| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/DefaultFocusTraversalPolicy.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/Cursor.html)   [**NEXT CLASS**](http://docs.google.com/java/awt/DefaultKeyboardFocusManager.html) | [**FRAMES**](http://docs.google.com/index.html?java/awt/DefaultFocusTraversalPolicy.html)    [**NO FRAMES**](http://docs.google.com/DefaultFocusTraversalPolicy.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | [CONSTR](#3znysh7) | [METHOD](#2et92p0) | DETAIL: FIELD | [CONSTR](#4d34og8) | [METHOD](#17dp8vu) |

## **java.awt**

Class DefaultFocusTraversalPolicy

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

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

public class **DefaultFocusTraversalPolicy**extends [ContainerOrderFocusTraversalPolicy](http://docs.google.com/java/awt/ContainerOrderFocusTraversalPolicy.html)

A FocusTraversalPolicy that determines traversal order based on the order of child Components in a Container. From a particular focus cycle root, the policy makes a pre-order traversal of the Component hierarchy, and traverses a Container's children according to the ordering of the array returned by Container.getComponents(). Portions of the hierarchy that are not visible and displayable will not be searched.

If client code has explicitly set the focusability of a Component by either overriding Component.isFocusTraversable() or Component.isFocusable(), or by calling Component.setFocusable(), then a DefaultFocusTraversalPolicy behaves exactly like a ContainerOrderFocusTraversalPolicy. If, however, the Component is relying on default focusability, then a DefaultFocusTraversalPolicy will reject all Components with non-focusable peers. This is the default FocusTraversalPolicy for all AWT Containers.

The focusability of a peer is implementation-dependent. Sun recommends that all implementations for a particular native platform construct peers with the same focusability. The recommendations for Windows and Unix are that Canvases, Labels, Panels, Scrollbars, ScrollPanes, Windows, and lightweight Components have non-focusable peers, and all other Components have focusable peers. These recommendations are used in the Sun AWT implementations. Note that the focusability of a Component's peer is different from, and does not impact, the focusability of the Component itself.

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.getComponents()](http://docs.google.com/java/awt/Container.html#getComponents()), [Component.isFocusable()](http://docs.google.com/java/awt/Component.html#isFocusable()), [Component.setFocusable(boolean)](http://docs.google.com/java/awt/Component.html#setFocusable(boolean)), [Serialized Form](http://docs.google.com/serialized-form.html#java.awt.DefaultFocusTraversalPolicy)

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

| **Method Summary** | |
| --- | --- |
| protected  boolean | [**accept**](http://docs.google.com/java/awt/DefaultFocusTraversalPolicy.html#accept(java.awt.Component))([Component](http://docs.google.com/java/awt/Component.html) aComponent)            Determines whether a Component is an acceptable choice as the new focus owner. |

| **Methods inherited from class java.awt.**[**ContainerOrderFocusTraversalPolicy**](http://docs.google.com/java/awt/ContainerOrderFocusTraversalPolicy.html) |
| --- |
| [getComponentAfter](http://docs.google.com/java/awt/ContainerOrderFocusTraversalPolicy.html#getComponentAfter(java.awt.Container,%20java.awt.Component)), [getComponentBefore](http://docs.google.com/java/awt/ContainerOrderFocusTraversalPolicy.html#getComponentBefore(java.awt.Container,%20java.awt.Component)), [getDefaultComponent](http://docs.google.com/java/awt/ContainerOrderFocusTraversalPolicy.html#getDefaultComponent(java.awt.Container)), [getFirstComponent](http://docs.google.com/java/awt/ContainerOrderFocusTraversalPolicy.html#getFirstComponent(java.awt.Container)), [getImplicitDownCycleTraversal](http://docs.google.com/java/awt/ContainerOrderFocusTraversalPolicy.html#getImplicitDownCycleTraversal()), [getLastComponent](http://docs.google.com/java/awt/ContainerOrderFocusTraversalPolicy.html#getLastComponent(java.awt.Container)), [setImplicitDownCycleTraversal](http://docs.google.com/java/awt/ContainerOrderFocusTraversalPolicy.html#setImplicitDownCycleTraversal(boolean)) |

| **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** |
| --- |

### DefaultFocusTraversalPolicy

public **DefaultFocusTraversalPolicy**()

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

### accept

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

Determines whether a Component is an acceptable choice as the new focus owner. The Component must be visible, displayable, and enabled to be accepted. If client code has explicitly set the focusability of the Component by either overriding Component.isFocusTraversable() or Component.isFocusable(), or by calling Component.setFocusable(), then the Component will be accepted if and only if it is focusable. If, however, the Component is relying on default focusability, then all Canvases, Labels, Panels, Scrollbars, ScrollPanes, Windows, and lightweight Components will be rejected.

**Overrides:**[accept](http://docs.google.com/java/awt/ContainerOrderFocusTraversalPolicy.html#accept(java.awt.Component)) in class [ContainerOrderFocusTraversalPolicy](http://docs.google.com/java/awt/ContainerOrderFocusTraversalPolicy.html) **Parameters:**aComponent - the Component whose fitness as a focus owner is to be tested **Returns:**true if aComponent meets the above requirements; false otherwise

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/DefaultFocusTraversalPolicy.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/Cursor.html)   [**NEXT CLASS**](http://docs.google.com/java/awt/DefaultKeyboardFocusManager.html) | [**FRAMES**](http://docs.google.com/index.html?java/awt/DefaultFocusTraversalPolicy.html)    [**NO FRAMES**](http://docs.google.com/DefaultFocusTraversalPolicy.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | [CONSTR](#3znysh7) | [METHOD](#2et92p0) | DETAIL: FIELD | [CONSTR](#4d34og8) | [METHOD](#17dp8vu) |

[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).