http://docs.google.com/javax/swing/SortingFocusTraversalPolicy.html#setImplicitDownCycleTraversal(boolean)) |

| **Methods inherited from class javax.swing.**[**InternalFrameFocusTraversalPolicy**](http://docs.google.com/javax/swing/InternalFrameFocusTraversalPolicy.html) |
| --- |
| [getInitialComponent](http://docs.google.com/javax/swing/InternalFrameFocusTraversalPolicy.html#getInitialComponent(javax.swing.JInternalFrame)) |

| **Methods inherited from class java.awt.**[**FocusTraversalPolicy**](http://docs.google.com/java/awt/FocusTraversalPolicy.html) |
| --- |
| [getInitialComponent](http://docs.google.com/java/awt/FocusTraversalPolicy.html#getInitialComponent(java.awt.Window)) |

| **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** |
| --- |

### LayoutFocusTraversalPolicy

public **LayoutFocusTraversalPolicy**()

Constructs a LayoutFocusTraversalPolicy.

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

### getComponentAfter

public [Component](http://docs.google.com/java/awt/Component.html) **getComponentAfter**([Container](http://docs.google.com/java/awt/Container.html) aContainer,  
 [Component](http://docs.google.com/java/awt/Component.html) aComponent)

Returns the Component that should receive the focus after aComponent. aContainer must be a focus cycle root of aComponent.

By default, LayoutFocusTraversalPolicy implicitly transfers focus down- cycle. That is, during normal focus traversal, the Component traversed after a focus cycle root will be the focus-cycle-root's default Component to focus. This behavior can be disabled using the setImplicitDownCycleTraversal method.

If aContainer is [focus traversal policy provider](http://docs.google.com/java/awt/doc-files/FocusSpec.html#FocusTraversalPolicyProviders), the focus is always transferred down-cycle.

**Overrides:**[getComponentAfter](http://docs.google.com/javax/swing/SortingFocusTraversalPolicy.html#getComponentAfter(java.awt.Container,%20java.awt.Component)) in class [SortingFocusTraversalPolicy](http://docs.google.com/javax/swing/SortingFocusTraversalPolicy.html) **Parameters:**aContainer - a focus cycle root of aComponent or a focus traversal policy provideraComponent - a (possibly indirect) child of aContainer, or aContainer itself **Returns:**the Component that should receive the focus after aComponent, or null if no suitable Component can be found **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if aContainer is not a focus cycle root of aComponent or a focus traversal policy provider, or if either aContainer or aComponent is null

### getComponentBefore

public [Component](http://docs.google.com/java/awt/Component.html) **getComponentBefore**([Container](http://docs.google.com/java/awt/Container.html) aContainer,  
 [Component](http://docs.google.com/java/awt/Component.html) aComponent)

Returns the Component that should receive the focus before aComponent. aContainer must be a focus cycle root of aComponent.

By default, LayoutFocusTraversalPolicy implicitly transfers focus down- cycle. That is, during normal focus traversal, the Component traversed after a focus cycle root will be the focus-cycle-root's default Component to focus. This behavior can be disabled using the setImplicitDownCycleTraversal method.

If aContainer is [focus traversal policy provider](http://docs.google.com/java/awt/doc-files/FocusSpec.html#FocusTraversalPolicyProviders), the focus is always transferred down-cycle.

**Overrides:**[getComponentBefore](http://docs.google.com/javax/swing/SortingFocusTraversalPolicy.html#getComponentBefore(java.awt.Container,%20java.awt.Component)) in class [SortingFocusTraversalPolicy](http://docs.google.com/javax/swing/SortingFocusTraversalPolicy.html) **Parameters:**aContainer - a focus cycle root of aComponent or a focus traversal policy provideraComponent - a (possibly indirect) child of aContainer, or aContainer itself **Returns:**the Component that should receive the focus before aComponent, or null if no suitable Component can be found **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if aContainer is not a focus cycle root of aComponent or a focus traversal policy provider, or if either aContainer or aComponent is null

### getFirstComponent

public [Component](http://docs.google.com/java/awt/Component.html) **getFirstComponent**([Container](http://docs.google.com/java/awt/Container.html) aContainer)

Returns the first Component in the traversal cycle. This method is used to determine the next Component to focus when traversal wraps in the forward direction.

**Overrides:**[getFirstComponent](http://docs.google.com/javax/swing/SortingFocusTraversalPolicy.html#getFirstComponent(java.awt.Container)) in class [SortingFocusTraversalPolicy](http://docs.google.com/javax/swing/SortingFocusTraversalPolicy.html) **Parameters:**aContainer - a focus cycle root of aComponent or a focus traversal policy provider whose first Component is to be returned **Returns:**the first Component in the traversal cycle of aContainer, or null if no suitable Component can be found **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if aContainer is null

### getLastComponent

public [Component](http://docs.google.com/java/awt/Component.html) **getLastComponent**([Container](http://docs.google.com/java/awt/Container.html) aContainer)

Returns the last Component in the traversal cycle. This method is used to determine the next Component to focus when traversal wraps in the reverse direction.

**Overrides:**[getLastComponent](http://docs.google.com/javax/swing/SortingFocusTraversalPolicy.html#getLastComponent(java.awt.Container)) in class [SortingFocusTraversalPolicy](http://docs.google.com/javax/swing/SortingFocusTraversalPolicy.html) **Parameters:**aContainer - a focus cycle root of aComponent or a focus traversal policy provider whose last Component is to be returned **Returns:**the last Component in the traversal cycle of aContainer, or null if no suitable Component can be found **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if aContainer is null

### accept

protected boolean **accept**([Component](http://docs.google.com/java/awt/Component.html) aComponent)

Determines whether the specified Component is an acceptable choice as the new focus owner. This method performs the following sequence of operations:

1. Checks whether aComponent is visible, displayable, enabled, and focusable. If any of these properties is false, this method returns false.
2. If aComponent is an instance of JTable, returns true.
3. If aComponent is an instance of JComboBox, then returns the value of aComponent.getUI().isFocusTraversable(aComponent).
4. If aComponent is a JComponent with a JComponent.WHEN\_FOCUSED InputMap that is neither null nor empty, returns true.
5. Returns the value of DefaultFocusTraversalPolicy.accept(aComponent).

**Overrides:**[accept](http://docs.google.com/javax/swing/SortingFocusTraversalPolicy.html#accept(java.awt.Component)) in class [SortingFocusTraversalPolicy](http://docs.google.com/javax/swing/SortingFocusTraversalPolicy.html) **Parameters:**aComponent - the Component whose fitness as a focus owner is to be tested **Returns:**true if aComponent is a valid choice for a focus owner; otherwise false**See Also:**[Component.isVisible()](http://docs.google.com/java/awt/Component.html#isVisible()), [Component.isDisplayable()](http://docs.google.com/java/awt/Component.html#isDisplayable()), [Component.isEnabled()](http://docs.google.com/java/awt/Component.html#isEnabled()), [Component.isFocusable()](http://docs.google.com/java/awt/Component.html#isFocusable()), [ComboBoxUI.isFocusTraversable(javax.swing.JComboBox)](http://docs.google.com/javax/swing/plaf/ComboBoxUI.html#isFocusTraversable(javax.swing.JComboBox)), [JComponent.getInputMap(int)](http://docs.google.com/javax/swing/JComponent.html#getInputMap(int)), [DefaultFocusTraversalPolicy.accept(java.awt.Component)](http://docs.google.com/java/awt/DefaultFocusTraversalPolicy.html#accept(java.awt.Component))

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

[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).