| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/BasicTreeUI.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/plaf/basic/BasicToolTipUI.html)   [**NEXT CLASS**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.CellEditorHandler.html) | [**FRAMES**](http://docs.google.com/index.html?javax/swing/plaf/basic/BasicTreeUI.html)    [**NO FRAMES**](http://docs.google.com/BasicTreeUI.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: [NESTED](#3znysh7) | [FIELD](#2et92p0) | [CONSTR](#tyjcwt) | [METHOD](#3dy6vkm) | DETAIL: [FIELD](#2s8eyo1) | [CONSTR](#41mghml) | [METHOD](#vx1227) |

## **javax.swing.plaf.basic**

Class BasicTreeUI

[java.lang.Object](http://docs.google.com/java/lang/Object.html)  
 [javax.swing.plaf.ComponentUI](http://docs.google.com/javax/swing/plaf/ComponentUI.html)  
 [javax.swing.plaf.TreeUI](http://docs.google.com/javax/swing/plaf/TreeUI.html)  
 **javax.swing.plaf.basic.BasicTreeUI**

**Direct Known Subclasses:** [MetalTreeUI](http://docs.google.com/javax/swing/plaf/metal/MetalTreeUI.html)

public class **BasicTreeUI**extends [TreeUI](http://docs.google.com/javax/swing/plaf/TreeUI.html)

The basic L&F for a hierarchical data structure.

| **Nested Class Summary** | |
| --- | --- |
| class | [**BasicTreeUI.CellEditorHandler**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.CellEditorHandler.html)            Listener responsible for getting cell editing events and updating the tree accordingly. |
| class | [**BasicTreeUI.ComponentHandler**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.ComponentHandler.html)            Updates the preferred size when scrolling (if necessary). |
| class | [**BasicTreeUI.FocusHandler**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.FocusHandler.html)            Repaints the lead selection row when focus is lost/gained. |
| class | [**BasicTreeUI.KeyHandler**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.KeyHandler.html)            This is used to get mutliple key down events to appropriately generate events. |
| class | [**BasicTreeUI.MouseHandler**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.MouseHandler.html)            TreeMouseListener is responsible for updating the selection based on mouse events. |
| class | [**BasicTreeUI.MouseInputHandler**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.MouseInputHandler.html)            MouseInputHandler handles passing all mouse events, including mouse motion events, until the mouse is released to the destination it is constructed with. |
| class | [**BasicTreeUI.NodeDimensionsHandler**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.NodeDimensionsHandler.html)            Class responsible for getting size of node, method is forwarded to BasicTreeUI method. |
| class | [**BasicTreeUI.PropertyChangeHandler**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.PropertyChangeHandler.html)            PropertyChangeListener for the tree. |
| class | [**BasicTreeUI.SelectionModelPropertyChangeHandler**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.SelectionModelPropertyChangeHandler.html)            Listener on the TreeSelectionModel, resets the row selection if any of the properties of the model change. |
| class | [**BasicTreeUI.TreeCancelEditingAction**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.TreeCancelEditingAction.html)            ActionListener that invokes cancelEditing when action performed. |
| class | [**BasicTreeUI.TreeExpansionHandler**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.TreeExpansionHandler.html)            Updates the TreeState in response to nodes expanding/collapsing. |
| class | [**BasicTreeUI.TreeHomeAction**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.TreeHomeAction.html)            TreeHomeAction is used to handle end/home actions. |
| class | [**BasicTreeUI.TreeIncrementAction**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.TreeIncrementAction.html)            TreeIncrementAction is used to handle up/down actions. |
| class | [**BasicTreeUI.TreeModelHandler**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.TreeModelHandler.html)            Forwards all TreeModel events to the TreeState. |
| class | [**BasicTreeUI.TreePageAction**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.TreePageAction.html)            TreePageAction handles page up and page down events. |
| class | [**BasicTreeUI.TreeSelectionHandler**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.TreeSelectionHandler.html)            Listens for changes in the selection model and updates the display accordingly. |
| class | [**BasicTreeUI.TreeToggleAction**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.TreeToggleAction.html)            For the first selected row expandedness will be toggled. |
| class | [**BasicTreeUI.TreeTraverseAction**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.TreeTraverseAction.html)            TreeTraverseAction is the action used for left/right keys. |

| **Field Summary** | |
| --- | --- |
| protected  [TreeCellEditor](http://docs.google.com/javax/swing/tree/TreeCellEditor.html) | [**cellEditor**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.html#cellEditor)            Editor for the tree. |
| protected  [Icon](http://docs.google.com/javax/swing/Icon.html) | [**collapsedIcon**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.html#collapsedIcon) |
| protected  boolean | [**createdCellEditor**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.html#createdCellEditor)            Set to true if editor that is currently in the tree was created by this instance. |
| protected  boolean | [**createdRenderer**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.html#createdRenderer)            Set to true if the renderer that is currently in the tree was created by this instance. |
| protected  [TreeCellRenderer](http://docs.google.com/javax/swing/tree/TreeCellRenderer.html) | [**currentCellRenderer**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.html#currentCellRenderer)            Renderer that is being used to do the actual cell drawing. |
| protected  int | [**depthOffset**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.html#depthOffset)            How much the depth should be offset to properly calculate x locations. |
| protected  [Hashtable](http://docs.google.com/java/util/Hashtable.html)<[TreePath](http://docs.google.com/javax/swing/tree/TreePath.html),[Boolean](http://docs.google.com/java/lang/Boolean.html)> | [**drawingCache**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.html#drawingCache)            Used for minimizing the drawing of vertical lines. |
| protected  [Component](http://docs.google.com/java/awt/Component.html) | [**editingComponent**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.html#editingComponent)            When editing, this will be the Component that is doing the actual editing. |
| protected  [TreePath](http://docs.google.com/javax/swing/tree/TreePath.html) | [**editingPath**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.html#editingPath)            Path that is being edited. |
| protected  int | [**editingRow**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.html#editingRow)            Row that is being edited. |
| protected  boolean | [**editorHasDifferentSize**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.html#editorHasDifferentSize)            Set to true if the editor has a different size than the renderer. |
| protected  [Icon](http://docs.google.com/javax/swing/Icon.html) | [**expandedIcon**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.html#expandedIcon) |
| protected  boolean | [**largeModel**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.html#largeModel)            True if doing optimizations for a largeModel. |
| protected  int | [**lastSelectedRow**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.html#lastSelectedRow)            Index of the row that was last selected. |
| protected  int | [**leftChildIndent**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.html#leftChildIndent)            Distance between left margin and where vertical dashes will be drawn. |
| protected  [AbstractLayoutCache.NodeDimensions](http://docs.google.com/javax/swing/tree/AbstractLayoutCache.NodeDimensions.html) | [**nodeDimensions**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.html#nodeDimensions)            Reponsible for telling the TreeState the size needed for a node. |
| protected  [Dimension](http://docs.google.com/java/awt/Dimension.html) | [**preferredMinSize**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.html#preferredMinSize)            Minimum preferred size. |
| protected  [Dimension](http://docs.google.com/java/awt/Dimension.html) | [**preferredSize**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.html#preferredSize)            Size needed to completely display all the nodes. |
| protected  [CellRendererPane](http://docs.google.com/javax/swing/CellRendererPane.html) | [**rendererPane**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.html#rendererPane)            Used to paint the TreeCellRenderer. |
| protected  int | [**rightChildIndent**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.html#rightChildIndent)            Distance to add to leftChildIndent to determine where cell contents will be drawn. |
| protected  boolean | [**stopEditingInCompleteEditing**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.html#stopEditingInCompleteEditing)            Set to false when editing and shouldSelectCell() returns true meaning the node should be selected before editing, used in completeEditing. |
| protected  int | [**totalChildIndent**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.html#totalChildIndent)            Total distance that will be indented. |
| protected  [JTree](http://docs.google.com/javax/swing/JTree.html) | [**tree**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.html#tree)            Component that we're going to be drawing into. |
| protected  [TreeModel](http://docs.google.com/javax/swing/tree/TreeModel.html) | [**treeModel**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.html#treeModel)            Used to determine what to display. |
| protected  [TreeSelectionModel](http://docs.google.com/javax/swing/tree/TreeSelectionModel.html) | [**treeSelectionModel**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.html#treeSelectionModel)            Model maintaing the selection. |
| protected  [AbstractLayoutCache](http://docs.google.com/javax/swing/tree/AbstractLayoutCache.html) | [**treeState**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.html#treeState)            Object responsible for handling sizing and expanded issues. |
| protected  boolean | [**validCachedPreferredSize**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.html#validCachedPreferredSize)            Is the preferredSize valid? |

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

| **Method Summary** | |
| --- | --- |
| void | [**cancelEditing**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.html#cancelEditing(javax.swing.JTree))([JTree](http://docs.google.com/javax/swing/JTree.html) tree)            Cancels the current editing session. |
| protected  void | [**checkForClickInExpandControl**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.html#checkForClickInExpandControl(javax.swing.tree.TreePath,%20int,%20int))([TreePath](http://docs.google.com/javax/swing/tree/TreePath.html) path, int mouseX, int mouseY)            If the mouseX and mouseY are in the expand/collapse region of the row, this will toggle the row. |
| protected  void | [**completeEditing**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.html#completeEditing())()            Messages to stop the editing session. |
| protected  void | [**completeEditing**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.html#completeEditing(boolean,%20boolean,%20boolean))(boolean messageStop, boolean messageCancel, boolean messageTree)            Stops the editing session. |
| protected  void | [**completeUIInstall**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.html#completeUIInstall())()            Invoked from installUI after all the defaults/listeners have been installed. |
| protected  void | [**completeUIUninstall**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.html#completeUIUninstall())() |
| protected  void | [**configureLayoutCache**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.html#configureLayoutCache())()            Resets the TreeState instance based on the tree we're providing the look and feel for. |
| protected  [CellEditorListener](http://docs.google.com/javax/swing/event/CellEditorListener.html) | [**createCellEditorListener**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.html#createCellEditorListener())()            Creates a listener to handle events from the current editor. |
| protected  [CellRendererPane](http://docs.google.com/javax/swing/CellRendererPane.html) | [**createCellRendererPane**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.html#createCellRendererPane())()            Returns the renderer pane that renderer components are placed in. |
| protected  [ComponentListener](http://docs.google.com/java/awt/event/ComponentListener.html) | [**createComponentListener**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.html#createComponentListener())()            Creates and returns a new ComponentHandler. |
| protected  [TreeCellEditor](http://docs.google.com/javax/swing/tree/TreeCellEditor.html) | [**createDefaultCellEditor**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.html#createDefaultCellEditor())()            Creates a default cell editor. |
| protected  [TreeCellRenderer](http://docs.google.com/javax/swing/tree/TreeCellRenderer.html) | [**createDefaultCellRenderer**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.html#createDefaultCellRenderer())()            Returns the default cell renderer that is used to do the stamping of each node. |
| protected  [FocusListener](http://docs.google.com/java/awt/event/FocusListener.html) | [**createFocusListener**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.html#createFocusListener())()            Creates a listener that is responsible for updating the display when focus is lost/gained. |
| protected  [KeyListener](http://docs.google.com/java/awt/event/KeyListener.html) | [**createKeyListener**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.html#createKeyListener())()            Creates the listener reponsible for getting key events from the tree. |
| protected  [AbstractLayoutCache](http://docs.google.com/javax/swing/tree/AbstractLayoutCache.html) | [**createLayoutCache**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.html#createLayoutCache())()            Creates the object responsible for managing what is expanded, as well as the size of nodes. |
| protected  [MouseListener](http://docs.google.com/java/awt/event/MouseListener.html) | [**createMouseListener**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.html#createMouseListener())()            Creates the listener responsible for updating the selection based on mouse events. |
| protected  [AbstractLayoutCache.NodeDimensions](http://docs.google.com/javax/swing/tree/AbstractLayoutCache.NodeDimensions.html) | [**createNodeDimensions**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.html#createNodeDimensions())()            Creates an instance of NodeDimensions that is able to determine the size of a given node in the tree. |
| protected  [PropertyChangeListener](http://docs.google.com/java/beans/PropertyChangeListener.html) | [**createPropertyChangeListener**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.html#createPropertyChangeListener())()            Creates a listener that is responsible that updates the UI based on how the tree changes. |
| protected  [PropertyChangeListener](http://docs.google.com/java/beans/PropertyChangeListener.html) | [**createSelectionModelPropertyChangeListener**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.html#createSelectionModelPropertyChangeListener())()            Creates the listener responsible for getting property change events from the selection model. |
| protected  [TreeExpansionListener](http://docs.google.com/javax/swing/event/TreeExpansionListener.html) | [**createTreeExpansionListener**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.html#createTreeExpansionListener())()            Creates and returns the object responsible for updating the treestate when nodes expanded state changes. |
| protected  [TreeModelListener](http://docs.google.com/javax/swing/event/TreeModelListener.html) | [**createTreeModelListener**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.html#createTreeModelListener())()            Returns a listener that can update the tree when the model changes. |
| protected  [TreeSelectionListener](http://docs.google.com/javax/swing/event/TreeSelectionListener.html) | [**createTreeSelectionListener**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.html#createTreeSelectionListener())()            Creates the listener that updates the display based on selection change methods. |
| static [ComponentUI](http://docs.google.com/javax/swing/plaf/ComponentUI.html) | [**createUI**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.html#createUI(javax.swing.JComponent))([JComponent](http://docs.google.com/javax/swing/JComponent.html) x) |
| protected  void | [**drawCentered**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.html#drawCentered(java.awt.Component,%20java.awt.Graphics,%20javax.swing.Icon,%20int,%20int))([Component](http://docs.google.com/java/awt/Component.html) c, [Graphics](http://docs.google.com/java/awt/Graphics.html) graphics, [Icon](http://docs.google.com/javax/swing/Icon.html) icon, int x, int y) |
| protected  void | [**drawDashedHorizontalLine**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.html#drawDashedHorizontalLine(java.awt.Graphics,%20int,%20int,%20int))([Graphics](http://docs.google.com/java/awt/Graphics.html) g, int y, int x1, int x2) |
| protected  void | [**drawDashedVerticalLine**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.html#drawDashedVerticalLine(java.awt.Graphics,%20int,%20int,%20int))([Graphics](http://docs.google.com/java/awt/Graphics.html) g, int x, int y1, int y2) |
| protected  void | [**ensureRowsAreVisible**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.html#ensureRowsAreVisible(int,%20int))(int beginRow, int endRow)            Ensures that the rows identified by beginRow through endRow are visible. |
| int | [**getBaseline**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.html#getBaseline(javax.swing.JComponent,%20int,%20int))([JComponent](http://docs.google.com/javax/swing/JComponent.html) c, int width, int height)            Returns the baseline. |
| [Component.BaselineResizeBehavior](http://docs.google.com/java/awt/Component.BaselineResizeBehavior.html) | [**getBaselineResizeBehavior**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.html#getBaselineResizeBehavior(javax.swing.JComponent))([JComponent](http://docs.google.com/javax/swing/JComponent.html) c)            Returns an enum indicating how the baseline of the component changes as the size changes. |
| protected  [TreeCellEditor](http://docs.google.com/javax/swing/tree/TreeCellEditor.html) | [**getCellEditor**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.html#getCellEditor())() |
| protected  [TreeCellRenderer](http://docs.google.com/javax/swing/tree/TreeCellRenderer.html) | [**getCellRenderer**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.html#getCellRenderer())()            Return currentCellRenderer, which will either be the trees renderer, or defaultCellRenderer, which ever wasn't null. |
| [TreePath](http://docs.google.com/javax/swing/tree/TreePath.html) | [**getClosestPathForLocation**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.html#getClosestPathForLocation(javax.swing.JTree,%20int,%20int))([JTree](http://docs.google.com/javax/swing/JTree.html) tree, int x, int y)            Returns the path to the node that is closest to x,y. |
| [Icon](http://docs.google.com/javax/swing/Icon.html) | [**getCollapsedIcon**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.html#getCollapsedIcon())() |
| [TreePath](http://docs.google.com/javax/swing/tree/TreePath.html) | [**getEditingPath**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.html#getEditingPath(javax.swing.JTree))([JTree](http://docs.google.com/javax/swing/JTree.html) tree)            Returns the path to the element that is being edited. |
| [Icon](http://docs.google.com/javax/swing/Icon.html) | [**getExpandedIcon**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.html#getExpandedIcon())() |
| protected  [Color](http://docs.google.com/java/awt/Color.html) | [**getHashColor**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.html#getHashColor())() |
| protected  int | [**getHorizontalLegBuffer**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.html#getHorizontalLegBuffer())()            The horizontal element of legs between nodes starts at the right of the left-hand side of the child node by default. |
| protected  [TreePath](http://docs.google.com/javax/swing/tree/TreePath.html) | [**getLastChildPath**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.html#getLastChildPath(javax.swing.tree.TreePath))([TreePath](http://docs.google.com/javax/swing/tree/TreePath.html) parent)            Returns a path to the last child of parent. |
| int | [**getLeftChildIndent**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.html#getLeftChildIndent())() |
| [Dimension](http://docs.google.com/java/awt/Dimension.html) | [**getMaximumSize**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.html#getMaximumSize(javax.swing.JComponent))([JComponent](http://docs.google.com/javax/swing/JComponent.html) c)            Returns the maximum size for this component, which will be the preferred size if the instance is currently in a JTree, or 0, 0. |
| [Dimension](http://docs.google.com/java/awt/Dimension.html) | [**getMinimumSize**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.html#getMinimumSize(javax.swing.JComponent))([JComponent](http://docs.google.com/javax/swing/JComponent.html) c)            Returns the minimum size for this component. |
| protected  [TreeModel](http://docs.google.com/javax/swing/tree/TreeModel.html) | [**getModel**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.html#getModel())() |
| [Rectangle](http://docs.google.com/java/awt/Rectangle.html) | [**getPathBounds**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.html#getPathBounds(javax.swing.JTree,%20javax.swing.tree.TreePath))([JTree](http://docs.google.com/javax/swing/JTree.html) tree, [TreePath](http://docs.google.com/javax/swing/tree/TreePath.html) path)            Returns the Rectangle enclosing the label portion that the last item in path will be drawn into. |
| [TreePath](http://docs.google.com/javax/swing/tree/TreePath.html) | [**getPathForRow**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.html#getPathForRow(javax.swing.JTree,%20int))([JTree](http://docs.google.com/javax/swing/JTree.html) tree, int row)            Returns the path for passed in row. |
| [Dimension](http://docs.google.com/java/awt/Dimension.html) | [**getPreferredMinSize**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.html#getPreferredMinSize())()            Returns the minimum preferred size. |
| [Dimension](http://docs.google.com/java/awt/Dimension.html) | [**getPreferredSize**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.html#getPreferredSize(javax.swing.JComponent))([JComponent](http://docs.google.com/javax/swing/JComponent.html) c)            Returns the preferred size to properly display the tree, this is a cover method for getPreferredSize(c, false). |
| [Dimension](http://docs.google.com/java/awt/Dimension.html) | [**getPreferredSize**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.html#getPreferredSize(javax.swing.JComponent,%20boolean))([JComponent](http://docs.google.com/javax/swing/JComponent.html) c, boolean checkConsistancy)            Returns the preferred size to represent the tree in *c*. |
| int | [**getRightChildIndent**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.html#getRightChildIndent())() |
| int | [**getRowCount**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.html#getRowCount(javax.swing.JTree))([JTree](http://docs.google.com/javax/swing/JTree.html) tree)            Returns the number of rows that are being displayed. |
| int | [**getRowForPath**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.html#getRowForPath(javax.swing.JTree,%20javax.swing.tree.TreePath))([JTree](http://docs.google.com/javax/swing/JTree.html) tree, [TreePath](http://docs.google.com/javax/swing/tree/TreePath.html) path)            Returns the row that the last item identified in path is visible at. |
| protected  int | [**getRowHeight**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.html#getRowHeight())() |
| protected  int | [**getRowX**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.html#getRowX(int,%20int))(int row, int depth)            Returns the location, along the x-axis, to render a particular row at. |
| protected  [TreeSelectionModel](http://docs.google.com/javax/swing/tree/TreeSelectionModel.html) | [**getSelectionModel**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.html#getSelectionModel())() |
| protected  boolean | [**getShowsRootHandles**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.html#getShowsRootHandles())() |
| protected  int | [**getVerticalLegBuffer**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.html#getVerticalLegBuffer())()            The vertical element of legs between nodes starts at the bottom of the parent node by default. |
| protected  void | [**handleExpandControlClick**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.html#handleExpandControlClick(javax.swing.tree.TreePath,%20int,%20int))([TreePath](http://docs.google.com/javax/swing/tree/TreePath.html) path, int mouseX, int mouseY)            Messaged when the user clicks the particular row, this invokes toggleExpandState. |
| protected  void | [**installComponents**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.html#installComponents())()            Intalls the subcomponents of the tree, which is the renderer pane. |
| protected  void | [**installDefaults**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.html#installDefaults())() |
| protected  void | [**installKeyboardActions**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.html#installKeyboardActions())() |
| protected  void | [**installListeners**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.html#installListeners())() |
| void | [**installUI**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.html#installUI(javax.swing.JComponent))([JComponent](http://docs.google.com/javax/swing/JComponent.html) c)            Configures the specified component appropriate for the look and feel. |
| protected  boolean | [**isEditable**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.html#isEditable())() |
| boolean | [**isEditing**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.html#isEditing(javax.swing.JTree))([JTree](http://docs.google.com/javax/swing/JTree.html) tree)            Returns true if the tree is being edited. |
| protected  boolean | [**isLargeModel**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.html#isLargeModel())() |
| protected  boolean | [**isLeaf**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.html#isLeaf(int))(int row) |
| protected  boolean | [**isLocationInExpandControl**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.html#isLocationInExpandControl(javax.swing.tree.TreePath,%20int,%20int))([TreePath](http://docs.google.com/javax/swing/tree/TreePath.html) path, int mouseX, int mouseY)            Returns true if mouseX and mouseY fall in the area of row that is used to expand/collapse the node and the node at row does not represent a leaf. |
| protected  boolean | [**isMultiSelectEvent**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.html#isMultiSelectEvent(java.awt.event.MouseEvent))([MouseEvent](http://docs.google.com/java/awt/event/MouseEvent.html) event)            Returning true signifies a mouse event on the node should select from the anchor point. |
| protected  boolean | [**isRootVisible**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.html#isRootVisible())() |
| protected  boolean | [**isToggleEvent**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.html#isToggleEvent(java.awt.event.MouseEvent))([MouseEvent](http://docs.google.com/java/awt/event/MouseEvent.html) event)            Returning true indicates the row under the mouse should be toggled based on the event. |
| protected  boolean | [**isToggleSelectionEvent**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.html#isToggleSelectionEvent(java.awt.event.MouseEvent))([MouseEvent](http://docs.google.com/java/awt/event/MouseEvent.html) event)            Returning true signifies a mouse event on the node should toggle the selection of only the row under mouse. |
| void | [**paint**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.html#paint(java.awt.Graphics,%20javax.swing.JComponent))([Graphics](http://docs.google.com/java/awt/Graphics.html) g, [JComponent](http://docs.google.com/javax/swing/JComponent.html) c)            Paints the specified component appropriate for the look and feel. |
| protected  void | [**paintExpandControl**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.html#paintExpandControl(java.awt.Graphics,%20java.awt.Rectangle,%20java.awt.Insets,%20java.awt.Rectangle,%20javax.swing.tree.TreePath,%20int,%20boolean,%20boolean,%20boolean))([Graphics](http://docs.google.com/java/awt/Graphics.html) g, [Rectangle](http://docs.google.com/java/awt/Rectangle.html) clipBounds, [Insets](http://docs.google.com/java/awt/Insets.html) insets, [Rectangle](http://docs.google.com/java/awt/Rectangle.html) bounds, [TreePath](http://docs.google.com/javax/swing/tree/TreePath.html) path, int row, boolean isExpanded, boolean hasBeenExpanded, boolean isLeaf)            Paints the expand (toggle) part of a row. |
| protected  void | [**paintHorizontalLine**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.html#paintHorizontalLine(java.awt.Graphics,%20javax.swing.JComponent,%20int,%20int,%20int))([Graphics](http://docs.google.com/java/awt/Graphics.html) g, [JComponent](http://docs.google.com/javax/swing/JComponent.html) c, int y, int left, int right)            Paints a horizontal line. |
| protected  void | [**paintHorizontalPartOfLeg**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.html#paintHorizontalPartOfLeg(java.awt.Graphics,%20java.awt.Rectangle,%20java.awt.Insets,%20java.awt.Rectangle,%20javax.swing.tree.TreePath,%20int,%20boolean,%20boolean,%20boolean))([Graphics](http://docs.google.com/java/awt/Graphics.html) g, [Rectangle](http://docs.google.com/java/awt/Rectangle.html) clipBounds, [Insets](http://docs.google.com/java/awt/Insets.html) insets, [Rectangle](http://docs.google.com/java/awt/Rectangle.html) bounds, [TreePath](http://docs.google.com/javax/swing/tree/TreePath.html) path, int row, boolean isExpanded, boolean hasBeenExpanded, boolean isLeaf)            Paints the horizontal part of the leg. |
| protected  void | [**paintRow**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.html#paintRow(java.awt.Graphics,%20java.awt.Rectangle,%20java.awt.Insets,%20java.awt.Rectangle,%20javax.swing.tree.TreePath,%20int,%20boolean,%20boolean,%20boolean))([Graphics](http://docs.google.com/java/awt/Graphics.html) g, [Rectangle](http://docs.google.com/java/awt/Rectangle.html) clipBounds, [Insets](http://docs.google.com/java/awt/Insets.html) insets, [Rectangle](http://docs.google.com/java/awt/Rectangle.html) bounds, [TreePath](http://docs.google.com/javax/swing/tree/TreePath.html) path, int row, boolean isExpanded, boolean hasBeenExpanded, boolean isLeaf)            Paints the renderer part of a row. |
| protected  void | [**paintVerticalLine**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.html#paintVerticalLine(java.awt.Graphics,%20javax.swing.JComponent,%20int,%20int,%20int))([Graphics](http://docs.google.com/java/awt/Graphics.html) g, [JComponent](http://docs.google.com/javax/swing/JComponent.html) c, int x, int top, int bottom)            Paints a vertical line. |
| protected  void | [**paintVerticalPartOfLeg**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.html#paintVerticalPartOfLeg(java.awt.Graphics,%20java.awt.Rectangle,%20java.awt.Insets,%20javax.swing.tree.TreePath))([Graphics](http://docs.google.com/java/awt/Graphics.html) g, [Rectangle](http://docs.google.com/java/awt/Rectangle.html) clipBounds, [Insets](http://docs.google.com/java/awt/Insets.html) insets, [TreePath](http://docs.google.com/javax/swing/tree/TreePath.html) path)            Paints the vertical part of the leg. |
| protected  void | [**pathWasCollapsed**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.html#pathWasCollapsed(javax.swing.tree.TreePath))([TreePath](http://docs.google.com/javax/swing/tree/TreePath.html) path)            Messaged from the VisibleTreeNode after it has collapsed. |
| protected  void | [**pathWasExpanded**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.html#pathWasExpanded(javax.swing.tree.TreePath))([TreePath](http://docs.google.com/javax/swing/tree/TreePath.html) path)            Messaged from the VisibleTreeNode after it has been expanded. |
| protected  void | [**prepareForUIInstall**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.html#prepareForUIInstall())()            Invoked after the tree instance variable has been set, but before any defaults/listeners have been installed. |
| protected  void | [**prepareForUIUninstall**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.html#prepareForUIUninstall())() |
| protected  void | [**selectPathForEvent**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.html#selectPathForEvent(javax.swing.tree.TreePath,%20java.awt.event.MouseEvent))([TreePath](http://docs.google.com/javax/swing/tree/TreePath.html) path, [MouseEvent](http://docs.google.com/java/awt/event/MouseEvent.html) event)            Messaged to update the selection based on a MouseEvent over a particular row. |
| protected  void | [**setCellEditor**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.html#setCellEditor(javax.swing.tree.TreeCellEditor))([TreeCellEditor](http://docs.google.com/javax/swing/tree/TreeCellEditor.html) editor)            Sets the cell editor. |
| protected  void | [**setCellRenderer**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.html#setCellRenderer(javax.swing.tree.TreeCellRenderer))([TreeCellRenderer](http://docs.google.com/javax/swing/tree/TreeCellRenderer.html) tcr)            Sets the TreeCellRenderer to tcr. |
| void | [**setCollapsedIcon**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.html#setCollapsedIcon(javax.swing.Icon))([Icon](http://docs.google.com/javax/swing/Icon.html) newG) |
| protected  void | [**setEditable**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.html#setEditable(boolean))(boolean newValue)            Configures the receiver to allow, or not allow, editing. |
| void | [**setExpandedIcon**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.html#setExpandedIcon(javax.swing.Icon))([Icon](http://docs.google.com/javax/swing/Icon.html) newG) |
| protected  void | [**setHashColor**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.html#setHashColor(java.awt.Color))([Color](http://docs.google.com/java/awt/Color.html) color) |
| protected  void | [**setLargeModel**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.html#setLargeModel(boolean))(boolean largeModel)            Updates the componentListener, if necessary. |
| void | [**setLeftChildIndent**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.html#setLeftChildIndent(int))(int newAmount) |
| protected  void | [**setModel**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.html#setModel(javax.swing.tree.TreeModel))([TreeModel](http://docs.google.com/javax/swing/tree/TreeModel.html) model)            Sets the TreeModel. |
| void | [**setPreferredMinSize**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.html#setPreferredMinSize(java.awt.Dimension))([Dimension](http://docs.google.com/java/awt/Dimension.html) newSize)            Sets the preferred minimum size. |
| void | [**setRightChildIndent**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.html#setRightChildIndent(int))(int newAmount) |
| protected  void | [**setRootVisible**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.html#setRootVisible(boolean))(boolean newValue)            Sets the root to being visible. |
| protected  void | [**setRowHeight**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.html#setRowHeight(int))(int rowHeight)            Sets the row height, this is forwarded to the treeState. |
| protected  void | [**setSelectionModel**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.html#setSelectionModel(javax.swing.tree.TreeSelectionModel))([TreeSelectionModel](http://docs.google.com/javax/swing/tree/TreeSelectionModel.html) newLSM)            Resets the selection model. |
| protected  void | [**setShowsRootHandles**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.html#setShowsRootHandles(boolean))(boolean newValue)            Determines whether the node handles are to be displayed. |
| protected  boolean | [**shouldPaintExpandControl**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.html#shouldPaintExpandControl(javax.swing.tree.TreePath,%20int,%20boolean,%20boolean,%20boolean))([TreePath](http://docs.google.com/javax/swing/tree/TreePath.html) path, int row, boolean isExpanded, boolean hasBeenExpanded, boolean isLeaf)            Returns true if the expand (toggle) control should be drawn for the specified row. |
| protected  boolean | [**startEditing**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.html#startEditing(javax.swing.tree.TreePath,%20java.awt.event.MouseEvent))([TreePath](http://docs.google.com/javax/swing/tree/TreePath.html) path, [MouseEvent](http://docs.google.com/java/awt/event/MouseEvent.html) event)            Will start editing for node if there is a cellEditor and shouldSelectCell returns true. |
| void | [**startEditingAtPath**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.html#startEditingAtPath(javax.swing.JTree,%20javax.swing.tree.TreePath))([JTree](http://docs.google.com/javax/swing/JTree.html) tree, [TreePath](http://docs.google.com/javax/swing/tree/TreePath.html) path)            Selects the last item in path and tries to edit it. |
| boolean | [**stopEditing**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.html#stopEditing(javax.swing.JTree))([JTree](http://docs.google.com/javax/swing/JTree.html) tree)            Stops the current editing session. |
| protected  void | [**toggleExpandState**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.html#toggleExpandState(javax.swing.tree.TreePath))([TreePath](http://docs.google.com/javax/swing/tree/TreePath.html) path)            Expands path if it is not expanded, or collapses row if it is expanded. |
| protected  void | [**uninstallComponents**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.html#uninstallComponents())()            Uninstalls the renderer pane. |
| protected  void | [**uninstallDefaults**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.html#uninstallDefaults())() |
| protected  void | [**uninstallKeyboardActions**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.html#uninstallKeyboardActions())() |
| protected  void | [**uninstallListeners**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.html#uninstallListeners())() |
| void | [**uninstallUI**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.html#uninstallUI(javax.swing.JComponent))([JComponent](http://docs.google.com/javax/swing/JComponent.html) c)            Reverses configuration which was done on the specified component during installUI. |
| protected  void | [**updateCachedPreferredSize**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.html#updateCachedPreferredSize())()            Updates the preferredSize instance variable, which is returned from getPreferredSize(). |
| protected  void | [**updateCellEditor**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.html#updateCellEditor())()            Updates the cellEditor based on the editability of the JTree that we're contained in. |
| protected  void | [**updateDepthOffset**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.html#updateDepthOffset())()            Updates how much each depth should be offset by. |
| protected  void | [**updateExpandedDescendants**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.html#updateExpandedDescendants(javax.swing.tree.TreePath))([TreePath](http://docs.google.com/javax/swing/tree/TreePath.html) path)            Updates the expanded state of all the descendants of path by getting the expanded descendants from the tree and forwarding to the tree state. |
| protected  void | [**updateLayoutCacheExpandedNodes**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.html#updateLayoutCacheExpandedNodes())()            Makes all the nodes that are expanded in JTree expanded in LayoutCache. |
| protected  void | [**updateRenderer**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.html#updateRenderer())()            Messaged from the tree we're in when the renderer has changed. |
| protected  void | [**updateSize**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.html#updateSize())()            Marks the cached size as being invalid, and messages the tree with treeDidChange. |

| **Methods inherited from class javax.swing.plaf.**[**ComponentUI**](http://docs.google.com/javax/swing/plaf/ComponentUI.html) |
| --- |
| [contains](http://docs.google.com/javax/swing/plaf/ComponentUI.html#contains(javax.swing.JComponent,%20int,%20int)), [getAccessibleChild](http://docs.google.com/javax/swing/plaf/ComponentUI.html#getAccessibleChild(javax.swing.JComponent,%20int)), [getAccessibleChildrenCount](http://docs.google.com/javax/swing/plaf/ComponentUI.html#getAccessibleChildrenCount(javax.swing.JComponent)), [update](http://docs.google.com/javax/swing/plaf/ComponentUI.html#update(java.awt.Graphics,%20javax.swing.JComponent)) |

| **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** |
| --- |

### collapsedIcon

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

### expandedIcon

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

### leftChildIndent

protected int **leftChildIndent**

Distance between left margin and where vertical dashes will be drawn.

### rightChildIndent

protected int **rightChildIndent**

Distance to add to leftChildIndent to determine where cell contents will be drawn.

### totalChildIndent

protected int **totalChildIndent**

Total distance that will be indented. The sum of leftChildIndent and rightChildIndent.

### preferredMinSize

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

Minimum preferred size.

### lastSelectedRow

protected int **lastSelectedRow**

Index of the row that was last selected.

### tree

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

Component that we're going to be drawing into.

### currentCellRenderer

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

Renderer that is being used to do the actual cell drawing.

### createdRenderer

protected boolean **createdRenderer**

Set to true if the renderer that is currently in the tree was created by this instance.

### cellEditor

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

Editor for the tree.

### createdCellEditor

protected boolean **createdCellEditor**

Set to true if editor that is currently in the tree was created by this instance.

### stopEditingInCompleteEditing

protected boolean **stopEditingInCompleteEditing**

Set to false when editing and shouldSelectCell() returns true meaning the node should be selected before editing, used in completeEditing.

### rendererPane

protected [CellRendererPane](http://docs.google.com/javax/swing/CellRendererPane.html) **rendererPane**

Used to paint the TreeCellRenderer.

### preferredSize

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

Size needed to completely display all the nodes.

### validCachedPreferredSize

protected boolean **validCachedPreferredSize**

Is the preferredSize valid?

### treeState

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

Object responsible for handling sizing and expanded issues.

### drawingCache

protected [Hashtable](http://docs.google.com/java/util/Hashtable.html)<[TreePath](http://docs.google.com/javax/swing/tree/TreePath.html),[Boolean](http://docs.google.com/java/lang/Boolean.html)> **drawingCache**

Used for minimizing the drawing of vertical lines.

### largeModel

protected boolean **largeModel**

True if doing optimizations for a largeModel. Subclasses that don't support this may wish to override createLayoutCache to not return a FixedHeightLayoutCache instance.

### nodeDimensions

protected [AbstractLayoutCache.NodeDimensions](http://docs.google.com/javax/swing/tree/AbstractLayoutCache.NodeDimensions.html) **nodeDimensions**

Reponsible for telling the TreeState the size needed for a node.

### treeModel

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

Used to determine what to display.

### treeSelectionModel

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

Model maintaing the selection.

### depthOffset

protected int **depthOffset**

How much the depth should be offset to properly calculate x locations. This is based on whether or not the root is visible, and if the root handles are visible.

### editingComponent

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

When editing, this will be the Component that is doing the actual editing.

### editingPath

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

Path that is being edited.

### editingRow

protected int **editingRow**

Row that is being edited. Should only be referenced if editingComponent is not null.

### editorHasDifferentSize

protected boolean **editorHasDifferentSize**

Set to true if the editor has a different size than the renderer.

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

### BasicTreeUI

public **BasicTreeUI**()

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

### createUI

public static [ComponentUI](http://docs.google.com/javax/swing/plaf/ComponentUI.html) **createUI**([JComponent](http://docs.google.com/javax/swing/JComponent.html) x)

### getHashColor

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

### setHashColor

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

### setLeftChildIndent

public void **setLeftChildIndent**(int newAmount)

### getLeftChildIndent

public int **getLeftChildIndent**()

### setRightChildIndent

public void **setRightChildIndent**(int newAmount)

### getRightChildIndent

public int **getRightChildIndent**()

### setExpandedIcon

public void **setExpandedIcon**([Icon](http://docs.google.com/javax/swing/Icon.html) newG)

### getExpandedIcon

public [Icon](http://docs.google.com/javax/swing/Icon.html) **getExpandedIcon**()

### setCollapsedIcon

public void **setCollapsedIcon**([Icon](http://docs.google.com/javax/swing/Icon.html) newG)

### getCollapsedIcon

public [Icon](http://docs.google.com/javax/swing/Icon.html) **getCollapsedIcon**()

### setLargeModel

protected void **setLargeModel**(boolean largeModel)

Updates the componentListener, if necessary.

### isLargeModel

protected boolean **isLargeModel**()

### setRowHeight

protected void **setRowHeight**(int rowHeight)

Sets the row height, this is forwarded to the treeState.

### getRowHeight

protected int **getRowHeight**()

### setCellRenderer

protected void **setCellRenderer**([TreeCellRenderer](http://docs.google.com/javax/swing/tree/TreeCellRenderer.html) tcr)

Sets the TreeCellRenderer to tcr. This invokes updateRenderer.

### getCellRenderer

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

Return currentCellRenderer, which will either be the trees renderer, or defaultCellRenderer, which ever wasn't null.

### setModel

protected void **setModel**([TreeModel](http://docs.google.com/javax/swing/tree/TreeModel.html) model)

Sets the TreeModel.

### getModel

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

### setRootVisible

protected void **setRootVisible**(boolean newValue)

Sets the root to being visible.

### isRootVisible

protected boolean **isRootVisible**()

### setShowsRootHandles

protected void **setShowsRootHandles**(boolean newValue)

Determines whether the node handles are to be displayed.

### getShowsRootHandles

protected boolean **getShowsRootHandles**()

### setCellEditor

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

Sets the cell editor.

### getCellEditor

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

### setEditable

protected void **setEditable**(boolean newValue)

Configures the receiver to allow, or not allow, editing.

### isEditable

protected boolean **isEditable**()

### setSelectionModel

protected void **setSelectionModel**([TreeSelectionModel](http://docs.google.com/javax/swing/tree/TreeSelectionModel.html) newLSM)

Resets the selection model. The appropriate listener are installed on the model.

### getSelectionModel

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

### getPathBounds

public [Rectangle](http://docs.google.com/java/awt/Rectangle.html) **getPathBounds**([JTree](http://docs.google.com/javax/swing/JTree.html) tree,  
 [TreePath](http://docs.google.com/javax/swing/tree/TreePath.html) path)

Returns the Rectangle enclosing the label portion that the last item in path will be drawn into. Will return null if any component in path is currently valid.

**Specified by:**[getPathBounds](http://docs.google.com/javax/swing/plaf/TreeUI.html#getPathBounds(javax.swing.JTree,%20javax.swing.tree.TreePath)) in class [TreeUI](http://docs.google.com/javax/swing/plaf/TreeUI.html)

### getPathForRow

public [TreePath](http://docs.google.com/javax/swing/tree/TreePath.html) **getPathForRow**([JTree](http://docs.google.com/javax/swing/JTree.html) tree,  
 int row)

Returns the path for passed in row. If row is not visible null is returned.

**Specified by:**[getPathForRow](http://docs.google.com/javax/swing/plaf/TreeUI.html#getPathForRow(javax.swing.JTree,%20int)) in class [TreeUI](http://docs.google.com/javax/swing/plaf/TreeUI.html)

### getRowForPath

public int **getRowForPath**([JTree](http://docs.google.com/javax/swing/JTree.html) tree,  
 [TreePath](http://docs.google.com/javax/swing/tree/TreePath.html) path)

Returns the row that the last item identified in path is visible at. Will return -1 if any of the elements in path are not currently visible.

**Specified by:**[getRowForPath](http://docs.google.com/javax/swing/plaf/TreeUI.html#getRowForPath(javax.swing.JTree,%20javax.swing.tree.TreePath)) in class [TreeUI](http://docs.google.com/javax/swing/plaf/TreeUI.html)

### getRowCount

public int **getRowCount**([JTree](http://docs.google.com/javax/swing/JTree.html) tree)

Returns the number of rows that are being displayed.

**Specified by:**[getRowCount](http://docs.google.com/javax/swing/plaf/TreeUI.html#getRowCount(javax.swing.JTree)) in class [TreeUI](http://docs.google.com/javax/swing/plaf/TreeUI.html)

### getClosestPathForLocation

public [TreePath](http://docs.google.com/javax/swing/tree/TreePath.html) **getClosestPathForLocation**([JTree](http://docs.google.com/javax/swing/JTree.html) tree,  
 int x,  
 int y)

Returns the path to the node that is closest to x,y. If there is nothing currently visible this will return null, otherwise it'll always return a valid path. If you need to test if the returned object is exactly at x, y you should get the bounds for the returned path and test x, y against that.

**Specified by:**[getClosestPathForLocation](http://docs.google.com/javax/swing/plaf/TreeUI.html#getClosestPathForLocation(javax.swing.JTree,%20int,%20int)) in class [TreeUI](http://docs.google.com/javax/swing/plaf/TreeUI.html)

### isEditing

public boolean **isEditing**([JTree](http://docs.google.com/javax/swing/JTree.html) tree)

Returns true if the tree is being edited. The item that is being edited can be returned by getEditingPath().

**Specified by:**[isEditing](http://docs.google.com/javax/swing/plaf/TreeUI.html#isEditing(javax.swing.JTree)) in class [TreeUI](http://docs.google.com/javax/swing/plaf/TreeUI.html)

### stopEditing

public boolean **stopEditing**([JTree](http://docs.google.com/javax/swing/JTree.html) tree)

Stops the current editing session. This has no effect if the tree isn't being edited. Returns true if the editor allows the editing session to stop.

**Specified by:**[stopEditing](http://docs.google.com/javax/swing/plaf/TreeUI.html#stopEditing(javax.swing.JTree)) in class [TreeUI](http://docs.google.com/javax/swing/plaf/TreeUI.html)

### cancelEditing

public void **cancelEditing**([JTree](http://docs.google.com/javax/swing/JTree.html) tree)

Cancels the current editing session.

**Specified by:**[cancelEditing](http://docs.google.com/javax/swing/plaf/TreeUI.html#cancelEditing(javax.swing.JTree)) in class [TreeUI](http://docs.google.com/javax/swing/plaf/TreeUI.html)

### startEditingAtPath

public void **startEditingAtPath**([JTree](http://docs.google.com/javax/swing/JTree.html) tree,  
 [TreePath](http://docs.google.com/javax/swing/tree/TreePath.html) path)

Selects the last item in path and tries to edit it. Editing will fail if the CellEditor won't allow it for the selected item.

**Specified by:**[startEditingAtPath](http://docs.google.com/javax/swing/plaf/TreeUI.html#startEditingAtPath(javax.swing.JTree,%20javax.swing.tree.TreePath)) in class [TreeUI](http://docs.google.com/javax/swing/plaf/TreeUI.html)

### getEditingPath

public [TreePath](http://docs.google.com/javax/swing/tree/TreePath.html) **getEditingPath**([JTree](http://docs.google.com/javax/swing/JTree.html) tree)

Returns the path to the element that is being edited.

**Specified by:**[getEditingPath](http://docs.google.com/javax/swing/plaf/TreeUI.html#getEditingPath(javax.swing.JTree)) in class [TreeUI](http://docs.google.com/javax/swing/plaf/TreeUI.html)

### installUI

public void **installUI**([JComponent](http://docs.google.com/javax/swing/JComponent.html) c)

**Description copied from class:** [**ComponentUI**](http://docs.google.com/javax/swing/plaf/ComponentUI.html#installUI(javax.swing.JComponent)) Configures the specified component appropriate for the look and feel. This method is invoked when the ComponentUI instance is being installed as the UI delegate on the specified component. This method should completely configure the component for the look and feel, including the following:

1. Install any default property values for color, fonts, borders, icons, opacity, etc. on the component. Whenever possible, property values initialized by the client program should *not* be overridden.
2. Install a LayoutManager on the component if necessary.
3. Create/add any required sub-components to the component.
4. Create/install event listeners on the component.
5. Create/install a PropertyChangeListener on the component in order to detect and respond to component property changes appropriately.
6. Install keyboard UI (mnemonics, traversal, etc.) on the component.
7. Initialize any appropriate instance data.

**Overrides:**[installUI](http://docs.google.com/javax/swing/plaf/ComponentUI.html#installUI(javax.swing.JComponent)) in class [ComponentUI](http://docs.google.com/javax/swing/plaf/ComponentUI.html) **Parameters:**c - the component where this UI delegate is being installed**See Also:**[ComponentUI.uninstallUI(javax.swing.JComponent)](http://docs.google.com/javax/swing/plaf/ComponentUI.html#uninstallUI(javax.swing.JComponent)), [JComponent.setUI(javax.swing.plaf.ComponentUI)](http://docs.google.com/javax/swing/JComponent.html#setUI(javax.swing.plaf.ComponentUI)), [JComponent.updateUI()](http://docs.google.com/javax/swing/JComponent.html#updateUI())

### prepareForUIInstall

protected void **prepareForUIInstall**()

Invoked after the tree instance variable has been set, but before any defaults/listeners have been installed.

### completeUIInstall

protected void **completeUIInstall**()

Invoked from installUI after all the defaults/listeners have been installed.

### installDefaults

protected void **installDefaults**()

### installListeners

protected void **installListeners**()

### installKeyboardActions

protected void **installKeyboardActions**()

### installComponents

protected void **installComponents**()

Intalls the subcomponents of the tree, which is the renderer pane.

### createNodeDimensions

protected [AbstractLayoutCache.NodeDimensions](http://docs.google.com/javax/swing/tree/AbstractLayoutCache.NodeDimensions.html) **createNodeDimensions**()

Creates an instance of NodeDimensions that is able to determine the size of a given node in the tree.

### createPropertyChangeListener

protected [PropertyChangeListener](http://docs.google.com/java/beans/PropertyChangeListener.html) **createPropertyChangeListener**()

Creates a listener that is responsible that updates the UI based on how the tree changes.

### createMouseListener

protected [MouseListener](http://docs.google.com/java/awt/event/MouseListener.html) **createMouseListener**()

Creates the listener responsible for updating the selection based on mouse events.

### createFocusListener

protected [FocusListener](http://docs.google.com/java/awt/event/FocusListener.html) **createFocusListener**()

Creates a listener that is responsible for updating the display when focus is lost/gained.

### createKeyListener

protected [KeyListener](http://docs.google.com/java/awt/event/KeyListener.html) **createKeyListener**()

Creates the listener reponsible for getting key events from the tree.

### createSelectionModelPropertyChangeListener

protected [PropertyChangeListener](http://docs.google.com/java/beans/PropertyChangeListener.html) **createSelectionModelPropertyChangeListener**()

Creates the listener responsible for getting property change events from the selection model.

### createTreeSelectionListener

protected [TreeSelectionListener](http://docs.google.com/javax/swing/event/TreeSelectionListener.html) **createTreeSelectionListener**()

Creates the listener that updates the display based on selection change methods.

### createCellEditorListener

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

Creates a listener to handle events from the current editor.

### createComponentListener

protected [ComponentListener](http://docs.google.com/java/awt/event/ComponentListener.html) **createComponentListener**()

Creates and returns a new ComponentHandler. This is used for the large model to mark the validCachedPreferredSize as invalid when the component moves.

### createTreeExpansionListener

protected [TreeExpansionListener](http://docs.google.com/javax/swing/event/TreeExpansionListener.html) **createTreeExpansionListener**()

Creates and returns the object responsible for updating the treestate when nodes expanded state changes.

### createLayoutCache

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

Creates the object responsible for managing what is expanded, as well as the size of nodes.

### createCellRendererPane

protected [CellRendererPane](http://docs.google.com/javax/swing/CellRendererPane.html) **createCellRendererPane**()

Returns the renderer pane that renderer components are placed in.

### createDefaultCellEditor

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

Creates a default cell editor.

### createDefaultCellRenderer

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

Returns the default cell renderer that is used to do the stamping of each node.

### createTreeModelListener

protected [TreeModelListener](http://docs.google.com/javax/swing/event/TreeModelListener.html) **createTreeModelListener**()

Returns a listener that can update the tree when the model changes.

### uninstallUI

public void **uninstallUI**([JComponent](http://docs.google.com/javax/swing/JComponent.html) c)

**Description copied from class:** [**ComponentUI**](http://docs.google.com/javax/swing/plaf/ComponentUI.html#uninstallUI(javax.swing.JComponent)) Reverses configuration which was done on the specified component during installUI. This method is invoked when this UIComponent instance is being removed as the UI delegate for the specified component. This method should undo the configuration performed in installUI, being careful to leave the JComponent instance in a clean state (no extraneous listeners, look-and-feel-specific property objects, etc.). This should include the following:

1. Remove any UI-set borders from the component.
2. Remove any UI-set layout managers on the component.
3. Remove any UI-added sub-components from the component.
4. Remove any UI-added event/property listeners from the component.
5. Remove any UI-installed keyboard UI from the component.
6. Nullify any allocated instance data objects to allow for GC.

**Overrides:**[uninstallUI](http://docs.google.com/javax/swing/plaf/ComponentUI.html#uninstallUI(javax.swing.JComponent)) in class [ComponentUI](http://docs.google.com/javax/swing/plaf/ComponentUI.html) **Parameters:**c - the component from which this UI delegate is being removed; this argument is often ignored, but might be used if the UI object is stateless and shared by multiple components**See Also:**[ComponentUI.installUI(javax.swing.JComponent)](http://docs.google.com/javax/swing/plaf/ComponentUI.html#installUI(javax.swing.JComponent)), [JComponent.updateUI()](http://docs.google.com/javax/swing/JComponent.html#updateUI())

### prepareForUIUninstall

protected void **prepareForUIUninstall**()

### completeUIUninstall

protected void **completeUIUninstall**()

### uninstallDefaults

protected void **uninstallDefaults**()

### uninstallListeners

protected void **uninstallListeners**()

### uninstallKeyboardActions

protected void **uninstallKeyboardActions**()

### uninstallComponents

protected void **uninstallComponents**()

Uninstalls the renderer pane.

### getBaseline

public int **getBaseline**([JComponent](http://docs.google.com/javax/swing/JComponent.html) c,  
 int width,  
 int height)

Returns the baseline.

**Overrides:**[getBaseline](http://docs.google.com/javax/swing/plaf/ComponentUI.html#getBaseline(javax.swing.JComponent,%20int,%20int)) in class [ComponentUI](http://docs.google.com/javax/swing/plaf/ComponentUI.html) **Parameters:**c - JComponent baseline is being requested forwidth - the width to get the baseline forheight - the height to get the baseline for **Returns:**baseline or a value < 0 indicating there is no reasonable baseline **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if c is null [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if width or height is < 0**Since:** 1.6 **See Also:**[JComponent.getBaseline(int, int)](http://docs.google.com/javax/swing/JComponent.html#getBaseline(int,%20int))

### getBaselineResizeBehavior

public [Component.BaselineResizeBehavior](http://docs.google.com/java/awt/Component.BaselineResizeBehavior.html) **getBaselineResizeBehavior**([JComponent](http://docs.google.com/javax/swing/JComponent.html) c)

Returns an enum indicating how the baseline of the component changes as the size changes.

**Overrides:**[getBaselineResizeBehavior](http://docs.google.com/javax/swing/plaf/ComponentUI.html#getBaselineResizeBehavior(javax.swing.JComponent)) in class [ComponentUI](http://docs.google.com/javax/swing/plaf/ComponentUI.html) **Parameters:**c - JComponent to return baseline resize behavior for **Returns:**an enum indicating how the baseline changes as the component size changes **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if c is null**Since:** 1.6 **See Also:**[JComponent.getBaseline(int, int)](http://docs.google.com/javax/swing/JComponent.html#getBaseline(int,%20int))

### paint

public void **paint**([Graphics](http://docs.google.com/java/awt/Graphics.html) g,  
 [JComponent](http://docs.google.com/javax/swing/JComponent.html) c)

**Description copied from class:** [**ComponentUI**](http://docs.google.com/javax/swing/plaf/ComponentUI.html#paint(java.awt.Graphics,%20javax.swing.JComponent)) Paints the specified component appropriate for the look and feel. This method is invoked from the ComponentUI.update method when the specified component is being painted. Subclasses should override this method and use the specified Graphics object to render the content of the component.

**Overrides:**[paint](http://docs.google.com/javax/swing/plaf/ComponentUI.html#paint(java.awt.Graphics,%20javax.swing.JComponent)) in class [ComponentUI](http://docs.google.com/javax/swing/plaf/ComponentUI.html) **Parameters:**g - the Graphics context in which to paintc - the component being painted; this argument is often ignored, but might be used if the UI object is stateless and shared by multiple components**See Also:**[ComponentUI.update(java.awt.Graphics, javax.swing.JComponent)](http://docs.google.com/javax/swing/plaf/ComponentUI.html#update(java.awt.Graphics,%20javax.swing.JComponent))

### paintHorizontalPartOfLeg

protected void **paintHorizontalPartOfLeg**([Graphics](http://docs.google.com/java/awt/Graphics.html) g,  
 [Rectangle](http://docs.google.com/java/awt/Rectangle.html) clipBounds,  
 [Insets](http://docs.google.com/java/awt/Insets.html) insets,  
 [Rectangle](http://docs.google.com/java/awt/Rectangle.html) bounds,  
 [TreePath](http://docs.google.com/javax/swing/tree/TreePath.html) path,  
 int row,  
 boolean isExpanded,  
 boolean hasBeenExpanded,  
 boolean isLeaf)

Paints the horizontal part of the leg. The receiver should NOT modify clipBounds, or insets.

NOTE: parentRow can be -1 if the root is not visible.

### paintVerticalPartOfLeg

protected void **paintVerticalPartOfLeg**([Graphics](http://docs.google.com/java/awt/Graphics.html) g,  
 [Rectangle](http://docs.google.com/java/awt/Rectangle.html) clipBounds,  
 [Insets](http://docs.google.com/java/awt/Insets.html) insets,  
 [TreePath](http://docs.google.com/javax/swing/tree/TreePath.html) path)

Paints the vertical part of the leg. The receiver should NOT modify clipBounds, insets.

### paintExpandControl

protected void **paintExpandControl**([Graphics](http://docs.google.com/java/awt/Graphics.html) g,  
 [Rectangle](http://docs.google.com/java/awt/Rectangle.html) clipBounds,  
 [Insets](http://docs.google.com/java/awt/Insets.html) insets,  
 [Rectangle](http://docs.google.com/java/awt/Rectangle.html) bounds,  
 [TreePath](http://docs.google.com/javax/swing/tree/TreePath.html) path,  
 int row,  
 boolean isExpanded,  
 boolean hasBeenExpanded,  
 boolean isLeaf)

Paints the expand (toggle) part of a row. The receiver should NOT modify clipBounds, or insets.

### paintRow

protected void **paintRow**([Graphics](http://docs.google.com/java/awt/Graphics.html) g,  
 [Rectangle](http://docs.google.com/java/awt/Rectangle.html) clipBounds,  
 [Insets](http://docs.google.com/java/awt/Insets.html) insets,  
 [Rectangle](http://docs.google.com/java/awt/Rectangle.html) bounds,  
 [TreePath](http://docs.google.com/javax/swing/tree/TreePath.html) path,  
 int row,  
 boolean isExpanded,  
 boolean hasBeenExpanded,  
 boolean isLeaf)

Paints the renderer part of a row. The receiver should NOT modify clipBounds, or insets.

### shouldPaintExpandControl

protected boolean **shouldPaintExpandControl**([TreePath](http://docs.google.com/javax/swing/tree/TreePath.html) path,  
 int row,  
 boolean isExpanded,  
 boolean hasBeenExpanded,  
 boolean isLeaf)

Returns true if the expand (toggle) control should be drawn for the specified row.

### paintVerticalLine

protected void **paintVerticalLine**([Graphics](http://docs.google.com/java/awt/Graphics.html) g,  
 [JComponent](http://docs.google.com/javax/swing/JComponent.html) c,  
 int x,  
 int top,  
 int bottom)

Paints a vertical line.

### paintHorizontalLine

protected void **paintHorizontalLine**([Graphics](http://docs.google.com/java/awt/Graphics.html) g,  
 [JComponent](http://docs.google.com/javax/swing/JComponent.html) c,  
 int y,  
 int left,  
 int right)

Paints a horizontal line.

### getVerticalLegBuffer

protected int **getVerticalLegBuffer**()

The vertical element of legs between nodes starts at the bottom of the parent node by default. This method makes the leg start below that.

### getHorizontalLegBuffer

protected int **getHorizontalLegBuffer**()

The horizontal element of legs between nodes starts at the right of the left-hand side of the child node by default. This method makes the leg end before that.

### drawCentered

protected void **drawCentered**([Component](http://docs.google.com/java/awt/Component.html) c,  
 [Graphics](http://docs.google.com/java/awt/Graphics.html) graphics,  
 [Icon](http://docs.google.com/javax/swing/Icon.html) icon,  
 int x,  
 int y)

### drawDashedHorizontalLine

protected void **drawDashedHorizontalLine**([Graphics](http://docs.google.com/java/awt/Graphics.html) g,  
 int y,  
 int x1,  
 int x2)

### drawDashedVerticalLine

protected void **drawDashedVerticalLine**([Graphics](http://docs.google.com/java/awt/Graphics.html) g,  
 int x,  
 int y1,  
 int y2)

### getRowX

protected int **getRowX**(int row,  
 int depth)

Returns the location, along the x-axis, to render a particular row at. The return value does not include any Insets specified on the JTree. This does not check for the validity of the row or depth, it is assumed to be correct and will not throw an Exception if the row or depth doesn't match that of the tree.

**Parameters:**row - Row to return x location fordepth - Depth of the row **Returns:**amount to indent the given row.**Since:** 1.5

### updateLayoutCacheExpandedNodes

protected void **updateLayoutCacheExpandedNodes**()

Makes all the nodes that are expanded in JTree expanded in LayoutCache. This invokes updateExpandedDescendants with the root path.

### updateExpandedDescendants

protected void **updateExpandedDescendants**([TreePath](http://docs.google.com/javax/swing/tree/TreePath.html) path)

Updates the expanded state of all the descendants of path by getting the expanded descendants from the tree and forwarding to the tree state.

### getLastChildPath

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

Returns a path to the last child of parent.

### updateDepthOffset

protected void **updateDepthOffset**()

Updates how much each depth should be offset by.

### updateCellEditor

protected void **updateCellEditor**()

Updates the cellEditor based on the editability of the JTree that we're contained in. If the tree is editable but doesn't have a cellEditor, a basic one will be used.

### updateRenderer

protected void **updateRenderer**()

Messaged from the tree we're in when the renderer has changed.

### configureLayoutCache

protected void **configureLayoutCache**()

Resets the TreeState instance based on the tree we're providing the look and feel for.

### updateSize

protected void **updateSize**()

Marks the cached size as being invalid, and messages the tree with treeDidChange.

### updateCachedPreferredSize

protected void **updateCachedPreferredSize**()

Updates the preferredSize instance variable, which is returned from getPreferredSize().

For left to right orientations, the size is determined from the current AbstractLayoutCache. For RTL orientations, the preferred size becomes the width minus the minimum x position.

### pathWasExpanded

protected void **pathWasExpanded**([TreePath](http://docs.google.com/javax/swing/tree/TreePath.html) path)

Messaged from the VisibleTreeNode after it has been expanded.

### pathWasCollapsed

protected void **pathWasCollapsed**([TreePath](http://docs.google.com/javax/swing/tree/TreePath.html) path)

Messaged from the VisibleTreeNode after it has collapsed.

### ensureRowsAreVisible

protected void **ensureRowsAreVisible**(int beginRow,  
 int endRow)

Ensures that the rows identified by beginRow through endRow are visible.

### setPreferredMinSize

public void **setPreferredMinSize**([Dimension](http://docs.google.com/java/awt/Dimension.html) newSize)

Sets the preferred minimum size.

### getPreferredMinSize

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

Returns the minimum preferred size.

### getPreferredSize

public [Dimension](http://docs.google.com/java/awt/Dimension.html) **getPreferredSize**([JComponent](http://docs.google.com/javax/swing/JComponent.html) c)

Returns the preferred size to properly display the tree, this is a cover method for getPreferredSize(c, false).

**Overrides:**[getPreferredSize](http://docs.google.com/javax/swing/plaf/ComponentUI.html#getPreferredSize(javax.swing.JComponent)) in class [ComponentUI](http://docs.google.com/javax/swing/plaf/ComponentUI.html) **Parameters:**c - the component whose preferred size is being queried; this argument is often ignored, but might be used if the UI object is stateless and shared by multiple components**See Also:**[JComponent.getPreferredSize()](http://docs.google.com/javax/swing/JComponent.html#getPreferredSize()), [LayoutManager.preferredLayoutSize(java.awt.Container)](http://docs.google.com/java/awt/LayoutManager.html#preferredLayoutSize(java.awt.Container))

### getPreferredSize

public [Dimension](http://docs.google.com/java/awt/Dimension.html) **getPreferredSize**([JComponent](http://docs.google.com/javax/swing/JComponent.html) c,  
 boolean checkConsistancy)

Returns the preferred size to represent the tree in *c*. If *checkConsistancy* is true **checkConsistancy** is messaged first.

### getMinimumSize

public [Dimension](http://docs.google.com/java/awt/Dimension.html) **getMinimumSize**([JComponent](http://docs.google.com/javax/swing/JComponent.html) c)

Returns the minimum size for this component. Which will be the min preferred size or 0, 0.

**Overrides:**[getMinimumSize](http://docs.google.com/javax/swing/plaf/ComponentUI.html#getMinimumSize(javax.swing.JComponent)) in class [ComponentUI](http://docs.google.com/javax/swing/plaf/ComponentUI.html) **Parameters:**c - the component whose minimum size is being queried; this argument is often ignored, but might be used if the UI object is stateless and shared by multiple components **Returns:**a Dimension object or null**See Also:**[JComponent.getMinimumSize()](http://docs.google.com/javax/swing/JComponent.html#getMinimumSize()), [LayoutManager.minimumLayoutSize(java.awt.Container)](http://docs.google.com/java/awt/LayoutManager.html#minimumLayoutSize(java.awt.Container)), [ComponentUI.getPreferredSize(javax.swing.JComponent)](http://docs.google.com/javax/swing/plaf/ComponentUI.html#getPreferredSize(javax.swing.JComponent))

### getMaximumSize

public [Dimension](http://docs.google.com/java/awt/Dimension.html) **getMaximumSize**([JComponent](http://docs.google.com/javax/swing/JComponent.html) c)

Returns the maximum size for this component, which will be the preferred size if the instance is currently in a JTree, or 0, 0.

**Overrides:**[getMaximumSize](http://docs.google.com/javax/swing/plaf/ComponentUI.html#getMaximumSize(javax.swing.JComponent)) in class [ComponentUI](http://docs.google.com/javax/swing/plaf/ComponentUI.html) **Parameters:**c - the component whose maximum size is being queried; this argument is often ignored, but might be used if the UI object is stateless and shared by multiple components **Returns:**a Dimension object or null**See Also:**[JComponent.getMaximumSize()](http://docs.google.com/javax/swing/JComponent.html#getMaximumSize()), [LayoutManager2.maximumLayoutSize(java.awt.Container)](http://docs.google.com/java/awt/LayoutManager2.html#maximumLayoutSize(java.awt.Container))

### completeEditing

protected void **completeEditing**()

Messages to stop the editing session. If the UI the receiver is providing the look and feel for returns true from getInvokesStopCellEditing, stopCellEditing will invoked on the current editor. Then completeEditing will be messaged with false, true, false to cancel any lingering editing.

### completeEditing

protected void **completeEditing**(boolean messageStop,  
 boolean messageCancel,  
 boolean messageTree)

Stops the editing session. If messageStop is true the editor is messaged with stopEditing, if messageCancel is true the editor is messaged with cancelEditing. If messageTree is true the treeModel is messaged with valueForPathChanged.

### startEditing

protected boolean **startEditing**([TreePath](http://docs.google.com/javax/swing/tree/TreePath.html) path,  
 [MouseEvent](http://docs.google.com/java/awt/event/MouseEvent.html) event)

Will start editing for node if there is a cellEditor and shouldSelectCell returns true.

This assumes that path is valid and visible.

### checkForClickInExpandControl

protected void **checkForClickInExpandControl**([TreePath](http://docs.google.com/javax/swing/tree/TreePath.html) path,  
 int mouseX,  
 int mouseY)

If the mouseX and mouseY are in the expand/collapse region of the row, this will toggle the row.

### isLocationInExpandControl

protected boolean **isLocationInExpandControl**([TreePath](http://docs.google.com/javax/swing/tree/TreePath.html) path,  
 int mouseX,  
 int mouseY)

Returns true if mouseX and mouseY fall in the area of row that is used to expand/collapse the node and the node at row does not represent a leaf.

### handleExpandControlClick

protected void **handleExpandControlClick**([TreePath](http://docs.google.com/javax/swing/tree/TreePath.html) path,  
 int mouseX,  
 int mouseY)

Messaged when the user clicks the particular row, this invokes toggleExpandState.

### toggleExpandState

protected void **toggleExpandState**([TreePath](http://docs.google.com/javax/swing/tree/TreePath.html) path)

Expands path if it is not expanded, or collapses row if it is expanded. If expanding a path and JTree scrolls on expand, ensureRowsAreVisible is invoked to scroll as many of the children to visible as possible (tries to scroll to last visible descendant of path).

### isToggleSelectionEvent

protected boolean **isToggleSelectionEvent**([MouseEvent](http://docs.google.com/java/awt/event/MouseEvent.html) event)

Returning true signifies a mouse event on the node should toggle the selection of only the row under mouse.

### isMultiSelectEvent

protected boolean **isMultiSelectEvent**([MouseEvent](http://docs.google.com/java/awt/event/MouseEvent.html) event)

Returning true signifies a mouse event on the node should select from the anchor point.

### isToggleEvent

protected boolean **isToggleEvent**([MouseEvent](http://docs.google.com/java/awt/event/MouseEvent.html) event)

Returning true indicates the row under the mouse should be toggled based on the event. This is invoked after checkForClickInExpandControl, implying the location is not in the expand (toggle) control

### selectPathForEvent

protected void **selectPathForEvent**([TreePath](http://docs.google.com/javax/swing/tree/TreePath.html) path,  
 [MouseEvent](http://docs.google.com/java/awt/event/MouseEvent.html) event)

Messaged to update the selection based on a MouseEvent over a particular row. If the event is a toggle selection event, the row is either selected, or deselected. If the event identifies a multi selection event, the selection is updated from the anchor point. Otherwise the row is selected, and if the event specified a toggle event the row is expanded/collapsed.

### isLeaf

protected boolean **isLeaf**(int row)

**Returns:**true if the node at row is a leaf.

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/BasicTreeUI.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/plaf/basic/BasicToolTipUI.html)   [**NEXT CLASS**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.CellEditorHandler.html) | [**FRAMES**](http://docs.google.com/index.html?javax/swing/plaf/basic/BasicTreeUI.html)    [**NO FRAMES**](http://docs.google.com/BasicTreeUI.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: [NESTED](#3znysh7) | [FIELD](#2et92p0) | [CONSTR](#tyjcwt) | [METHOD](#3dy6vkm) | DETAIL: [FIELD](#2s8eyo1) | [CONSTR](#41mghml) | [METHOD](#vx1227) |

[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).