| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/DefaultTreeCellEditor.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/tree/DefaultMutableTreeNode.html)   [**NEXT CLASS**](http://docs.google.com/javax/swing/tree/DefaultTreeCellEditor.DefaultTextField.html) | [**FRAMES**](http://docs.google.com/index.html?javax/swing/tree/DefaultTreeCellEditor.html)    [**NO FRAMES**](http://docs.google.com/DefaultTreeCellEditor.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: [NESTED](#3znysh7) | [FIELD](#2et92p0) | [CONSTR](#tyjcwt) | [METHOD](#3dy6vkm) | DETAIL: [FIELD](#4d34og8) | [CONSTR](#2xcytpi) | [METHOD](#2bn6wsx) |

## **javax.swing.tree**

Class DefaultTreeCellEditor

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

**All Implemented Interfaces:** [ActionListener](http://docs.google.com/java/awt/event/ActionListener.html), [EventListener](http://docs.google.com/java/util/EventListener.html), [CellEditor](http://docs.google.com/javax/swing/CellEditor.html), [TreeSelectionListener](http://docs.google.com/javax/swing/event/TreeSelectionListener.html), [TreeCellEditor](http://docs.google.com/javax/swing/tree/TreeCellEditor.html)

public class **DefaultTreeCellEditor**extends [Object](http://docs.google.com/java/lang/Object.html)implements [ActionListener](http://docs.google.com/java/awt/event/ActionListener.html), [TreeCellEditor](http://docs.google.com/javax/swing/tree/TreeCellEditor.html), [TreeSelectionListener](http://docs.google.com/javax/swing/event/TreeSelectionListener.html)

A TreeCellEditor. You need to supply an instance of DefaultTreeCellRenderer so that the icons can be obtained. You can optionally supply a TreeCellEditor that will be layed out according to the icon in the DefaultTreeCellRenderer. If you do not supply a TreeCellEditor, a TextField will be used. Editing is started on a triple mouse click, or after a click, pause, click and a delay of 1200 miliseconds.

**Warning:** Serialized objects of this class will not be compatible with future Swing releases. The current serialization support is appropriate for short term storage or RMI between applications running the same version of Swing. As of 1.4, support for long term storage of all JavaBeansTM has been added to the java.beans package. Please see [XMLEncoder](http://docs.google.com/java/beans/XMLEncoder.html).

**See Also:**[JTree](http://docs.google.com/javax/swing/JTree.html)

| **Nested Class Summary** | |
| --- | --- |
| class | [**DefaultTreeCellEditor.DefaultTextField**](http://docs.google.com/javax/swing/tree/DefaultTreeCellEditor.DefaultTextField.html)            TextField used when no editor is supplied. |
| class | [**DefaultTreeCellEditor.EditorContainer**](http://docs.google.com/javax/swing/tree/DefaultTreeCellEditor.EditorContainer.html)            Container responsible for placing the editingComponent. |

| **Field Summary** | |
| --- | --- |
| protected  [Color](http://docs.google.com/java/awt/Color.html) | [**borderSelectionColor**](http://docs.google.com/javax/swing/tree/DefaultTreeCellEditor.html#borderSelectionColor)            True if the border selection color should be drawn. |
| protected  boolean | [**canEdit**](http://docs.google.com/javax/swing/tree/DefaultTreeCellEditor.html#canEdit)            As of Java 2 platform v1.4 this field should no longer be used. |
| protected  [Component](http://docs.google.com/java/awt/Component.html) | [**editingComponent**](http://docs.google.com/javax/swing/tree/DefaultTreeCellEditor.html#editingComponent)            Component used in editing, obtained from the editingContainer. |
| protected  [Container](http://docs.google.com/java/awt/Container.html) | [**editingContainer**](http://docs.google.com/javax/swing/tree/DefaultTreeCellEditor.html#editingContainer)            Editing container, will contain the editorComponent. |
| protected  [Icon](http://docs.google.com/javax/swing/Icon.html) | [**editingIcon**](http://docs.google.com/javax/swing/tree/DefaultTreeCellEditor.html#editingIcon)            Icon to use when editing. |
| protected  [Font](http://docs.google.com/java/awt/Font.html) | [**font**](http://docs.google.com/javax/swing/tree/DefaultTreeCellEditor.html#font)            Font to paint with, null indicates font of renderer is to be used. |
| protected  [TreePath](http://docs.google.com/javax/swing/tree/TreePath.html) | [**lastPath**](http://docs.google.com/javax/swing/tree/DefaultTreeCellEditor.html#lastPath)            Last path that was selected. |
| protected  int | [**lastRow**](http://docs.google.com/javax/swing/tree/DefaultTreeCellEditor.html#lastRow)            Row that was last passed into getTreeCellEditorComponent. |
| protected  int | [**offset**](http://docs.google.com/javax/swing/tree/DefaultTreeCellEditor.html#offset)            Used in editing. |
| protected  [TreeCellEditor](http://docs.google.com/javax/swing/tree/TreeCellEditor.html) | [**realEditor**](http://docs.google.com/javax/swing/tree/DefaultTreeCellEditor.html#realEditor)            Editor handling the editing. |
| protected  [DefaultTreeCellRenderer](http://docs.google.com/javax/swing/tree/DefaultTreeCellRenderer.html) | [**renderer**](http://docs.google.com/javax/swing/tree/DefaultTreeCellEditor.html#renderer)            Renderer, used to get border and offsets from. |
| protected  [Timer](http://docs.google.com/javax/swing/Timer.html) | [**timer**](http://docs.google.com/javax/swing/tree/DefaultTreeCellEditor.html#timer)            Used before starting the editing session. |
| protected  [JTree](http://docs.google.com/javax/swing/JTree.html) | [**tree**](http://docs.google.com/javax/swing/tree/DefaultTreeCellEditor.html#tree)            JTree instance listening too. |

| **Constructor Summary** | |
| --- | --- |
| [**DefaultTreeCellEditor**](http://docs.google.com/javax/swing/tree/DefaultTreeCellEditor.html#DefaultTreeCellEditor(javax.swing.JTree,%20javax.swing.tree.DefaultTreeCellRenderer))([JTree](http://docs.google.com/javax/swing/JTree.html) tree, [DefaultTreeCellRenderer](http://docs.google.com/javax/swing/tree/DefaultTreeCellRenderer.html) renderer)            Constructs a DefaultTreeCellEditor object for a JTree using the specified renderer and a default editor. |
| [**DefaultTreeCellEditor**](http://docs.google.com/javax/swing/tree/DefaultTreeCellEditor.html#DefaultTreeCellEditor(javax.swing.JTree,%20javax.swing.tree.DefaultTreeCellRenderer,%20javax.swing.tree.TreeCellEditor))([JTree](http://docs.google.com/javax/swing/JTree.html) tree, [DefaultTreeCellRenderer](http://docs.google.com/javax/swing/tree/DefaultTreeCellRenderer.html) renderer, [TreeCellEditor](http://docs.google.com/javax/swing/tree/TreeCellEditor.html) editor)            Constructs a DefaultTreeCellEditor object for a JTree using the specified renderer and the specified editor. |

| **Method Summary** | |
| --- | --- |
| void | [**actionPerformed**](http://docs.google.com/javax/swing/tree/DefaultTreeCellEditor.html#actionPerformed(java.awt.event.ActionEvent))([ActionEvent](http://docs.google.com/java/awt/event/ActionEvent.html) e)            Messaged when the timer fires, this will start the editing session. |
| void | [**addCellEditorListener**](http://docs.google.com/javax/swing/tree/DefaultTreeCellEditor.html#addCellEditorListener(javax.swing.event.CellEditorListener))([CellEditorListener](http://docs.google.com/javax/swing/event/CellEditorListener.html) l)            Adds the CellEditorListener. |
| void | [**cancelCellEditing**](http://docs.google.com/javax/swing/tree/DefaultTreeCellEditor.html#cancelCellEditing())()            Messages cancelCellEditing to the realEditor and removes it from this instance. |
| protected  boolean | [**canEditImmediately**](http://docs.google.com/javax/swing/tree/DefaultTreeCellEditor.html#canEditImmediately(java.util.EventObject))([EventObject](http://docs.google.com/java/util/EventObject.html) event)            Returns true if event is null, or it is a MouseEvent with a click count > 2 and inHitRegion returns true. |
| protected  [Container](http://docs.google.com/java/awt/Container.html) | [**createContainer**](http://docs.google.com/javax/swing/tree/DefaultTreeCellEditor.html#createContainer())()            Creates the container to manage placement of editingComponent. |
| protected  [TreeCellEditor](http://docs.google.com/javax/swing/tree/TreeCellEditor.html) | [**createTreeCellEditor**](http://docs.google.com/javax/swing/tree/DefaultTreeCellEditor.html#createTreeCellEditor())()            This is invoked if a TreeCellEditor is not supplied in the constructor. |
| protected  void | [**determineOffset**](http://docs.google.com/javax/swing/tree/DefaultTreeCellEditor.html#determineOffset(javax.swing.JTree,%20java.lang.Object,%20boolean,%20boolean,%20boolean,%20int))([JTree](http://docs.google.com/javax/swing/JTree.html) tree, [Object](http://docs.google.com/java/lang/Object.html) value, boolean isSelected, boolean expanded, boolean leaf, int row) |
| [Color](http://docs.google.com/java/awt/Color.html) | [**getBorderSelectionColor**](http://docs.google.com/javax/swing/tree/DefaultTreeCellEditor.html#getBorderSelectionColor())()            Returns the color the border is drawn. |
| [CellEditorListener](http://docs.google.com/javax/swing/event/CellEditorListener.html)[] | [**getCellEditorListeners**](http://docs.google.com/javax/swing/tree/DefaultTreeCellEditor.html#getCellEditorListeners())()            Returns an array of all the CellEditorListeners added to this DefaultTreeCellEditor with addCellEditorListener(). |
| [Object](http://docs.google.com/java/lang/Object.html) | [**getCellEditorValue**](http://docs.google.com/javax/swing/tree/DefaultTreeCellEditor.html#getCellEditorValue())()            Returns the value currently being edited. |
| [Font](http://docs.google.com/java/awt/Font.html) | [**getFont**](http://docs.google.com/javax/swing/tree/DefaultTreeCellEditor.html#getFont())()            Gets the font used for editing. |
| [Component](http://docs.google.com/java/awt/Component.html) | [**getTreeCellEditorComponent**](http://docs.google.com/javax/swing/tree/DefaultTreeCellEditor.html#getTreeCellEditorComponent(javax.swing.JTree,%20java.lang.Object,%20boolean,%20boolean,%20boolean,%20int))([JTree](http://docs.google.com/javax/swing/JTree.html) tree, [Object](http://docs.google.com/java/lang/Object.html) value, boolean isSelected, boolean expanded, boolean leaf, int row)            Configures the editor. |
| protected  boolean | [**inHitRegion**](http://docs.google.com/javax/swing/tree/DefaultTreeCellEditor.html#inHitRegion(int,%20int))(int x, int y)            Returns true if the passed in location is a valid mouse location to start editing from. |
| boolean | [**isCellEditable**](http://docs.google.com/javax/swing/tree/DefaultTreeCellEditor.html#isCellEditable(java.util.EventObject))([EventObject](http://docs.google.com/java/util/EventObject.html) event)            If the realEditor returns true to this message, prepareForEditing is messaged and true is returned. |
| protected  void | [**prepareForEditing**](http://docs.google.com/javax/swing/tree/DefaultTreeCellEditor.html#prepareForEditing())()            Invoked just before editing is to start. |
| void | [**removeCellEditorListener**](http://docs.google.com/javax/swing/tree/DefaultTreeCellEditor.html#removeCellEditorListener(javax.swing.event.CellEditorListener))([CellEditorListener](http://docs.google.com/javax/swing/event/CellEditorListener.html) l)            Removes the previously added CellEditorListener. |
| void | [**setBorderSelectionColor**](http://docs.google.com/javax/swing/tree/DefaultTreeCellEditor.html#setBorderSelectionColor(java.awt.Color))([Color](http://docs.google.com/java/awt/Color.html) newColor)            Sets the color to use for the border. |
| void | [**setFont**](http://docs.google.com/javax/swing/tree/DefaultTreeCellEditor.html#setFont(java.awt.Font))([Font](http://docs.google.com/java/awt/Font.html) font)            Sets the font to edit with. |
| protected  void | [**setTree**](http://docs.google.com/javax/swing/tree/DefaultTreeCellEditor.html#setTree(javax.swing.JTree))([JTree](http://docs.google.com/javax/swing/JTree.html) newTree)            Sets the tree currently editing for. |
| boolean | [**shouldSelectCell**](http://docs.google.com/javax/swing/tree/DefaultTreeCellEditor.html#shouldSelectCell(java.util.EventObject))([EventObject](http://docs.google.com/java/util/EventObject.html) event)            Messages the realEditor for the return value. |
| protected  boolean | [**shouldStartEditingTimer**](http://docs.google.com/javax/swing/tree/DefaultTreeCellEditor.html#shouldStartEditingTimer(java.util.EventObject))([EventObject](http://docs.google.com/java/util/EventObject.html) event)            Returns true if event is a MouseEvent and the click count is 1. |
| protected  void | [**startEditingTimer**](http://docs.google.com/javax/swing/tree/DefaultTreeCellEditor.html#startEditingTimer())()            Starts the editing timer. |
| boolean | [**stopCellEditing**](http://docs.google.com/javax/swing/tree/DefaultTreeCellEditor.html#stopCellEditing())()            If the realEditor will allow editing to stop, the realEditor is removed and true is returned, otherwise false is returned. |
| void | [**valueChanged**](http://docs.google.com/javax/swing/tree/DefaultTreeCellEditor.html#valueChanged(javax.swing.event.TreeSelectionEvent))([TreeSelectionEvent](http://docs.google.com/javax/swing/event/TreeSelectionEvent.html) e)            Resets lastPath. |

| **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)) |

| **Field Detail** |
| --- |

### realEditor

protected [TreeCellEditor](http://docs.google.com/javax/swing/tree/TreeCellEditor.html) **realEditor**

Editor handling the editing.

### renderer

protected [DefaultTreeCellRenderer](http://docs.google.com/javax/swing/tree/DefaultTreeCellRenderer.html) **renderer**

Renderer, used to get border and offsets from.

### editingContainer

protected [Container](http://docs.google.com/java/awt/Container.html) **editingContainer**

Editing container, will contain the editorComponent.

### editingComponent

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

Component used in editing, obtained from the editingContainer.

### canEdit

protected boolean **canEdit**

As of Java 2 platform v1.4 this field should no longer be used. If you wish to provide similar behavior you should directly override isCellEditable.

### offset

protected transient int **offset**

Used in editing. Indicates x position to place editingComponent.

### tree

protected transient [JTree](http://docs.google.com/javax/swing/JTree.html) **tree**

JTree instance listening too.

### lastPath

protected transient [TreePath](http://docs.google.com/javax/swing/tree/TreePath.html) **lastPath**

Last path that was selected.

### timer

protected transient [Timer](http://docs.google.com/javax/swing/Timer.html) **timer**

Used before starting the editing session.

### lastRow

protected transient int **lastRow**

Row that was last passed into getTreeCellEditorComponent.

### borderSelectionColor

protected [Color](http://docs.google.com/java/awt/Color.html) **borderSelectionColor**

True if the border selection color should be drawn.

### editingIcon

protected transient [Icon](http://docs.google.com/javax/swing/Icon.html) **editingIcon**

Icon to use when editing.

### font

protected [Font](http://docs.google.com/java/awt/Font.html) **font**

Font to paint with, null indicates font of renderer is to be used.

| **Constructor Detail** |
| --- |

### DefaultTreeCellEditor

public **DefaultTreeCellEditor**([JTree](http://docs.google.com/javax/swing/JTree.html) tree,  
 [DefaultTreeCellRenderer](http://docs.google.com/javax/swing/tree/DefaultTreeCellRenderer.html) renderer)

Constructs a DefaultTreeCellEditor object for a JTree using the specified renderer and a default editor. (Use this constructor for normal editing.)

**Parameters:**tree - a JTree objectrenderer - a DefaultTreeCellRenderer object

### DefaultTreeCellEditor

public **DefaultTreeCellEditor**([JTree](http://docs.google.com/javax/swing/JTree.html) tree,  
 [DefaultTreeCellRenderer](http://docs.google.com/javax/swing/tree/DefaultTreeCellRenderer.html) renderer,  
 [TreeCellEditor](http://docs.google.com/javax/swing/tree/TreeCellEditor.html) editor)

Constructs a DefaultTreeCellEditor object for a JTree using the specified renderer and the specified editor. (Use this constructor for specialized editing.)

**Parameters:**tree - a JTree objectrenderer - a DefaultTreeCellRenderer objecteditor - a TreeCellEditor object

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

### setBorderSelectionColor

public void **setBorderSelectionColor**([Color](http://docs.google.com/java/awt/Color.html) newColor)

Sets the color to use for the border.

**Parameters:**newColor - the new border color

### getBorderSelectionColor

public [Color](http://docs.google.com/java/awt/Color.html) **getBorderSelectionColor**()

Returns the color the border is drawn.

**Returns:**the border selection color

### setFont

public void **setFont**([Font](http://docs.google.com/java/awt/Font.html) font)

Sets the font to edit with. null indicates the renderers font should be used. This will NOT override any font you have set in the editor the receiver was instantied with. If null for an editor was passed in a default editor will be created that will pick up this font.

**Parameters:**font - the editing Font**See Also:**[getFont()](http://docs.google.com/javax/swing/tree/DefaultTreeCellEditor.html#getFont())

### getFont

public [Font](http://docs.google.com/java/awt/Font.html) **getFont**()

Gets the font used for editing.

**Returns:**the editing Font**See Also:**[setFont(java.awt.Font)](http://docs.google.com/javax/swing/tree/DefaultTreeCellEditor.html#setFont(java.awt.Font))

### getTreeCellEditorComponent

public [Component](http://docs.google.com/java/awt/Component.html) **getTreeCellEditorComponent**([JTree](http://docs.google.com/javax/swing/JTree.html) tree,  
 [Object](http://docs.google.com/java/lang/Object.html) value,  
 boolean isSelected,  
 boolean expanded,  
 boolean leaf,  
 int row)

Configures the editor. Passed onto the realEditor.

**Specified by:**[getTreeCellEditorComponent](http://docs.google.com/javax/swing/tree/TreeCellEditor.html#getTreeCellEditorComponent(javax.swing.JTree,%20java.lang.Object,%20boolean,%20boolean,%20boolean,%20int)) in interface [TreeCellEditor](http://docs.google.com/javax/swing/tree/TreeCellEditor.html) **Parameters:**tree - the JTree that is asking the editor to edit; this parameter can be nullvalue - the value of the cell to be editedisSelected - true if the cell is to be rendered with selection highlightingexpanded - true if the node is expandedleaf - true if the node is a leaf noderow - the row index of the node being edited **Returns:**the component for editing

### getCellEditorValue

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

Returns the value currently being edited.

**Specified by:**[getCellEditorValue](http://docs.google.com/javax/swing/CellEditor.html#getCellEditorValue()) in interface [CellEditor](http://docs.google.com/javax/swing/CellEditor.html) **Returns:**the value currently being edited

### isCellEditable

public boolean **isCellEditable**([EventObject](http://docs.google.com/java/util/EventObject.html) event)

If the realEditor returns true to this message, prepareForEditing is messaged and true is returned.

**Specified by:**[isCellEditable](http://docs.google.com/javax/swing/CellEditor.html#isCellEditable(java.util.EventObject)) in interface [CellEditor](http://docs.google.com/javax/swing/CellEditor.html) **Parameters:**event - the event the editor should use to consider whether to begin editing or not **Returns:**true if editing can be started**See Also:**[CellEditor.shouldSelectCell(java.util.EventObject)](http://docs.google.com/javax/swing/CellEditor.html#shouldSelectCell(java.util.EventObject))

### shouldSelectCell

public boolean **shouldSelectCell**([EventObject](http://docs.google.com/java/util/EventObject.html) event)

Messages the realEditor for the return value.

**Specified by:**[shouldSelectCell](http://docs.google.com/javax/swing/CellEditor.html#shouldSelectCell(java.util.EventObject)) in interface [CellEditor](http://docs.google.com/javax/swing/CellEditor.html) **Parameters:**event - the event the editor should use to start editing **Returns:**true if the editor would like the editing cell to be selected; otherwise returns false**See Also:**[CellEditor.isCellEditable(java.util.EventObject)](http://docs.google.com/javax/swing/CellEditor.html#isCellEditable(java.util.EventObject))

### stopCellEditing

public boolean **stopCellEditing**()

If the realEditor will allow editing to stop, the realEditor is removed and true is returned, otherwise false is returned.

**Specified by:**[stopCellEditing](http://docs.google.com/javax/swing/CellEditor.html#stopCellEditing()) in interface [CellEditor](http://docs.google.com/javax/swing/CellEditor.html) **Returns:**true if editing was stopped; false otherwise

### cancelCellEditing

public void **cancelCellEditing**()

Messages cancelCellEditing to the realEditor and removes it from this instance.

**Specified by:**[cancelCellEditing](http://docs.google.com/javax/swing/CellEditor.html#cancelCellEditing()) in interface [CellEditor](http://docs.google.com/javax/swing/CellEditor.html)

### addCellEditorListener

public void **addCellEditorListener**([CellEditorListener](http://docs.google.com/javax/swing/event/CellEditorListener.html) l)

Adds the CellEditorListener.

**Specified by:**[addCellEditorListener](http://docs.google.com/javax/swing/CellEditor.html#addCellEditorListener(javax.swing.event.CellEditorListener)) in interface [CellEditor](http://docs.google.com/javax/swing/CellEditor.html) **Parameters:**l - the listener to be added

### removeCellEditorListener

public void **removeCellEditorListener**([CellEditorListener](http://docs.google.com/javax/swing/event/CellEditorListener.html) l)

Removes the previously added CellEditorListener.

**Specified by:**[removeCellEditorListener](http://docs.google.com/javax/swing/CellEditor.html#removeCellEditorListener(javax.swing.event.CellEditorListener)) in interface [CellEditor](http://docs.google.com/javax/swing/CellEditor.html) **Parameters:**l - the listener to be removed

### getCellEditorListeners

public [CellEditorListener](http://docs.google.com/javax/swing/event/CellEditorListener.html)[] **getCellEditorListeners**()

Returns an array of all the CellEditorListeners added to this DefaultTreeCellEditor with addCellEditorListener().

**Returns:**all of the CellEditorListeners added or an empty array if no listeners have been added**Since:** 1.4

### valueChanged

public void **valueChanged**([TreeSelectionEvent](http://docs.google.com/javax/swing/event/TreeSelectionEvent.html) e)

Resets lastPath.

**Specified by:**[valueChanged](http://docs.google.com/javax/swing/event/TreeSelectionListener.html#valueChanged(javax.swing.event.TreeSelectionEvent)) in interface [TreeSelectionListener](http://docs.google.com/javax/swing/event/TreeSelectionListener.html) **Parameters:**e - the event that characterizes the change.

### actionPerformed

public void **actionPerformed**([ActionEvent](http://docs.google.com/java/awt/event/ActionEvent.html) e)

Messaged when the timer fires, this will start the editing session.

**Specified by:**[actionPerformed](http://docs.google.com/java/awt/event/ActionListener.html#actionPerformed(java.awt.event.ActionEvent)) in interface [ActionListener](http://docs.google.com/java/awt/event/ActionListener.html)

### setTree

protected void **setTree**([JTree](http://docs.google.com/javax/swing/JTree.html) newTree)

Sets the tree currently editing for. This is needed to add a selection listener.

**Parameters:**newTree - the new tree to be edited

### shouldStartEditingTimer

protected boolean **shouldStartEditingTimer**([EventObject](http://docs.google.com/java/util/EventObject.html) event)

Returns true if event is a MouseEvent and the click count is 1.

**Parameters:**event - the event being studied

### startEditingTimer

protected void **startEditingTimer**()

Starts the editing timer.

### canEditImmediately

protected boolean **canEditImmediately**([EventObject](http://docs.google.com/java/util/EventObject.html) event)

Returns true if event is null, or it is a MouseEvent with a click count > 2 and inHitRegion returns true.

**Parameters:**event - the event being studied

### inHitRegion

protected boolean **inHitRegion**(int x,  
 int y)

Returns true if the passed in location is a valid mouse location to start editing from. This is implemented to return false if x is <= the width of the icon and icon gap displayed by the renderer. In other words this returns true if the user clicks over the text part displayed by the renderer, and false otherwise.

**Parameters:**x - the x-coordinate of the pointy - the y-coordinate of the point **Returns:**true if the passed in location is a valid mouse location

### determineOffset

protected void **determineOffset**([JTree](http://docs.google.com/javax/swing/JTree.html) tree,  
 [Object](http://docs.google.com/java/lang/Object.html) value,  
 boolean isSelected,  
 boolean expanded,  
 boolean leaf,  
 int row)

### prepareForEditing

protected void **prepareForEditing**()

Invoked just before editing is to start. Will add the editingComponent to the editingContainer.

### createContainer

protected [Container](http://docs.google.com/java/awt/Container.html) **createContainer**()

Creates the container to manage placement of editingComponent.

### createTreeCellEditor

protected [TreeCellEditor](http://docs.google.com/javax/swing/tree/TreeCellEditor.html) **createTreeCellEditor**()

This is invoked if a TreeCellEditor is not supplied in the constructor. It returns a TextField editor.

**Returns:**a new TextField editor

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/DefaultTreeCellEditor.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/tree/DefaultMutableTreeNode.html)   [**NEXT CLASS**](http://docs.google.com/javax/swing/tree/DefaultTreeCellEditor.DefaultTextField.html) | [**FRAMES**](http://docs.google.com/index.html?javax/swing/tree/DefaultTreeCellEditor.html)    [**NO FRAMES**](http://docs.google.com/DefaultTreeCellEditor.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: [NESTED](#3znysh7) | [FIELD](#2et92p0) | [CONSTR](#tyjcwt) | [METHOD](#3dy6vkm) | DETAIL: [FIELD](#4d34og8) | [CONSTR](#2xcytpi) | [METHOD](#2bn6wsx) |

[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).