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

## **java.awt**

Class FocusTraversalPolicy

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

**Direct Known Subclasses:** [ContainerOrderFocusTraversalPolicy](http://docs.google.com/java/awt/ContainerOrderFocusTraversalPolicy.html), [InternalFrameFocusTraversalPolicy](http://docs.google.com/javax/swing/InternalFrameFocusTraversalPolicy.html)

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

A FocusTraversalPolicy defines the order in which Components with a particular focus cycle root are traversed. Instances can apply the policy to arbitrary focus cycle roots, allowing themselves to be shared across Containers. They do not need to be reinitialized when the focus cycle roots of a Component hierarchy change.

The core responsibility of a FocusTraversalPolicy is to provide algorithms determining the next and previous Components to focus when traversing forward or backward in a UI. Each FocusTraversalPolicy must also provide algorithms for determining the first, last, and default Components in a traversal cycle. First and last Components are used when normal forward and backward traversal, respectively, wraps. The default Component is the first to receive focus when traversing down into a new focus traversal cycle. A FocusTraversalPolicy can optionally provide an algorithm for determining a Window's initial Component. The initial Component is the first to receive focus when a Window is first made visible.

FocusTraversalPolicy takes into account [focus traversal policy providers](http://docs.google.com/doc-files/FocusSpec.html#FocusTraversalPolicyProviders). When searching for first/last/next/previous Component, if a focus traversal policy provider is encountered, its focus traversal policy is used to perform the search operation.

Please see  [How to Use the Focus Subsystem](http://java.sun.com/docs/books/tutorial/uiswing/misc/focus.html), a section in *The Java Tutorial*, and the [Focus Specification](http://docs.google.com/java/awt/doc-files/FocusSpec.html) for more information.

**Since:** 1.4 **See Also:**[Container.setFocusTraversalPolicy(java.awt.FocusTraversalPolicy)](http://docs.google.com/java/awt/Container.html#setFocusTraversalPolicy(java.awt.FocusTraversalPolicy)), [Container.getFocusTraversalPolicy()](http://docs.google.com/java/awt/Container.html#getFocusTraversalPolicy()), [Container.setFocusCycleRoot(boolean)](http://docs.google.com/java/awt/Container.html#setFocusCycleRoot(boolean)), [Container.isFocusCycleRoot(java.awt.Container)](http://docs.google.com/java/awt/Container.html#isFocusCycleRoot(java.awt.Container)), [Container.setFocusTraversalPolicyProvider(boolean)](http://docs.google.com/java/awt/Container.html#setFocusTraversalPolicyProvider(boolean)), [Container.isFocusTraversalPolicyProvider()](http://docs.google.com/java/awt/Container.html#isFocusTraversalPolicyProvider()), [KeyboardFocusManager.setDefaultFocusTraversalPolicy(java.awt.FocusTraversalPolicy)](http://docs.google.com/java/awt/KeyboardFocusManager.html#setDefaultFocusTraversalPolicy(java.awt.FocusTraversalPolicy)), [KeyboardFocusManager.getDefaultFocusTraversalPolicy()](http://docs.google.com/java/awt/KeyboardFocusManager.html#getDefaultFocusTraversalPolicy())

| **Constructor Summary** | |
| --- | --- |
| [**FocusTraversalPolicy**](http://docs.google.com/java/awt/FocusTraversalPolicy.html#FocusTraversalPolicy())() |

| **Method Summary** | |
| --- | --- |
| abstract  [Component](http://docs.google.com/java/awt/Component.html) | [**getComponentAfter**](http://docs.google.com/java/awt/FocusTraversalPolicy.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. |
| abstract  [Component](http://docs.google.com/java/awt/Component.html) | [**getComponentBefore**](http://docs.google.com/java/awt/FocusTraversalPolicy.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. |
| abstract  [Component](http://docs.google.com/java/awt/Component.html) | [**getDefaultComponent**](http://docs.google.com/java/awt/FocusTraversalPolicy.html#getDefaultComponent(java.awt.Container))([Container](http://docs.google.com/java/awt/Container.html) aContainer)            Returns the default Component to focus. |
| abstract  [Component](http://docs.google.com/java/awt/Component.html) | [**getFirstComponent**](http://docs.google.com/java/awt/FocusTraversalPolicy.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) | [**getInitialComponent**](http://docs.google.com/java/awt/FocusTraversalPolicy.html#getInitialComponent(java.awt.Window))([Window](http://docs.google.com/java/awt/Window.html) window)            Returns the Component that should receive the focus when a Window is made visible for the first time. |
| abstract  [Component](http://docs.google.com/java/awt/Component.html) | [**getLastComponent**](http://docs.google.com/java/awt/FocusTraversalPolicy.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 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** |
| --- |

### FocusTraversalPolicy

public **FocusTraversalPolicy**()

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

### getComponentAfter

public abstract [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 or a focus traversal policy provider.

**Parameters:**aContainer - a focus cycle root of aComponent or 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 abstract [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 or a focus traversal policy provider.

**Parameters:**aContainer - a focus cycle root of aComponent or 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 abstract [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.

**Parameters:**aContainer - the focus cycle root or 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 abstract [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.

**Parameters:**aContainer - the focus cycle root or 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

### getDefaultComponent

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

Returns the default Component to focus. This Component will be the first to receive focus when traversing down into a new focus traversal cycle rooted at aContainer.

**Parameters:**aContainer - the focus cycle root or focus traversal policy provider whose default Component is to be returned **Returns:**the default 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

### getInitialComponent

public [Component](http://docs.google.com/java/awt/Component.html) **getInitialComponent**([Window](http://docs.google.com/java/awt/Window.html) window)

Returns the Component that should receive the focus when a Window is made visible for the first time. Once the Window has been made visible by a call to show() or setVisible(true), the initial Component will not be used again. Instead, if the Window loses and subsequently regains focus, or is made invisible or undisplayable and subsequently made visible and displayable, the Window's most recently focused Component will become the focus owner. The default implementation of this method returns the default Component.

**Parameters:**window - the Window whose initial Component is to be returned **Returns:**the Component that should receive the focus when window is made visible for the first time, or null if no suitable Component can be found **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if window is null**See Also:**[getDefaultComponent(java.awt.Container)](http://docs.google.com/java/awt/FocusTraversalPolicy.html#getDefaultComponent(java.awt.Container)), [Window.getMostRecentFocusOwner()](http://docs.google.com/java/awt/Window.html#getMostRecentFocusOwner())

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/FocusTraversalPolicy.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/awt/FlowLayout.html)   [**NEXT CLASS**](http://docs.google.com/java/awt/Font.html) | [**FRAMES**](http://docs.google.com/index.html?java/awt/FocusTraversalPolicy.html)    [**NO FRAMES**](http://docs.google.com/FocusTraversalPolicy.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).